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

Intra-Personal Kernel Space for Face Recognition

Shaohua Zhou
Rama Chellappa
Baback Moghaddam

TR2004-025 April 2004

Abstract

Intra-personal space modeling proposed by Moghaddam *et al*. has been successfully applied in face recognition. In their work the regular principal subspaces are derived from the intra-personal space using a principal component analysis and embedded in a probabilistic formulation. In this paper, we derive the principal subspace from the intra-personal kernel space by developing a probabilistic analysis of kernel principal components for face recognition. We test this new algorithm on a subset of the FERET database with illumination and expression variations. The recognition performance demonstrates its advantage over other traditional subspace approaches.

Appears in: IEEE Int'l Conf. on Automatic Face & Gesutre Recognition (FG'04)

Publication History:–

1. First printing, TR2004-025, April 2004

INTRA-PERSONAL KERNEL SPACE FOR FACE RECOGNITION

Shaohua Kevin Zhou, Rama Chellappa[*]

Center for Automation Research
University of Maryland
College Park, MD, 20742

Baback Moghaddam

Mitsubishi Electric Research Laboratories
201 Broadway
Cambridge, MA 02139

ABSTRACT

Intra-personal space modeling proposed by Moghaddam *et. al.* has been successfully applied in face recognition. In their work the regular principal subspaces are derived from the intra-personal space using a principal component analysis and embedded in a probabilistic formulation. In this paper, we derive the principal subspace from the intra-personal kernel space by developing a probabilistic analysis of kernel principal components for face recognition. We test this new algorithm on a subset of the FERET database with illumination and facial expression variations. The recognition performance demonstrates its advantage over other traditional subspace approaches.

1. INTRODUCTION

Subspace representations have been widely used for face recognition task. For a recent review on face recognition, refer to [4]. Among them, two famous examples are the 'Eigenface' [18] and 'Fisherface' [3, 7] approaches. The 'Eigenface' approach derives its subspace from a principal component analysis (PCA) while the 'Fisherface' approach from a Fisher discriminant analysis (FDA). Both approaches attained satisfactory performances in the FERET test as documented in [14].

Recently, there is an increasing trend of applying kernel subspace representations to face recognition [19, 20, 11], where kernel methods such as the kernel PCA (KPCA) [15] and the kernel FDA (KFDA) [2], corresponding to the 'kernelized' versions of the PCA and the FDA respectively, are invoked to derive the subspace. By mapping the original data into a high-dimensional, or even infinite-dimensional feature space, the kernel methods are able to capture higher-order statistical dependencies, which typically abound in human facial images captured under different scenarios with variations in pose, illumination and facial expression, etc. However, the computation involved in the kernel methods is still maintained almost at the same level as that in the non-kernel methods, as guaranteed by the 'kernel trick'. This feature space is known as the reproducing kernel Hilbert space (RKHS) [15].

In this paper, we investigate a 'kernelized' version of the intra-personal space (IPS) algorithm, which was originally proposed by Moghaddam *et. al.* [12]. An intra-personal space is constructed by collecting all the difference images between any two image pairs belonging to the same individual, to capture all intra-personal variations. Using the PCA, the IPS is decomposed into two subspaces, a principal subspace and an error residual subspace and these two subspaces are embedded in a probabilistic formulation. However, the PCA only accounts for the second-order statistics of the IPS and the role of the higher-order statistics of the IPS is not clear. This paper attempts to address this issue by replacing the PCA with the KPCA. However, this replacement is nontrivial as the ordinary KPCA does not accomodate a probabilistic analysis. We propose a probabilistic analysis of the kernel principal components, which integrates a probabilistic PCA (PPCA) [17] into the KPCA.

This paper is structured as follows. In section 2, we demonstrate the importance of the intra-personal space by comparing it with regular subspace algorithms. We review the relevant theoretical issues regarding the KPCA in section 3, and present a probabilistic analysis of the KPC's in section 4. Section 5 applies the proposed algorithm to a subset of the FERET database [14] and presents the obtained experimental results. Section 6 concludes the paper.

1.1. Notations

x is a scalar, x a vector, and X a matrix. X^{T} represents the matrix transpose and $\mathtt{tr}(\mathsf{X})$ the matrix trace. $\mathtt{I}_m$ denotes an $m \times m$ identity matrix. $\mathbf{1}$ denotes a vector or matrix of ones. $\mathtt{D}[a_1, a_2, \ldots, a_m]$ means an $m \times m$ diagonal matrix with diagonal elements $a_1, a_2, \ldots, a_m$. $\mathtt{p}(.)$ is a general probability density function. $\mathtt{N}(\mu, \Sigma)$ means a normal density with a mean μ and a covariance matrix Σ.

[*]Partially supported by the DARPA/ONR Grant N00014-03-1-0520.

2. INTRA-PERSONAL SPACE MODELING

2.1. Intra-Personal Space(IPS)

In this paper, we are only interested in testing the generalization capability of our algorithm. It is our hope that the training stage can learn the intrinsic characteristics of the target space. We follow [14] to define three sets, namely the training, gallery and probe sets. There is no overlap between the training set and the gallery set in terms of the identity.

Assume that in the training set each class c, $c = 1, ..., C$, possesses J_c observations, indexed by $j_c = 1, ..., J_c$. In total we have $N = \sum_{c=1}^{C} J_c$ images $\{\mathsf{x}_{c,j_c}\}$ in the training set. Typically $N < d$ where d is the number of pixels in one image. Note that, when the class information is not important in a particular context, we simply drop the notation c and denote the training set by $\mathsf{X}_{d\times N} = [\mathsf{x}_1, ..., \mathsf{x}_N]$. The gallery set is denoted by $\mathsf{Y}_{d\times N} = [\mathsf{y}_1, ..., \mathsf{y}_M]$, where each individual m, $m = 1, ..., M$, possess only one observation y_m. The recognition algorithm determines the identity of a given probe image z as $\hat{m}$ among the set $\{1, ..., M\}$.

Regular subspace algorithms for face recognition proceed as follows:

- In the training stage, from the training set X, q basis vectors ($q < N$) forming a *subspace projection* matrix $\mathsf{U}_{d\times q} = [\mathsf{u}_1, \cdots, \mathsf{u}_q]$ in $\mathcal{R}^d$ are learned such that the new representation $\hat{\mathsf{X}} = \mathsf{U}^{\mathsf{T}}\mathsf{X}$ satisfies certain properties. Different properties give rise to different kinds of analysis methods such as the PCA, the FDA, and the independent component analysis (ICA) [8].

- In the testing stage, the algorithms usually determine the identity of the probe z as follows:

$$\hat{m} = \arg \min_{m=1,...,M} ||\mathsf{U}^{\mathsf{T}}(\mathsf{z} - \mathsf{y}_m)||, \qquad (1)$$

  where $||.||$ is a certain norm metric.

In (1), $\mathsf{z} - \mathsf{y}_m$ plays a crucial role. However, its projection onto the A matrix is not guaranteed to be small even when z and y_m are two members belonging to the same class because the learning algorithm is not geared towards minimizing such distance. This is true even for the FDA as the minimization in the FDA is with respect to the class center, not the class member itself. This is a significant difference between the class center and the class member as pose/illuminatin/expression variations might severely deviate the class member from the class center.

To efficiently capture the characteristic of the difference between class members, Moghaddam *et. al.* [12] introduced the intra-personal space (IPS). The IPS is constructed by collecting all the difference images between any two image pairs belonging to the same individual. The construction of the IPS is meant to capture all the possible intra-personal variations introduced during the image acquisition.

Denote the IPS by Δ. Its construction proceeds as follows: From the training set $\{\mathsf{x}_{c,j_c}; c = 1, ..., C\}$, we can construct $\delta_{c,k_c} = \mathsf{x}_{c,j_{1c}} - \mathsf{x}_{c,j_{2c}}; j_{1c} \neq j_{2c}$. Hence for the same individual c, we have $K_c = J_c(J_c - 1)/2$ difference images. Now, we have reached $\Delta = \{\delta_{c,k_c}; c = 1, ..., C, k_c = 1, ..., K_c\}$, with each δ_{c,k_c} treated as an i.i.d. realization. With the availability of the training sample for the IPS Δ, we can learn a probabilistic density function (PDF) on it, say $\mathsf{p}_\Delta(\mathsf{x})$, where x is an arbitrary point lying in the space Δ. Now, given the gallery set Y and the density $\mathsf{p}_\Delta(\mathsf{x})$, the identity $\hat{m}$ of the probe image z is determined by a maximum likelihood (ML) rule:

$$\hat{m} = \arg \max_{m=1,...,C} \mathsf{p}_\Delta(\mathsf{z} - \mathsf{y}_m). \qquad (2)$$

Similar to the FDA, an extra-personal space (EPS) can be constructed to mimic the between-class difference and then the recognition mechanism follows a maximum a posteriori (MAP) rule. See [12] for details. Therefore, this study can be regarded as a 'generalized' discriminant analysis. However, as commented in [12], using only the IPS modeling does not sacrifice the recognition performance.

2.2. Probabilistic subspace density and probabilistic principal component analysis

In [12], a probabilistic subspace (PS) density p_Δ is used [13]. The probabilistic subspace (PS) density decomposes the data space into two subspaces, a principal subspace and an error residual subspace. Suppose that the covariance matrix of the data space is C, whose eigenpairs are given by $\{(\lambda_i, \mathsf{u}_i)\}_{i=1}^{d}$ with d being the dimensionality of the data space, the PS density is written as:

$$\mathsf{p}_\Delta(\mathsf{x}) = \{\frac{\exp(-\frac{1}{2}\sum_{i=1}^{q}\frac{(\mathsf{u}_i^{\mathsf{T}}\mathsf{x})^2}{\lambda_i})}{(2\pi)^{q/2}\prod_{i=1}^{q}\lambda_i^{1/2}}\}\{\frac{\exp(-\frac{\epsilon^2(\mathsf{x})}{2\rho})}{(2\pi\rho)^{(d-q)/2}}\}, \qquad (3)$$

where $\epsilon^2(\mathsf{x}) = \|x\|^2 - \sum_{i=1}^{q} y_i^2$ is the reconstruction error, and

$$\rho = \frac{1}{d-q}\sum_{i=q+1}^{d}\lambda_i = \frac{1}{d-q}\{\mathsf{tr}(\Sigma) - \sum_{i=1}^{q}\lambda_i\}. \qquad (4)$$

In practice, we cannot compute all eigenpairs due to 'curse of dimensionality'. However, in the PS density, we are only interested in the top q eigenpairs.

It is very interesting to note that the probabilistic PCA (PPCA) [17] is very similar to the PS density. The theory of PPCA is briefly reviewed in Section 4. The key observation is that PPCA relates to the ordinary PCA by the fact that the top q eigenpairs of the covariance matrix are maintained.

We implemented both the PS and PPCA in the experiments and found that their performances were similar. Thus, in the sequel, we use the PPCA instead due to its probabilistic interpretation.

3. KERNEL PRINCIPAL COMPONENT ANALYSIS

3.1. PCA in the feature space

Suppose that $\{\mathsf{x}_1, \mathsf{x}_2, \ldots, \mathsf{x}_N\}$ are the given training samples in the original data space $\mathcal{R}^d$. The KPCA operates in a higher-dimensional feature space $\mathcal{R}^f$ induced by a nonlinear mapping function $\phi : \mathcal{R}^d \rightarrow \mathcal{R}^f$, where $f > d$ and f could even be infinite. The training samples in $\mathcal{R}^f$ are denoted by $\Phi_{f\times N} = [\phi_1, \phi_2, ..., \phi_N]$, where $\phi_n \doteq \phi(\mathsf{x}_n) \in \mathcal{R}^f$. Denote the sample mean in the feature space as

$$\bar{\phi}_0 \doteq N^{-1}\sum_{n=1}^{N}\phi(\mathsf{x}_n) = \Phi\mathsf{s},\ \ \mathsf{s}_{N\times 1} \doteq N^{-1}\mathbf{1}. \quad (5)$$

The covariance matrix in the feature space denoted by $\mathsf{C}_{f\times f}$ is given as

$$\mathsf{C} \doteq N^{-1}\sum_{n=1}^{N}(\phi_n - \bar{\phi}_0)(\phi_n - \bar{\phi}_0)^{\mathsf{T}} = \Phi\mathsf{J}\mathsf{J}^{\mathsf{T}}\Phi^{\mathsf{T}} = \Psi\Psi^{\mathsf{T}}, \quad (6)$$

where

$$\mathsf{J} \doteq N^{-1/2}(\mathtt{I}_N - \mathsf{s}\mathbf{1}^{\mathsf{T}}),\ \ \Psi \doteq \Phi\mathsf{J}. \quad (7)$$

The KPCA performs an eigen-decomposition of the covariance matrix C in the feature space. Due to the high dimensionality of the feature space, we commonly possess insufficient number of samples, i.e., the rank of the C matrix is maximally N instead of f. However, computing eigensystem is still possible using the method presented in [18]. Before that, we first show how to avoid the explicit knowledge of the nonlinear feature mapping.

3.2. Kernel trick

Define

$$\bar{\mathsf{K}} \doteq \Psi^{\mathsf{T}}\Psi = \mathsf{J}^{\mathsf{T}}\Phi^{\mathsf{T}}\Phi\mathsf{J} = \mathsf{J}^{\mathsf{T}}\mathsf{K}\mathsf{J}, \quad (8)$$

where $\mathsf{K} \doteq \Phi^{\mathsf{T}}\Phi$ is the grand matrix or the dot product matrix and can be evaluated using the 'kernel trick'; thus the explicit knowledge of the mapping function ϕ is avoided. Given a kernel function k satisfying

$$k(\mathsf{x}, \mathsf{y}) = \phi(\mathsf{x})^{\mathsf{T}}\phi(\mathsf{y});\ \ \forall \mathsf{x}, \mathsf{y} \in \mathcal{R}^d, \quad (9)$$

the $(i, j)^{th}$ entry of the grand matrix K can be calculated as follows:

$$\mathsf{K}^{ij} = \phi(\mathsf{x}_i)^{\mathsf{T}}\phi(\mathsf{x}_j) = k(\mathsf{x}_i, \mathsf{x}_j). \quad (10)$$

The existence of such kernel functions is guaranteed by the Mercer's Theorem [10]. One example is the Gaussian kernel (or the RBF kernel) which has been widely studied in the literature and the focus of this paper. It is defined as

$$k(\mathsf{x}, \mathsf{y}) = \exp(-(2\sigma^2)^{-1}\|\mathsf{x} - \mathsf{y}\|^2)\ \ \forall \mathsf{x}, \mathsf{y} \in \mathcal{R}^d, \quad (11)$$

where σ controls the kernel width. In this case we have $f = \infty$.

The use of the 'kernel trick' (or kernel embedding) [15] captures high-order statistical information since the ϕ function coming from the nonlinear kernel function is nonlinear. We also note that, as long as the computations of interest can be cast in terms of dot products, we can safely use the 'kernel trick' to embed our operations into the feature space. This is the essence of all kernel methods including this work.

3.3. Computing eigensystem for the C matrix

As shown in [9, 18], the eigensystem for C can be derived from $\bar{\mathsf{K}}$. Suppose that the eigenpairs for $\bar{\mathsf{K}}$ are $\{(\lambda_n, \mathsf{v}_n)\}_{n=1}^{N}$, where λ_n's are sorted in a non-increasing order. We now have

$$\bar{\mathsf{K}}\mathsf{v}_n = \Psi^{\mathsf{T}}\Psi\mathsf{v}_n = \lambda_n\mathsf{v}_n;\ \ n = 1, ..., N. \quad (12)$$

Pre-multiplying (12) by Ψ gives rises to

$$\Psi\Psi^{\mathsf{T}}(\Psi\mathsf{v}_n) = \mathsf{C}(\Psi\mathsf{v}_n) = \lambda_n(\Psi\mathsf{v}_n);\ \ n = 1, ..., N. \quad (13)$$

Hence λ_n is the desired eigenvalue of C, with its corresponding eigenvector $\Psi\mathsf{v}_n$. To get the normalized eigenvector u_n for C, we only need to normalize $\Psi\mathsf{v}_n$.

$$(\Psi\mathsf{v}_n)^{\mathsf{T}}(\Psi\mathsf{v}_n) = \mathsf{v}_n^{\mathsf{T}}\Psi^{\mathsf{T}}\Psi\mathsf{v}_n = \mathsf{v}_n^{\mathsf{T}}\lambda_n\mathsf{v}_n = \lambda_n. \quad (14)$$

So,

$$\mathsf{u}_n = (\lambda_n)^{-1/2}\Psi\mathsf{v}_n,\ \ n = 1, ..., N, \quad (15)$$

In a matrix form (if only top q eigenvectors are retained),

$$\mathsf{U}_q \doteq [\mathsf{u}_1, ..., \mathsf{u}_q] = \Psi\mathsf{V}_q\Lambda_q^{-1/2}, \quad (16)$$

where $\mathsf{V}_q \doteq [\mathsf{v}_1, ..., \mathsf{v}_q]$ and $\Lambda_q \doteq \mathsf{D}[\lambda_1, ..., \lambda_q]$.

It is clear that we are not operating in the full feature space, but in a low-dimensional subspace of it, which is spanned by the training samples. It seems that the modeling capacity is limited by the subspace dimensionality, or by the number of the samples. In reality, it however turns out that even in this subspace the smallest eigenvalues are very close to zero, which means that the full feature space can be further captured by a subspace with an even-lower dimensionality. This motivates the use of the latent model.

4. PROBABILISTIC ANALYSIS OF KERNEL PRINCIPAL COMPONENTS

In this section, we present the theory of probabilistic analysis of kernel principal components, which unifies the PPCA and the KPCA in one treatment. We call this analysis as probabilistic kernel principal component analysis (PKPCA). We then present how to compute the Mahalanobis distance and study its limiting behavior.

4.1. Theory of PKPCA

Probabilistic analysis assumes that the data in the feature space follows a special factor analysis model which relates an f-dimensional data $\phi(\mathsf{x})$ to a latent q-dimensional variable z as

$$\phi(\mathsf{x}) = \mu + \mathsf{W}\mathsf{z} + \epsilon, \tag{17}$$

where $\mathsf{z} \sim \mathsf{N}(0, \mathtt{I}_q)$, $\epsilon \sim \mathsf{N}(0, \rho\mathtt{I}_f)$, and W is a $f \times q$ *loading matrix*. Therefore, $\phi(\mathsf{x}) \sim \mathsf{N}(\mu, \Sigma)$, where $\Sigma = \mathsf{W}\mathsf{W}^\mathsf{T} + \rho\mathtt{I}_f$. Typically, we have $q << N << f$.

The maximum likelihood estimates (MLE's) for μ and W are given by

$$\mu = \bar{\phi}_0 = N^{-1}\sum_{n=1}^{N}\phi(\mathsf{x}_n) = \Phi\mathsf{s},\ \ \mathsf{W} = \mathsf{U}_q(\Lambda_q - \rho\mathtt{I}_q)^{1/2}\mathsf{R}, \tag{18}$$

where R is any $q \times q$ orthogonal matrix, and U_q and Λ_q contain the top q eigenvectors and eigenvalues of the C matrix.

Substituting (16) into (18), we obtain the following:

$$\mathsf{W} = \Psi\mathsf{V}_q\Lambda_q^{-1/2}(\Lambda_q - \rho\mathtt{I}_q)^{1/2}\mathsf{R} = \Psi\mathsf{Q} = \Phi\mathsf{J}\mathsf{Q}, \tag{19}$$

where the $N \times q$ matrix Q is defined as

$$\mathsf{Q} \doteq \mathsf{V}_q(\mathtt{I}_q - \rho\Lambda_q^{-1})^{1/2}\mathsf{R}. \tag{20}$$

Since the matrix $(\mathtt{I}_q - \rho\Lambda_q^{-1})$ in Q is diagonal, additional savings in computing its square root are realized. Without loss of generality, we assume that $\mathsf{R} = \mathtt{I}$.

The MLE for ρ is given as

$$\begin{aligned}\rho &= (f-q)^{-1}\{\mathtt{tr}(\mathsf{C}) - \mathtt{tr}(\Lambda_q)\} \\ &\simeq (f-q)^{-1}\{\mathtt{tr}(\mathsf{K}) - \mathtt{tr}(\Lambda_q)\}.\end{aligned} \tag{21}$$

In (21), the approximation needs the assumption that the remaining eigenvalues are zero. This is a reasonable assumption supported by empirical evidences only when f is finite. When f is infinite, this is doubtful since this always gives $\rho = 0$. In such a case, we temporarily set a manual choice $\rho > 0$. Later we show that we can actually let ρ be zero as a limiting case. However, even if a fixed ρ is used, the optimal estimate for W is still same as in (20). It is interesting that (21) is the same as (4).

Now, the covariance matrix is given by

$$\Sigma = \Phi\mathsf{J}\mathsf{Q}\mathsf{Q}^\mathsf{T}\mathsf{J}^\mathsf{T}\Phi^\mathsf{T} + \rho\mathtt{I}_f = \Phi\mathsf{A}\Phi^\mathsf{T} + \rho\mathtt{I}_f, \tag{22}$$

where A is a $N \times N$ matrix given by

$$\mathsf{A} \doteq \mathsf{J}\mathsf{Q}\mathsf{Q}^\mathsf{T}\mathsf{J}^\mathsf{T} = \mathsf{J}\mathsf{V}_q(\mathtt{I}_q - \rho\Lambda_q^{-1})\mathsf{V}_q^\mathsf{T}\mathsf{J}^\mathsf{T}. \tag{23}$$

This offers a regularized approximation to the covariance matrix $\mathsf{C} = \Phi\mathsf{J}\mathsf{J}^\mathsf{T}\Phi^\mathsf{T}$. Especially the top q eigenvalues/vectors of the Σ and C matrices are equivalent [1]. Another approximation often seen in the literature is $\Sigma = \mathsf{C} + \rho\mathtt{I}_f$. However, this approximation changes the eigenvalues while leaving the eigenvectors unchanged. It is interesting to note that Tipping [16] used a similar technique to approximate the covariance matrix C as $\Sigma = \Phi\mathsf{J}\mathsf{D}\mathsf{J}^\mathsf{T}\Phi^\mathsf{T} + \rho\mathtt{I}_f$, where D is a diagonal matrix with many diagonal entries being zero, i.e., D is rank deficient. This can be interpreted in our approach since in our computation $\mathsf{D} = \mathsf{Q}\mathsf{Q}^\mathsf{T}$ is also rank deficient. However, we do not enforce D to be a diagonal matrix. Also, Tipping's approximation might change both the eigenvalues and eigenvectors.

A useful matrix denoted by $\mathsf{M}_{q\times q}$, which can be thought as a 'reciprocal' matrix for Σ is defined as

$$\mathsf{M} \doteq \rho\mathtt{I}_q + \mathsf{W}^\mathsf{T}\mathsf{W} = \rho\mathtt{I}_q + \mathsf{Q}^\mathsf{T}\mathsf{K}\mathsf{Q}. \tag{24}$$

If (20) is substituted into (24), it is easy to show (refer to the Appendix) that $\mathsf{M} = \Lambda_q$.

4.2. Mahalanobis distance

Given a vector $\mathsf{y} \in \mathcal{R}^d$, we are often interested in computing the Mahalanobis distance (see Sec. 5) $\mathsf{L}(\mathsf{y}) \doteq (\phi(\mathsf{y}) - \bar{\phi}_0)^\mathsf{T}\Sigma^{-1}(\phi(\mathsf{y}) - \bar{\phi}_0)$. Firstly, Σ^{-1} is computed as

$$\begin{aligned}\Sigma^{-1} &= (\rho\mathtt{I}_f + \mathsf{W}\mathsf{W}^\mathsf{T})^{-1} = \rho^{-1}(\mathtt{I}_f - \mathsf{W}\mathsf{M}^{-1}\mathsf{W}^\mathsf{T}) \\ &= \rho^{-1}(\mathtt{I}_f - \Phi\mathsf{J}\mathsf{Q}\mathsf{M}^{-1}\mathsf{Q}^\mathsf{T}\mathsf{J}^\mathsf{T}\Phi^\mathsf{T}) \\ &= \rho^{-1}(\mathtt{I}_f - \Phi\mathsf{B}\Phi^\mathsf{T}),\end{aligned} \tag{25}$$

where B is an $N \times N$ matrix given by (refer to Appendix)

$$\mathsf{B} = \mathsf{J}\mathsf{Q}\mathsf{M}^{-1}\mathsf{Q}^\mathsf{T}\mathsf{J}^\mathsf{T} = \mathsf{J}\mathsf{V}_r(\Lambda_r^{-1} - \rho\Lambda_r^{-2})\mathsf{V}_r^\mathsf{T}\mathsf{J}^\mathsf{T}. \tag{26}$$

Then, the Mahalanobis distance is calculated as follows:

$$\begin{aligned}\mathsf{L}(\mathsf{y}) &= (\phi(\mathsf{y}) - \bar{\phi}_0)^\mathsf{T}\Sigma^{-1}(\phi(\mathsf{y}) - \bar{\phi}_0) \\ &= \rho^{-1}\{g_\mathsf{y} - \mathsf{h}_\mathsf{y}^\mathsf{T}\mathsf{B}\mathsf{h}_\mathsf{y}\},\end{aligned} \tag{27}$$

where g_y and h_y are defined by:

$$g_\mathsf{y} \doteq (\phi(\mathsf{y}) - \bar{\phi}_0)^\mathsf{T}(\phi(\mathsf{y}) - \bar{\phi}_0) = k(\mathsf{y},\mathsf{y}) - 2k_\mathsf{y}^\mathsf{T}\mathsf{s} + \mathsf{s}^\mathsf{T}\mathsf{K}\mathsf{s}, \tag{28}$$

[1] In fact, the remaining eigenvectors are unchanged though those eigenvalues are changed.

$$\mathsf{h_y} \doteq \Phi^{\mathsf{T}}(\phi(\mathsf{y}) - \bar{\phi}_0) = k_{\mathsf{y}} - \mathsf{Ks}, \quad (29)$$

$$k_{\mathsf{y}} \doteq \Phi^{\mathsf{T}}\phi(\mathsf{y}) = [k(\mathsf{x}_1, \mathsf{y}), ..., k(\mathsf{x}_N, \mathsf{y})]^{\mathsf{T}}. \quad (30)$$

We now observe that when ρ approaches zero, the quantity $\rho\mathsf{L}(\mathsf{y})$ has a limit $\hat{\mathsf{L}}(\mathsf{y})$ given by

$$\hat{\mathsf{L}}(\mathsf{y}) = g_{\mathsf{y}} - \mathsf{h}_{\mathsf{y}}^{\mathsf{T}}\hat{\mathsf{B}}\mathsf{h}_{\mathsf{y}}, \quad (31)$$

where

$$\hat{\mathsf{B}} = \mathsf{JQM}^{-1}\mathsf{Q}^{\mathsf{T}}\mathsf{J}^{\mathsf{T}} = \mathsf{J}\mathsf{V}_r\Lambda_r^{-1}\mathsf{V}_r^{\mathsf{T}}\mathsf{J}^{\mathsf{T}}. \quad (32)$$

Notice that this limiting Mahalanobis distance does not depend on the choice ρ. Thus, we use this limiting Mahalanobis distance in the followup experiments. Also, this also closes the loop for using a zero ρ.

5. EXPERIMENTAL RESULTS ON FACE RECOGNITION

We perform face recognition using a subset of the FERET database [14] with 200 subjects only. Each subject has 3 images: (a) one taken under controlled lighting condition with neutral expression; (b) one taken under the same lighting condition as above but with different facial expressions (mostly smiling); and (c) one taken under different lighting condition and mostly with a neutral expression. Fig. 1 shows some face examples in this database. All images are pre-processed using zero-mean-unit-variance operation and manually registered using the eye positions.

We randomly divide the 200 objects into two sets, with one set for training and the other one for testing. We focus on the effects of two different variations in facial expression and illumination. For one particular variation, say illumination variation, we use 200 images belonging to the first 100 subjects as the training set for learning and the remaining 200 images as the gallery and probe sets for testing, with images in the category (a) as the gallery set, and those in the category (c) as the probe set. This random division is repeated 20 times and we take their averages as the final result.

We perform our probabilistic analysis of kernel principal components on the IPS. This actually derives the intrapersonal kernel subspace as shown in section 4. It turns out that (2) is equivalent to

$$\hat{m} = \arg \min_{m=1,...,C} \hat{\mathsf{L}}(\mathsf{z} - \mathsf{y}_m), \quad (33)$$

where $\mathsf{L}(.)$ is the Mahalanobis distance defined in (27).

For comparison, we have implemented the following eight methods: the PKPCA and the PPCA [17] with the IPS modeling, the KFDA [2] and the FDA [6], the KPCA [15] and the PCA [18], and the kernel ICA (KICA) [1] and the ICA [8]. For the PKPCA/IPS and the PPCA/IPS, the IPS is constructed based on the training set and the PKPCA/PPCA density is fitted on top of that. For the KPCA, the PCA, the KICA and the ICA, all 200 training images are regarded lying in one face space (FS) and then the learning algorithms are applied on that FS. For the KFDA and the FDA, the identity information of the training set is employed.

Table 1 lists the recognition rate, averaging those of 20 simulations, using the top 1 match. The PKPCA/IPS algorithm attains the best performance since it combines the discriminative power of the IPS modeling and the merit of the PKPCA. As mentioned earlier, using the PS density with the IPS modeling produces the same results as PPCA/IPS. Also, using the dual IPS/EPS modeling does not further improve the results. Compared to the PPCA/IPS, the improvement is not significant, indicating that second-order statistics might be enough after the IPS modeling for the face recognition problem. However, using the PKPCA may be more effective since it also takes into account the higher-order statistics besides the second-order ones. Another observation is that variations in illumination are easier to model than facial expression using subspace methods.

6. CONCLUSION

In this paper, we illustrated the importance of the intrapersonal space for a recognition problem. Then, we proposed a probabilistic analysis of kernel principal components and computed the Mahalanobis distance and its limiting distance. Finally, we have applied this proposed probabilistic approach with IPS modeling to a face dataset and highlighted its advantages. A final note is that our analysis is quite general and is applicable to other learning and recognition tasks.

7. REFERENCES

[1] F. Bach and M. I. Jordan. Kernel independent component analysis. *Technical Report CSD-01-1166, Computer Science Division, University of California, Berkeley*, 2001.

[2] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12(10):2385–2404, 2000.

[3] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. PAMI*, 19, 1997.

[4] R. Chellappa, C. L. Wilson, and S. Sirohey. Human and machine recognition of faces, a survey. *Proceedings of IEEE*, 83:705–740, 1995.

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. Roy. Statist. Soc. B*, 1977.

[6] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, 2001.

Fig. 1. Top row: neutral faces. Middle row: faces with facial expression. Bottom row: faces under different illumination. Image size is 24 by 21 in pixels.

	PKPCA/IPS	PPCA/IPS	KFDA	FDA	KPCA	PCA	KICA	ICA
Expression	79%	78%	73%	72%	64%	68%	61%	53%
Illumination	84%	82%	65%	75%	52%	73%	61%	57%

Table 1. The recognition rates.

[7] K. Etemad and R. Chellappa. Discriminant analysis for recognition of human face images. *Journal of Optical Society of America A*, pages 1724–1733, 1997.

[8] A. Hyvarinen. Survey on independent component analysis. *Neural Computing Surveys*, 2:94–128, 1999.

[9] M. Kirby and L. Sirovich. Application of karhunen-loéve procedure of the characterization of human faces. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 12(1):103–108, 1990.

[10] J. Mercer. Functions of positive and negative type and their connection with the thoery of integral equations. *Philos. Trans. Roy. Soc. London*, A 209:415–446, 1909.

[11] B. Moghaddam. Principal manifolds and probabilistic subspaces for visual recognition. *IEEE Trans. PAMI*, 24(6):780–788, 2002.

[12] B. Moghaddam, T. Jebara, and A. Pentland. Bayesian modeling of facial similarity. *Advances in Neural Information Processing System*, 1998.

[13] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Trans. PAMI*, 19(7):696–710, 1997.

[14] P. J. Philipps, H. Moon, S. Rivzi, and P. Ross. The FERET evaluation methodology for face-recognition algorithms. *IEEE Trans. PAMI*, 22:1090–1104, 2000.

[15] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[16] M. Tipping. Sparse kernel prinicipal component analysis. *NIPS*, 2001.

[17] M. Tipping and C. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61(3):611–622, 1999.

[18] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neutoscience*, 3:72–86, 1991.

[19] M.-H. Yang. Face recognition using kernel methods. *NIPS*, 2001.

[20] M.-H. Yang. Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods. *Proc. of Intl. Conf. on Face and Gesture Recognition*, 2002.

8. APPENDIX – SOME USEFUL COMPUTATIONS

8.1. Computation related to M

We first compute $\mathsf{Q}^\mathsf{T}\bar{\mathsf{K}}\mathsf{Q}$ and then M.

$$\begin{aligned}&\mathsf{Q}^\mathsf{T}\bar{\mathsf{K}}\mathsf{Q}\\ &= (\mathtt{I}_r - \rho\Lambda_r^{-1})^{1/2}\mathsf{V}_r^\mathsf{T}\bar{\mathsf{K}}\mathsf{V}_r(\mathtt{I}_r - \rho\Lambda_r^{-1})^{1/2}\\ &= (\mathtt{I}_r - \rho\Lambda_r^{-1})^{1/2}\Lambda_r(\mathtt{I}_r - \rho\Lambda_r^{-1})^{1/2}\\ &= \Lambda_r - \rho\mathtt{I}_r,\end{aligned} \tag{34}$$

where the fact that $\mathsf{V}_r^\mathsf{T}\bar{\mathsf{K}}\mathsf{V}_r = \mathsf{V}_r^\mathsf{T}\mathsf{J}^\mathsf{T}\mathsf{K}\mathsf{J}\mathsf{V}_r = \Lambda_r$ is used. Therefore,

$$\mathsf{M} = \rho\mathtt{I}_r + \mathsf{Q}^\mathsf{T}\bar{\mathsf{K}}\mathsf{Q} = \rho\mathtt{I}_r + (\Lambda_r - \rho\mathtt{I}_r) = \Lambda_r. \tag{35}$$

$$|\mathsf{M}| = |\Lambda_r| = \prod_{i=1}^{q}\lambda_i,\ \ \mathsf{M}^{-1} = \Lambda_r^{-1}. \tag{36}$$

8.2. Computation related to A and B

$$\begin{aligned}\mathsf{A} &= \mathsf{J}\mathsf{Q}\mathsf{Q}^\mathsf{T}\mathsf{J}^\mathsf{T}\\ &= \mathsf{J}\mathsf{V}_r(\mathtt{I}_r - \rho\Lambda_r^{-1})^{1/2}(\mathtt{I}_r - \rho\Lambda_r^{-1})^{1/2}\mathsf{V}_r^\mathsf{T}\mathsf{J}^\mathsf{T}\\ &= \mathsf{J}\mathsf{V}_r(\mathtt{I}_r - \rho\Lambda_r^{-1})\mathsf{V}_r^\mathsf{T}\mathsf{J}^\mathsf{T}\end{aligned} \tag{37}$$

$$\begin{aligned}\mathsf{B} &= \mathsf{J}\mathsf{Q}\mathsf{M}^{-1}\mathsf{Q}^\mathsf{T}\mathsf{J}^\mathsf{T}\\ &= \mathsf{J}\mathsf{V}_r(\mathtt{I}_r - \rho\Lambda_r^{-1})^{1/2}\Lambda_r^{-1}(\mathtt{I}_r - \rho\Lambda_r^{-1})^{1/2}\mathsf{V}_r^\mathsf{T}\mathsf{J}^\mathsf{T}\\ &= \mathsf{J}\mathsf{V}_r(\Lambda_r^{-1} - \rho\Lambda_r^{-2})\mathsf{V}_r^\mathsf{T}\mathsf{J}^\mathsf{T}\end{aligned} \tag{38}$$