MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Face Recognition in Subspaces

Gregory Shakhnarovich
Baback Moghaddam

TR2004-041 May 2004

Abstract

Images of faces, represented as high-dimensional pixel arrays, often belong to a manifold of intrinsically low dimension. Face recognition, and computer vision research in general, has witnessed a growing interest in techniques that capitalize on this observation, and apply algebraic and statistical tools for extraction and analysis of the underlying manifold. In this chapter we describe in roughly chronological order techniques that identify, parameterize and analyze linear and nonlinear subspaces, from the original Eigenfaces technique to the recently introduced Bayesian method for probabilistic similarity analysis, and discuss comparative experimental evaluation of some of these techniques. We also discuss practical issues related to the application of subspace methods for varying pose, illumination and expression.

Published in: **Handbook of Face Recognition**, *Eds. Stan Z. Li & Anil K. Jain, Springer-Verlag, 2004*

Publication History:–

1. First printing, TR2004-041, May 2004

Face Recognition in Subspaces

Gregory Shakhnarovich[1] and Baback Moghaddam[2]

[1] Massachusetts Institute of Technology, Cambridge MA, 02139, USA.
gregory@ai.mit.edu
[2] Mitsubishi Electric Research Laboratories, Cambridge MA, 02139, USA.
baback@merl.com

Images of faces, represented as high-dimensional pixel arrays, often belong to a manifold of intrinsically low dimension. Face recognition, and computer vision research in general, has witnessed a growing interest in techniques that capitalize on this observation, and apply algebraic and statistical tools for extraction and analysis of the underlying manifold. In this chapter we describe in roughly chronological order techniques that identify, parameterize and analyze linear and nonlinear subspaces, from the original Eigenfaces technique to the recently introduced Bayesian method for probabilistic similarity analysis, and discuss comparative experimental evaluation of some of these techniques. We also discuss practical issues related to the application of subspace methods for varying pose, illumination and expression.

1 Face Space and its Dimensionality

Computer analysis of face images deals with a visual signal (light reflected off the surface of a face) that is registered by a digital sensor as an array of pixel values. The pixels may encode color or only intensity; In this chapter we will assume the latter case, i.e. gray-level imagery. After proper normalization and resizing to a fixed m-by-n size, the pixel array can be represented as a point (i.e. vector) in an mn-dimensional *image space* by simply writing its pixel values in a fixed (typically raster) order. A critical issue in the analysis of such multi-dimensional data is the *dimensionality*, the number of coordinates necessary to specify a data point. Below we discuss the factors affecting this number in the case of face images.

1.1 Image Space vs. Face Space

In order to specify an arbitrary image in the image space, one needs to specify every pixel value. Thus the "nominal" dimensionality of the space, dictated by the pixel representation, is mn - a very high number even for images of modest

size. Recognition methods that operate on this representation suffer from a number of potential disadvantages, most of them rooted in the so-called curse of dimensionality:

- Handling high-dimensional examples, especially in the context of similarity- / matching-based recognition, is computationally expensive.
- For parametric methods, the number of parameters one needs to estimate typically grows exponentially with the dimensionality. Often this number is much higher than the number of images available for training, making the estimation task in the image space ill-posed.
- Similarly, for non-parametric methods, the sample complexity – the number of examples needed to efficiently represent the underlying distribution of the data – is prohibitively high.

However, much of the surface of a face is smooth and has regular texture. Therefore, per-pixel sampling is in fact unnecessarily dense: The value of a pixel is typically highly correlated with the values of the surrounding pixels. Moreover, the appearance of faces is highly constrained; for example, any frontal view of a face is roughly symmetrical, has eyes on the sides, nose in the middle, *etc.* A vast proportion of the points in the image space does not represent physically possible faces.

Thus, the natural constraints dictate that the face images will in fact be confined to a subspace, which is referred to as the *face space*.

1.2 The Principal Manifold and Basis Functions

It is common to model the face space as a (possibly disconnected) *principal manifold*, embedded in the high-dimensional image space. Its *intrinsic* dimensionality is determined by the number of degrees of freedom within the face space; the goal of subspace analysis is to determine this number, and to extract the *principal modes* of the manifold. The principal modes are computed as functions of the pixel values and referred to as *basis functions* of the principal manifold.

To make these concepts concrete, consider a straight line in $\mathbb{R}^3$, passing through the origin and parallel to the vector $\mathbf{a} = [a_1, a_2, a_3]^T$. Any point on the line can be described by 3 coordinates; nevertheless, the subspace that consists of all points on the line has a single degree of freedom, with the principal mode corresponding to translation along the direction of $\mathbf{a}$. Consequently, representing the points in this subspace requires a single basis function: $\phi(x_1, x_2, x_3) = \sum_{j=1}^{3} a_j x_j$. The analogy here is between the line and the face space, and between $\mathbb{R}^3$ and the image space.

Note that in theory, according to the described model any face image should fall in the face space. In practice, due to sensor noise, the signal usually will have a non-zero component outside of the face space. This introduces uncertainty into the model and requires algebraic and statistical techniques

capable of extracting the basis functions of the principal manifold in the presence of noise. In Section 1.3 we briefly describe Principal Component Analysis, that plays an important role in many of such techniques. For a more detailed discussion, see [12, 17].

1.3 Principal Component Analysis

Principal Component Analysis (PCA) [17] is a dimensionality reduction technique based on extracting the desired number of *principal components* of the multi-dimensional data. The first principal component is the linear combination of the original dimensions that has the maximum variance; the n-th principal component is the linear combination with the highest variance, subject to being orthogonal to the $n-1$ first principal components.

The idea of PCA is illustrated in Figure 1(a); the axis labeled ϕ_1 corresponds to the direction of maximum variance and is chosen as the first principal component. In a 2D case, the second principal component is then determined uniquely by the orthogonality constraints; in a higher-dimensional space the selection process would continue, guided by the variances of the projections.

PCA is closely related to the Karhunen-Loève Transform (KLT) [21], which was derived in the signal processing context as the orthogonal transform with the basis $\mathbf{\Phi} = [\phi_1, \ldots, \phi_N]^T$ that for any $k \leq N$ minimizes the average L_2 *reconstruction error* for data points $\mathbf{x}$

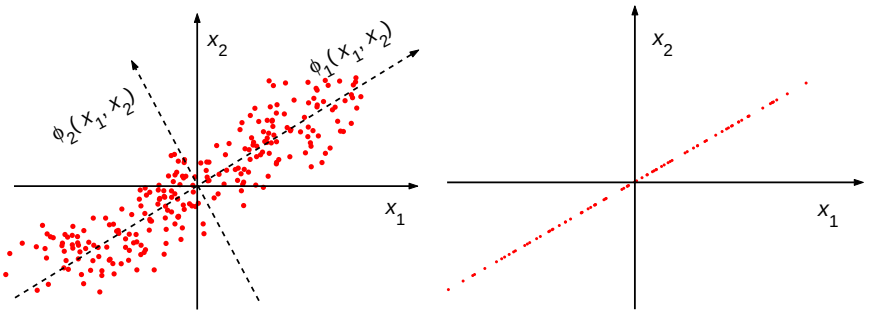
$$\epsilon(\mathbf{x}) = \left\| \mathbf{x} - \sum_{i=1}^{k} \left(\phi_i^T \mathbf{x}\right) \phi_i \right\|. \tag{1}$$

One can show [12] that, under the assumption that the data is zero-mean, the formulations of PCA and KLT are identical. Without loss of generality we will hereafter assume that the data is indeed zero-mean, that is, the mean face $\bar{\mathbf{x}}$ is always subtracted from the data.

The basis vectors in KLT can be calculated in the following way. Let $\mathbf{X}$ be the $N \times M$ data matrix whose columns $\mathbf{x}_1, \ldots, \mathbf{x}_M$ are *observations* of a signal embedded in $\mathbb{R}^N$; in the context of face recognition, M is the number of available face images and $N = mn$ is the number of pixels in an image. The KLT basis $\mathbf{\Phi}$ is obtained by solving the eigenvalue problem $\mathbf{\Lambda} = \mathbf{\Phi}^T \mathbf{\Sigma} \mathbf{\Phi}$, where $\mathbf{\Sigma}$ is the covariance matrix of the data

$$\mathbf{\Sigma} \;=\; \frac{1}{M} \sum_{i=1}^{M} \mathbf{x}_i \mathbf{x}_i^T, \tag{2}$$

$\mathbf{\Phi} = [\phi_1, \ldots, \phi_m]^T$ is the eigenvector matrix of $\mathbf{\Sigma}$, and $\mathbf{\Lambda}$ is the diagonal matrix with eigenvalues $\lambda_1 \geq \ldots \geq \lambda_N$ of $\mathbf{\Sigma}$ on its main diagonal, so that ϕ_j is the eigenvector corresponding to the j-th largest eigenvalue. Then it can be shown that the eigenvalue λ_i is the variance of the data projected on ϕ_i.



(a) PCA basis (b) PCA reduction to 1D

Fig. 1. The concept of PCA/KLT. (a) Solid lines: the original basis; dashed lines: the KLT basis. The dots are selected at regularly spaced locations on a straight line rotated at 30°, and then perturbed by isotropic 2D Gaussian noise. (b) The projection (1D reconstruction) of the data using only the first principal component.

Thus, to perform PCA and extract k principal components of the data, one must project the data onto $\mathbf{\Phi}_k$ – the first k columns of the KLT basis $\mathbf{\Phi}$, which correspond to the k highest eigenvalues of $\mathbf{\Sigma}$. This can be seen as a linear projection $\mathbb{R}^N \rightarrow \mathbb{R}^k$ that retains the maximum energy (i.e. variance) of the signal. Another important property of PCA is that it *decorrelates* the data: the covariance matrix of $\mathbf{\Phi}_k^T \mathbf{X}$ is always diagonal.

The main properties of PCA are summarized by the following:

$$\mathbf{x} \approx \mathbf{\Phi}_k \mathbf{y} \ , \ \mathbf{\Phi}_k^T \mathbf{\Phi}_k = \mathbf{I} \ , \ E\{y_i y_j\}_{i \neq j} = 0 \tag{3}$$

namely, approximate reconstruction, orthonormality of the basis $\mathbf{\Phi}_k$ and decorrelated principal components $y_i = \phi_i^T \mathbf{x}$, respectively. These properties are illustrated in Figure 1, where PCA is successful in finding the principal manifold, and in Figure 8(a) where it is less successful, due to clear non-linearity of the principal manifold.

PCA may be implemented via Singular Value Decomposition (SVD): The SVD of an $M \times N$ matrix $\mathbf{X}$ ($M \geq N$) is given by

$$\mathbf{X} = \mathbf{U}\ \mathbf{D}\ \mathbf{V}^T, \tag{4}$$

where the $M \times N$ matrix $\mathbf{U}$ and the $N \times N$ matrix $\mathbf{V}$ have orthonormal columns, and the $N \times N$ matrix $\mathbf{D}$ has the singular values[3] of $\mathbf{X}$ on its main diagonal and zero elsewhere.

It can be shown that $\mathbf{U} = \mathbf{\Phi}$, so that SVD allows efficient and robust computation of PCA without the need to estimate the data covariance matrix

[3] A singular value of a matrix $\mathbf{X}$ is the square root of an eigenvalue of $\mathbf{X}\mathbf{X}^T$.

$\mathbf{\Sigma}$ (2). When the number of examples M is much smaller than the dimension N, this is a crucial advantage.

1.4 Eigenspectrum and Dimensionality

An important, and largely unsolved problem in dimensionality reduction is the choice of k – the intrinsic dimensionality of the principal manifold. No analytical derivation of this number for a complex natural visual signal is available to date. To simplify this problem, it is common to assume that in the noisy embedding of the signal of interest (in our case, a point sampled from the face space) in a high-dimensional space, the *signal-to-noise ratio* is high. Statistically, that means that the variance of the data along the principal modes of the manifold is high compared to the variance within the complementary space.

This assumption relates to the *eigenspectrum* - the set of the eigenvalues of the data covariance matrix $\mathbf{\Sigma}$. Recall that the i-th eigenvalue is equal to the variance along the i-th principal component; thus, a reasonable algorithm for detecting k is to search for the location along the decreasing eigenspectrum where the value of λ_i drops significantly. A typical eigenspectrum for a face recognition problem, and the natural choice of k for such a spectrum, is shown in Figure 3(b).

In practice the choice of k is also guided by computational constraints, related to the cost of matching within the extracted principal manifold and the number of available face images; please see [29] as well as Sections 2.2, 2.4 for more discussion on this issue.

2 Linear Subspaces

Perhaps the simplest case of principal manifold analysis arises under the assumption that the principal manifold is linear. After the origin has been translated to the *mean face* (the average image in the database) by subtracting it from every image, this means that the face space is a linear subspace of the image space. In this section we describe methods that operate under this assumption and its generalization – a multi-linear manifold.

2.1 Eigenfaces and Related Techniques

In their ground-breaking work in 1991 Kirby and Sirovich [19] proposed the use of PCA for face analysis and representation. Their paper was followed by the "Eigenfaces" technique by Turk and Pentland [35], the first application of PCA to face recognition. Since the basis vectors constructed by PCA had the same dimension as the input face images, they were named "Eigenfaces". Figure 2 shows an example of the mean face and a few of the top Eigenfaces.

Fig. 2. Eigenfaces (from [36]): average face on the left, followed by 7 top eigenfaces.

Every face image was projected (after subtracting the mean face) into the principal subspace; the coefficients of the PCA expansion were averaged for each subject, resulting in a single k-dimensional representation of that subject. When a test image was projected into the subspace, Euclidean distances between its coefficient vector and those representing each subject were computed. Depending on the distance to the subject for which this distance would be minimized, and the PCA reconstruction error (1), the image was classified as belonging to one of the familiar subjects, as a new face, or as non-face. The latter demonstrates the dual use of subspace techniques for *detection*: when the appearance of an object class (e.g. faces) is modeled by a subspace, the distance from this subspace can serve to classify an object as a member or non-member of the class.

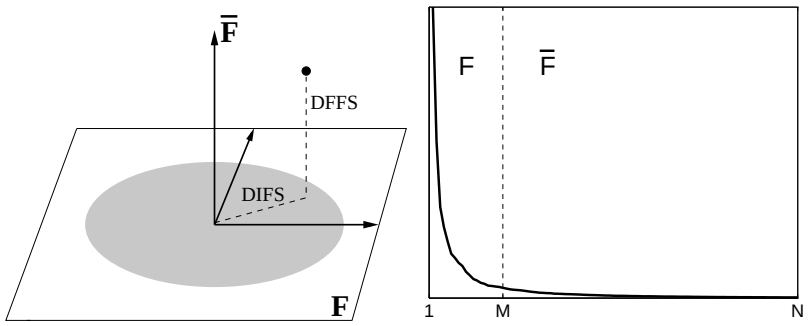
2.2 Probabilistic Eigenspaces

The role of PCA in the original Eigenfaces was largely confined to dimensionality reduction. The similarity between images $\mathbf{I}_1, \mathbf{I}_2$ was measured in terms of the Euclidean norm of the difference $\mathbf{\Delta} = \mathbf{I}_1 - \mathbf{I}_2$ projected to the subspace, essentially ignoring the variation modes both within the subspace and outside of it. This was improved in the extension of Eigenfaces proposed by Moghaddam and Pentland [26, 27] that uses a *probabilistic* similarity measure, based on a parametric estimate of the probability density $p(\mathbf{\Delta}|\Omega)$.

A major difficulty in such estimation is that normally there is not nearly enough data to estimate the parameters of the density in a high dimensional space. Moghaddam and Pentland overcome this problem by using PCA to divide the vector space $\mathbb{R}^N$ into two subspaces as shown in Figure 3: the principal subspace F, obtained by $\mathbf{\Phi}_k$ (the first k columns of $\mathbf{\Phi}$) and its orthogonal complement $\bar{F}$ spanned by the remaining columns of $\mathbf{\Phi}$. The operating assumption here is that the data have intrinsic dimensionality k (at most) and thus reside in F, with the exception of additive white Gaussian noise within $\bar{F}$. Every image can be decomposed into two orthogonal components by projection into these two spaces. Figure 3(a) shows the decomposition of $\mathbf{\Delta}$ into distance *within* face space (DIFS) and distance *from* face space (DFFS). Moreover, the probability density can be decomposed into two orthogonal components:

$$P(\mathbf{\Delta}|\Omega) = P_F(\mathbf{\Delta}|\Omega) \cdot P_{\bar{F}}(\mathbf{\Delta}|\Omega). \tag{5}$$

In the simplest case, $P(\mathbf{\Delta}|\Omega)$ is a Gaussian density. As derived in [26], the complete likelihood estimate in this case can be written as the product of two



(a) (b)

Fig. 3. (a) Decomposition of $\mathbb{R}^N$ into the principal subspace F and its orthogonal complement $\bar{F}$ for a Gaussian density, (b) a typical eigenvalue spectrum and its division into the two orthogonal subspaces.

independent marginal Gaussian densities

$$\hat{P}(\mathbf{\Delta}|\Omega) = \left[\frac{\exp\left(-\frac{1}{2}\sum_{i=1}^{k}\frac{y_i^2}{\lambda_i}\right)}{(2\pi)^{k/2}\prod_{i=1}^{k}\lambda_i^{1/2}} \right] \cdot \left[\frac{\exp\left(-\frac{\epsilon^2(\Delta)}{2\rho}\right)}{(2\pi\rho)^{(N-k)/2}} \right] \tag{6}$$

$$= P_F(\mathbf{\Delta}|\Omega)\ \hat{P}_{\bar{F}}(\mathbf{\Delta}|\Omega;\rho)\ ,$$

where $P_F(\mathbf{\Delta}|\Omega)$ is the true marginal density in F, $\hat{P}_{\bar{F}}(\mathbf{\Delta}|\Omega;\rho)$ is the estimated marginal density in $\bar{F}$, $y_i = \phi_i^T\mathbf{\Delta}$ are the principal components of $\mathbf{\Delta}$ and $\epsilon(\mathbf{\Delta})$ is the PCA reconstruction error (1). The information-theoretic optimal value for the noise density parameter ρ is derived by minimizing the Kullback-Leibler (KL) divergence [8] and can be shown to be simply the average of the $N-k$ smallest eigenvalues

$$\rho\ =\ \frac{1}{N-k}\sum_{i=k+1}^{N}\lambda_i\ . \tag{7}$$

This is a special case of the recent, more general factor analysis model called Probabilistic PCA (PPCA) proposed by Tipping & Bishop [34]. In their formulation, the above expression for ρ is the maximum-likelihood solution of a latent variable model as opposed to the minimal-divergence solution derived in [26].

In practice, the majority of the eigenvalues in $\bar{F}$ can not be computed due to insufficient data, but they can be estimated, for example, by fitting a

nonlinear function to the available portion of the eigenvalue spectrum and estimating the average of the eigenvalues beyond the principal subspace. Fractal power law spectra of the form f^{-n} are thought to be typical of "natural" phenomenon and are often a good fit to the decaying nature of the eigenspectrum, as illustrated by Figure 3(b).

In this probabilistic framework, the recognition of a test image $\mathbf{x}$ is carried out in terms of computing for every database example $\mathbf{x}_i$ the difference $\mathbf{\Delta} = \mathbf{x} - \mathbf{x}_i$ and its decomposition into the F and $\bar{F}$ components, and then ranking the examples according to the value in (6).

2.3 Linear Discriminants: Fisherfaces

When substantial changes in illumination and expression are present, much of the variation in the data is due to these changes. The PCA techniques essentially select a subspace which retains most of that variation, and consequently the similarity in the face space is not necessarily determined by the identity.

In [2], Belhumeur *et al.* propose to solve this problem with "Fisherfaces" – an application of Fisher's Linear Discriminant (FLD). FLD selects the linear subspace $\mathbf{\Phi}$ which maximizes the ratio

$$\frac{\left|\mathbf{\Phi}^T \mathbf{S_b} \mathbf{\Phi}\right|}{\left|\mathbf{\Phi}^T \mathbf{S_w} \mathbf{\Phi}\right|} \tag{8}$$

where

$$\mathbf{S_b} = \sum_{i=1}^{m} N_i (\bar{\mathbf{x}}_i - \bar{\mathbf{x}})(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})^T,$$

is the *between-class* scatter matrix, and

$$\mathbf{S_w} = \sum_{i=1}^{m} \sum_{\mathbf{x} \in \mathbf{X}_i} (\mathbf{x} - \bar{\mathbf{x}}_i)(\mathbf{x} - \bar{\mathbf{x}}_i)^T$$

is the *within-class* scatter matrix; m is the number of subjects (classes) in the database. Intuitively, FLD finds the projection of the data in which the classes are most linearly separable. It can be shown that the dimension of $\mathbf{\Phi}$ is at most $m - 1$.[4]

Since in practice $\mathbf{S_w}$ is usually singular, the Fisherfaces algorithm first reduces the dimensionality of the data with PCA so that (8) can be computed, and then applies FLD to further reduce the dimensionality to $m - 1$. The recognition is then accomplished by a NN classifier in this final subspace. The experiments reported in [2] were performed on data sets containing frontal face images of 5 people with drastic lighting variations and another set with faces of 16 people with varying expressions and again drastic illumination changes. In all the reported experiments Fisherfaces achieve lower error rate than Eigenfaces.

[4] For comparison, note that the objective of PCA can bee seen as maximizing the total scatter across all the images in the database.

2.4 Bayesian Methods

Consider now a feature space of $\mathbf{\Delta}$ vectors, the differences between two images ($\mathbf{\Delta} = \mathbf{I}_j - \mathbf{I}_k$). One can define two classes of facial image variations: *intrapersonal* variations Ω_I (corresponding, for example, to different facial expressions, illuminations, *etc* of the *same* individual) and *extrapersonal* variations Ω_E (corresponding to variations between *different* individuals). The similarity measure $S(\mathbf{\Delta})$ can then be expressed in terms of the intrapersonal *a posteriori* probability of $\mathbf{\Delta}$ belonging to Ω_I given by the Bayes rule:

$$S(\mathbf{\Delta}) \;=\; P(\Omega_I|\mathbf{\Delta}) \;=\; \frac{P(\mathbf{\Delta}|\Omega_I)P(\Omega_I)}{P(\mathbf{\Delta}|\Omega_I)P(\Omega_I) + P(\mathbf{\Delta}|\Omega_E)P(\Omega_E)} \tag{9}$$

Note that this particular Bayesian formulation, proposed by Moghaddam *et al.* in [25], casts the standard face recognition task (essentially an m-ary classification problem for m individuals) into a *binary* pattern classification problem with Ω_I and Ω_E.

The densities of both classes are modeled as high-dimensional Gaussians, using an efficient PCA-based method described in Section 2.2:

$$\begin{aligned} P(\mathbf{\Delta}|\Omega_E) &= \frac{e^{-\frac{1}{2}\mathbf{\Delta}^T \Sigma_E^{-1}\mathbf{\Delta}}}{(2\pi)^{D/2}|\Sigma_E|^{1/2}} \\ P(\mathbf{\Delta}|\Omega_I) &= \frac{e^{-\frac{1}{2}\mathbf{\Delta}^T \Sigma_I^{-1}\mathbf{\Delta}}}{(2\pi)^{D/2}|\Sigma_I|^{1/2}} \end{aligned} \tag{10}$$

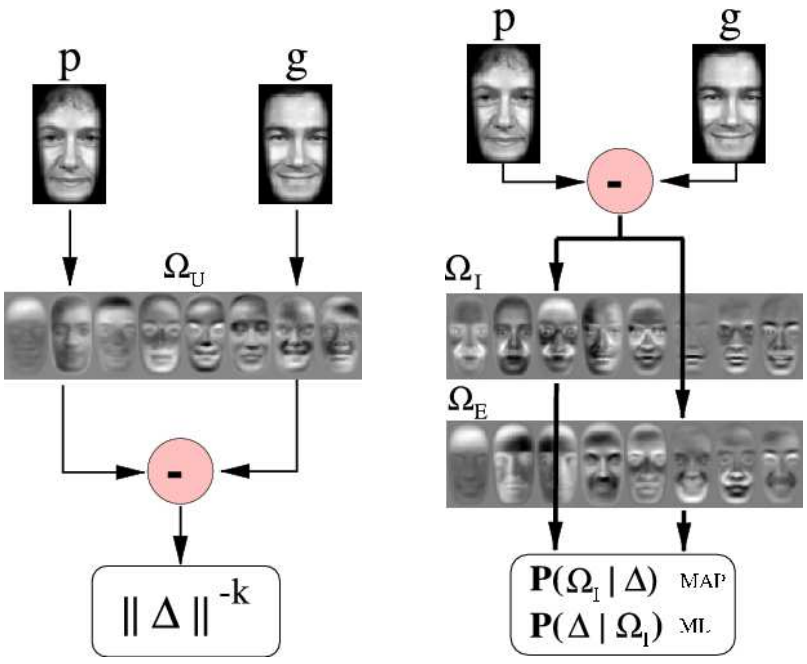
These densities are zero-mean, since for each $\mathbf{\Delta} = \mathbf{I}_j - \mathbf{I}_i$, there exists a $\mathbf{I}_i - \mathbf{I}_j$.

By PCA, the Gaussians are known to only occupy a subspace of image space (face space) and thus, only the top few eigenvectors of the Gaussian densities are relevant for modeling. These densities are used to evaluate the similarity in (9). Computing the similarity involves first subtracting a candidate image $\mathbf{I}$ from a database example $\mathbf{I}_j$. The resulting $\mathbf{\Delta}$ image is then projected onto the eigenvectors of the extrapersonal Gaussian and also the eigenvectors of the intrapersonal Gaussian. The exponentials are computed, normalized and then combined as in (9). This operation is iterated over all examples in the database, and the example that achieves the maximum score is considered the match. For large databases, such evaluations are expensive and it is desirable to simplify them by off-line transformations.

To compute the likelihoods $P(\mathbf{\Delta}|\Omega_I)$ and $P(\mathbf{\Delta}|\Omega_E)$, the database images $\mathbf{I}_j$ are pre-processed with *whitening* transformations [11]. Each image is converted and stored as a set of two whitened subspace coefficients; $\mathbf{y}_{\Phi_I}$ for intrapersonal space and $\mathbf{y}_{\Phi_E}$ for extrapersonal space:

$$\mathbf{y}^j_{\Phi_I} = \mathbf{\Lambda}_I^{-\frac{1}{2}}\mathbf{V}_I\mathbf{I}_j \;, \qquad \mathbf{y}^j_{\Phi_E} = \mathbf{\Lambda}_E^{-\frac{1}{2}}\mathbf{V}_E\mathbf{I}_j \;, \tag{11}$$

where $\mathbf{\Lambda}_X$ and $\mathbf{V}_X$ are matrices of the largest eigenvalues and eigenvectors, respectively, of $\mathbf{\Sigma}_X$ (X being a substituting symbol for I or E).



(a) (b)

Fig. 4. Signal flow diagrams for computing the similarity **g** between two images: (a) The original Eigenfaces. (b) Bayesian similarity. The difference image is projected through both sets of (intra/extra) eigenfaces in order to obtain the two likelihoods.

After this pre-processing, evaluating the Gaussians can be reduced to simple Euclidean distances as in (12). Denominators are of course pre-computed. These likelihoods are evaluated and used to compute the *maximum a-posteriori* (MAP) similarity $S(\mathbf{\Delta})$ in (9). Euclidean distances are computed between the k_I-dimensional $\mathbf{y}_{\Phi_I}$ vectors as well as the k_E-dimensional $\mathbf{y}_{\Phi_E}$ vectors. Thus, roughly $2 \times (k_E + k_I)$ arithmetic operations are required for each similarity computation, avoiding repeated image differencing and projections:

$$\begin{aligned} P(\mathbf{\Delta}|\Omega_I) \;&=\; P(\mathbf{I} - \mathbf{I}_j|\Omega_I) \;=\; \frac{e^{-\|\mathbf{y}_{\Phi_I} - \mathbf{y}^j_{\Phi_I}\|^2/2}}{(2\pi)^{k_I/2}|\;\mathbf{\Sigma}_I\;|^{1/2}}\;, \\ P(\mathbf{\Delta}|\Omega_E) \;&=\; P(\mathbf{I} - \mathbf{I}_j|\Omega_E) \;=\; \frac{e^{-\|\mathbf{y}_{\Phi_E} - \mathbf{y}^j_{\Phi_E}\|^2/2}}{(2\pi)^{k_E/2}|\;\mathbf{\Sigma}_E\;|^{1/2}}\;. \end{aligned} \tag{12}$$

The *maximum likelihood* (ML) similarity matching is even simpler since only the intra-personal class is evaluated, leading to the following modified

form for the similarity measure

$$S'(\mathbf{\Delta}) \;=\; P(\mathbf{\Delta}|\Omega_I) \;=\; \frac{e^{-\|\mathbf{y}_{\Phi_I} - \mathbf{y}^j_{\Phi_I}\|^2/2}}{(2\pi)^{k_I/2}|\;\mathbf{\Sigma}_I\;|^{1/2}}\;. \tag{13}$$

The approach described above requires two projections of the difference vector $\mathbf{\Delta}$, from which likelihoods can be estimated for the Bayesian similarity measure. The computation flow is illustrated in Figure 4(b). The projection steps are linear while the posterior computation is nonlinear. Because of the double PCA projections required, this approach has been called a "dual eigenspace" technique. Note the projection of the difference vector $\mathbf{\Delta}$ onto the "dual eigenfaces" (Ω_I and Ω_E) for computation of the posterior in (9).

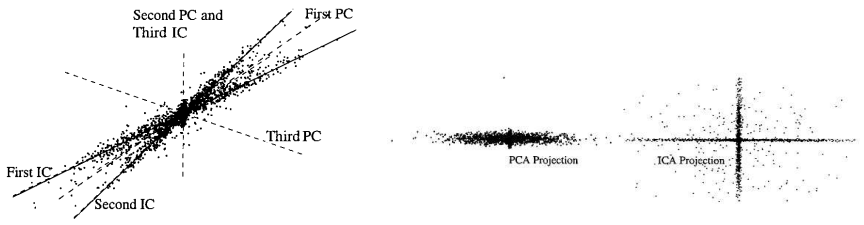
It is instructive to compare and contrast LDA (Fisherfaces) and the dual subspace Bayesian technique by noting the similar roles played by the between-class/within-class and extrapersonal/intrapersonal subspaces. However, there are key differences between the two techniques and LDA can in fact be viewed as a special case of the dual subspace Bayesian approach. One such analysis is presented in [39] wherein PCA, LDA and Bayesian matching are "unified" under a 3-parameter subspace approach and compared in terms of performance. Likewise, other experimental studies in recent years have shown that the intra/extra Bayesian matching technique out-performs LDA. One should bear in mind that ultimately the only optimal probabilistic justification for the use of LDA is for the case of two Guassian distributions of *equal* covariance (although LDA tends to perform well even when this condition is not strictly true). In contrast, the dual subspace Bayesian formulation is completely general and is probabilistic *by definition* and as such it makes no appeals to Gaussianity, geometry or the symmmetry of the underlying data or the two "meta-classes" (intra and extra). The intra/extra probability distributions can take on *any* form (eg. arbitrary mixture models) and not just single Gaussians – although the latter case does allow for easy visualization (by diagonalizing the dual covariances as two sets of "eigenfaces").

2.5 ICA & Source Separation

While PCA minimizes the sample covariance (second-order dependency) of the data, Independent Component Analysis (ICA) [18, 6] minimizes higher-order dependencies as well, and the components found by ICA are designed to be non-Gaussian. Like PCA, ICA also yields a linear projection $\mathbb{R}^N \rightarrow \mathbb{R}^M$ but with different properties:

$$\mathbf{x} \approx \mathbf{A}\mathbf{y}\;,\;\; \mathbf{A}^T\mathbf{A} \neq \mathbf{I}\;,\;\; P(\mathbf{y}) \approx \prod\; p(y_i)\;, \tag{14}$$

that is, approximate reconstruction, *non-orthogonality* of the basis $\mathbf{A}$ and the near factorization of the joint distribution $P(\mathbf{y})$ into marginal distributions of the (non-Gaussian) ICs.



(a) (b)

Fig. 5. ICA vs. PCA decomposition of a 3D data set. (a) the bases of PCA (orthogonal) and ICA (non-orthogonal). (b) Left: the projection of the data onto the top two principal components (PCA). Right: the projection onto the top two independent components (ICA). From [1].

An example of ICA basis is shown in Figure 5, where it is computed from a set of 3D points. The 2D subspace recovered by ICA appears to reflect the distribution of the data much better than the subspace obtained with PCA. Another example of an ICA basis is shown in Figure 8(b) where we see two unordered non-orthogonal IC vectors, one of which is roughly aligned with the first principal component vector in Figure 8(a) — *i.e.,* the direction of maximum variance. Note that the actual non-Gaussianity and statistical independence achieved in this toy example are minimal at best, and so is the success of ICA in recovering the principal modes of the data.

ICA is intimately related to the *blind source separation* problem: decomposition of the input signal (image) $\mathbf{x}$ into a linear combination (mixture) of independent source signals. Formally, the assumption is that $\mathbf{x}^T = \mathbf{A}\mathbf{s}^\mathbf{T}$, with $\mathbf{A}$ the unknown mixing matrix. ICA algorithms [5] try to find $\mathbf{A}$ or the *separating matrix* $\mathbf{W}$ such that $\mathbf{u}^T = \mathbf{W}\mathbf{x}^T = \mathbf{W}\mathbf{A}\mathbf{s}^T$. When the data consist of M observations with N variables, the input to ICA is arranged in an $N \times M$ matrix $\mathbf{X}$.

Bartlett *et al.* [1, 10] investigated the use of ICA framework for face recognition in two fundamentally different architectures:

Architecture I Rows of $\mathbf{S}$ are *independent basis images*, which combined by $\mathbf{A}$ yield the input images $\mathbf{X}$. Learning $\mathbf{W}$ allows to estimate the basis images in the rows of $\mathbf{U}$. In practice, for reasons of computational tractability, PCA is first performed on the input data $\mathbf{X}$ to find the top K eigenfaces; these are arranged in the columns of a matrix $\mathbf{E}$. [6] Then ICA is performed on $\mathbf{E}^T$ - that is, the images are variables, and the pixel values

[5] A number of algorithms exist, most notably Jade [5], InfoMax and FastICA [16].
[6] These Eigenfaces are linear combination of the original images, which under the assumptions of ICA should not affect the resulting decomposition.

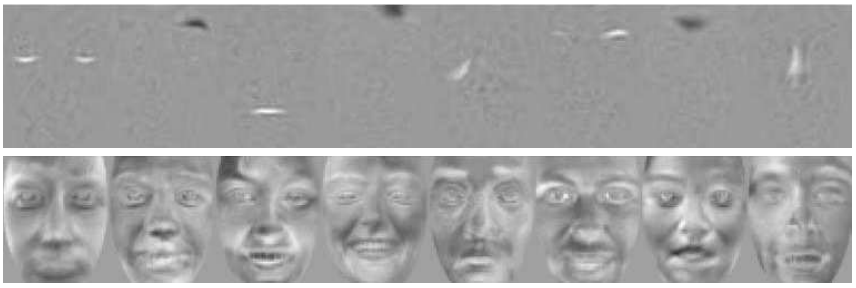


Fig. 6. Basis images of ICA: Architecture I (top) and II (bottom). From [10].

are observations. Let $\mathbf{C}$ be the PCA coefficient matrix, that is $\mathbf{X} = \mathbf{C}\mathbf{E}^T$. Then the k independent ICA basis images (Figure 6, top) are estimated by the rows of $\mathbf{U} = \mathbf{W}\mathbf{E}^T$, and the coefficients for the data are computed from $\mathbf{X} = \mathbf{E}\mathbf{W}^{-1}\mathbf{U}$.

Architecture II In this architecture algorithm assumes that the sources in $\mathbf{S}$ are independent coefficients, while the columns of the mixing matrix $\mathbf{A}$ are the basis images; that is, the variables in the source separation problem are the pixels. Similar to Architecture I, ICA is preceded by PCA; however, in this case the input to ICA is the coefficient matrix $\mathbf{C}$. The resulting ICA basis consists of the columns of $\mathbf{EA}$ (Figure 6, bottom), and the coefficients are found in the rows of $\mathbf{U} = \mathbf{W}\mathbf{C}^T$. These coefficients give the *factorial representation* of the data.

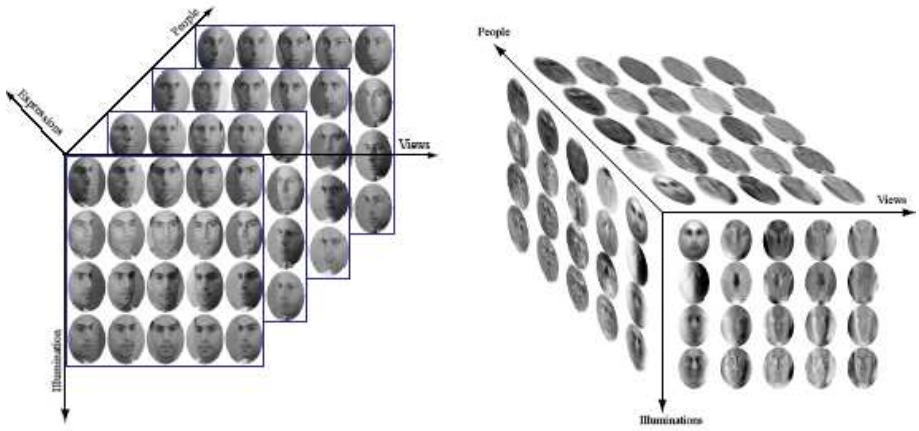
Generally, the bases obtained with Architecture I reflect more local properties of the faces, while the bases in Architecture II have global properties and much more resemble faces (see Figure 6).

2.6 Multi-Linear SVD: "Tensorfaces"

The linear analysis methods discussed above have been shown to be suitable when pose, illumination or expression are fixed across the face database. When any of these parameters is allowed to vary, the linear subspace representation does not capture this variation well (see Section 5.1). In Section 3 we discuss recognition with nonlinear subspaces. An alternative, *multi-linear* approach, called "Tensorfaces", has been proposed by Vasilescu and Terzopoulos in [38, 37].

Tensor is a multidimensional generalization of a matrix: a *n-order tensor* $\mathcal{A}$ is an object with n indices, with elements denoted by $a_{i_1,\ldots,i_n} \in \mathbb{R}$. Note that there are n ways to *flatten* this tensor, i.e. to rearrange the elements in a matrix: the i-th row of $\mathcal{A}_{(s)}$ is obtained by concatenating all the elements of $\mathcal{A}$ of the form $a_{i_1,\ldots,i_{s-1},i,i_{s+1},\ldots,i_n}$.

A generalization of matrix multiplication for tensors is the *l-mode* product $\mathcal{A} \times_l \mathbf{M}$ of a tensor $\mathcal{A}$ and an $m \times k$ matrix $\mathbf{M}$, where k is the l-th dimension



(a) (b)

Fig. 7. Tensorfaces. (a) The data tensor; the four dimensions visualized are identity, illumination, pose and the pixel vector. The fifth dimension corresponds to expression (only sub-tensor for neutral expression is shown). (b) The Tensorfaces decomposition. From [37].

of $\mathcal{A}$:

$$(\mathcal{A} \times_l \mathbf{M})_{i_1,\ldots i_{l-1},j,i_{l+1},\ldots i_n} = \sum_{i=1}^{k} a_{i_1,\ldots i_{l-1},i,i_{l+1},\ldots i_n} m_{ji}. \tag{15}$$

Under this definition, Vasilescu and Terzopoulos propose in [38] an algorithm they call *n-mode SVD*, that decomposes an n-dimensional tensor $\mathcal{A}$ into

$$\mathcal{A} = \mathcal{Z} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_n \mathbf{U}_n. \tag{16}$$

The role of the *core tensor* $\mathcal{Z}$ in this decomposition is similar to the role of the singular value matrix $\mathbf{\Sigma}$ in SVD (4): it governs the interactions between the *mode matrices* $\mathbf{U}_1, \ldots, \mathbf{U}_n$ which contain the orthonormal bases for the spaces spanned by the corresponding dimensions of the data tensor. The mode matrices can be obtained by flattening the tensor across the corresponding dimension and performing PCA on the columns of the resulting matrix; then the core tensor is computed as

$$\mathcal{Z} = \mathcal{A} \times_1 \mathbf{U}_1^T \times_2 \mathbf{U}_2^T \cdots \times_N \mathbf{U}_n^T.$$

The notion of tensor can be applied to a face image ensemble in the following way [38]: consider a set of N-pixel images of N_p people's faces, each photographed in N_v viewpoints, with N_i illuminations and N_e expressions. The entire set may be arranged in a $N_\mathrm{p} \times N_\mathrm{v} \times N_\mathrm{i} \times N_\mathrm{e} \times N$ tensor of order

5. Figure 7(a) illustrates this concept: only 4 dimensions are shown; to visualize the fifth one (expression), imagine that the four-dimensional tensors for different expressions are "stacked".

In this context, the face image tensor can be decomposed into

$$\mathcal{A} = \mathcal{Z} \times_1 \mathbf{U}_\mathrm{p} \times_2 \mathbf{U}_\mathrm{v} \times_3 \mathbf{U}_\mathrm{i} \times_4 \mathbf{U}_\mathrm{e} \times_5 \mathbf{U}_\mathrm{pixels}. \tag{17}$$

Each mode matrix represents a parameter of the object appearance. For example, the columns of the $N_\mathrm{e} \times N_\mathrm{e}$ matrix $\mathbf{U}_e$ span the space of expression parameters. The columns of $\mathbf{U}_\mathrm{pixels}$ span the image space; these are exactly the eigenfaces which would be obtained by direct PCA on the entire data set.

Every person in the database can be represented by a single N_p vector, which contains coefficients with respect to the bases comprising the tensor

$$\mathcal{B} = \mathcal{Z} \times_2 \mathbf{U}_\mathrm{v} \times_3 \mathbf{U}_\mathrm{i} \times_4 \mathbf{U}_\mathrm{e} \times_5 \mathbf{U}_\mathrm{pixels}.$$

For a given viewpoint v, illumination i and expression e, an $N_\mathrm{p} \times N$ matrix $\mathbf{B}_\mathrm{v,i,e}$ can be obtained by indexing into $\mathcal{B}$ for v, i, e and flattening the resulting $N_\mathrm{p} \times 1 \times 1 \times 1 \times N$ sub-tensor along the identity (people) mode. Now a training image $\mathbf{x}_\mathrm{p,v,e,i}$ of a person j under the given conditions can be written as

$$\mathbf{x}_\mathrm{p,v,e,i} = \mathbf{B}^T_\mathrm{v,i,e}\, \mathbf{c}_\mathrm{p}, \tag{18}$$

where $\mathbf{c}_j$ is the j-th row vector of $\mathbf{U}_\mathrm{p}$.

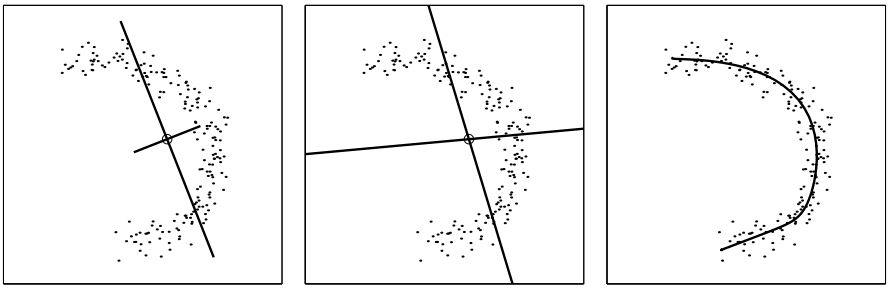
Given an input image $\mathbf{x}$, a candidate coefficient vector $\mathbf{c}_\mathrm{v,i,e}$ is computed for all combinations of viewpoint, expression and illumination, solving the equation in (18). The recognition is carried out by finding the value of j that yields the minimum Euclidean distance between $\mathbf{c}$ and the vectors $\mathbf{c}_j$ across all illuminations, expressions and viewpoints.[7]

In [38] the authors report experiments involving the data tensor consisting of images of $N_p = 28$ subjects photographed in $N_i = 3$ illumination conditions from $N_v = 5$ viewpoints, with $N_e = 3$ different expressions; the images were resized and cropped so that they contain $N = 7493$ pixels. The performance of TensorFaces is reported to be significantly better than that of standard Eigenfaces described in Section 2.1.

3 Nonlinear Subspaces

In this section we describe a number of modeling techniques for principal manifolds which are strictly nonlinear. We must emphasize that while the mathematics of these methods are readily applicable to all types of data, in practice one should always distinguish between the intrinsic nonlinearity of the data

[7] This technique can also be used to estimate the parameters (of illumination, etc.) associated with the variability of the input images.



(a) (b) (c)

Fig. 8. (a) PCA basis (linear, ordered and orthogonal) (b) ICA basis (linear, unordered and non-orthogonal), (c) Principal Curve (parameterized nonlinear manifold). The circle shows the data mean.

and the nonlinearity which arises due to the (improper) choice of parameterization. For example, object translation is linear but its visual representation (as spatially sampled in the image, for example) can be highly nonlinear. A judicious choice of the coordinate frame (very often an *ojbect-centered* one) will linearize the data manifold, thus obviating the need for computationally difficult and intractable nonlinear modeling techniques. Therefore, whenever possible one should seek the "right" parameterization for a given problem.

3.1 Principal Curves and Nonlinear PCA

The defining property of nonlinear principal manifolds is that the *inverse image* of the manifold in the original space $\mathbb{R}^N$ is a nonlinear (curved) lower-dimensional surface that "passes through the middle of the data" while minimizing the sum total distance between the data points and their projections on that surface. Often referred to as *principal curves* [14], this formulation is essentially a nonlinear regression on the data. An example of a principal curve is shown in Figure 8(c).

One of the simplest methods for computing nonlinear principal manifolds is the nonlinear PCA (NLPCA) auto-encoder multi-layer neural network [20, 9] shown in Figure 9. The so-called "bottleneck" layer forms a lower-dimensional manifold representation by means of a nonlinear *projection* function $f(\mathbf{x})$, implemented as a weighted sum-of-sigmoids. The resulting principal components $\mathbf{y}$ have an inverse mapping with a similar nonlinear *reconstruction* function $g(\mathbf{y})$, which reproduces the input data as accurately as possible. The NLPCA computed by such a multi-layer sigmoidal neural network is equivalent — with certain exceptions[8] — to a *principal surface* under the more general definition

[8] The class of functions attainable by this neural network restricts the projection function $f()$ to be smooth and differentiable, hence suboptimal in some cases [22].

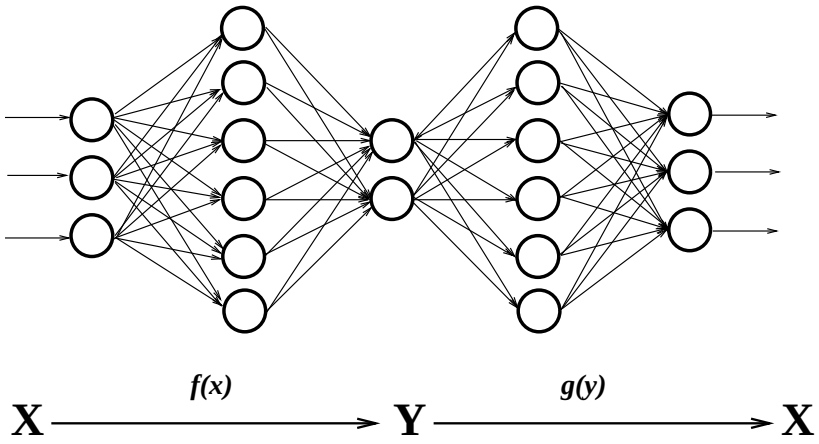


Fig. 9. An auto-associative ("bottleneck") neural network for computing principal manifolds $\mathbf{y} \in \mathbb{R}^k$ in the input space $\mathbf{x} \in \mathbb{R}^N$.

[13, 14]. To summarize, the main properties of NLPCA are:

$$\mathbf{y} = f(\mathbf{x}) \ , \ \mathbf{x} \approx g(\mathbf{y}) \ , \ P(\mathbf{y}) = ? \tag{19}$$

corresponding to nonlinear projection, approximate reconstruction and typically no prior knowledge regarding the joint distribution of the components, respectively (however, see Zemel [43] for an example of devising suitable priors in such cases). The principal curve in Figure 8(c) was generated with a 2-4-1-4-2 layer neural network of the type shown in Figure 9. Note how the principal curve yields a compact and relatively accurate representation of the data, in contrast to the linear models (PCA and ICA).

3.2 Kernel-PCA and Kernel-Fisher Methods

Recently nonlinear principal component analysis has been revived with the "kernel eigenvalue" method of Schölkopf *et al.* [32]. The basic methodology of KPCA is to apply a nonlinear mapping to the input $\Psi(\mathbf{x}) : \mathbb{R}^N \rightarrow \mathbb{R}^L$ and then solve for a linear PCA in the resulting feature space $\mathbb{R}^L$, where L is larger than N and possibly infinite. Because of this increase in dimensionality, the mapping $\Psi(\mathbf{x})$ is made implicit (and economical) by the use of kernel functions satisfying Mercer's theorem [7]

$$k(\mathbf{x_i}, \mathbf{x_j}) \ = \ (\Psi(\mathbf{x_i}) \cdot \Psi(\mathbf{x_j})), \tag{20}$$

where kernel evaluations $k(\mathbf{x_i}, \mathbf{x_j})$ in the input space correspond to dot-products in the higher dimensional feature space. Because computing covariance is based on dot-products, performing a PCA in the feature space can be formulated with kernels in the input space without the explicit (and possibly

prohibitively expensive) direct computation of $\Psi(\mathbf{x})$. Specifically, assuming that the projection of the data in feature space is zero-mean ("centered"), the covariance is given by

$$\mathbf{\Sigma}_K \;=\; < \Psi(\mathbf{x_i}) \, , \Psi(\mathbf{x_i})^T > \tag{21}$$

with the resulting eigenvector equation $\lambda \mathbf{V} = \Sigma_K \mathbf{V}$. Since the eigenvectors (columns of $\mathbf{V}$) must lie in the span of the training data $\Psi(\mathbf{x_i})$, it must be true that for each training point

$$\lambda(\Psi(\mathbf{x_i}) \cdot \mathbf{V}) \;=\; (\Psi(\mathbf{x_i}) \cdot \Sigma_K \mathbf{V}) \quad \text{for } i = 1, ... \, T \, , \tag{22}$$

and that there must exist coefficients $\{w_i\}$ such that

$$\mathbf{V} \;=\; \sum_{i=1}^{T} w_i \Psi(\mathbf{x_i}) \, . \tag{23}$$

Using the definition of Σ_K, substituting the above equation into (22) and defining the resulting T-by-T matrix $\mathbf{K}$ by $\mathbf{K}_{ij} = (\Psi(\mathbf{x_i}) \cdot \Psi(\mathbf{x_j}))$ leads to the equivalent eigenvalue problem formulated in terms of kernels in the input space:

$$T\lambda \mathbf{w} \;=\; \mathbf{K}\mathbf{w} \, , \tag{24}$$

where $\mathbf{w} = (w_1, ..., w_T)^T$ is the vector of expansion coefficients of a given eigenvector $\mathbf{V}$ as defined in (23). The kernel matrix $\mathbf{K}_{ij} = k(\mathbf{x_i}, \mathbf{x_j})$ is then diagonalized with a standard PCA[9]. Orthonormality of the eigenvectors, $(\mathbf{V}^n \cdot \mathbf{V}^n) = 1$, leads to the equivalent normalization of their respective expansion coefficients, $\lambda_n(\mathbf{w}^n \cdot \mathbf{w}^n) = 1$.

Subsequently, the KPCA principal components of any input vector can be efficiently computed with simple kernel evaluations against the dataset. The n-th principal component y_n of $\mathbf{x}$ is given by

$$y_n \;=\; (\mathbf{V}_n \cdot \Psi(\mathbf{x})) \;=\; \sum_{i=1}^{T} \; w_i^n \; k(\mathbf{x}, \mathbf{x_i}), \tag{25}$$

where $\mathbf{V}_n$ is the n-th eigenvector of the feature space defined by Ψ. As with PCA, the eigenvectors $\mathbf{V}_n$ can be ranked by decreasing order of their eigenvalues λ_n and an d-dimensional manifold projection of $\mathbf{x}$ is $\mathbf{y} = (y_1, ..., y_d)^T$, with individual components defined by (25).

A significant advantage of KPCA over neural network and principal curves is that KPCA does not require nonlinear optimization, is not subject to overfitting and does not require prior knowledge of network architecture or the number of dimensions. Furthermore, unlike traditional PCA, one can use more

[9] However, computing Σ_K in (21) requires "centering" the data by computing the mean of $\Psi(\mathbf{x_i})$. However, since there is no explicit computation of $\Psi(\mathbf{x_i})$, the covariance matrix $\mathbf{K}$ must be centered instead (for details see [32]).

eigenvector projections than the input dimensionality of the data (since KPCA is based on the matrix $\mathbf{K}$, the number of eigenvectors or features available is T). On the other hand, the selection of the optimal kernel (and its associated parameters) remains an "engineering problem." Typical kernels include Gaussians $\exp(-\|\mathbf{x_i} - \mathbf{x_j}\|)^2/\sigma^2)$, polynomials $(\mathbf{x_i} \cdot \mathbf{x_j})^d$ and sigmoids $\tanh(a(\mathbf{x_i} \cdot \mathbf{x_j}) + b)$, all of which satisfy Mercer's theorem [7].

Similar to the derivation of KPCA, one may extend the Fisherfaces method (see Section 2.3) by applying the FLD in the feature space. In [42] Yang derives the Kernel Fisherfaces algorithm, that maximizes the between-scatter to within-scatter ratio in the feature space through the use of the kernel matrix $\mathbf{K}$. In experiments on two data sets that contained images from 40 and 11 subjects, respectively, with varying pose, scale and illumination, this algorithm showed performance clearly superior to that of ICA, PCA and KPCA and somewhat better than that of the standard Fisherfaces.

4 Empirical Comparison of Subspace Methods

In [23] Moghaddam reports on an extensive evaluation of many of the subspace methods described above on a large subset of FERET dataset [31] (see also Chapter 13). The experimental data consisted of a training "gallery" of 706 individual FERET faces and 1,123 "probe" images containing one or more views of every person in the gallery. All these images were aligned and normalized as described in [27]. The multiple probe images reflected different expressions, lighting and with glasses on/off, *etc.* The study compared the Bayesian approach described in Section 2.4 to a number of other techniques, and tested the limits of the recognition algorithms with respect to image resolution or equivalently the amount of visible facial detail: since the Bayesian algorithm was independently evaluated in DARPA's 1996 FERET face recognition competition [31] with medium resolution images (84-by-44 pixels) — achieving an accuracy of $\approx 95\%$ on $O(10^3)$ individuals — it was decided to lower the resolution (the number of pixels) by a factor 16. Therefore, the aligned faces in the dataset were downsampled to 21-by-12 pixels, yielding input vectors in a $\mathbb{R}^{N=252}$ space. Several examples are shown in Figures 10(a) and 10(b).

The reported results were obtained with a 5-fold Cross-Validation (CV) analysis. The total dataset of 1829 faces (706 unique individuals and their collective 1123 probes) was randomly partitioned into 5 subsets with unique (non-overlapping) individuals and their associated probes. Each subset contained both gallery and probe images of ≈ 140 unique individuals. For each of the 5 subsets, the recognition task was correctly matching the multiple probes to the ≈ 140 gallery faces using the other 4 subsets as training data. Note that with $N = 252$ and using 80% of the entire dataset for training, there are nearly 3 times as many training samples than the data dimensionality, thus

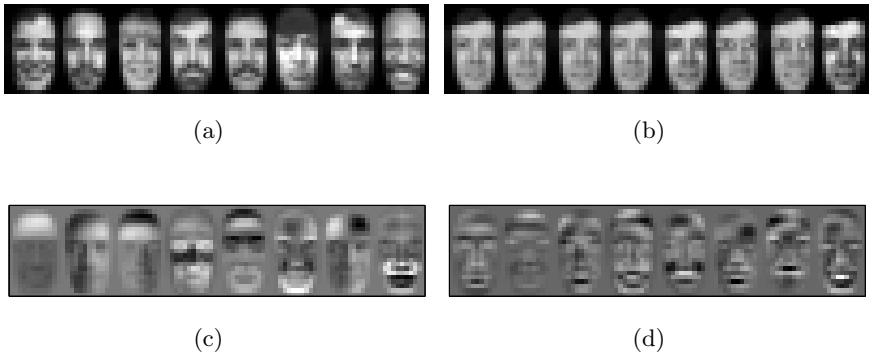


Fig. 10. Experiments on FERET data. (a) Several faces from the gallery. (b) Multiple probes for one individual, with different facial expressions, eye-glasses, variable ambient lighting and image contrast, etc. (c) Eigenfaces. (d) ICA basis images.

parameter estimations (for PCA, ICA, KPCA and the Bayesian method) were properly over-constrained.

The resulting 5 experimental trials were pooled to compute the mean and standard deviation of the recognition rates for each method. The fact that the training and testing sets had no overlap in terms of individual identities led to an evaluation of the algorithms' *generalization* performance — the ability to recognize new individuals which were not part of the manifold computation or density modeling with the training set.

The baseline recognition experiments used a default manifold dimensionality of $k = 20$. This choice of k was made for two reasons: it led to a reasonable PCA reconstruction error of MSE = 0.0012 (or 0.12% per pixel with a normalized intensity range of [0,1]) and a baseline PCA recognition rate of $\approx 80\%$ (on a different 50/50 partition of the dataset) thus leaving a sizeable margin for improvement. Note that since the recognition experiments were essentially a 140-way classification task, chance performance was approximately 0.7%.

4.1 PCA-based Recognition

The baseline algorithm for these face recognition experiments was standard PCA (Eigenface) matching. The first 8 principal eigenvectors computed from a single partition are shown in Figure 10(c). Projection of the test set probes onto the 20-dimensional linear manifold (computed with PCA on the training set only) followed by nearest-neighbor matching to the ≈ 140 gallery images using a Euclidean metric yielded a mean recognition rate of 77.31% with the highest rate achieved being 79.62% as shown in Table 1. The full image-vector nearest-neighbor (template matching) — *i.e.,* on $\mathbf{x} \in \mathbb{R}^{252}$ — yielded a recognition rate of 86.46% (see dashed line in Figure 11). Clearly, performance is degraded by the $252 \rightarrow 20$ dimensionality reduction, as expected.

4.2 ICA-based Recognition

For ICA-based recognition (Architecture II, see Section 2.5) two different algorithms based on 4th-order cumulants were tried: the "JADE" algorithm of Cardoso [5] and the fixed-point algorithm of Hyvärinen & Oja [15]. In both algorithms a PCA whitening step ("sphering") preceded the core ICA decomposition. The corresponding *non-orthogonal* JADE-derived ICA basis is shown in Figure 10(d). Similar basis faces were obtained with Hyvärinen's method. These basis faces are the columns of the matrix $\mathbf{A}$ in (14) and their linear combination (specified by the ICs) reconstructs the training data. The ICA manifold projection of the test set was obtained using $\mathbf{y} = \mathbf{A}^{-1}\mathbf{x}$. Nearest-neighbor matching with ICA using Euclidean L_2 norm resulted in a mean recognition rate of 77.30% with the highest rate being 82.90% as shown in Table 1. We found little difference between the two ICA algorithms and noted that ICA resulted in the largest performance variation in the 5 trials (7.66% std. dev.). Based on the mean recognition rates it is unclear whether ICA provides a systematic advantage over PCA and whether "more non-Gaussian" and/or "more independent" components result in a better manifold for *recognition* purposes with this dataset.

Note that the experimental results of Bartlett *et al.* [1] with FERET faces did favor ICA over PCA. This seeming disagreement can be reconciled if one considers the differences in the experimental setup and in the choice of the similarity measure. First, the advantage of ICA was seen primarily with more difficult time-separated images. In addition, compared to [1] the faces in this experiment were cropped much tighter, leaving no information regarding hair and face shape, and also were much lower in resolution; factors which when combined make the recognition task much harder.

The second factor is the choice of the distance function used to measure similarity in the subspace. This matter was further investigated by Draper *et al.* in [10]. They found that the best results for ICA are obtained using the cosine distance, while for Eigenfaces the L_1 metric appears to be optimal; with L_2 metric, which was also used in the experiments in [23], the performance of ICA (Architecture II) was very similar to that of Eigenfaces.

4.3 KPCA-based Recognition

For KPCA, the parameters of Gaussian, polynomial and sigmoidal kernels were first fine-tuned for best performance with a different 50/50 partition validation set, and Gaussian kernels were found to be the best for this dataset. For each trial, the kernel matrix was computed from the corresponding training data. Both the test set gallery and probes were projected onto the kernel eigenvector basis (25) in order to obtain the nonlinear principal components which were then used in nearest-neighbor matching of test set probes against the test set gallery images. The mean recognition rate was found to be 87.34%

Table 1. Recognition accuracies (in %) with $k = 20$ subspace projections using 5-fold Cross-Validation.

Partition	PCA	ICA	KPCA	Bayes
1	78.00	82.90	83.26	**95.46**
2	79.62	77.29	92.37	**97.87**
3	78.59	79.19	88.52	**94.49**
4	76.39	82.84	85.96	**92.90**
5	73.96	64.29	86.57	**93.45**
Mean	77.31	77.30	87.34	**94.83**
Std. Dev.	2.21	7.66	3.39	1.96

Table 2. Comparison of various techniques across multiple attributes ($k = 20$).

	PCA	ICA	KPCA	Bayes
Accuracy	77%	77%	87%	95%
Complexity	10^8	10^9	10^9	10^8
Uniqueness	yes	no	yes	yes
Projections	linear	linear	nonlinear	linear

with the highest rate being 92.37% as shown in Table 1. The standard deviation of the KPCA trials was slightly higher (3.39) than that of PCA (2.21) but Figure 11 indicates that KPCA does in fact do better than both PCA and ICA, hence justifying the use of nonlinear feature extraction.

4.4 MAP-based Recognition

For Bayesian similarity matching, appropriate training $\mathbf{\Delta}$s for the two classes Ω_I (Figure 10(b)) and Ω_E (Figure 10(a)a) were used for the dual PCA-based density estimates $P(\mathbf{\Delta}|\Omega_I)$ and $P(\mathbf{\Delta}|\Omega_E)$, which were both modeled as single Gaussians with subspace dimensions of k_I and k_E, respectively. The total subspace dimensionality k was divided evenly between the two densities by setting $k_I = k_E = k/2$ for modeling.[10]

With $k = 20$, Gaussian subspace dimensions of $k_I = 10$ and $k_E = 10$ were used for $P(\mathbf{\Delta}|\Omega_I)$ and $P(\mathbf{\Delta}|\Omega_E)$, respectively. Note that $k_I + k_E = 20$, thus matching the total number of projections used with the 3 principal manifold techniques. Using the maximum *a posteriori* (MAP) similarity in (9), the Bayesian matching technique yielded a mean recognition rate of 94.83% with the highest rate achieved being 97.87% as shown in Table 1. The standard

[10] In practice, $k_I > k_E$ yields good results. In fact as $k_E \to 0$, one obtains a maximum-likelihood similarity $S = P(\mathbf{\Delta}|\Omega_I)$ with $k_I = k$, which for this dataset is only few percent less accurate than MAP [24].

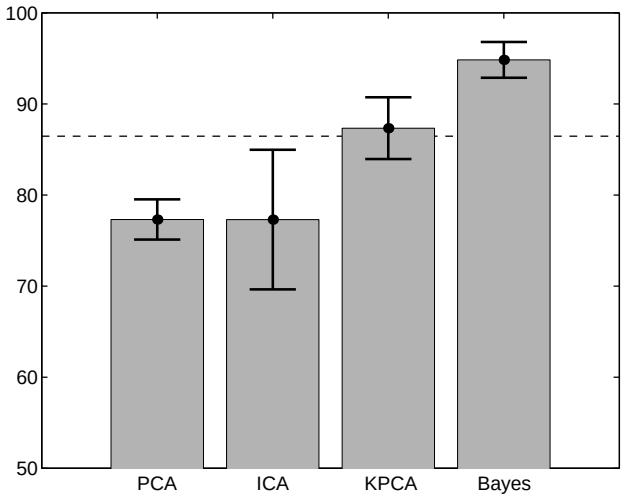


Fig. 11. Recognition performance of PCA, ICA, and KPCA manifolds vs. Bayesian (MAP) similarity matching with a $k = 20$ dimensional subspace (dashed line is performance of nearest-neighbor matching with the full-dimensional image vectors).

deviation of the 5 partitions for this algorithm was also the lowest (1.96) — see Figure 11.

4.5 Compactness of Manifolds

The performance of different methods with different size manifolds can be compared by plotting their recognition rates $R(k)$ as a function of the first k principal components. For the manifold matching techniques, this simply means using a subspace dimension of k (the first k components of PCA/ICA/KPCA), whereas for the Bayesian matching technique this means that the subspace Gaussian dimensions should satisfy $k_I + k_E = k$. Thus all methods used the same number of subspace projections. This test was the premise for one of the key points investigated in [23]: given the *same* number of subspace projections, which of these techniques is better at data modeling and subsequent recognition? The presumption being that the one achieving the highest recognition rate with the smallest dimension is preferred.

For this particular dimensionality test, the total dataset of 1829 images was partitioned (split) in half: a training set of 353 gallery images (randomly selected) along with their corresponding 594 probes and a testing set containing the remaining 353 gallery images and their corresponding 529 probes. The training and test sets had no overlap in terms of individuals' identities. As in the previous experiments, the test set probes were matched to the test set gallery images based on the projections (or densities) computed with the

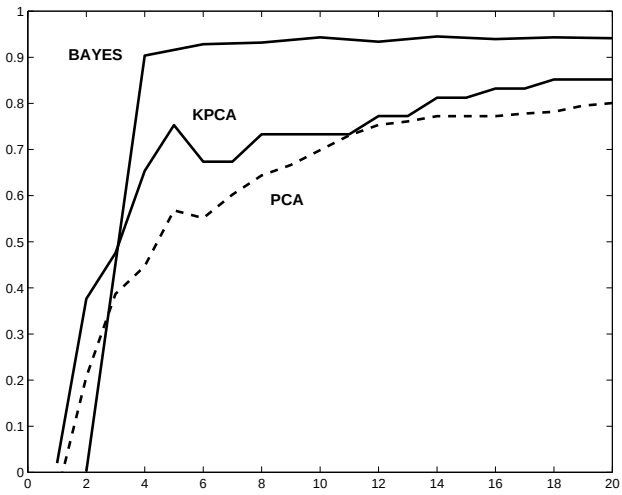


Fig. 12. Recognition accuracy $R(k)$ of PCA, KPCA and Bayesian similarity with increasing dimensionality k of the principal subspace (ICA results, not shown, are similar to PCA).

training set. The results of this experiment are shown in Figure 12 which plots the recognition rates as a function of the dimensionality of the subspace k. This is a more revealing comparison of the relative performance of the different methods since *compactness* of the manifolds — defined by the lowest acceptable value of k — is an important consideration in regards to both generalization error (over-fitting) and computational requirements.

4.6 Performance of Manifolds

The relative performance of the principal manifold techniques and Bayesian matching is summarized in Table 1 and Figure 11. The advantage of probabilistic matching over metric matching on both linear and nonlinear manifolds is quite evident (≈18% increase over PCA and ≈8% over KPCA). Note that the dimensionality test results in Figure 12 indicate that KPCA out-performs PCA by a ≈10% margin, and even more so with only few principal components (a similar effect is reported by Schölkopf [32] where KPCA out-performs PCA in low-dimensional manifolds). However, Bayesian matching achieves ≈90% with only four projections — 2 for each $P(\mathbf{\Delta}|\Omega)$ — and dominates both PCA and KPCA throughout the entire range of subspace dimensions in Figure 12.

A comparison of the subspace techniques with respect to multiple criteria is shown in Table 2. Note that PCA, KPCA and the dual subspace density estimation are uniquely defined for a given training set (making experimental comparisons repeatable), whereas ICA is not unique due to the variety

of different techniques used to compute the basis and the iterative (stochastic) optimizations involved. Considering the relative computation (of training), KPCA required $\approx 7 \times 10^9$ floating-point operations compared to PCA's $\approx 2 \times 10^8$ operations. On the average, ICA computation was one order of magnitude larger than PCA. Since the Bayesian similarity method's learning stage involves two separate PCAs, its computation is merely twice that of PCA (the same order of magnitude).

Considering its significant performance advantage (at low subspace dimensionality) and its relative simplicity, the dual-eigenface Bayesian matching method is a highly effective subspace modeling technique for face recognition. In independent FERET tests conducted by the US Army Laboratory [31], the Bayesian similarity technique out-performed PCA and other subspace techniques such as Fisher's Linear Discriminant (by a margin of at least 10%). Experimental results described above show that a similar recognition accuracy can be achieved using mere "thumbnails" with 16 times fewer pixels than in the images used in the FERET test. These results demonstrate the Bayesian matching technique's robustness with respect to image resolution, thus revealing the surprisingly small amount of facial detail that is required for high accuracy performance with this learning technique.

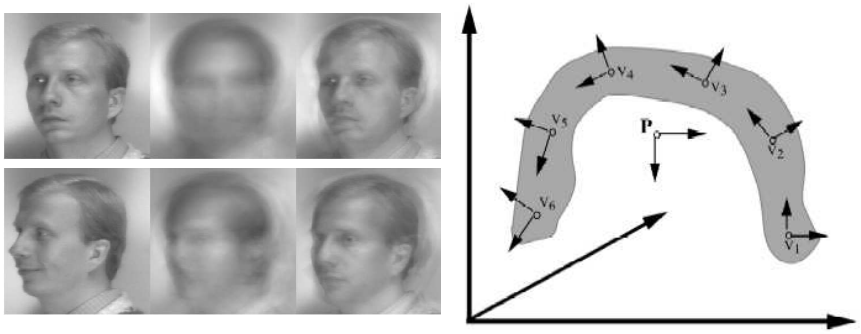
5 Methodology and Usage

In this section we discuss issues that require special care from the practitioner, in particular, the approaches designed to handle database with varying imaging conditions. We also present a number of extensions and modifications of the subspace methods.

5.1 Multi-View Approach for Pose

The problem of face recognition under general viewing conditions (change in pose) can also be approached using an eigenspace formulation. There are essentially two ways of approaching this problem using an eigenspace framework. Given M individuals under C different views, one can do recognition and pose estimation in a universal eigenspace computed from the combination of MC images. In this way, a single parametric eigenspace will encode both identity as well as pose. Such an approach, for example, has been used by Murase and Nayar [28] for general 3D object recognition.

Alternatively, given M individuals under C different views, we can build a view-based set of C distinct eigenspaces, each capturing the variation of the M individuals in a common view. The view-based eigenspace is essentially an extension of the eigenface technique to multiple sets of eigenvectors, one for each combination of scale and orientation. One can view this architecture as a set of parallel observers, each trying to explain the image data with their set of eigenvectors. In this view-based, multiple-observer approach, the first step is



(a) (b)

Fig. 13. Parametric vs. view-based eigenspace methods. (a) Reconstructions of the input image (left) with parametric (middle) and view-based (right) eigenspaces. Top - training image, bottom - novel (test) image. (b) Schematic illustration of the difference in the way the two approaches span the manifold.

to determine the location and orientation of the target object by selecting the eigenspace which best describes the input image. This can be accomplished by calculating the likelihood estimate using each viewspace's eigenvectors and then selecting the maximum.

The key difference between the view-based and parametric representations can be understood by considering the geometry of face space, schematically illustrated in Figure 13(b). In the high-dimensional vector space of an input image, multiple-orientation training images are represented by a set of C distinct regions, each defined by the scatter of M individuals. Multiple views of a face form non-convex (yet connected) regions in image space [3]. Therefore, the resulting ensemble is a highly complex and nonseparable manifold.

The parametric eigenspace attempts to describe this ensemble with a projection onto a single low-dimensional linear subspace (corresponding to the first k eigenvectors of the MC training images). In contrast, the view-based approach corresponds to C independent subspaces, each describing a particular region of the face space (corresponding to a particular view of a face); The principal manifold $\mathbf{v}_c$ of each region c is extracted separately. The relevant analogy here is that of modeling a complex distribution by a single cluster model or by the union of several component clusters. Naturally, the latter (view-based) representation can yield a more accurate representation of the underlying geometry.

This difference in representation becomes evident when considering the quality of reconstructed images using the two different methods. Fig. 13 compares reconstructions obtained with the two methods when trained on images

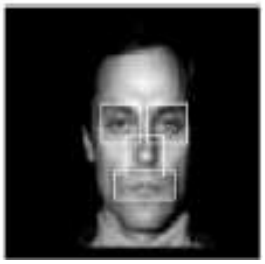
Fig. 14. An example of multi-view face image data used in the experiments described in Section 5.1. From [27].

of faces at multiple orientations. In the top row of Fig. 13(a), we see first an image in the training set, followed by reconstructions of this image using, first, the parametric eigenspace, and then, the view-based eigenspace. Note that in the parametric reconstruction, neither the pose nor the identity of the individual is adequately captured. The view-based reconstruction, on the other hand, provides a much better characterization of the object. Similarly, in the bottom row of Fig. 13(a), we see a novel view ($+68^o$) with respect to the training set (-90^o to $+45^o$). Here, both reconstructions correspond to the nearest view in the training set ($+45^o$), but the view-based reconstruction is seen to be more representative of the individual's identity. Although the quality of the reconstruction is not a direct indicator of the recognition power, from an information- theoretic point-of-view, the multiple eigenspace representation is a more accurate representation of the signal content.

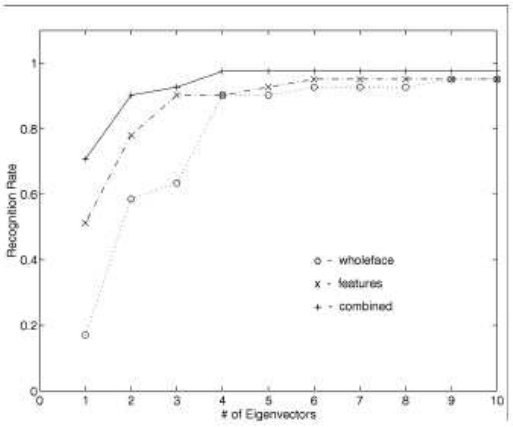
In [27] the view-based approach was evaluated on data similar to that shown in Fig. 14 that consisted of 189 images — nine views of 21 people. The viewpoints were evenly spaced from -90^o to $+90^o$ along the horizontal plane. In the first series of experiments, the interpolation performance was tested by training on a subset of the available views 90^o, 45^o, 0^o and testing on the intermediate views 68^o, 23^o. A 90 percent average recognition rate was obtained. A second series of experiments tested the extrapolation performance by training on a range of views (e.g., -90^o to $+45^o$) and testing on novel views outside the training range (e.g., $+68^o$ and $+90^o$). For testing views separated by 23^o from the training range, the average recognition rates were 83 percent. For 45^o testing views, the average recognition rates were 50 percent.

5.2 Modular Recognition

The Eigenface recognition method is easily extended to facial features [30], as shown in Figure 15(a). This leads to an improvement in recognition performance by incorporating an additional layer of description in terms of facial



(a)



(b)

Fig. 15. Modular eigenspaces. (a) The rectangular patches whose appearance is modeled with Eigenfeatures. (b) Performance of Eigenfaces, Eigenfeatures and the layered combination of both as a function of subspace dimension. From [30]

features. This can be viewed as either a modular or layered representation of a face, where a coarse (low-resolution) description of the whole head is augmented by additional (higher resolution) details in terms of salient facial features. Pentland *et al.* [30] called the latter component *Eigenfeatures*. The utility of this layered representation (Eigenface plus Eigenfeatures) was tested on a small subset of a large face database: a representative sample of 45 individuals with two views per person, corresponding to different facial expressions (neutral vs. smiling). This set of images was partitioned into a training set (neutral) and a testing set (smiling). Since the difference between these particular facial expressions is primarily articulated in the mouth, this feature was discarded for recognition purposes.

Fig. 15(b) shows the recognition rates as a function of the number of eigenvectors for Eigenface-only, Eigenfeature only, and the combined representation. What is surprising is that (for this small dataset at least) the Eigenfeatures alone were sufficient in achieving an (asymptotic) recognition rate of 95 percent (equal to that of the Eigenfaces).

More surprising, perhaps, is the observation that in the lower dimensions of eigenspace, Eigenfeatures outperformed the Eigenface recognition. Finally, by using the combined representation, one gains a slight improvement in the asymptotic recognition rate (98 percent). A similar effect was reported by Brunelli and Poggio [4], where the cumulative normalized correlation scores

of templates for the face, eyes, nose, and mouth showed improved performance over the face-only templates.

A potential advantage of the Eigenfeature layer is the ability to overcome the shortcomings of the standard Eigenface method. A pure eigenface recognition system can be fooled by gross variations in the input image (hats, beards, etc.). However, the feature-based representation may still find the correct match by focusing on the characteristic non-occluded features, e.g. the eyes and the nose.

5.3 Recognition with Sets

An interesting recognition paradigm involves the scenario in which the input consists not of a single image but of a *set* of images of an unknown person. The set may consist of a contiguous *sequence* of frames from a video, or of a non-contiguous, and perhaps unordered, set of photographs, extracted from a video or obtained from a individual snapshots. The former case is discussed in Chapter 8 (recognition from video). In the latter case, which we consider here, no temporal information is available. A possible approach, and in fact the one often taken until recently, has been to apply standard recognition methods to every image in the input set, and then combine the results - typically, by means of voting.

However, a large set of images contains more information than every individual image in it: it provides a clue not only on possible appearance on one's face, but also on the typical patterns of variation. Technically, just as a set of images known to contain an individual's face allows one to represent that individual by an estimated intrinsic subspace, so the unlabeled input set leads to a subspace estimate that represents the unknown subject. The recognition task can then be formulated in terms of matching the subspaces.

One of the first approaches to this task has been the Mutual Subspace Method (MSM) [41] which extracts the principal linear subspace of fixed dimension (via PCA), and measures the distance between subspaces by means of *principal angles* – the minimal angle between any two vectors in the subspaces. MSM has the desirable feature that it builds a compact model of the distribution of observations. However, it ignores important statistical characteristics of the data, since the eigenvalues corresponding to the principal components, as well as the means of the samples, are disregarded in the comparison. Thus its decisions may be statistically sub-optimal.

A probabilistic approach to measuring subspace similarity has been proposed in [33]. The underlying statistical model assumes that images of the j-th person's face have probability density p_j; the density of the unknown subject's face is denoted by p_0. The task of the recognition system is then to find the class label j^* satisfying

$$j^* = \operatorname*{argmax}_j \Pr(p_0 = p_j), \tag{26}$$

Therefore, given a set of images distributed by p_0, solving (26) amounts to optimally choosing between M hypotheses of the form which in statistics is sometimes referred to as the two-sample hypothesis: that two sets of examples come from the same distribution. A principled way of solving this task is to choose the hypothesis j for which the *Kullback-Leibler divergence* between p_0 and p_j is minimized.

In reality the distributions p_j are unknown and need to be estimated from data, as well as p_0. Shakhnarovich *et al.* model these distributions as Gaussians (one per subject), which are estimated according to the method described in Section 2.2 above; the KL divergence is then computed in closed form. In the experiments reported in [33], this method significantly outperforms the MSM.

Modeling the distributions by a single Gaussian is somewhat limiting; in [40], Wolf and Shashua extend this approach and propose a non-parametric discriminative method: *kernel principal angles*. They devise a positive definite kernel that operates on pairs of data matrices by projecting the data (columns) into a feature space of arbitrary dimension, in which principal angles can be calculated by computing inner products between the examples (i.e., application of the kernel). Note that this approach corresponds to nonlinear subspace analysis in the original space; for instance, one can use polynomial kernels of arbitrary degree. In experiments that included face recognition task on a set of 9 subjects, this method significantly outperformed both MSM and the Gaussian-based KL-divergence model of [33].

6 Conclusion

Subspace methods have been shown to be highly successful in face recognition, as they have in many other vision tasks. The exposition in this chapter roughly follows the chronological order in which these methods have evolved. Two most notable directions in this evolution can be discerned: the transition from linear to general, possibly non-linear and disconnected manifolds; and the introduction of probabilistic and specifically Bayesian methods for dealing with the uncertainty and with similarity. All of these methods share the same core assumption: that such ostensibly complex visual phenomena such as images of human faces, represented in a high-dimensional measurement space, are often intrinsically low-dimensional. Exploiting this low dimensionality allows a face recognition system to simplify computations and to focus the attention on the features of the data relevant for the identity of a person.

Acknowledgements

We would like to thank M. S. Bartlett and M. A. O. Vasilescu for use of figures from their papers and also for their helpful comments. We also would like to acknowledge all those who contributed to the research described in this chapter.

References

1. M. S. Bartlett, H. M. Lades, and T. J. Sejnowski. Independent component representations for face recognition. In *Proceedings of the SPIE: Conference on Human Vision and Electronic Imaging III*, volume 3299, pages 528–539, 1998.
2. V. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, July 1997.
3. M. Bichsel and A. Pentland. Human face recognition and the face image set's topology. *CVGIP: Image Understanding*, 59(2):254–261, 1994.
4. R. Brunelli and T. Poggio. Face recognition: Features vs. templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, 1993.
5. J.-F. Cardoso. High-order contrasts for independent component analysis. *Neural Computation*, 11(1):157–192, 1999.
6. P. Comon. Independent component analysis - a new concept? *Signal Processing*, 36:287–314, 1994.
7. R. Courant and D. Hilbert. *Methods of Mathematical Physics*, volume 1. Interscience, New-York, 1953.
8. M. Cover and J. Thomas. *Elements of Information Theory*. John Wiley & Sons, New York, 1994.
9. D. DeMers and G. Cottrell. Nonlinear dimensionality reduction. In *Advances in Neural Information Processing Systems*, pages 580–587. Morgan Kaufmann, 1993.
10. B. A. Draper, K. Baek, M. S. Bartlett, and J. R. Beveridge. Recognizing faces with PCA and ICA. *Computer Vision and Image Understanding*, 91(1–2):115–137, July/Aug. 2003.
11. K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, second edition, 1990.
12. J. J. Gerbrands. On the relationships between SVD, KLT and PCA. *Pattern Recognition*, 14:375–381, 1981.
13. T. Hastie. *Principal Curves and Surfaces*. PhD thesis, Stanford University, 1984.
14. T. Hastie and W. Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84(406):502–516, 1989.
15. A. Hyvärinen and E. Oja. A family of fixed-point algorithms for independent component analysis. Technical Report A40, Helsinki University of Technology, 1996.
16. A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4-5):411–430, 2000.
17. I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1986.
18. C. Jutten and J. Herault. Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24:1–10, 1991.
19. M. Kirby and L. Sirovich. Application of the karhunen-loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):103–108, Jan. 1990.
20. M. A. Kramer. Nonlinear principal components analysis using autoassociative neural networks. *AIChE Journal*, 32(2):233–243, 1991.
21. M. M. Loève. *Probability Theory*. Van Nostrand, Princeton, 1955.

22. E. C. Malthouse. Some theoretical results on nonlinear principal component analysis. Technical report, Northwestern University, 1998.
23. B. Moghaddam. Principal manifolds and bayesian subspaces for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6):780–788, June 2002.
24. B. Moghaddam, T. Jebara, and A. Pentland. Efficient MAP/ML similarity matching for face recognition. In *Proceedings of International Conference on Pattern Recognition*, pages 876–881, Brisbane, Australia, Aug. 1998.
25. B. Moghaddam, T. Jebara, and A. Pentland. Bayesian face recognition. *Pattern Recognition*, 33(11):1771–1782, Nov. 2000.
26. B. Moghaddam and A. Pentland. Probabilistic visual learning for object detection. In *Proceedings of IEEE International Conference on Computer Vision*, pages 786–793, Cambridge, USA, June 1995.
27. B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):696–710, July 1997.
28. H. Murase and S. K. Nayar. Visual learning and recognition of 3D objects from appearance. *International Journal of Computer Vision*, 14(1):5–24, Jan. 1995.
29. P. Penev and L.Sirovich. The global dimensionality of face space. In *Proc. of IEEE Internation Conf. on Face and Gesture Recognition*, pages 264–270, Grenoble, France, 2000.
30. A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, pages 84–91, Seattle, WA, June 1994. IEEE Computer Society Press.
31. P. J. Phillips, H. Moon, P. Rauss, and S. Rizvi. The FERET evaluation methodology for face-recognition algorithms. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, pages 137–143, June 1997.
32. B. Schölkopf, A. Smola, and K.-R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.
33. G. Shakhnarovich, J. W. Fisher, and T. Darrell. Face recognition from long-term observations. In *Proceedings of European Conference on Computer Vision*, pages 851–865, Copenhagen, Denmark, May 2002.
34. M. Tipping and C. Bishop. Probabilistic principal component analysis. Technical Report NCRG/97/010, Aston University, Sept. 1997.
35. M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.
36. M. Turk and A. Pentland. Face recognition using eigenfaces. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, pages 586–590, Maui, Hawaii, Dec. 1991.
37. M. Vasilescu and D. Terzopoulos. Multilinear Subspace Analysis of Image Ensembles. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, pages 93–99, Madison, WI, June 2003.
38. M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: TensorFaces. In *Proceedings of European Conference on Computer Vision*, pages 447–460, Copenhagen, Denmark, May 2002.
39. X. Wang and X. Tang. Unified subspace analysis for face recognition. In *Proceedings of IEEE International Conference on Computer Vision*, Nice, France, June 2003.

40. L. Wolf and A. Shashua. Learning over Sets using Kernel Principal Angles. *Journal of Machine Learning Research*, 4:913–931, Oct. 2003.
41. O. Yamaguchi, K. Fukui, and K.-i. Maeda. Face recognition using temporal image sequence. In *Proc. of IEEE Internation Conf. on Face and Gesture Recognition*, pages 318–323, Nara, Japan, Apr. 1998.
42. M.-H. Yang. Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods. In *Proc. of IEEE Internation Conf. on Face and Gesture Recognition*, pages 215–220, Washington, DC, May 2002.
43. R. S. Zemel and G. E. Hinton. Developing population codes by minimizing description length. In J. D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems*, volume 6, pages 11–18. Morgan Kaufmann Publishers, Inc., 1994.