Chapter 2
Face Recognition in Subspaces

Gregory Shakhnarovich and Baback Moghaddam

2.1 Introduction

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 chronologic 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. We also discuss comparative experimental evaluation of some of these techniques as well as practical issues related to the application of subspace methods for varying pose, illumination, and expression.

2.2 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 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 multidimensional data is the

G. Shakhnarovich (✉)
Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge, MA 02139, USA
e-mail: gregory@ai.mit.edu

B. Moghaddam
Mitsubishi Electric Research Labs, Cambridge, MA 02139, USA
e-mail: baback@merl.com

S.Z. Li, A.K. Jain (eds.), *Handbook of Face Recognition*,
DOI 10.1007/978-0-85729-932-1_2,

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.

2.2.1 Image Space Versus Face Space

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 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- and 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 nonparametric methods, the sample complexity—the number of examples needed to represent the underlying distribution of the data efficiently—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, and so on. 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 are in fact confined to a subspace referred to as the *face subspace*.

2.2.2 Principal Manifold and Basis Functions

It is common to model the face subspace 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 subspace; 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 $\boldsymbol{a} = [a_1, a_2, a_3]^{\mathrm{T}}$. Any point on the line can be described by three 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 $\boldsymbol{a}$. 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 subspace 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 subspace. In practice, owing to sensor noise, the signal usually has a nonzero component outside the face subspace. 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 Sect. 2.2.3, we briefly describe principal component analysis, which plays an important role in many of such techniques. For a more detailed discussion, see Gerbrands [12] and Joliffe [17].

2.2.3 Principal Component Analysis

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

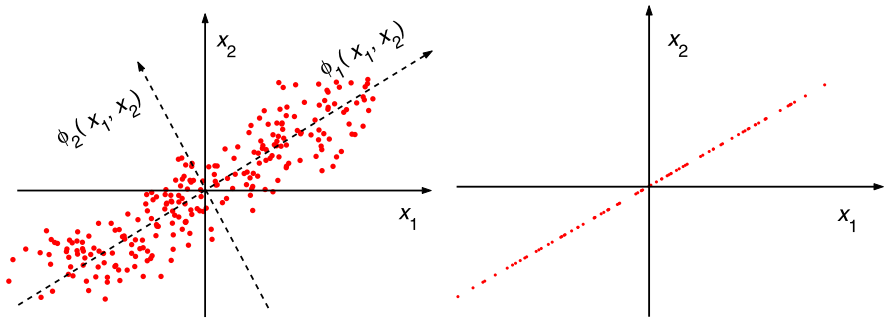
The idea of PCA is illustrated in Fig. 2.1a; the axis labeled $\boldsymbol{\phi}_1$ corresponds to the direction of maximum variance and is chosen as the first principal component. In a two-dimensional case, the second principal component is then determined uniquely by the orthogonality constraints; 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 $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_N]^{\mathrm{T}}$ that for any $k \leq N$ minimizes the average L_2 *reconstruction error* for data points $\boldsymbol{x}$

$$\varepsilon(\boldsymbol{x}) = \left\| \boldsymbol{x} - \sum_{i=1}^{k} (\boldsymbol{\phi}_i^{\mathrm{T}} \boldsymbol{x}) \phi_i \right\|. \tag{2.1}$$

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

The basis vectors in KLT can be calculated in the following way. Let $\boldsymbol{X}$ be the $N \times M$ data matrix whose columns $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M$ 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 $\boldsymbol{\Phi}$ is obtained by solving the eigenvalue problem $\boldsymbol{\Lambda} = \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Sigma} \boldsymbol{\Phi}$, where $\boldsymbol{\Sigma}$ is the covariance



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

Fig. 2.1 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

matrix of the data

$$\boldsymbol{\Sigma} = \frac{1}{M} \sum_{i=1}^{M} \boldsymbol{x}_i \boldsymbol{x}_i^{\mathrm{T}} \tag{2.2}$$

$\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_m]^{\mathrm{T}}$ is the eigenvector matrix of $\boldsymbol{\Sigma}$, and $\boldsymbol{\Lambda}$ is the diagonal matrix with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_N$ of $\boldsymbol{\Sigma}$ on its main diagonal, so ϕ_j is the eigenvector corresponding to the jth largest eigenvalue. Then it can be shown that the eigenvalue λ_i is the variance of the data projected on $\boldsymbol{\phi}_i$.

Thus, to perform PCA and extract k principal components of the data, one must project the data onto $\boldsymbol{\Phi}_k$, the first k columns of the KLT basis $\boldsymbol{\Phi}$, which correspond to the k highest eigenvalues of $\boldsymbol{\Sigma}$. This can be seen as a linear projection $\mathbb{R}^N \rightarrow \mathbb{R}^k$, which 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 $\boldsymbol{\Phi}_k^{\mathrm{T}} \boldsymbol{X}$ is always diagonal.

The main properties of PCA are summarized by the following

$$\boldsymbol{x} \approx \boldsymbol{\Phi}_k \boldsymbol{y}, \qquad \boldsymbol{\Phi}_k^{\mathrm{T}} \boldsymbol{\Phi}_k = \boldsymbol{I}, \qquad E\{y_i y_j\}_{i \neq j} = 0 \tag{2.3}$$

namely, approximate reconstruction, orthonormality of the basis $\boldsymbol{\Phi}_k$, and decorrelated principal components $y_i = \boldsymbol{\phi}_i^{\mathrm{T}} \boldsymbol{x}$, respectively. These properties are illustrated in Fig. 2.1, where PCA is successful in finding the principal manifold, and in Fig. 2.8a (see later), where it is less successful, owing to clear nonlinearity of the principal manifold.

PCA may be implemented via singular value decomposition (SVD). The SVD of an $M \times N$ matrix $\boldsymbol{X}$ ($M \geq N$) is given by

$$\boldsymbol{X} = \boldsymbol{U} \boldsymbol{D} \boldsymbol{V}^{\mathrm{T}} \tag{2.4}$$

where the $M \times N$ matrix $\boldsymbol{U}$ and the $N \times N$ matrix $\boldsymbol{V}$ have orthonormal columns, and the $N \times N$ matrix $\boldsymbol{D}$ has the singular values[1] of $\boldsymbol{X}$ on its main diagonal and zero elsewhere.

It can be shown that $\boldsymbol{U} = \boldsymbol{\Phi}$, so SVD allows efficient and robust computation of PCA without the need to estimate the data covariance matrix $\boldsymbol{\Sigma}$ (2.2). When the number of examples M is much smaller than the dimension N, this is a crucial advantage.

2.2.4 Eigenspectrum and Dimensionality

An important 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 subspace) 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 eigenvalues of the data covariance matrix $\boldsymbol{\Sigma}$. Recall that the ith eigenvalue is equal to the variance along the ith 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 Fig. 2.3b (see later).

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. See Penev and Sirovich [29] as well as Sects. 2.3.2 and 2.3.4 for more discussion on this issue.

2.3 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, the face subspace is a linear subspace of the image space. In this section, we describe methods that operate under this assumption and its generalization, a multilinear manifold.

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

Fig. 2.2 Eigenfaces: the average face on the left, followed by seven top eigenfaces. From Turk and Pentland [36], with permission

2.3.1 Eigenfaces and Related Techniques

In their ground-breaking work in 1990, 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. Because the basis vectors constructed by PCA had the same dimension as the input face images, they were named "eigenfaces." Figure 2.2 shows an example of the mean face and a few of the top eigenfaces. Each 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 (2.1), the image was classified as belonging to one of the familiar subjects, as a new face, or as a nonface. 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 a nonmember of the class.

2.3.2 Probabilistic Eigenspaces

The role of PCA in the original Eigenfaces was largely confined to dimensionality reduction. The similarity between images $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$ was measured in terms of the Euclidean norm of the difference $\boldsymbol{\Delta} = \boldsymbol{I}_1 - \boldsymbol{I}_2$ projected to the subspace, essentially ignoring the variation modes within the subspace and outside it. This was improved in the extension of eigenfaces proposed by Moghaddam and Pentland [24, 25], which uses a *probabilistic* similarity measure based on a parametric estimate of the probability density $p(\boldsymbol{\Delta} \mid \Omega)$.

A major difficulty with such estimation is that normally there are not nearly enough data to estimate the parameters of the density in a high dimensional space. Moghaddam and Pentland overcame this problem by using PCA to divide the vector space $\mathbb{R}^N$ into two subspaces, as shown in Fig. 2.3: the principal subspace F, obtained by $\boldsymbol{\Phi}_k$ (the first k columns of $\boldsymbol{\Phi}$) and its orthogonal complement $\bar{F}$ spanned by the remaining columns of $\boldsymbol{\Phi}$. The operating assumption here is that the data have

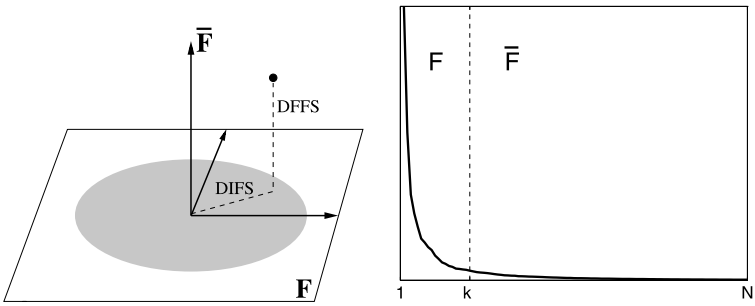


Fig. 2.3 **a** Decomposition of $\mathbb{R}^N$ into the principal subspace F and its orthogonal complement $\bar{F}$ for a Gaussian density. **b** Typical eigenvalue spectrum and its division into the two orthogonal subspaces

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 2.3a shows the decomposition of $\boldsymbol{\Delta}$ into distance *within* face subspace (DIFS) and the distance *from* face subspace (DFFS). Moreover, the probability density can be decomposed into two orthogonal components.

$$P(\boldsymbol{\Delta} \mid \Omega) = P_F(\boldsymbol{\Delta} \mid \Omega) \cdot P_{\bar{F}}(\boldsymbol{\Delta} \mid \Omega). \tag{2.5}$$

In the simplest case, $P(\boldsymbol{\Delta} \mid \Omega)$ is a Gaussian density. As derived by Moghaddam and Pentland [24], the complete likelihood estimate in this case can be written as the product of two independent marginal Gaussian densities

$$\begin{aligned} \hat{P}(\boldsymbol{\Delta} \mid \Omega) &= \left[\frac{\exp(-\frac{1}{2}\sum_{i=1}^{k} \frac{y_i^2}{\lambda_i})}{(2\pi)^{k/2} \prod_{i=1}^{k} \lambda_i^{1/2}} \right] \cdot \left[\frac{\exp(-\frac{\epsilon^2(\Delta)}{2\rho})}{(2\pi\rho)^{(N-k)/2}} \right] \\ &= P_F(\boldsymbol{\Delta} \mid \Omega) \hat{P}_{\bar{F}}(\boldsymbol{\Delta} \mid \Omega; \rho) \end{aligned} \tag{2.6}$$

where $P_F(\boldsymbol{\Delta} \mid \Omega)$ is the true marginal density in F; $\hat{P}_{\bar{F}}(\boldsymbol{\Delta} \mid \Omega; \rho)$ is the estimated marginal density in $\bar{F}$; $y_i = \boldsymbol{\phi}_i^{\mathrm{T}} \boldsymbol{\Delta}$ are the principal components of $\boldsymbol{\Delta}$; and $\epsilon(\boldsymbol{\Delta})$ is the PCA reconstruction error (2.1). The information-theoretical 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

$$\rho = \frac{1}{N-k} \sum_{i=k+1}^{N} \lambda_i. \tag{2.7}$$

This is a special case of the recent, more general factor analysis model called probabilistic PCA (PPCA) proposed by Tipping and Bishop [34]. In their formulation,

the above expression for ρ is the maximum-likelihood solution of a latent variable model in contrast to the minimal-divergence solution derived by Moghaddam and Pentland [24].

In practice, most of the eigenvalues in $\bar{F}$ cannot be computed owing 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 Fig. 2.3b.

In this probabilistic framework, the recognition of a test image $\boldsymbol{x}$ is carried out in terms of computing for every database example $\boldsymbol{x}_i$ the difference $\boldsymbol{\Delta} = \boldsymbol{x} - \boldsymbol{x}_i$ and its decomposition into the F and $\bar{F}$ components and then ranking the examples according to the value in (2.6).

2.3.3 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 that retains most of that variation, and consequently the similarity in the face subspace is not necessarily determined by the identity.

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

$$\frac{|\boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{S}_{\mathrm{b}} \boldsymbol{\Phi}|}{|\boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{S}_{\mathrm{w}} \boldsymbol{\Phi}|} \tag{2.8}$$

where

$$\boldsymbol{S}_{\mathrm{b}} = \sum_{i=1}^{m} N_i (\bar{\boldsymbol{x}}_i - \bar{\boldsymbol{x}})(\bar{\boldsymbol{x}}_i - \bar{\boldsymbol{x}})^{\mathrm{T}}$$

is the *between-class* scatter matrix, and

$$\boldsymbol{S}_{\mathrm{w}} = \sum_{i=1}^{m} \sum_{\boldsymbol{x} \in \boldsymbol{X}_i} (\boldsymbol{x} - \bar{\boldsymbol{x}}_i)(\boldsymbol{x} - \bar{\boldsymbol{x}}_i)^{\mathrm{T}}$$

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 $\boldsymbol{\Phi}$ is at most $m - 1$.[2]

Because in practice $\boldsymbol{S}_{\mathrm{w}}$ is usually singular, the Fisherfaces algorithm first reduces the dimensionality of the data with PCA so (2.8) can be computed and then

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

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 by Belhumeur et al. [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 a lower error rate than eigenfaces.

2.3.4 Bayesian Methods

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

$$S(\boldsymbol{\Delta}) = P(\Omega_I \mid \boldsymbol{\Delta}) = \frac{P(\boldsymbol{\Delta} \mid \Omega_I)P(\Omega_I)}{P(\boldsymbol{\Delta} \mid \Omega_I)P(\Omega_I) + P(\boldsymbol{\Delta} \mid \Omega_E)P(\Omega_E)}. \tag{2.9}$$

Note that this particular Bayesian formulation, proposed by Moghaddam et al. [27], 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 Sect. 2.3.2.

$$\begin{aligned} P(\boldsymbol{\Delta} \mid \Omega_E) &= \frac{e^{-\frac{1}{2}\boldsymbol{\Delta}^{\mathrm{T}}\Sigma_E^{-1}\boldsymbol{\Delta}}}{(2\pi)^{D/2}|\Sigma_E|^{1/2}}, \\ P(\boldsymbol{\Delta}|\Omega_I) &= \frac{e^{-\frac{1}{2}\boldsymbol{\Delta}^{\mathrm{T}}\Sigma_I^{-1}\boldsymbol{\Delta}}}{(2\pi)^{D/2}|\Sigma_I|^{1/2}}. \end{aligned} \tag{2.10}$$

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

By PCA, the Gaussians are known to occupy only a subspace of image space (face subspace); thus, only the top few eigenvectors of the Gaussian densities are relevant for modeling. These densities are used to evaluate the similarity in (2.9). Computing the similarity involves first subtracting a candidate image $\boldsymbol{I}$ from a database example $\boldsymbol{I}_j$. The resulting $\boldsymbol{\Delta}$ 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 (2.9). 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 $P(\boldsymbol{\Delta} \mid \Omega_I)$ and $P(\boldsymbol{\Delta} \mid \Omega_E)$, the database images $\boldsymbol{I}_j$ are preprocessed with *whitening* transformations [11]. Each image is converted and stored as a set of two whitened subspace coefficients: $\boldsymbol{y}_{\Phi_I}$ for intrapersonal space and $\boldsymbol{y}_{\Phi_E}$ for extrapersonal space

$$\boldsymbol{y}^j_{\Phi_I} = \boldsymbol{\Lambda}_I^{-\frac{1}{2}} \boldsymbol{V}_I \boldsymbol{I}_j, \qquad \boldsymbol{y}^j_{\Phi_E} = \boldsymbol{\Lambda}_E^{-\frac{1}{2}} \boldsymbol{V}_E \boldsymbol{I}_j \tag{2.11}$$

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

After this preprocessing, evaluating the Gaussians can be reduced to simple Euclidean distances as in (2.12). Denominators are of course precomputed. These likelihoods are evaluated and used to compute the *maximum a posteriori* (MAP) similarity $S(\boldsymbol{\Delta})$ in (2.9). Euclidean distances are computed between the k_I-dimensional $\boldsymbol{y}_{\Phi_I}$ vectors as well as the k_E-dimensional $\boldsymbol{y}_{\Phi_E}$ vectors. Thus, roughly $2 \times (k_E + k_I)$ arithmetic operations are required for each similarity computation, avoiding repeated image differencing and projections

$$\begin{aligned} P(\boldsymbol{\Delta} \mid \Omega_I) = P(\boldsymbol{I} - \boldsymbol{I}_j \mid \Omega_I) &= \frac{e^{-\|\boldsymbol{y}_{\Phi_I} - \boldsymbol{y}^j_{\Phi_I}\|^2/2}}{(2\pi)^{k_I/2} |\boldsymbol{\Sigma}_I|^{1/2}}, \\ P(\boldsymbol{\Delta} \mid \Omega_E) = P(\boldsymbol{I} - \boldsymbol{I}_j \mid \Omega_E) &= \frac{e^{-\|\boldsymbol{y}_{\Phi_E} - \boldsymbol{y}^j_{\Phi_E}\|^2/2}}{(2\pi)^{k_E/2} |\boldsymbol{\Sigma}_E|^{1/2}}. \end{aligned} \tag{2.12}$$

The *maximum likelihood* (ML) similarity matching is even simpler, as only the intrapersonal class is evaluated, leading to the following modified form for the similarity measure.

$$S'(\boldsymbol{\Delta}) = P(\boldsymbol{\Delta} \mid \Omega_I) = \frac{e^{-\|\boldsymbol{y}_{\Phi_I} - \boldsymbol{y}^j_{\Phi_I}\|^2/2}}{(2\pi)^{k_I/2} |\boldsymbol{\Sigma}_I|^{1/2}}. \tag{2.13}$$

The approach described above requires two projections of the difference vector $\boldsymbol{\Delta}$, from which likelihoods can be estimated for the Bayesian similarity measure. The computation flow is illustrated in Fig. 2.4b. 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 $\boldsymbol{\Delta}$ onto the "dual eigenfaces" (Ω_I and Ω_E) for computation of the posterior in (2.9).

It is instructive to compare and contrast LDA (Fisherfaces) and the dual subspace technique by noting the similar roles of the between-class/within-class and extrapersonal/intrapersonal subspaces. One such analysis was presented by Wang and Tang [39] where PCA, LDA, and Bayesian methods were "unified" under a three-parameter subspace method. Ultimately, the optimal probabilistic justification of LDA is for the case of two Gaussian distributions of equal covariance (although LDA tends to perform rather well even when this condition is not strictly true). In contrast, the dual formulation is entirely general and probabilistic by definition, and it makes no appeals to geometry, Gaussianity, or symmetry of the underlying data

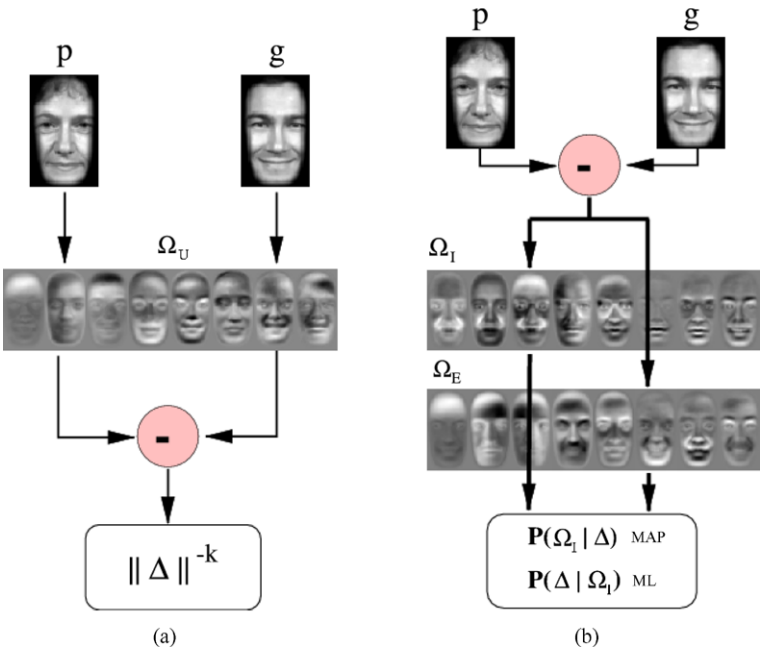


Fig. 2.4 Signal flow diagrams for computing the similarity g between two images. **a** Original eigenfaces. **b** Bayesian similarity. The difference image is projected through both sets of (intra/extra) eigenfaces to obtain the two likelihoods

or, in fact, the two "meta classes" (intra-, and extrapersonal). These two probability distributions can take on any form (e.g., arbitrary mixture models), not just single Gaussians, although the latter case does make for easy visualization by diagonalizing the dual covariances as two sets of "eigenfaces".

2.3.5 Independent Component Analysis and Source Separation

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

$$\boldsymbol{x} \approx \boldsymbol{A}\boldsymbol{y}, \qquad \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} \neq \boldsymbol{I}, \qquad P(\boldsymbol{y}) \approx \prod p(y_i) \tag{2.14}$$

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

An example of ICA basis is shown in Fig. 2.5, where it is computed from a set of 3D points. The 2D subspace recovered by ICA appears to reflect the distribution

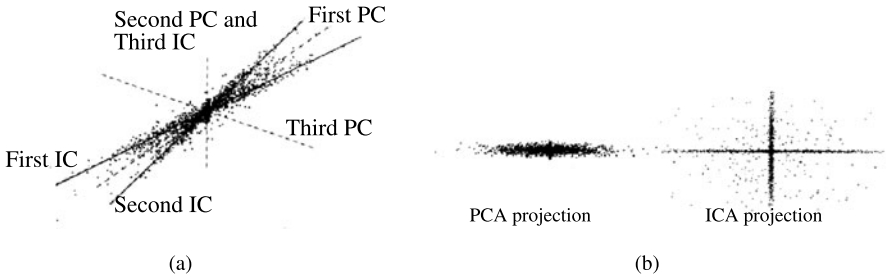


Fig. 2.5 ICA vs. PCA decomposition of a 3D data set. **a** The bases of PCA (orthogonal) and ICA (nonorthogonal). **b** *Left*: the projection of the data onto the top two principal components (PCA). *Right*: the projection onto the top two independent components (ICA). (From Bartlett et al. [1], with permission)

of the data much better than the subspace obtained with PCA. Another example of an ICA basis is shown in Fig. 2.8b where we see two unordered nonorthogonal IC vectors, one of which is roughly aligned with the first principal component vector in Fig. 2.8a (see later), (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) $\boldsymbol{x}$ into a linear combination (mixture) of independent source signals. Formally, the assumption is that $\boldsymbol{x}^{\mathrm{T}} = \boldsymbol{A}\boldsymbol{s}^{\mathbf{T}}$, with $\boldsymbol{A}$ the unknown mixing matrix. ICA algorithms[3] try to find $\boldsymbol{A}$ or the *separating matrix* $\boldsymbol{W}$ such that $\boldsymbol{u}^{\mathrm{T}} = \boldsymbol{W}\boldsymbol{x}^{\mathrm{T}} = \boldsymbol{W}\boldsymbol{A}\boldsymbol{s}^{\mathrm{T}}$. When the data consist of M observations with N variables, the input to ICA is arranged in an $N \times M$ matrix $\boldsymbol{X}$.

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

Architecture I Rows of $\boldsymbol{S}$ are *independent basis images*, which combined by $\boldsymbol{A}$ yield the input images $\boldsymbol{X}$. Learning $\boldsymbol{W}$ allows us to estimate the basis images in the rows of $\boldsymbol{U}$. In practice, for reasons of computational tractability, PCA is first performed on the input data $\boldsymbol{X}$ to find the top K eigenfaces; these are arranged in the columns of a matrix $\boldsymbol{E}$.[4] Then ICA is performed on $\boldsymbol{E}^{\mathrm{T}}$; that is, the images are variables, and the pixel values are observations. Let $\boldsymbol{C}$ be the PCA coefficient matrix, that is, $\boldsymbol{X} = \boldsymbol{C}\boldsymbol{E}^{\mathrm{T}}$. Then the k independent ICA basis images (Fig. 2.6, top) are estimated by the rows of $\boldsymbol{U} = \boldsymbol{W}\boldsymbol{E}^{\mathrm{T}}$, and the coefficients for the data are computed from $\boldsymbol{X} = \boldsymbol{E}\boldsymbol{W}^{-1}\boldsymbol{U}$.

Architecture II This architecture assumes that the sources in $\boldsymbol{S}$ are independent coefficients, and the columns of the mixing matrix $\boldsymbol{A}$ are the basis images; that is, the

[3] A number of algorithms exist; most notable are Jade [5], InfoMax, and FastICA [16].

[4] These eigenfaces are linear combination of the original images, which under the assumptions of ICA should not affect the resulting decomposition.

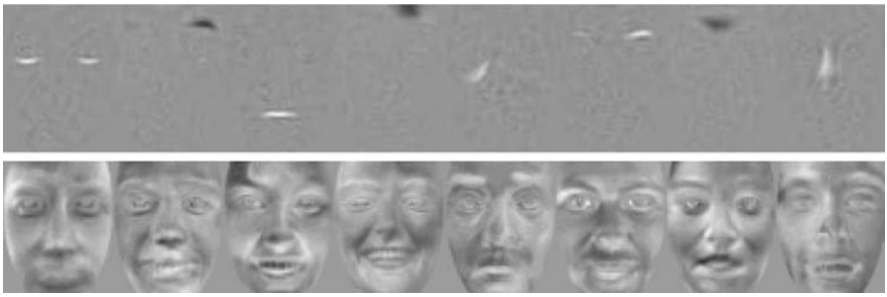


Fig. 2.6 Basis images obtained with ICA: Architecture I (*top*) and II (*bottom*). (From Draper et al. [10], with permission)

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 $\boldsymbol{C}$. The resulting ICA basis consists of the columns of $\boldsymbol{EA}$ (Fig. 2.6, bottom), and the coefficients are found in the rows of $\boldsymbol{U} = \boldsymbol{W}\boldsymbol{C}^{\mathrm{T}}$. These coefficients give the *factorial representation* of the data.

Generally, the bases obtained with Architecture I reflect more local properties of the faces, whereas the bases in Architecture II have global properties and much more resemble faces (Fig. 2.6).

2.3.6 Multilinear 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 Sect. 2.6.1). In Sect. 2.4, we discuss recognition with nonlinear subspaces. An alternative, *multilinear* approach, called "tensorfaces," has been proposed by Vasilescu and Terzopoulos in [37, 38].

Tensor is a multidimensional generalization of a matrix: a *n-order tensor* $\mathscr{A}$ 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 ith row of $\mathscr{A}_{(s)}$ is obtained by concatenating all the elements of $\mathscr{A}$ 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 $\mathscr{A} \times_l \boldsymbol{M}$ of a tensor $\mathscr{A}$ and an $m \times k$ matrix $\boldsymbol{M}$, where k is the lth dimension of $\mathscr{A}$.

$$(\mathscr{A} \times_l \boldsymbol{M})_{i_1,\ldots i_{l-1},j,i_{l+1},\ldots i_n} = \sum_{i=1}^{k} a_{i_1,\ldots i_{l-1},i,i_{l+1},\ldots i_n} m_{ji}. \tag{2.15}$$

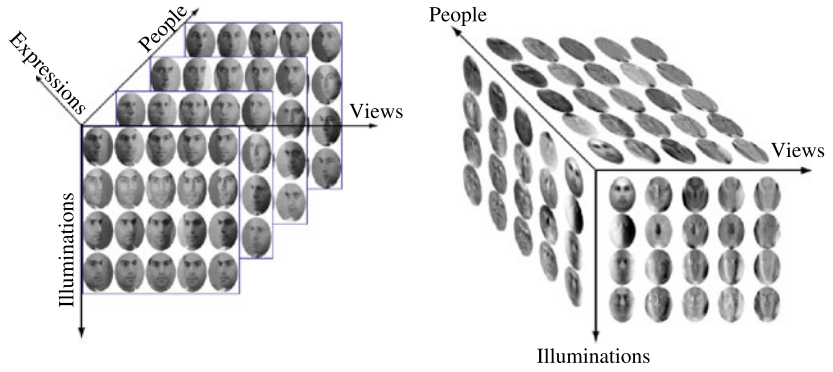


Fig. 2.7 Tensorfaces. **a** Data tensor; the four dimensions visualized are identity, illumination, pose, and the pixel vector. The fifth dimension corresponds to expression (only the subtensor for neutral expression is shown). **b** Tensorfaces decomposition. (From Vasilescu and Terzopoulos [37], with permission)

Under this definition, Vasilescu and Terzopoulos proposed [38] an algorithm they called *n-mode SVD*, which decomposes an n-dimensional tensor $\mathscr{A}$ into

$$\mathscr{A} = \mathscr{Z} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \cdots \times_n \boldsymbol{U}_n. \tag{2.16}$$

The role of the *core tensor* $\mathscr{Z}$ in this decomposition is similar to the role of the singular value matrix $\boldsymbol{\Sigma}$ in SVD (2.4): It governs the interactions between the *mode matrices* $\boldsymbol{U}_1, \ldots, \boldsymbol{U}_n$, 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

$$\mathscr{Z} = \mathscr{A} \times_1 \boldsymbol{U}_1^{\mathrm{T}} \times_2 \boldsymbol{U}_2^{\mathrm{T}} \cdots \times_N \boldsymbol{U}_n^{\mathrm{T}}.$$

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 an $N_\mathrm{p} \times N_\mathrm{v} \times N_\mathrm{i} \times N_\mathrm{e} \times N$ tensor of order 5. Figure 2.7a illustrates this concept: Only four 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

$$\mathscr{A} = \mathscr{Z} \times_1 \boldsymbol{U}_\mathrm{p} \times_2 \boldsymbol{U}_\mathrm{v} \times_3 \boldsymbol{U}_\mathrm{i} \times_4 \boldsymbol{U}_\mathrm{e} \times_5 \boldsymbol{U}_\mathrm{pixels}. \tag{2.17}$$

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

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

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

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

$$\boldsymbol{x}_{\mathrm{p,v,e,i}} = \boldsymbol{B}^\mathrm{T}_{\mathrm{v,i,e}} \boldsymbol{c}_\mathrm{p} \tag{2.18}$$

where $\boldsymbol{c}_j$ is the jth row vector of $\boldsymbol{U}_\mathrm{p}$.

Given an input image $\boldsymbol{x}$, a candidate coefficient vector $\boldsymbol{c}_{\mathrm{v,i,e}}$ is computed for all combinations of viewpoint, expression, and illumination, solving (2.18). The recognition is carried out by finding the value of j that yields the minimum Euclidean distance between $\boldsymbol{c}$ and the vectors $\boldsymbol{c}_j$ across all illuminations, expressions, and viewpoints.[5]

Vasilescu and Terzopoulos [38] reported experiments involving the data tensor consisting of images of $N_\mathrm{p} = 28$ subjects photographed in $N_\mathrm{i} = 3$ illumination conditions from $N_\mathrm{v} = 5$ viewpoints, with $N_\mathrm{e} = 3$ different expressions; the images were resized and cropped so they contain $N = 7493$ pixels. The performance of tensorfaces is reported to be significantly better than that of standard eigenfaces described in Sect. 2.3.1.

2.4 Nonlinear Subspaces

In this section, we describe a number of techniques that do not assume that the principal manifold is linear.

2.4.1 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 Fig. 2.8c.

One of the simplest methods for computing nonlinear principal manifolds is the nonlinear PCA (NLPCA) autoencoder multilayer neural network [9, 20] shown in

[5] This also provides an estimate of the parameters (e.g., illumination) for the input image.

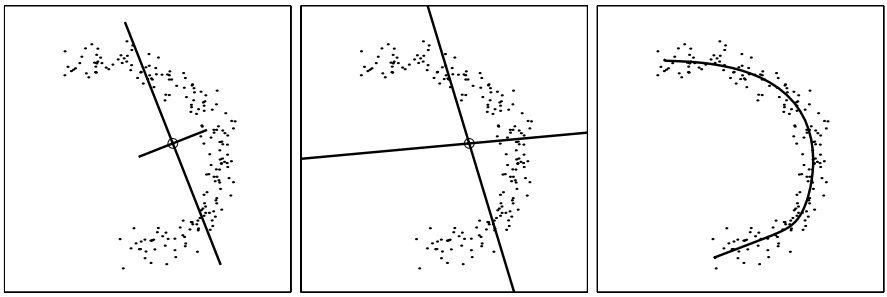


Fig. 2.8 **a** PCA basis (linear, ordered, and orthogonal). **b** ICA basis (linear, unordered, and nonorthogonal). **c** Principal curve (parameterized nonlinear manifold). The *circle* shows the data mean

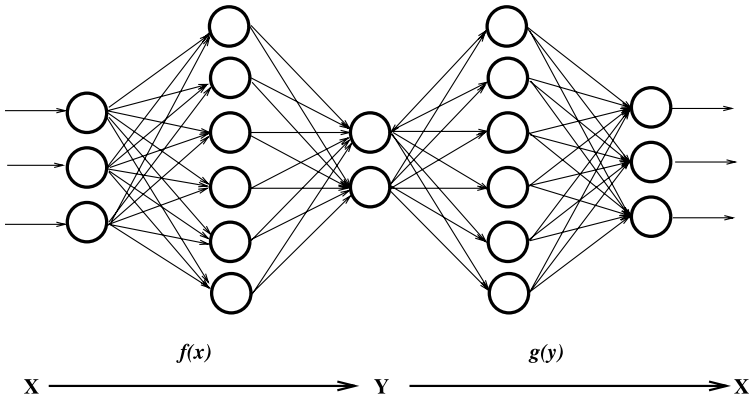


Fig. 2.9 Autoassociative ("bottleneck") neural network for computing principal manifolds $\boldsymbol{y} \in \mathbb{R}^k$ in the input space $\boldsymbol{x} \in \mathbb{R}^N$

Fig. 2.9. The "bottleneck" layer forms a lower-dimensional manifold representation by means of a nonlinear *projection* function $f(\boldsymbol{x})$, implemented as a weighted sum-of-sigmoids. The resulting principal components $\boldsymbol{y}$ have an inverse mapping with a similar nonlinear *reconstruction* function $g(\boldsymbol{y})$, which reproduces the input data as accurately as possible. The NLPCA computed by such a multilayer sigmoidal neural network is equivalent (with certain exceptions[6]) to a *principal surface* under the more general definition [13, 14]. To summarize, the main properties of NLPCA are

$$\boldsymbol{y} = f(\boldsymbol{x}), \qquad \boldsymbol{x} \approx g(\boldsymbol{y}), \qquad P(\boldsymbol{y}) = ? \tag{2.19}$$

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

corresponding to nonlinear projection, approximate reconstruction, and typically no prior knowledge regarding the joint distribution of the components, respectively (however, see Zemel and Hinton [43] for an example of devising suitable priors in such cases). The principal curve in Fig. 2.8c was generated with a 2-4-1-4-2 layer neural network of the type shown in Fig. 2.9. Note how the principal curve yields a compact, relatively accurate representation of the data, in contrast to the linear models (PCA and ICA).

2.4.2 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 $\Psi(\boldsymbol{x}) : \mathbb{R}^N \rightarrow \mathbb{R}^L$ 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 $\Psi(\boldsymbol{x})$ is made implicit (and economical) by the use of kernel functions satisfying Mercer's theorem [7]

$$k(\boldsymbol{x_i}, \boldsymbol{x_j}) = \left[\Psi(\boldsymbol{x_i}) \cdot \Psi(\boldsymbol{x_j})\right] \tag{2.20}$$

where kernel evaluations $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ 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 $\Psi(\boldsymbol{x})$. Specifically, assuming that the projection of the data in feature space is zero-mean ("centered"), the covariance is given by

$$\boldsymbol{\Sigma}_K = \left\langle \Psi(\boldsymbol{x}_i), \Psi(\boldsymbol{x}_i)^{\mathrm{T}} \right\rangle \tag{2.21}$$

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

$$\lambda\left(\Psi(\boldsymbol{x}_i) \cdot \boldsymbol{V}\right) = \left(\Psi(\boldsymbol{x}_i) \cdot \boldsymbol{\Sigma}_K \boldsymbol{V}\right) \quad \text{for } i = 1, \ldots, T \tag{2.22}$$

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

$$\boldsymbol{V} = \sum_{i=1}^{\mathrm{T}} w_i \Psi(\boldsymbol{x}_i). \tag{2.23}$$

Using the definition of $\boldsymbol{\Sigma}_K$, substituting the above equation into (2.22) and defining the resulting T-by-T matrix $\boldsymbol{K}$ by $\boldsymbol{K}_{ij} = [\Psi(\boldsymbol{x}_i) \cdot \Psi(\boldsymbol{x}_j)]$ leads to the equivalent eigenvalue problem formulated in terms of kernels in the input space

$$T\lambda \boldsymbol{w} = \boldsymbol{K}\boldsymbol{w} \tag{2.24}$$

where $\boldsymbol{w} = (w_1, \ldots, w_T)^{\mathrm{T}}$ is the vector of expansion coefficients of a given eigenvector $\boldsymbol{V}$ as defined in (2.23). The kernel matrix $\boldsymbol{K}_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is then diagonalized with a standard PCA.[7] Orthonormality of the eigenvectors, $(\boldsymbol{V}^n \cdot \boldsymbol{V}^n) = 1$, leads to the equivalent normalization of their respective expansion coefficients, $\lambda_n(\boldsymbol{w}^n \cdot \boldsymbol{w}^n) = 1$.

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

$$y_n = \left(\boldsymbol{V}_n \cdot \Psi(\boldsymbol{x})\right) = \sum_{i=1}^{\mathrm{T}} w_i^n k(\boldsymbol{x}, \boldsymbol{x}_i) \tag{2.25}$$

where $\boldsymbol{V}_n$ is the nth eigenvector of the feature space defined by Ψ. As with PCA, the eigenvectors $\boldsymbol{V}_n$ can be ranked by decreasing order of their eigenvalues λ_n and a d-dimensional manifold projection of $\boldsymbol{x}$ is $\boldsymbol{y} = (y_1, \ldots, y_d)^{\mathrm{T}}$, with individual components defined by (2.25).

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 eigenvector projections than the input dimensionality of the data (because KPCA is based on the matrix $\boldsymbol{K}$, 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 $\exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|)^2/\sigma^2)$, polynomials $(\boldsymbol{x}_i \cdot \boldsymbol{x}_j)^d$ and sigmoids $\tanh(a(\boldsymbol{x}_i \cdot \boldsymbol{x}_j) + b)$, all of which satisfy Mercer's theorem [7].

Similar to the derivation of KPCA, one may extend the Fisherfaces method (see Sect. 2.3.3) by applying the FLD in the feature space. Yang [42] derived the kernel Fisherfaces algorithm, which maximizes the between-scatter to within-scatter ratio in the feature space through the use of the kernel matrix $\boldsymbol{K}$. 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.

2.5 Empirical Comparison of Subspace Methods

Moghaddam [23] reported on an extensive evaluation of many of the subspace methods described above on a large subset of the FERET data set [31] (see also Chap. 13).

[7] However, computing Σ_K in (2.21) requires "centering" the data by computing the mean of $\Psi(\boldsymbol{x}_i)$. Because there is no explicit computation of $\Psi(\boldsymbol{x}_i)$, the equivalent must be carried out when computing the kernel matrix $\boldsymbol{K}$. For details on "centering" $\boldsymbol{K}$, see Schölkopf et al. [32].

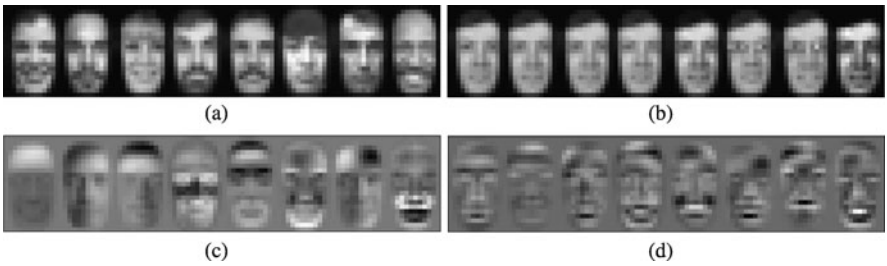


Fig. 2.10 Experiments on FERET data. **a** Several faces from the gallery. **b** Multiple probes for one individual, with different facial expressions, eyeglasses, variable ambient lighting, and image contrast. **c** Eigenfaces. **d** ICA basis images

The experimental data consisted of a training "gallery" of 706 individual FERET faces and 1123 "probe" images containing one or more views of every person in the gallery. All these images were aligned and normalized as described by Moghaddam and Pentland [25]. The multiple probe images reflected various expressions, lighting, glasses on/off, and so on. The study compared the Bayesian approach described in Sect. 2.3.4 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. Because the Bayesian algorithm was independently evaluated in DARPA's 1996 FERET face recognition competition [31] with medium resolution images (84×44 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 data set were downsampled to 21×12 pixels, yielding input vectors in a $\mathbb{R}^{N=252}$ space. Several examples are shown in Fig. 2.10a, b.

The reported results were obtained with a fivefold Cross-Validation (CV) analysis. The total data set of 1829 faces (706 unique individuals and their collective 1123 probes) was randomly partitioned into five subsets with unique (nonoverlapping) individuals and their associated probes. Each subset contained both gallery and probe images of $\approx$140 unique individuals. For each of the five subsets, the recognition task was correctly matching the multiple probes to the $\approx$140 gallery faces using the other four subsets as training data. Note that with $N = 252$ and using 80% of the entire dataset for training, there are nearly three times as many training samples than the data dimensionality; thus, parameter estimations (for PCA, ICA, KPCA, and the Bayesian method) were properly overconstrained.

The resulting five 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 who 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 ≈80% (on a different 50/50 partition of the dataset), thereby leaving a sizable margin for improvement. Note that because the recognition experiments were essentially a 140-way classification task, chance performance was approximately 0.7%.

2.5.1 PCA-Based Recognition

The baseline algorithm for these face recognition experiments was standard PCA (eigenface) matching. The first eight principal eigenvectors computed from a single partition are shown in Fig. 2.10c. 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% (Table 2.1). The full image-vector nearest-neighbor (template matching) (i.e., on $\boldsymbol{x} \in \mathbb{R}^{252}$) yielded a recognition rate of 86.46% (see dashed line in Fig. 2.11). Clearly, performance is degraded by the $252 \rightarrow 20$ dimensionality reduction, as expected.

2.5.2 ICA-Based Recognition

For ICA-based recognition (Architecture II, see Sect. 2.3.5) two algorithms based on fourth-order cumulants were tried: the "JADE" algorithm of Cardoso [5] and the fixed-point algorithm of Hyvärinen and Oja [15]. In both algorithms a PCA whitening step ("sphering") preceded the core ICA decomposition. The corresponding *nonorthogonal* JADE-derived ICA basis is shown in Fig. 2.10d. Similar basis faces were obtained with the method of Hyvärinen and Oja. These basis faces are the columns of the matrix $\boldsymbol{A}$ in (2.14), and their linear combination (specified by the ICs) reconstructs the training data. The ICA manifold projection of the test set was obtained using $\boldsymbol{y} = \boldsymbol{A}^{-1}\boldsymbol{x}$. Nearest-neighbor matching with ICA using the Euclidean L_2 norm resulted in a mean recognition rate of 77.30% with the highest rate being 82.90% (Table 2.1). We found little difference between the two ICA algorithms and noted that ICA resulted in the largest performance variation in the five trials (7.66% SD). Based on the mean recognition rates it is unclear whether ICA provides a systematic advantage over PCA or 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 the results of Bartlett et al. [1] the faces in this

Table 2.1 Recognition accuracies with $k = 20$ subspace projections using fivefold cross validation. Results are in percents

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
SD	2.21	7.66	3.39	1.96

experiment were cropped much tighter, leaving no information regarding hair and face shape, and they were much lower in resolution, factors that when combined make the recognition task much more difficult.

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. [10]. They found that the best results for ICA are obtained using the cosine distance, whereas for eigenfaces the L_1 metric appears to be optimal; with L_2 metric, which was also used in the experiments of Moghaddam [23], the performance of ICA (Architecture II) was similar to that of eigenfaces.

2.5.3 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 data set. 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 (2.25) 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%, with the highest rate being 92.37% (Table 2.1). The standard deviation of the KPCA trials was slightly higher (3.39) than that of PCA (2.21), but Fig. 2.11 indicates that KPCA does in fact do better than both PCA and ICA, hence justifying the use of nonlinear feature extraction.

2.5.4 MAP-Based Recognition

For Bayesian similarity matching, appropriate training $\boldsymbol{\Delta}$s for the two classes Ω_I (Fig. 2.10b) and Ω_E (Fig. 2.10a) were used for the dual PCA-based density estimates $P(\boldsymbol{\Delta} \mid \Omega_I)$ and $P(\boldsymbol{\Delta} \mid \Omega_E)$, which were both modeled as single Gaussians

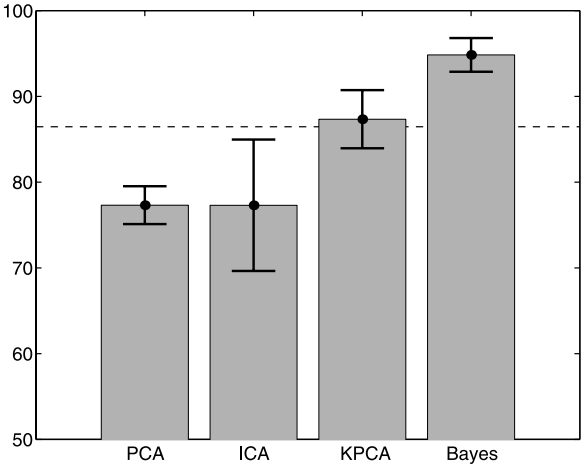


Fig. 2.11 Recognition performance of PCA, ICA, and KPCA manifolds versus Bayesian (MAP) similarity matching with a $k = 20$ dimensional subspace. *Dashed line* indicates the performance of nearest-neighbor matching with the full-dimensional image vectors

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.[8]

With $k = 20$, Gaussian subspace dimensions of $k_I = 10$ and $k_E = 10$ were used for $P(\boldsymbol{\Delta} \mid \Omega_I)$ and $P(\boldsymbol{\Delta} \mid \Omega_E)$, respectively. Note that $k_I + k_E = 20$, thus matching the total number of projections used with the three principal manifold techniques. Using the maximum *a posteriori* (MAP) similarity in (2.9), the Bayesian matching technique yielded a mean recognition rate of 94.83%, with the highest rate achieved being 97.87% (Table 2.1). The standard deviation of the five partitions for this algorithm was also the lowest (1.96) (Fig 2.11).

2.5.5 Compactness of Manifolds

The performance of various 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 by Moghaddam [23]: Given the *same* number of subspace projections, which of these techniques is better at data modeling and subsequent recognition? The presumption is that the one achieving the highest recognition rate with the smallest dimension is preferred.

[8] In practice, $k_I > k_E$ often works just as well. In fact, as $k_E \to 0$, one obtains a maximum-likelihood similarity $S = P(\boldsymbol{\Delta} \mid \Omega_I)$ with $k_I = k$, which for this data set is only a few percent less accurate than MAP [26].

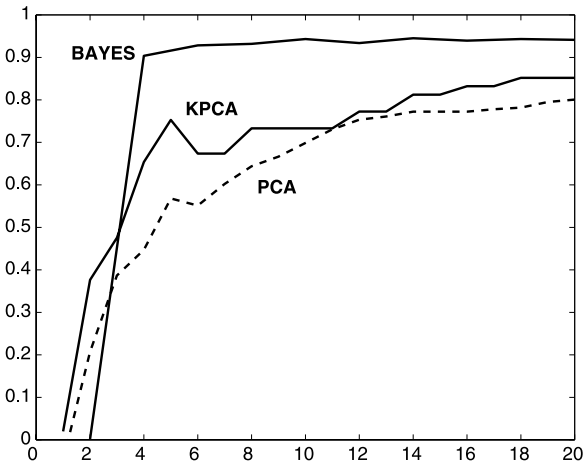


Fig. 2.12 Recognition accuracy $R(k)$ of PCA, KPCA, and Bayesian similarity with increasing dimensionality k of the principal subspace. ICA results, not shown, are similar to those of PCA

For this particular dimensionality test, the total data set 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 training set. The results of this experiment are shown in Fig. 2.12, 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 methods, as *compactness* of the manifolds—defined by the lowest acceptable value of k—is an important consideration in regard to both generalization error (overfitting) and computational requirements.

2.5.6 Discussion

The relative performance of the principal manifold techniques and Bayesian matching is summarized in Table 2.1 and Fig. 2.11. The advantage of probabilistic matching over metric matching on both linear and nonlinear manifolds is quite evident ($\approx$18% increase over PCA and $\approx$8% over KPCA). Note that the dimensionality test results in Fig. 2.12 indicate that KPCA outperforms PCA by a $\approx$10% margin, and even more so with only few principal components (a similar effect was reported by Schölkopf et al. [32] where KPCA outperforms PCA in low-dimensional manifolds). However, Bayesian matching achieves $\approx$90% with only four projections—two for each $P(\boldsymbol{\Delta} \mid \Omega)$—and dominates both PCA and KPCA throughout the entire range of subspace dimensions in Fig. 2.12.

A comparison of the subspace techniques with respect to multiple criteria is shown in Table 2.2. Note that PCA, KPCA, and the dual subspace density estimation are uniquely defined for a given training set (making experimental comparisons

Table 2.2 Comparison of the subspace techniques across multiple attributes ($k = 20$)

	PCA	ICA	KPCA	Bayes
Accuracy	77%	77%	87%	95%
Computation	10^8	10^9	10^9	10^8
Uniqueness	Yes	No	Yes	Yes
Projections	Linear	Linear	Nonlinear	Linear

repeatable), whereas ICA is not unique owing to the variety of 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 that of PCA. Because 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 U.S. Army Laboratory [31], the Bayesian similarity technique outperformed 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, revealing the surprisingly small amount of facial detail required for high accuracy performance with this learning technique.

2.6 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.

2.6.1 Multiple View-Based 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 to approach 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

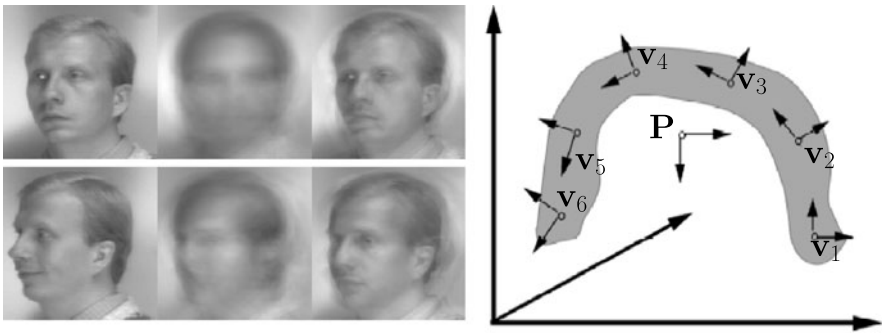


Fig. 2.13 Parametric versus view-based eigenspace methods. **a** Reconstructions of the input image (*left*) with parametric (*middle*) and view-based (*right*) eigenspaces. *Top*: training image; *bottom*: novel (test) image. **b** Difference in the way the two approaches span the manifold

parametric eigenspace encodes 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 to determine the location and orientation of the target object by selecting the eigenspace that 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 subspace, illustrated in Fig. 2.13b. 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 nonconvex (yet connected) regions in image space [3]. Therefore, the resulting ensemble is a highly complex and nonseparable manifold.

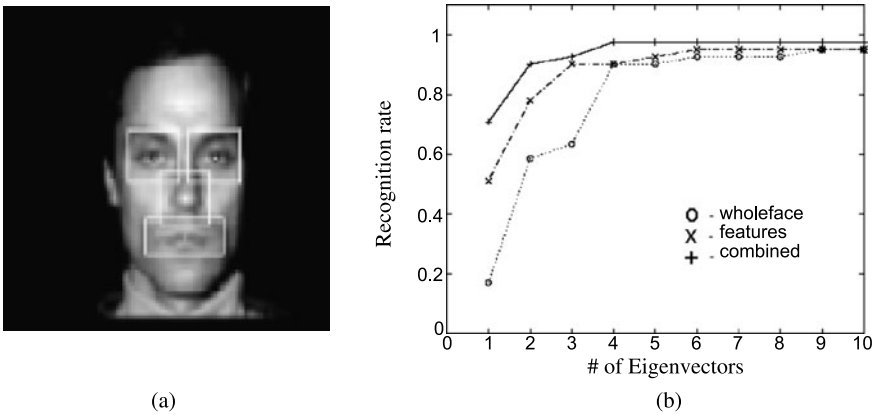
The parametric eigenspace attempts to describe this ensemble with a projection onto a single low-dimensional linear subspace (corresponding to the first k eigenvectors of the MC training images). In contrast, the view-based approach corresponds to C independent subspaces, each describing a particular region of the face subspace (corresponding to a particular view of a face). The principal manifold $\boldsymbol{v}_c$ 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 methods. Figure 2.13 compares reconstruc-

Fig. 2.14 Multiview face image data used in the experiments described in Sect. 2.6.1. (From Moghaddam and Pentland [25], with permission)

tions obtained with the two methods when trained on images of faces at multiple orientations. In the top row of Fig. 2.13a, 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. 2.13a, we see a novel view ($+68°$) with respect to the training set ($-90°$ to $+45°$). Here, both reconstructions correspond to the nearest view in the training set ($+45°$), 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-theoretical point-of-view, the multiple eigenspace representation is a more accurate representation of the signal content.

The view-based approach was evaluated [25] on data similar to that shown in Fig. 2.14 which consisted of 189 images: nine views of 21 people. The viewpoints were evenly spaced from $-90°$ to $+90°$ along the horizontal plane. In the first series of experiments, the interpolation performance was tested by training on a subset of the available views ($\pm 90°$, $\pm 45°$, $0°$) and testing on the intermediate views ($\pm 68°$, $\pm 23°$). A 90% average recognition rate was obtained. A second series of experiments tested the extrapolation performance by training on a range of views (e.g., $-90°$ to $+45°$) and testing on novel views outside the training range (e.g., $+68°$ and $+90°$). For testing views separated by $\pm 23°$ from the training range, the average recognition rate was 83%. For $\pm 45°$ testing views, the average recognition rate was 50%.



(a) (b)

Fig. 2.15 Modular eigenspaces. **a** 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 Pentland et al. [30], with permission)

2.6.2 Modular Recognition

The eigenface recognition method is easily extended to facial features [30], as shown in Fig. 2.15a. This leads to an improvement in recognition performance by incorporating an additional layer of description in terms of facial 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). Because the difference between these particular facial expressions is primarily articulated in the mouth, this feature was discarded for recognition purposes.

Figure 2.15b 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 to achieve an (asymptotic) recognition rate of 95% (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%). 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 sys-

tem can be fooled by gross variations in the input image (e.g., hats, beards). However, the feature-based representation may still find the correct match by focusing on the characteristic nonoccluded features (e.g., the eyes and nose).

2.6.3 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 a noncontiguous, perhaps unordered, set of photographs extracted from a video or obtained from individual snapshots. The former case is discussed in Chap. 13 (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 clues not only on the 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, as 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 suboptimal.

A probabilistic approach to measuring subspace similarity has been proposed [33]. The underlying statistical model assumes that images of the jth 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

$$j^* = \operatorname*{argmax}_j \Pr(p_0 = p_j). \tag{2.26}$$

Therefore, given a set of images distributed by p_0, solving (2.26) amounts to choosing optimally between M hypotheses of the form in statistics is sometimes referred to as the two-sample hypothesis: that two sets of examples come from the same distribution. A principled way to solve 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 must be estimated from data, as well as p_0. Shakhnarovich et al. [33] modeled these distributions as Gaussians (one per subject), which are estimated according to the method described in Sect. 2.3.2. The KL divergence is then computed in closed form. In the experiments reported by these authors [33], this method significantly outperformed the MSM.

Modeling the distributions by a single Gaussian is somewhat limiting; Wolf and Shashua [40] extended this approach and proposed a nonparametric discriminative method: *kernel principal angles*. They devised 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 a face recognition task on a set of nine subjects, this method significantly outperformed both MSM and the Gaussian-based KL-divergence model of Shakhnarovich et al. [33].

2.7 Conclusions

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 chronologic order in which these methods have evolved. Two most notable directions in this evolution can be discerned: (1) the transition from linear to general, possibly nonlinear, and disconnected manifolds; and (2) 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 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 thank M.S. Bartlett and M.A.O. Vasilescu for kind permission to use figures from their published work and for their comments. We also acknowledge all who contributed to the research described in this chapter.

References

1. Bartlett, M., Lades, H., Sejnowski, T.: Independent component representations for face recognition. In: Proceedings of the SPIE: Conference on Human Vision and Electronic Imaging III, vol. 3299, pp. 528–539 (1998)
2. Belhumeur, V., Hespanha, J., Kriegman, D.: Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. IEEE Trans. Pattern Anal. Mach. Intell. **19**(7), 711–720 (1997)

3. Bichsel, M., Pentland, A.: Human face recognition and the face image set's topology. CVGIP, Image Underst. **59**(2), 254–261 (1994)
4. Brunelli, R., Poggio, T.: Face recognition: Features vs. templates. IEEE Trans. Pattern Anal. Mach. Intell. **15**(10), 1042–1052 (1993)
5. Cardoso, J.-F.: High-order contrasts for independent component analysis. Neural Comput. **11**(1), 157–192 (1999)
6. Comon, P.: Independent component analysis—a new concept? Signal Process. **36**, 287–314 (1994)
7. Courant, R., Hilbert, D.: Methods of Mathematical Physics, vol. 1. Interscience, New York (1953)
8. Cover, M., Thomas, J.: Elements of Information Theory. Wiley, New York (1994)
9. DeMers, D., Cottrell, G.: Nonlinear dimensionality reduction. In: Advances in Neural Information Processing Systems, pp. 580–587. Morgan Kaufmann, San Francisco (1993)
10. Draper, B.A., Baek, K., Bartlett, M.S., Beveridge, J.R.: Recognizing faces with PCA and ICA. Comput. Vis. Image Underst. **91**(1–2), 115–137 (2003)
11. Fukunaga, K.: Introduction to Statistical Pattern Recognition, 2nd edn. Academic Press, San Diego (1990)
12. Gerbrands, J.J.: On the relationships between SVD, KLT and PCA. Pattern Recognit. **14**, 375–381 (1981)
13. Hastie, T.: Principal curves and surfaces. PhD thesis, Stanford University (1984)
14. Hastie, T., Stuetzle, W.: Principal curves. J. Am. Stat. Assoc. **84**(406), 502–516 (1989)
15. Hyvärinen, A., Oja, E.: A family of fixed-point algorithms for independent component analysis. Technical Report A40, Helsinki University of Technology (1996)
16. Hyvärinen, A., Oja, E.: Independent component analysis: algorithms and applications. Neural Netw. **13**(4–5), 411–430 (2000)
17. Jolliffe, I.T.: Principal Component Analysis. Springer, New York (1986)
18. Jutten, C., Herault, J.: Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. Signal Process. **24**, 1–10 (1991)
19. Kirby, M., Sirovich, L.: Application of the Karhunen–Loéve procedure for the characterization of human faces. IEEE Trans. Pattern Anal. Mach. Intell. **12**(1), 103–108 (1990)
20. Kramer, M.A.: Nonlinear principal components analysis using autoassociative neural networks. AIChE J. **32**(2), 233–243 (1991)
21. Loève, M.M.: Probability Theory. Van Nostrand, Princeton (1955)
22. Malthouse, E.C.: Some theoretical results on nonlinear principal component analysis. Technical report, Northwestern University (1998)
23. Moghaddam, B.: Principal manifolds and Bayesian subspaces for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. **24**(6), 780–788 (2002)
24. Moghaddam, B., Pentland, A.: Probabilistic visual learning for object detection. In: Proceedings of IEEE International Conference on Computer Vision, pp. 786–793, Cambridge, MA, June 1995
25. Moghaddam, B., Pentland, A.: Probabilistic visual learning for object representation. IEEE Trans. Pattern Anal. Mach. Intell. **19**(7), 696–710 (1997)
26. Moghaddam, B., Jebara, T., Pentland, A.: Efficient MAP/ML similarity matching for face recognition. In: Proceedings of International Conference on Pattern Recognition, pp. 876–881, Brisbane, Australia, August 1998
27. Moghaddam, B., Jebara, T., Pentland, A.: Bayesian face recognition. Pattern Recognit. **33**(11), 1771–1782 (2000)
28. Murase, H., Nayar, S.K.: Visual learning and recognition of 3D objects from appearance. Int. J. Comput. Vis. **14**(1), 5–24 (1995)
29. Penev, P., Sirovich, L.: The global dimensionality of face space. In: Proc. of IEEE Internation Conf. on Face and Gesture Recognition, pp. 264–270. Grenoble, France (2000)
30. Pentland, A., Moghaddam, B., Starner, T.: View-based and modular eigenspaces for face recognition. In: Proceedings of IEEE Computer Vision and Pattern Recognition, pp. 84–91, Seattle, WA, June 1994. IEEE Computer Society Press, Los Alamitos (1994)

31. Phillips, P.J., Moon, H., Rauss, P., Rizvi, S.: The FERET evaluation methodology for face-recognition algorithms. In: Proceedings of IEEE Computer Vision and Pattern Recognition, pp. 137–143, June 1997
32. Schölkopf, B., Smola, A., Muller, K.-R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput. **10**(5), 1299–1319 (1998)
33. Shakhnarovich, G., Fisher, J.W., Darrell, T.: Face recognition from long-term observations. In: Proceedings of European Conference on Computer Vision, pp. 851–865, Copenhagen, Denmark, May 2002
34. Tipping, M., Bishop, C.: Probabilistic principal component analysis. Technical Report NCRG/97/010, Aston University, September 1997
35. Turk, M., Pentland, A.: Eigenfaces for recognition. J. Cogn. Neurosci. **3**(1), 71–86 (1991)
36. Turk, M., Pentland, A.: Face recognition using eigenfaces. In: Proceedings of IEEE Computer Vision and Pattern Recognition, pp. 586–590, Maui, Hawaii, December 1991
37. Vasilescu, M., Terzopoulos, D.: Multilinear Subspace Analysis of Image Ensembles. In: Proceedings of IEEE Computer Vision and Pattern Recognition, pp. 93–99, Madison, WI, June 2003
38. Vasilescu, M.A.O., Terzopoulos, D.: Multilinear analysis of image ensembles: TensorFaces. In: Proceedings of European Conference on Computer Vision, pp. 447–460, Copenhagen, Denmark, May 2002
39. Wang, X., Tang, X.: Unified subspace analysis for face recognition. In: Proceedings of IEEE International Conference on Computer Vision, pp. 318–323, Nice, France, October 2003
40. Wolf, L., Shashua, A.: Learning over Sets using Kernel Principal Angles. J. Mach. Learn. Res. **4**, 913–931 (2003)
41. Yamaguchi, O., Fukui, K., Maeda, K.-I.: Face recognition using temporal image sequence. In: Proc. of IEEE Internation Conf. on Face and Gesture Recognition, pp. 318–323, Nara, Japan, April 1998
42. Yang, M.-H.: Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods. In: Proc. of IEEE Internation Conf. on Face and Gesture Recognition, pp. 215–220, Washington, DC, May 2002
43. Zemel, R.S., Hinton, G.E.: Developing population codes by minimizing description length. In: Cowan, J.D., Tesauro, G., Alspector, J. (eds.) Advances in Neural Information Processing Systems, vol. 6, pp. 11–18. Morgan Kaufmann, San Francisco (1994)

http://www.springer.com/978-0-85729-931-4

Handbook of Face Recognition
Li, S.Z.; Jain, A. (Eds.)
2011, XXV, 699 p. 293 illus., 190 illus. in color.,
Hardcover
ISBN: 978-0-85729-931-4