# Principal Components Null Space Analysis for Image and Video Classification

Namrata Vaswani, *Member, IEEE,* and Rama Chellappa, *Fellow, IEEE*

***Abstract*—We present a new classification algorithm, principal component null space analysis (PCNSA), which is designed for classification problems like object recognition where different classes have unequal and nonwhite noise covariance matrices. PCNSA first obtains a principal components subspace (PCA space) for the entire data. In this PCA space, it finds for each class "$i$," an $M_i$-dimensional subspace along which the class' intraclass variance is the smallest. We call this subspace an approximate null space (ANS) since the lowest variance is usually "much smaller" than the highest. A query is classified into class "$i$" if its distance from the class' mean in the class' ANS is a minimum. We derive upper bounds on classification error probability of PCNSA and use these expressions to compare classification performance of PCNSA with that of subspace linear discriminant analysis (SLDA). We propose a practical modification of PCNSA called progressive-PCNSA that also detects "new" (untrained classes). Finally, we provide an experimental comparison of PCNSA and progressive PCNSA with SLDA and PCA and also with other classification algorithms—linear SVMs, kernel PCA, kernel discriminant analysis, and kernel SLDA, for object recognition and face recognition under large pose/expression variation. We also show applications of PCNSA to two classification problems in video—an action retrieval problem and abnormal activity detection.**

## I. Introduction

WITHIN the last several years, many algorithms have been proposed for object and face recognition problems; for a detailed survey, see [3] and [4]. While much progress has been made toward recognizing faces under small variations in lighting and pose, reliable techniques for more extreme variations and for the more difficult image classification problems like object recognition have proved elusive. For classification problems like face recognition, different classes have similar class covariance matrices (in particular, similar directions of low and high intraclass variance; see Fig. 5) while, for problems like object recognition (for example, the COIL database; see Fig. 4), the different classes can have very different class covariance matrix structures. As an extreme case of this situation, the minimum variance direction of one class could be a maximum variance direction for another. We propose, in this paper, a subspace based classification algorithm, called principal components null space analysis (PCNSA), for this situation of unequal covariance matrices.

Manuscript received March 2, 2004; revised March 29, 2005. This work was supported in part by ARDA/VACE under Contract 2004H840200000. Parts of this paper were also presented at ICPR 2002 [1] and ICPR 2004 [2]. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Truong T. Nguyen.

N. Vaswani is with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011 USA (e-mail: namrata@iastate.edu).

R. Chellappa is with the Department of Electrical and Computer Engineering and the Center for Automation Research, University of Maryland, College Park, MD 20742 USA (e-mail: rama@cfar.umd.edu).

Digital Object Identifier 10.1109/TIP.2006.873449

### A. Related Work

The two linear subspace algorithms to which PCNSA is most closely related are principal component analysis (PCA) [5] and subspace linear discriminant analysis (SLDA) [6], but these are both optimal for problems with similar directions of minimum and maximum variance (made precise in Section I-B). PCA [5] yields projection directions that maximize the total scatter but do not minimize the within class variance of each class and also sometimes retains directions with unwanted large variations due to variation in lighting etc. linear discriminant analysis (LDA) [7] encodes discriminatory information by finding directions that maximize the ratio of between class scatter to within-class (or intraclass) scatter. In subspace LDA (SLDA) [6], PCA and LDA are combined to yield a classification algorithm for face recognition which uses PCA first for dimensionality reduction and then LDA. Subspace LDA is also used in [8] for view based image retrieval. Independent component analysis (ICA) [9] is a generalization of PCA which searches for a linear transformation to express the given data as a linear combination of statistically independent source variables, but like PCA, ICA is actually optimal for data representation and not classification, and, hence, we do not discuss it in this work.

A support vector machine (SVM) [10] is another linear two class classifier which finds a separating hyperplane between the training data of the two classes such that the "margin" (worst case distance of either class from the separating hyperplane) is maximized, while keeping all training data correctly classified. For data which is not strictly linearly separable, it finds the hyperplane that maximizes a sum of the number of points correctly classified and the margin. Many strategies to extend SVMs to multiclass classification have been proposed. In [11], the authors use the rules of a tennis tournament to classify 32 objects from the COIL-100 database. We compare results of PCNSA with that of [11] in Section VII-A.

Note that, even though PCNSA utilizes linear subspaces for classification, its classification boundaries are not hyperplanes, i.e., the set $\{Y \in \mathbb{R}^P : d_1(Y) = d_2(Y)\}$ where $d_i(Y)$ is defined in (5) is not a hyperplane. The reason for this is that the PCNSA classification distance defined in (5), unlike other linear algorithms, uses different subspaces for different classes. Two other subspace based classification algorithms which also share this property are BiasMap [12] and multispace KL (MKL) [13]. BiasMap [12] performs a class specific LDA for a two class problem, i.e., for the set of "positive" samples it finds a direction that maximizes their distance from the "negative" samples and

minimizes their within class variance. We discuss in Section V the relation between MKL and our algorithm and how our error probability analysis can be extended to analyze MKL.

Several nonlinear classification methods have been proposed in the literature. In [14], Murase and Nayar propose a representation of object appearance in the PCA space parameterized by pose and illumination. Each object class is represented in the PCA space by a B-spline interpolated manifold. A query image is recognized based on the manifold that it is closest to in the PCA space. Mao and Jain [15] describe neural network algorithms for PCA, LDA, Sammon's nonlinear projection, and nonlinear discriminant analysis. Kernel PCA [16] and Kernel discriminant analysis (KLDA) [17]–[19] use the "kernel trick" [17] to transform nonlinearly separable data into a higher dimensional space, called the "feature space," where it becomes linearly separable. The feature space, $F$, obtained by the mapping $\phi : \mathbb{R}^P \rightarrow F$, may even be infinite dimensional, but since PCA or LDA projections can be written as inner products, these can be evaluated without explicitly projecting the data into the feature space [16], [17]. We define a kernel PCNSA method in Section VI-C and compare its performance with kernel LDA and kernel PCA in Section VII.

### B. Problem Formulation

Consider a $P$-dimensional data sample $\mathbf{Y}$ from class $i$ (denote class $i$ by $C_i$). Then

$$(\mathbf{Y})_{P\times 1}|\{\mathbf{Y}\in C_i\}\sim\mathcal{N}(\mu_{\text{full,i}},\Sigma_{\text{full,i}}) \tag{1}$$

where $\mu_{\text{full,i}}$ and $\Sigma_{\text{full,i}}$ are the class conditional mean and covariance of $\mathbf{Y}$. For high-dimensional data-like images, the real dimensionality of data (with noise removed) is much smaller than $P$. Thus, we first perform PCA which, as explained below, attempts to remove directions with only noise and retain directions with large between class variance [6]. PCA takes data from all classes as a single sample set and evaluates the common mean, $\bar{\mu}_{\text{full}}$, and common covariance matrix, $\bar{\Sigma}_{\text{full}}$. It chooses the $L$ leading eigenvectors of $\bar{\Sigma}_{\text{full}}$ as the principal component subspace (PCA space). Given that the data sample, $\mathbf{Y}$, belongs to class $i$, its projection in the $L$-dimensional PCA space with projection matrix, $(W^{\text{PCA}})_{P\times L}$, is distributed as

$$\begin{aligned}(\mathbf{X})_{L\times 1} &\triangleq {W^{\text{PCA}}}^T(\mathbf{Y}-\bar{\mu}_{\text{full}})\sim\mathcal{N}(\mu_i,\Sigma_i)\quad\text{where}\\ (\mu_i)_{L\times 1} &\triangleq {W^{\text{PCA}}}^T(\mu_{\text{full,i}}-\bar{\mu}_{\text{full}})\\ (\Sigma_i)_{L\times L} &\triangleq {W^{\text{PCA}}}^T\Sigma_{\text{full,i}}W^{\text{PCA}}.\end{aligned} \tag{2}$$

In this paper, we address the classification problem for the most general class covariance matrices (unequal, nonwhite) in the PCA space with eigenvalue decomposition $\Sigma_i = U_i\Lambda_i U_i^T$ where $U_i$ is the matrix of eigenvectors arranged in decreasing order of eigenvalues and $\Lambda_i$ is the diagonal matrix of eigenvalues. We propose an algorithm called PCNSA, which first performs PCA on the entire data set and, then, for each class $i$, finds the directions of least within class covariance.

*1) Need for PCA:* The total scatter matrix $\bar{\Sigma}_{\text{full}} = \text{cov}(\mathbf{Y})$ can be written as $\bar{\Sigma}_{\text{full}} = \Sigma_{\text{full,w}} + \Sigma_{\text{full,b}}$ [17] where $\Sigma_{\text{full,w}} = (1/K)\sum_{i=1}^{K}\Sigma_{\text{full,i}}$ is the average within class covariance matrix and $\Sigma_{\text{full,b}} = (1/K)\sum_{i=1}^{K}(\mu_{\text{full,i}}-\bar{\mu}_{\text{full}})(\mu_{\text{full,i}}-\bar{\mu}_{\text{full}})^T$ is the between class covariance. PCA finds the principal eigenvectors of $\bar{\Sigma}_{\text{full}}$. Under the assumption that the total within class variance is much smaller than the between class variance, these are also approximately the principal eigenvectors of $\Sigma_{\text{full,b}}$, i.e., PCA approximately finds directions $W$ along which $\text{Trace}(W^T\Sigma_{\text{full},b}W)$ is maximized. Thus, in PCA space, $(1/K)\sum_{i=1}^{K}\|\mu_i-\bar{\mu}\|^2$ [with $\mu_i$ defined in (2)] is the maximum over all possible choices of $W$.[1] Since $\|\mu_i-\bar{\mu}\| = \|(1/K)\sum_{j=1}^{K}(\mu_i-\mu_j)\| \leq (1/K)\sum_{j=1}^{K}\|\mu_i-\mu_j\|$, this also implies that the average $\|\mu_i-\mu_j\|$ is large in the PCA space or that means are well separated in PCA space. Note that while the assumption of within class variance being much smaller than between class variance may not hold in all directions, linear separability requires it to hold in some directions at least. By keeping enough PCA projections, one ensures that the PCA space does contain directions of large between class variance. One way to ensure this would be to keep taking more PCA directions until the total between class variance in the PCA space is more than a certain percentage of the total between class variance.

*2) Relation to LRT:* The likelihood ratio test (LRT) [20] (maximum likelihood solution) for this problem is to choose the class $c$ as

$$c = \arg\min_i d_i^{\text{LRT}}(\mathbf{X}),\ \ d_i^{\text{LRT}}(\mathbf{X}) = (\mathbf{X}-\mu_i)^T\Sigma_i^{-1}(\mathbf{X}-\mu_i). \tag{3}$$

The PCA distance, $d_i^{\text{PCA}}(\mathbf{X}) = \|\mathbf{X}-\mu_i\|$ is equivalent to the LRT under the assumption that $\Sigma_i = \sigma^2 I$ (which implies that $U_i = I$, i.e., principal eigenvectors of every class covariance matrix are the same as the PCA directions and have equal eigenvalues). Also, SLDA approximates the LRT when $\Sigma_i \approx \Sigma$ and $\Sigma$ is ill conditioned (has an approximate null space). Thus, both PCA and SLDA are suited for classification problems where classes have similar within class variance directions.

Assuming that the within class covariance matrices, $\Sigma_i$, are ill conditioned (happens very often in real applications), the dominant terms in the LRT expression (3) are those along the trailing eigenvectors of $\Sigma_i$. Hence, under this assumption, $d_i^{\text{LRT}}(\mathbf{X}) \approx d_i^{\text{PCNSA}}(\mathbf{X})$ where $d_i^{\text{PCNSA}}(\mathbf{X})$ is defined in (5).

### C. Paper Organization

The rest of the paper is organized as follows. The PCNSA algorithm and assumptions required for it are discussed in Section II. Bounds on its classification error probability are derived in Section III. These error probability bounds are used to compare performance of PCNSA with that of SLDA[2] in Section IV. We also discuss conditions under which PCNSA would outperform SLDA and when it would fail. The connection with Multispace KL [13] is discussed in Section V. New class detection and some modifications of PCNSA are discussed in Section VI. Experimental results on image and video classification problems—object recognition, face recognition under large pose variations, action video retrieval, and abnormal activity detection—are given in Section VII. Performance of PCNSA is compared with that of SLDA, PCA, SVMs, kernel PCA, and SLDA. Conclusions and future directions are discussed in Section VIII.

[1] $\bar{\mu}$ in the PCA space is actually zero, i.e., $\bar{\mu} = {W^{\text{PCA}}}^T(\bar{\mu}_{\text{full}}-\bar{\mu}_{\text{full}}) = 0$.

[2] In the entire paper, we use SLDA and LDA interchangeably both always refer to SLDA.

## II. Principal Components Null Space Analysis

PCNSA first performs PCA on the entire data for dimensionality reduction and to retain directions of large between class variance [6] (discussed above in Section I-B). In PCA space, it finds for each class $i$, an $M_i$-dimensional subspace along which the class' intraclass variance is smallest. We call this subspace the *approximate null space (ANS)* of class $i$ since for most applications, the lowest variance(s) are usually "much smaller" than the highest (the class covariance matrix is usually ill conditioned). A query is classified into class $i$ if its distance from the class' mean in the class' ANS is a minimum. We first discuss below the assumptions required for PCNSA to work as a classification algorithm (have low classification error probability) and then provide the stepwise algorithm.

### A. Assumptions

1) For all classes $i$, the class covariance matrix, $\Sigma_i$, has a high enough condition number, i.e., $R_i = \lambda_{\max,i}/\lambda_{\min,i} > \gamma_1$ (where $\lambda_{\max,i}$ and $\lambda_{\min,i}$ are the maximum and minimum eigenvalues of $\Sigma_i$), with $\gamma_1$ large. This ensures that an approximate null space (ANS) exists. The within class covariance matrix is ill conditioned for most real classification problems.
2) Any class $i$ is linearly separable from all other classes $j \neq i$ in its own ANS. A sufficient condition for this is: The distance between class means in the ANS space of any class $i$, denoted by $N_i$, is at least $\gamma_2$ times the square root of $M_i$ times the maximum eigenvalue of any other class $j$, i.e., $\|N_i^T(\mu_i - \mu_j)\| \geq \gamma_2\sqrt{M_i\lambda_{\max,j}}\ \forall i \neq j$. Here, $M_i$ is the dimension of the ANS of class $i$.

Note that, as we shall see later, for low error probability, we either need $\gamma_2$ to be large (say $\gamma_2 = 3$) or we need $\sqrt{\gamma_1}\gamma_2$ to be large (e.g., $\gamma_1 = 10^6$, $\gamma_2 = 0.1$ can also work).

### B. Algorithm [1]

1) **Obtain PCA Space**: Evaluate the sample mean, $\bar{\mu}_{\text{full}}$ and covariance, $\bar{\Sigma}_{\text{full}}$ of the training data of all classes taken together as one sample set. Obtain the PCA projection matrix, $(W^{\text{PCA}})_{P\times L}$ whose columns are the $L$ leading eigenvectors of $\bar{\Sigma}_{\text{full}}$. We discuss the choice of $L$ in Section VII-A.
2) Project the training data samples of each class into PCA space. Evaluate for each class $i$, the class mean, $\mu_i$, and the class covariance, $\Sigma_i$, in PCA space.
3) **Obtain Class ANS**: Evaluate the approximate null space, $(N_i)_{L\times M_i}$, for each class $i$ as the $M_i$ trailing eigenvectors of $\Sigma_i$ (choose $M_i$ so that the eigenvalues in ANS satisfy, $\lambda \leq (1/\gamma_1)\lambda_{\max}$, $\gamma_1 = 10^4$), where $\lambda_{\max} = \max_i \lambda_{\max,i}$. Assumption 1 ensures that it exists.
4) **Obtain Valid Classification Directions in ANS**: Let $N_i = [e_{i,1}|e_{i,2}|\ldots e_{i,k}\ldots|e_{i,M_i}]$. A null space direction, $e$, is a valid classification direction if $|e^T(\mu_i - \mu_j)| > \gamma_2\sqrt{e^T\Sigma_j e}$. If assumption 2 holds, it guarantees that this is always possible to do for pairs of classes.[3] Thus, the PCNSA projection matrix for class $i$ ($W_i^{\text{NSA}}$) is chosen as those columns, $e$, of $N_i$ which satisfy

$$\min_{j\neq i} \frac{\left|e^T(\mu_i - \mu_j)\right|}{\sqrt{e^T\Sigma_j e}} > \gamma_2. \tag{4}$$

Note, in practice, the above may not be satisfied for any one direction when the number of classes is large but it is still possible to find a subset of directions $N_i$ that satisfy assumption 2. This idea forms the basis of progresive PCNSA discussed in Section VI-B.
5) **Classification**: Project the query $\mathbf{Y}$ into the PCA space as $\mathbf{X} = W^{\text{PCA}^{\text{T}}}(\mathbf{Y} - \bar{\mu}_{\text{full}})$. PCNSA chooses the query class to be $c = \arg\min_i d_i(\mathbf{X})$ where

$$d_i(\mathbf{X}) \triangleq \left\|W_i^{\text{NSA}^T}(\mathbf{X} - \mu_i)\right\|. \tag{5}$$

[3]Assumption 2 is equivalent to $\sum_{k=1}^{M_i}[(e_{i,k}^T(\mu_i - \mu_j))^2 - \gamma_2^2\lambda_{\max,j}] > 0$. Now, $\sum_k \beta_k > 0$ implies that there exists at least one $k$ for which $\beta_k \triangleq [(e_{i,k}^T(\mu_i - \mu_j))^2 - \gamma_2^2\lambda_{\max,j}] > 0$. Since $\lambda_{\max,j} > e_{i,k}^T\Sigma_j e_{i,k}$, this implies that $e_{i,k}$ is a valid direction.

## III. Two Class Classification Error Probability

We obtain the error probability bound for classification using PCNSA for a two class problem. We first evaluate the error probability assuming Gaussian distributed classes (each class has a Gaussian class conditional distribution) and a one-dimensional (1-D) ANS per class so that $W_i^{\text{NSA}} = (N_i)_{L\times 1}$. We then show how this can be extended to the general case of Gaussian distributed classes and $M_i$-dimensional ANS per class. We discuss in Section III-C how the error probability analysis can be extended to non-Gaussian distributions. The two class error probability expressions can be used to obtain a union bound [20] for the multiclass error probability.

### A. One-Dimensional ANS Per Class, Gaussian Distributions

We assume a Gaussian class conditional distribution of the query $\mathbf{X}$ in this and the next subsection. Define $E_i$ as the event that error occurs given query $\mathbf{X} \in C_i$ (class $i$). The average error probability assuming that both classes are equally likely, is $P_{e,\text{avg}} = (P(E_1) + P(E_2))/2$. Using PCNSAs class specific metric defined in (5), the error event $E_1$ is

$$E_1 \triangleq \left\{d_2^2(\mathbf{X}) < d_1^2(\mathbf{X}) | \boldsymbol{X} \in C_1\right\}. \tag{6}$$

Since, ANS is 1-D, $W_1^{\text{NSA}} = N_1$ and $d_1(\mathbf{X}) = |N_1^T(\mathbf{X} - \mu_1)|$ is a scalar. Then, we have the following theorem [1], [2].

*Theorem 1:* The error probability $P(E_1)$ is upper bounded as

$$\begin{aligned} P(E_1) &\leq \min_{k>0}\left[\Phi\left(\frac{\alpha_2^1 + \Delta_1}{\sigma_2^1}\right) - \Phi\left(\frac{\alpha_2^1 - \Delta_1}{\sigma_2^1}\right) + 2\left(1 - \Phi(k)\right)\right] \\ &= \min_{k>0}\left[\int_{\frac{\alpha_2^1}{\sigma_2^1}\left(1+\frac{k\sqrt{\lambda_{\text{ANS},1}}}{\alpha_2^1}\right)}^{\frac{\alpha_2^1}{\sigma_2^1}\left(1-\frac{k\sqrt{\lambda_{\text{ANS},1}}}{\alpha_2^1}\right)} \mathcal{N}(z;0,1)dz\right] \\ &\quad + 2\int_k^\infty \mathcal{N}(z;0,1)dz \end{aligned} \tag{7}$$

$$\text{where} \quad \alpha_2^1 \triangleq \left|N_2^T(\mu_2-\mu_1)\right|, \quad \sigma_2^1 \triangleq \sqrt{N_2^T\Sigma_1 N_2} \tag{8}$$

$$\text{and} \quad \Delta_1 = k\sqrt{\lambda_{\text{ANS},1}} \tag{9}$$

and $\Phi$ is the cdf of a standard normal ($\mathcal{N}(0,1)$) random variable. Symmetric expressions can be obtained for $P(E_2)$.

*Proof:* See Appendix

### B. $M_i$-Dimensional ANS Per Class, Gaussian Distributions

In this case, $N_1$ and $N_2$ are $L\times M_i, i=1,2$-dimensional matrices. The error upper bounds are stated in the theorem below.

*Theorem 2:* Let

$$\Delta_1 \triangleq k\sqrt{\left(\sum_{j=1}^{M_1}\lambda_{\text{ANS},1,j}^2\right)}, \quad \beta_2^1 \triangleq N_2^T(\mu_2-\mu_1)$$

$$\text{and} \quad \Sigma_2^1 \triangleq N_2^T\Sigma_1 N_2 \tag{10}$$

and let the eigenvalue decomposition of $\Sigma_2^1$ be $\Sigma_2^1 = US_2^1U^T$. Then, defining $(\sigma_{2,j}^1)^2 = (S_2^1)_{j,j}$ and $\alpha_2^1 = |U^T\beta_2^1|$ where $|.|$ is component-wise magnitude, the error probability $P(E_1)$ is upper bounded as

$$P(E_1) \le \min_{k>0}\left[\prod_{j=1}^{M_2}\left[\Phi\left(\frac{\alpha_{2,j}^1+\Delta_1}{\sigma_{2,j}^1}\right)-\Phi\left(\frac{\alpha_{2,j}^1-\Delta_1}{\sigma_{2,j}^1}\right)\right] + \left[1-(2-2\Phi(k))^{M_1}\right]\right]$$

$$= \min_{k>0}\left[\left[\prod_{j=1}^{M_2}\int_{\frac{\alpha_{2,j}^1-\Delta_1}{\sigma_{2,j}^1}}^{\frac{\alpha_{2,j}^1+\Delta_1}{\sigma_{2,j}^1}}\mathcal{N}(z;0,1)dz\right] + \left[1-\left(\int_{-k}^{k}\mathcal{N}(z;0,1)dz\right)^{M_1}\right]\right]. \tag{11}$$

Symmetric expressions can be obtained for $P(E_2)$.

*Proof:* See Appendix.

### C. Extension to Non-Gaussian Distributions

The analysis for 1-D ANS can be extended to the case of non-Gaussian distributions.[4] Assume that the distribution of $\mathbf{X}$ has mean $\mu_1$, and has covariance matrix $\Sigma_1$. Let $F_1(.)$ be the cumulative distribution function (cdf) and $f_1(.)$ the probability distribution function (pdf) of $\mathbf{Z}_1 \triangleq (N_1^T(X-\mu_1))/\sqrt{N_1^T\Sigma_1 N_1}$, i.e., it is the cdf of $N_1^T\mathbf{X}$ after location normalization to zero mean and scale normalization to unit variance. Similarly, let $F_2(.)$ and $f_2(.)$ be the cdf and pdf of $\mathbf{Z}_2 \triangleq (N_2^T(X-\mu_1))/\sqrt{N_2^T\Sigma_1 N_2}$. Then, $P(d_1^2(\mathbf{X}) > \Delta_1^2) = P(\mathbf{Z}_1 > k \text{ or } \mathbf{Z}_1 < -k) = (1-F_1(k)) + F_1(-k)$. Also, $P(d_2^2(X) \le \Delta_1^2))$ is defined as explained in the Appendix (proof of Theorem 1), with $\Phi$ replaced by $F_2$. Using (30), we get

$$P(E_1) \le P\left(d_2^2(\mathbf{X}) \le \Delta_1^2\right) + P(d_1^2(\mathbf{X}) > \Delta_1^2) = \begin{cases} T^+ & \text{if } N_2^T(\mu_2-\mu_1) \ge 0 \\ T^- & \text{if } N_2^T(\mu_2-\mu_1) < 0 \end{cases} \tag{12}$$

where

$$T^+ \triangleq \min_{k>0}\left[F_2\left(\frac{\alpha_2^1+k\sqrt{\lambda_{\text{ANS},1}}}{\sigma_2^1}\right)-F_2\left(\frac{\alpha_2^1-k\sqrt{\lambda_{\text{ANS},1}}}{\sigma_2^1}\right) + (1-F_1(k))+F_1(-k)\right] \tag{13}$$

$$T^- \triangleq \min_{k>0}\left[F_2\left(\frac{-\alpha_2^1+k\sqrt{\lambda_{\text{ANS},1}}}{\sigma_2^1}\right)-F_2\left(\frac{-\alpha_2^1-k\sqrt{\lambda_{\text{ANS},1}}}{\sigma_2^1}\right) + (1-F_1(k))+F_1(-k)\right] \tag{14}$$

where $\alpha_2^1$ and $\sigma_2^1$ are defined in (8). If the distribution $F_2$ is symmetric about zero, the two different cases in the equation above will be equal. If $F_1$ is symmetric then the last two terms of (13) and of (14) add up to $2(1-F_1(k))$ (like in the Gaussian case). If $f_1$ and $f_2$ are unimodal and not heavy tailed, and the assumptions of Section II-A are true, the error probability bound can be shown to be small (by repeating the analysis of Section IV-B).

## IV. Comparison With Subspace LDA (SLDA)

We first explain the Subspace LDA algorithm and its classification error probability in the next subsection. A qualitative and quantitative performance comparison of PCNSA with SLDA is given in Section IV-B. We also compare the training data size requirement, ability to detect untrained classes and computational complexity in later subsections.

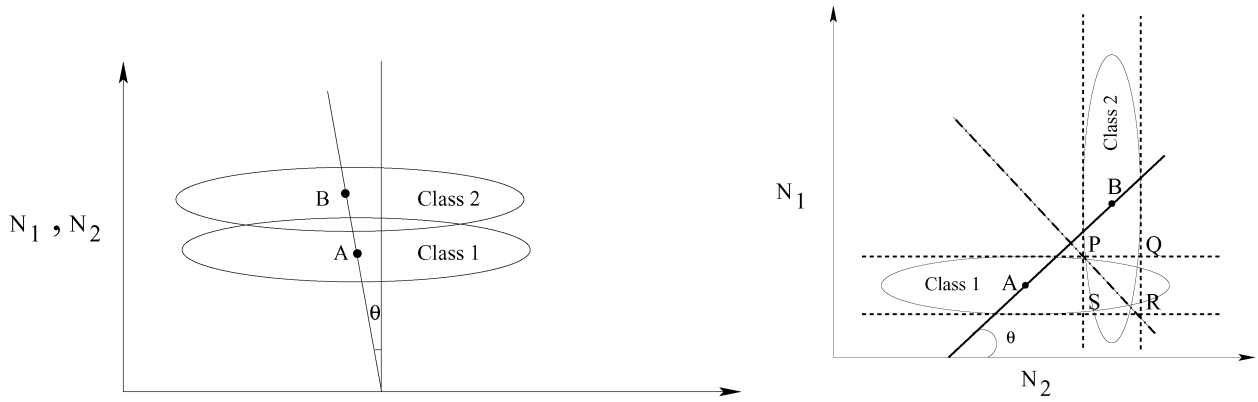
### A. Subspace Linear Discriminant Analysis (SLDA)

As discussed in Section I, SLDA [6] first computes a PCA space for the training data of all classes taken together as one sample. In PCA space, it performs linear discriminant analysis, i.e., it computes the most discriminant directions $W^{\text{LDA}}$ as

$$W^{\text{LDA}} = \arg\max_W \frac{(W^T\Sigma_b W)}{(W^T\Sigma_w W)} \tag{15}$$

where $\Sigma_b = (\sum_{i=1}^K(\mu_i-\bar{\mu}))/K$ and $\Sigma_w = (\sum_{i=1}^K\Sigma_i)/K$. The solution of (15) is obtained by finding the principal eigenvectors of the generalized eigenvalue problem $\Sigma_b W = \Lambda\Sigma_w W$, and,

[4]The $M_i$-dimensional ANS analysis is more difficult to extend because it hinges on the assumption that dependent Gaussian variables can be made independent by a linear transformation.




(a) Case 1 ($\psi = 0^o$) with $\theta \approx 0^o$ (b) Case 2 ($\psi = 90^o$) with $\theta \approx 45^o$

Fig. 1. $\theta$ is the angle between the line AB and the $Y$ axis in (a) and between AB and the $X$ axis in (b). Case 1 with ANS directions ($Y$ axis) of both classes coinciding is shown in (a). Case 2 where the $Y$ axis is ANS for class 1 and maximum variance direction for class 2; vice versa for the $X$ axis, shown in (b). (a) Case 1 ($\psi = 0°$) with $\theta \approx 0°$. (b) Case 2 ($\psi = 90°$) with $\theta \approx 45°$.

hence, the maximum number of LDA directions for a $K$ class problem is limited by the rank of $\Sigma_b$ which is $(K-1)$. The LDA classification metric is

$$d_i(\mathbf{X}) = \left\| W^{\mathrm{LDA}^T}(\mathbf{X} - \mu_i) \right\|. \tag{16}$$

The error event for a two class problem (1-D $W^{\mathrm{LDA}}$) is $E_1 \triangleq \{d_2^2(\mathbf{X}) < d_1^2(\mathbf{X}) | \mathbf{X} \in C_1\}$. The error probability follows directly using Gaussian hypothesis testing [20]:

$$P(E_1) = 1 - \Phi\left(\frac{\hat{\alpha}}{\hat{\sigma}^1}\right) = \int_{\frac{\hat{\alpha}}{\hat{\sigma}^1}}^{\infty} \mathcal{N}(z;0,1)dz \quad \text{where}$$

$$\hat{\alpha} \triangleq \frac{\left| W^{\mathrm{LDA}^T}(\mu_2 - \mu_1) \right|}{2}$$

$$\hat{\sigma}^1 \triangleq \sqrt{W^{\mathrm{LDA}^T}\Sigma_1 W^{\mathrm{LDA}}}. \tag{17}$$

This results has also been discussed in [21]. Now, the above analysis can also be extended to situations where $\mathbf{X}$, and, hence, $W^{\mathrm{LDA}^{\mathrm{T}}}\mathbf{X}$ have a non-Gaussian distribution. Let $(W^{\mathrm{LDA}^{\mathrm{T}}}(\mathbf{X} - \mu_1))/\sqrt{W^{\mathrm{LDA}^{\mathrm{T}}}\Sigma_1 W^{\mathrm{LDA}}}$ have a cdf $F_1$ and pdf $f_1$. Then for nonsymmetric distribution, $F_1$ we have two cases: If $W^{\mathrm{LDA}^{\mathrm{T}}}(\mu_2 - \mu_1) > 0$, $P(E_1) = 1 - F_1(\hat{\alpha}/\hat{\sigma}_1) = \int_{\hat{\alpha}/\hat{\sigma}^1}^{\infty} f_1(z)dz$. If $W^{\mathrm{LDA}^{\mathrm{T}}}(\mu_2 - \mu_1) < 0$, then $P(E_1) = F_1(-(\hat{\alpha}/\hat{\sigma}_1))$. If the distribution $F_1$ is symmetric about the origin, then both cases are equal. Once again, $f_1$ unimodal and not heavy tailed is required for small error.

### B. Classification Performance Comparison

We analyze the error probability expressions (7) and (17). First, note that in (7), the second term reduces very fast as $k$ increases, e.g., if $k = 3$, $g(k) \triangleq 2(1 - \Phi(k)) = 0.0027$ and if $k = 10$, $g(k) = 10^{-23}$. Now, in (7), if $\sqrt{\lambda_{\mathrm{ANS},1}}/\alpha_2^1$, and also $\sqrt{\lambda_{\mathrm{ANS},2}}/\alpha_1^2$ (for $P(E_2)$) tend to zero, the lower and upper limits of the first integral tend to each other, and, hence, the first term tends to zero. Choosing $k = 10$, the second term, and, hence, the total error probability bound is of the order of $10^{-23}$, but if this does not hold, i.e., if either of $\sqrt{\lambda_{\mathrm{ANS},1}}$ or $\sqrt{\lambda_{\mathrm{ANS},2}}$ are of the order of $\alpha_2^1$ or $\alpha_1^2$ (ANS space does not exist for either class), then choosing $k = 10$ makes the first term of (7) comparable to the LDA error expression (17). In this situation, to make (7) zero, one requires $\alpha_2^1/\sigma_2^1$ and $\alpha_1^2/\sigma_1^2$ to go to infinity. Also, for (17) to go to zero, $\hat{\alpha}/\hat{\sigma}_1$ and $\hat{\alpha}/\hat{\sigma}_2$ need to go to infinity.

SLDA evaluates $W^{\mathrm{LDA}}$ to maximize the between class variance in the PCA space, while also minimizing the average within class variance, so that $\hat{\alpha}/\hat{\sigma}_1$ is large. Also, assumption 2 (step 4 of the algorithm in Section II-B) ensures that $\alpha_i^j/\sigma_i^j > \gamma_2$ (large)[5] while assumptions 1 and 2 together (steps 3 and 4 of the algorithm) ensure that $\sqrt{\lambda_{\mathrm{ANS},i}}/\alpha_j^i < 1/\sqrt{\gamma_1}\gamma_2$ (small).[6] Thus, PCNSA error will be small if either $\gamma_2$ is large or $\sqrt{\gamma_1}\gamma_2$ is large.

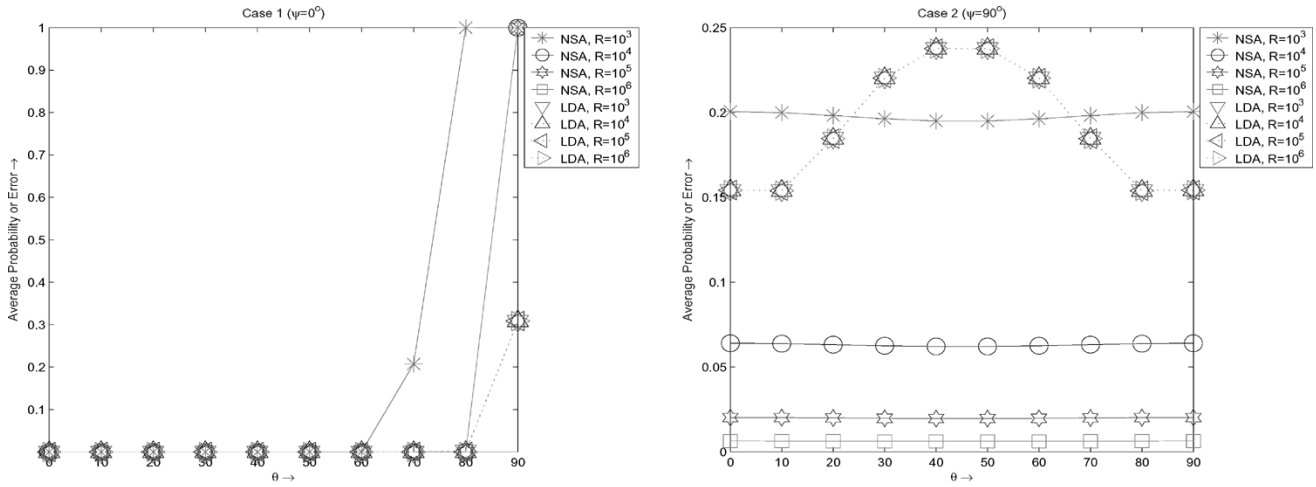
We now discuss some example situations which demonstrate when PCNSA outperforms SLDA and vice versa. First, we make some assumptions to reduce the number of variables to analyze. We consider the two situations shown in Fig. 1 and study the error probability variation as the angle $\theta$ is varied between zero and 90°, and the logarithm of the condition number is varied between 3 and 7.

*1) Simplifying Assumptions:* We assume a two-dimensional PCA space and each class having a 1-D ANS and one direction of maximum variance. Also, we assume that the eigenvalues of covariance matrices of both classes are equal, i.e., $\lambda_{\max,1} = \lambda_{\max,2} = \lambda_{\max}$ and $\lambda_{\mathrm{ANS},1} = \lambda_{\mathrm{ANS},2} = \lambda_{\min}$. We take $\|\mu_1 - \mu_2\| = \sqrt{\lambda_{\max}}$. With these assumptions, the error probability expressions can be reduced to a function of three variables: the condition number, $R = \lambda_{\max}/\lambda_{\min}$, the angle between $N_1$ and $N_2$, denoted by $\psi$ and the angle made by the vector $(\mu_1 - \mu_2)$ (line joining the means) with $N_2$, denoted by $\theta$. In two dimensions these two angles automatically fix the angle between the direction of $(\mu_1 - \mu_2)$ and $N_1$.

We study the variation of error probability as a function of $R$ and $\theta$ for two extreme values of $\psi$, $\psi = 0°$ (case 1) and $\psi = 90°$ (case 2). We show that PCNSA works well in both these extreme

[5]Since $\sigma_i^{j2} = N_i^T \Sigma_j N_i < \lambda_{\max,j}$ (by definition of maximum eigenvalue), we have $\alpha_i^j/\sigma_i^j \triangleq |N_i^T(\mu - \mu_j)|/\sigma_i^j > |N_i^T(\mu - \mu_j)|/\sqrt{\lambda_{\max,j}}$. Thus, assumption 2 implies that $\alpha_i^j/\sigma_i^j > \gamma_2$.

[6]$\lambda_{\max,j}/\lambda_{\min,j} > \gamma_1$ and $|N_i^T(\mu_i - \mu_j)|/\sqrt{\lambda_{\max,j}} > \gamma_2$ together imply that $(\sqrt{\lambda_{\min,j}}/|N_i^T(\mu_i - \mu_j)|) = (\sqrt{\lambda_{\mathrm{ANS},j}}/\alpha_j^i) < (1/\sqrt{\gamma_1}\gamma_2)$.




(a) Case 1 (b) Case 2

Fig. 2. Average probability of error as a function of $\theta$ for different values of condition number $R$ for (a) Case 1 and (b) Case 2. As can be seen the LDA, error probability does not vary much with $R$ in either case (curves for all $R$ values are coincident) and also does not degrade much as $\theta \rightarrow 90^\circ$.

cases as long as the assumptions of Section II-A are satisfied and fails completely when they are not.

*2) Qualitative Comparison:* We first provide a qualitative comparison of the two cases using Fig. 1(a) ($\psi = 0^\circ$) and (b) ($\psi = 90^\circ$) for small $\theta$. In both figures, the condition number $R$ is set to a large value (assumption 1 of Section II-A). We have $\theta \approx 0$ in Fig. 1(a) and $\theta \approx 45^\circ$ in Fig. 1(b), both being far from 90° (assumption 2 of Section II-A satisfied). Case 1 with $\theta \approx 0$, shown in Fig. 1(a) is a best case scenario for both PCNSA and LDA since $Y$ axis is the ANS direction for both classes and the common LDA direction ($W^{\mathrm{LDA}}$) is also close to the $Y$ axis. Thus, the variances of both classes along $W^{\mathrm{LDA}}$ are small; hence, LDA works very well. Also variances of both classes along the common ANS direction ($Y$ axis) are small and the distance between class means along the $Y$ axis is large. Hence, the performance of PCNSA will also be very good in this case.

However, for case 2 with $\theta \approx 45^\circ$, shown in Fig. 1(b), the maximum variance direction of one class coincides with the ANS of the other. This is the worst case for LDA but PCNSA works very well in this case. In fact, this case demonstrates the need for the PCNSA algorithm. Here, the $Y$ axis is ANS direction for class 1, but a maximum variance direction for class 2 and vice versa for $X$ axis. Thus, $W^{\mathrm{LDA}}$ is along the direction $(\mu_1 - \mu_2)$ (direction AB in the figure). Along $W^{\mathrm{LDA}}$, both classes have a large enough variance. So, LDA has a high error probability in this case. The region for the LDA error event $E_1^{\mathrm{LDA}}$ is the region of ellipse 1 to the right of line PR and for $E_2^{\mathrm{LDA}}$ it is the region of ellipse 2 below line PR, but PCNSA still works well because the integration region for $E_1^{\mathrm{NSA}}$ is only those parts of ellipse 1 that are closer to $\mu_2$ (point B) along $N_2$ ($X$ axis) than to $\mu_1$ (point A) along $N_1$ ($Y$ axis) and similarly for $E_2^{\mathrm{NSA}}$. Thus, the error region is the small overlap region of the two ellipses (region PQRS) for both $E_1^{\mathrm{NSA}}$ and $E_2^{\mathrm{NSA}}$.

*3) Quantitative Comparison—Error Probabilities as a Function of $R$ and $\theta$:* In case 1 ($\psi = 0^\circ$), $N_1 = N_2 = [0\ 1]^T$. Using the simplifying assumptions and definitions (8), $\Sigma_1 = \Sigma_2 = \mathrm{diag}\{\lambda_{\max}, \lambda_{\min}\}$, $\alpha = \sqrt{\lambda_{\max}}\cos\theta$ and $\sigma = \sqrt{\lambda_{\min}}$. The condition number of either class' covariance matrix is $R = \lambda_{\max}/\lambda_{\min}$. Substituting in (7), we get

$$P\left(E_1^{\mathrm{NSA}}\right) \leq \int_{\sqrt{R}\cos\theta - k}^{\sqrt{R}\cos\theta + k} \mathcal{N}(z;0,1)dz \triangleq P\left(E^{\mathrm{NSA\ bound}}\right) \tag{18}$$

and the same expression for $P(E_2^{\mathrm{NSA}})$ so that $P(E_{\mathrm{avg}}^{\mathrm{NSA}}) = P(E_1^{\mathrm{NSA}})$. We also evaluate $P(E^{\mathrm{LDA}})$ using (17). MATLAB is used to evaluate $W^{\mathrm{LDA}}$ for different values of $R$ and $\theta$. Both $P(E^{\mathrm{NSA,bound}})$ and $P(E^{\mathrm{LDA}})$ are plotted in Fig. 2(a), for $\theta \in [0, 90^\circ]$, and $R = 10^3, 10^4, 10^5$. This is a best case scenario for both SLDA and PCNSA as long as $\theta$ is bounded away from 90° (distance between class means along both classes' ANS is nonzero). We have for both NSA and LDA

$$\begin{aligned} &\lim_{R\to\infty} P(E^{\mathrm{NSA/LDA}}, R, \theta) = 0, \quad \forall \quad |\theta| < \theta_0 < 90^\circ \\ &\text{but} \quad \lim_{\theta\to 90^\circ}\lim_{R\to\infty} P(E^{\mathrm{NSA\ bound}}, R, \theta) = 1 \\ &\text{while} \quad \lim_{\theta\to 90^\circ}\lim_{R\to\infty} P(E^{\mathrm{LDA}}, R, \theta) \approx 0.31 \end{aligned} \tag{19}$$

i.e., when $\theta$ tends to 90°, PCNSA fails completely while the performance of LDA degrades gracefully.[7]

Now, in case 2 ($\psi = 90^\circ$), $N_1 \perp N_2$, i.e., $N_1 = [0\ 1]^T$ and $N_2 = [1\ 0]^T$. So $\Sigma_1 = \mathrm{diag}\{\lambda_{\max}, \lambda_{\min}\}$ while $\Sigma_2 = \mathrm{diag}\{\lambda_{\min}, \lambda_{\max}\}$. Again using the simplifying assumptions and (8), $\sigma = \sqrt{N_2^T \Sigma_1 N_2} = \sqrt{\lambda_{\max}}$ and $\alpha = \sqrt{\lambda_{\max}}\cos\theta$. This gives

$$P\left(E_1^{\mathrm{NSA}}\right) \leq \int_{\cos\theta - \frac{k}{\sqrt{R}}}^{\cos\theta + \frac{k}{\sqrt{R}}} \mathcal{N}(z;0,1)dz. \tag{20}$$

[7]The LDA limit is an approximate numerically evaluated value.

For LDA, $\Sigma_w = (\Sigma_1 + \Sigma_2)/2 = \text{diag}\{((\lambda_{\max} + \lambda_{\min})/2),((\lambda_{\max}+\lambda_{\min})/2)\}$ so that $W^{\text{LDA}}$ is along $(\mu_1-\mu_2)$, i.e., $W^{\text{LDA}} = [\cos\theta \sin\theta]^T$. Thus, we have

$$P\left(E_1^{\text{LDA}}\right) = \int_{\frac{\sqrt{R}}{2(\sqrt{R\cos^2\theta+\sin^2\theta})}}^{\infty} \mathcal{N}(z;0,1)dz. \quad (21)$$

The expressions for $P(E_2)$ for both PCNSA and LDA have the "cos" replaced by "sin." Case 2, as also discussed earlier, is the worst case for LDA. The average error probabilities are plotted in Fig. 2(b). The LDA error probability in this case converges to a nonzero value which depends on $\theta$, i.e., we get [using (21)]

$$\lim_{R\to\infty} P(E^{\text{LDA}}, R, \theta) = \frac{\int_{\frac{\sec\theta}{2}}^{\infty} \mathcal{N}(z;0,1)dz + \int_{\frac{\text{cosec}\theta}{2}}^{\infty} \mathcal{N}(z;0,1)dz}{2}, \ \forall\theta. \quad (22)$$

The above limit is approximately the LDA curve (dotted line) shown in Fig. 2(b). PCNSA still works very well in this case, i.e., we have [using (20)]

$$\lim_{R\to\infty} P(E^{\text{NSA}}, R, \theta) = 0 \quad \forall \quad \theta \quad (23)$$

although the rate of convergence is much slower than in case 1. Note that, in this case, $P(E^{\text{NSA}})$ converges to zero even for $\theta = 90°$ or $\theta = 0°$. This is because variance of class 2 along ANS-1 and vice versa is large. Hence, in (7), even when $\alpha = 0$, for $R \to \infty$, we get $\Delta/\sigma = 1/R \to 0$. In Case 1 on the other hand, $\Delta/\sigma = k$, and, hence, it relies on the $\alpha/\sigma$ term going to infinity to minimize error.

*4) Discussion:* Thus, from the above analysis, we conclude that PCNSA fails for small values of $R$ (no approximate null space) or when the distance between class means projected along ANS becomes small ($\theta \to 90°$). We have included checks in steps 3 and 4 of our algorithm to avoid these two situations.

### C. Comparing Size of Training Data

In real applications, the model is never exact and so the ANS calculation is never exact. Finding the approximate null space directions requires a large amount of training data to correctly find directions along which there is almost no variation. The size of the training data set per class should be at least two to three times the dimension of the PCA space to correctly estimate the lowest eigenvalues (and corresponding eigenvectors) of the class covariance matrix. SLDA can do with lesser training data and PCA requires the least. This fact has been observed experimentally and is plotted in Fig. 3. Note the figure is for a facial feature matching problem [22] where training and test data was very different. On the other hand, for the object recognition applications (Section VII-A), even for 36 training samples per class, PCNSA outperformed SLDA and PCA.

### D. Comparing "New" (Untrained) Class Detection Ability

Since PCNSA defines a class specific metric, "new" (untrained) classes can be detected most easily using PCNSA. When a query belongs to a trained class its distance from the class mean along that class' approximate null space is a very sharp minimum while a query belonging to a new class will have no such sharp minimum. This idea has been used in Section VI-A to design a new class detection algorithm. Detecting new classes is more difficult with LDA because trained classes may not have very sharp minimum distances from their own class means along the LDA directions.

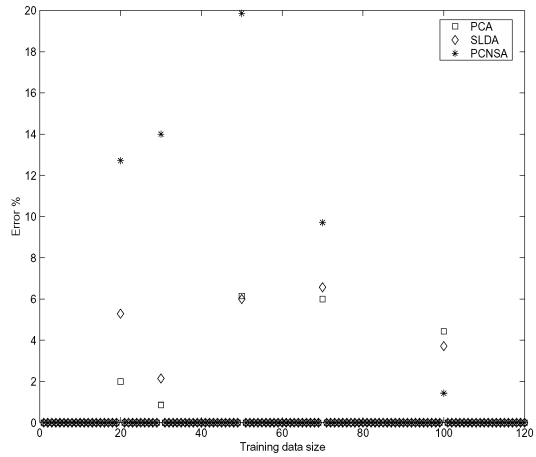



Fig. 3. Error probability variation with reduced training data sizes per class.

Another advantage of PCNSA over SLDA is that the PCNSA class-specific metric does not require any knowledge of the second class and so can be used for binary hypothesis testing problems where the statistics of the alternate hypothesis $(H_1)$ are not known. We have discussed its application to abnormal activity detection (where "abnormality" is not characterized) in Section VII-D.

### E. Comparing Computational Complexity

The extra overhead for obtaining PCNSA or LDA subspace in PCA space (highly reduced dimension data) is negligible compared to the initial principal eigenvectors calculation done on $P$-dimensional data (for images $P$ is the total number of pixels). Since training is done once and offline, this complexity is not very critical while classification is an online process. Query classification time is proportional to the number of inner products (equal to total number of projection directions) to be taken. For a K-class application, LDA requires a maximum of "$K-1$" $P$-dimensional inner products and PCNSA using $M$ ANS directions per class requires "$MK$" $P$-dimensional inner products (this assumes that PCNSA/LDA projection matrix from original $P$-dimensional space to SLDA or PCNSA space, $W^{\text{PCA}}W^{i^{\text{NSA}}}$ and $W^{\text{PCA}}W^{\text{LDA}}$ have been precalculated). If the principal component space is $L$-dimensional, PCA classification requires "$L$" $P$-dimensional inner products. "$L$" will be larger than "$MK$" for most applications. If it is not, one can first project the query to the principal component space ($L$ $P$-dimensional inner products) and then projecting to the ANS space of each class will require negligibly small extra time ("$MK$" extra $L$-dimensional inner products, $L \ll P$). To summarize, classification complexity for PCA is $L$ $P$-dimensional inner products which can be greater than or equal to that for PCNSA ($\min(MK, L)$, assuming M-dimensional ANS) but is always greater than that for LDA $(K-1)$.

The computational complexity of any kernel method is "T + 1" times more (where T is the total number of training samples). Projecting a query into the KPCA space (PCA of feature space), done using the kernel requires "T + 1" $P$-dimensional inner products instead of the usual one $P$-dimensional inner product.

## V. Relation to Multispace KL [13]

Multispace KL [13] is a subspace based classification and representation algorithm which appeared around the same time as our conference paper [1] on PCNSA. When used for classification, MKL can be thought of as a generalization of PCNSA. It separates all classes into subsets of similar classes and for each subset derives a principal component subspace representation. For classification of a query, it first finds the subspace (subset) from which the distance of the query is a minimum and in that subspace finds the class mean that is closest to the query in Euclidean norm. The distance from space defined in [13] is equivalent to the distance in ANS space defined by us. In fact MKL is exactly equivalent to performing null space analysis to choose the nearest subspace (subset) and then using PCA to choose the nearest class within the subset.

We can extend the error probability analysis of Section III to evaluating the classification error probability of MKL. The error in choosing the correct subspace is $P(E^{\mathrm{NSA}})$ with ANS dimension $M_i = n-k$ (Using notation from [13] where $k$ is the subspace dimension and $n$ is the original data dimension). The bound for this error, $P(E^{\mathrm{NSA\ bound}})$, for a two class problem is given by (11). The error in classification within the subspace is the error in classification using the Euclidean distance in PCA space. Thus, classification error (given query belongs to class $i$) using MKL would be

$$\begin{aligned} P\left(E_i^{MKL}\right) &= P\left(E_i^{\mathrm{NSA}}\right) + \left(1 - P\left(E_i^{\mathrm{NSA}}\right)\right) P\left(E_i^{\mathrm{PCA}}\right) \\ &\leq P\left(E_i^{\mathrm{NSA\ union\ bound}}\right) + P\left(E_i^{\mathrm{PCA}}\right). \end{aligned}$$

Now, MKL has been applied to an image retrieval problem in [13]. We can also use PCNSA for retrieval applications. We show in Section VII-C, application of PCNSA to a video retrieval problem from a small database. For a large database retrieval application, using the MKL idea, we can select subsets of classes with similar within-class covariance matrices and obtain ANS for each subset (as in [13]). PCNSA can be used to choose the subset to which the query is closest and LDA to classify within the subset.

## VI. New Class Detection and PCNSA Modifications

### A. New Class Detection

A common problem in most classification applications is to detect when a query does not belong to any of the classes for which the classifier has been trained. In this paper we refer to such a query as belonging to a "new" class. Since PCNSA uses a class-specific metric, its ability to detect "new" classes is better. We use the following idea to develop an algorithm for new class detection: If distances from two or more classes are roughly equal, we conclude that the query belongs to a "new" class. This is because a query will have a very sharp minimum in its own class' ANS and if there is no such sharp minimum, then one can say that it does not belong to any of the trained classes. We classify a query $\mathbf{X}$ as belonging to a "new" class if the minimum distance $(d_c(\mathbf{X}))$ is greater than a threshold $t$ times the distance from any other class $(d_i(\mathbf{X}), i \neq c)$, i.e.,

$$d_c(\mathbf{X}) > t d_i(\mathbf{X}) \quad \forall i \neq c, \quad t < 1 \tag{24}$$

$$\text{or equivalently } \frac{d_c(\mathbf{X})}{\min_{i \neq c} d_i(\mathbf{X})} > t. \tag{25}$$

The value of $t$ governs the false alarm and miss probabilities. If we define $H_0$ as the hypothesis that the query belongs to one of the $K$ trained classes and $H_1$ as the hypothesis that it belongs to an untrained ("new") class, then false alarm is the event that the algorithm decides in favor of $H_1$ ("new" class) when actually $H_0$ is true (query comes from a trained class) [23]. The value of $t$ can be set based on the requirements of the application, if it can tolerate false alarms but is sensitive to misses, $t$ is set to a small value. We vary the value of $t$ between 0 and 1 and plot the ROC curves (plot of new class detection probability against probability of false alarm, both evaluated experimentally) [20] for the different algorithms in Section VII.

### B. Progressive-PCNSA

Progressive-PCNSA is a modification of the PCNSA algorithm which chooses the number of ANS directions on the fly. In practice, when the number of classes is large, quite often, there is no one single direction of the ANS of class $i$ which satisfies assumption 2 of Section II-A for all $j \neq i$. As a practical solution to this problem, we vary the dimension of ANS of all classes between a value $M_{\mathrm{low}}$ to $M_{\mathrm{high}}$ (choice of $M_{\mathrm{low}}$, $M_{\mathrm{high}}$ discussed in Section VII-A) and evaluate the ratio given in the left hand side of (25) for each value of $M$. The stepwise classification procedure is as follows.

1) Vary ANS dimension from $M = M_{\mathrm{low}}, M_{\mathrm{low}} + 1, \ldots M_{\mathrm{high}}$. For each value of $M$:
   - — evaluate $d_i^M(\mathbf{X})$ for all classes using (5) and with $W_i^{\mathrm{NSA}}$ the $M$ trailing eigenvectors of $\Sigma_i$;
   - — find the minimum distance $d_c^M(\mathbf{X})$ and the corresponding class $c^M$;
   - — evaluate the ratio in the left hand side of (25).
2) Find the minimum value of the ratio and the corresponding ANS dimension $M_{best}$. If this minimum value is less than $t$, the class $c^{M_{best}}$ is the chosen class. If the minimum value is greater than $t$, then (25) is satisfied and so the query is declared as coming from a new class.

### C. Kernel PCNSA

Kernel PCNSA can be performed by performing Null Space Analysis in KPCA space instead of the PCA space, i.e., K-PCNSA finds an Approximate Null Space of the KPCA space. Now PCNSA requires the assumption that each class can be linearly separated from all other classes in its own ANS. The motivation for K-PCNSA (similarly to that for KPCA, KDA, or KSLDA) is that this assumption may not be satisfied in the PCA space of the data but by projecting the data into a higher dimensional "feature" space and taking its principal components, the assumption will (hopefully) be satisfied. Progressive-PCNSA can be implemented similarly.

### D. Combining PCNSA and SLDA Using Error Probabilities

We propose to combine PCNSA and SLDA using the error probability expressions derived in Sections III and IV-A. Since the expressions are for two classes at a time, we use the tennis tournament strategy described in [11] for classifying multiple classes. The algorithm is as follows.

1) In the training stage, for every pair of classes $i, j$.
   — Evaluate the two class error probabilities using PCNSA and SLDA and choose the algorithm with smaller error probability.
   — Also, store the corresponding SLDA direction, $W^{\mathrm{LDA}^{i,j}}$, or the best PCNSA classification directions, $W_i^{\mathrm{NSA}^j}$, $W_j^{\mathrm{NSA}^i}$ (obtained using step 4 of Section II-B) for the class pair.
2) Given a query, the tennis tournament strategy [11] is as follows. In the first round, the query is classified into one of every pair of classes using the algorithm chosen for the pair in the training stage. Thus, at the end of the first round, the query belongs to one of $\mathrm{floor}(K/2)+1$ classes. In the next round, the same algorithm is repeated to choose $\mathrm{floor}(K/2+1)+1$ classes and so on. Thus, after a total of $\mathrm{floor}(\log(K))+1$ rounds, the query is classified. The classification complexity of this algorithm is only as much as that for PCNSA.

## VII. Experimental Comparison

We have compared the performance of progressive-PCNSA and PCNSA with that of PCA, SLDA, KPCA, KLDA, K-SLDA, and SVMs for object recognition (COIL database [24]) and face recognition under large pose variation (UMIST [25] and AT&T databases [26]). We experimented with the following kernel choices—the Polynomial kernel, $k(x,y) = (a(x.y) + b)^d$, the Gaussian kernel, $k(x,y) = e^{-\|x-y\|^2/\sigma^2}$ and the recently proposed cosine kernels [19]. A cosine kernel can be defined for any Mercer kernel [17], $k(x,y)$, as $k_c(x,y) = k(x,y/\sqrt{k(x,x)k(y,y)}$ and is also shown to be a Mercel kernel [19]. The motivation for this kernel comes from the fact that similarity measures based on the cosine (normalized inner product) measurement should be more reliable than the inner product measurement [19].

In all experiments, we treat one image (arranged as a column vector) as one sample. We have also shown the superior performance of PCNSA for new class detection by leaving a few classes untrained and testing for data from those classes. There are three kinds of classification errors.

— Misclassification error given $H_0$: A query from trained class $i$ gets wrongly classified as trained class $j, i \neq j$.
— False Alarm (Type I error) given $H_0$: A query from any "trained" class gets wrongly classified as "new."
— Miss (Type II error) given $H_1$: A query from a "new" class gets wrongly classified as some "trained" class. New class detection probability is $P(\mathrm{Detect}) = 1 - P(\mathrm{Miss})$.

We varied the value of the new class detection threshold, $t$, between 0 and 1 and plotted the ROC curves for different algorithms. We also show application of PCNSA to action retrieval and abnormal activity detection.


Fig. 4. Object recognition: Some samples from the COIL-100 database.

### A. Image Classification: Object Recognition

We tested our algorithm on the Columbia Object Image Library (COIL-100) database [24] (shown in Fig. 4) which contains 100 different objects and 72 views of each object taken at five-degree-apart orientations. We compare the performance of prog-PCNSA and PCNSA with that of linear SVMs [10], [11], SLDA [6] and PCA [5] and also with KPCA [16], KDA [17], [18], and KSLDA performed with Gaussian and Cosine polynomial [19] kernels. To compare with SVM results on the COIL-100 database, we repeated the experimental setup discussed in [11]. A set of 32 classes was chosen randomly and 20 such iterations were run every time choosing a different set. 36 of the 72 images of each class were used for training and the other 36 for testing. The original $128 \times 128$ images were resampled to $32 \times 32$. Under this setup, a 0% error is reported with linear SVMs [11]. We show the misclassification error probability in Table I. As can be seen, Progressive-PCNSA had a 0.16% error while SLDA had a much higher 2.1% error. When applying kernel methods, the best results for all algorithms were obtained with the Gaussian kernel. We tried to implement both KDA and KSLDA but KDA had an error of 89% even when classifying between just ten classes, and, hence, we show results only with KSLDA.

For prog-PCNSA and PCNSA we used a 15-dimensional PCA space. The PCA space dimension can be chosen by retaining a large percentage, say 80%, of the total energy, but since we need to find the trailing eigenvectors in PCA space, we need PCA space to be small enough so that the training data size per class is at least 2–3 times the PCA space dimension. We choose the PCA space dimension to be $L = 15$ so that the training data size per class, $36 = 2.4 * L$. The SLDA results were obtained with a 100-dimensional PCA space (larger PCA space needed to be able to correctly estimate directions of low average within class variance and high between class variance) and 31 LDA directions. The SLDA results with a 15-dimensional PCA space were much worse with an error of 13.4% without kernels. For choosing the ANS dimension (for simplicity, we used the same ANS dimension for all classes), we ran a sequence of iterations to compare performance of PCNSA for increasing ANS dimension. We show the plot in Fig. 6. Based on this plot, we took $M_i = M = 4$ for PCNSA and $M_{\mathrm{low}} = 4$ and $M_{\mathrm{high}} = 9$ for prog-PCNSA.

TABLE I
OBJECT RECOGNITION (COIL DATABASE, 32 CLASSES): MISCLASSIFICATION ERROR PROBABILITY. THE FIRST ROW SHOWS RESULTS WITHOUT USING ANY KERNELS FOR ALL ALGORITHMS. THE SECOND AND THIRD ROWS SHOW RESULTS WITH THE COSINE POLYNOMIAL KERNEL OF DEGREE 10 AND THE GAUSSIAN KERNEL. WE HAVE HIGHLIGHTED THE BEST KERNEL CHOICE FOR EACH ALGORITHM BY UNDERLINING AND SHOWING THE ERROR IN BOLD

| Kernel Type | Prog-PCNSA | PCNSA | SLDA | PCA | SVM |
|---|---|---|---|---|---|
| $k(x,y) = x^T y$ (No Kernel) | 0.0016 | 0.0060 | 0.0211 | 0.4338 | 0.000 |
| Cosine Polynomial, d=10 | 0.0066 | 0.0154 | 0.0197 | 0.2526 | |
| Gaussian, $\sigma^2 = 4000$ | **0.0013** | **0.0042** | **0.0125** | **0.2190** | |

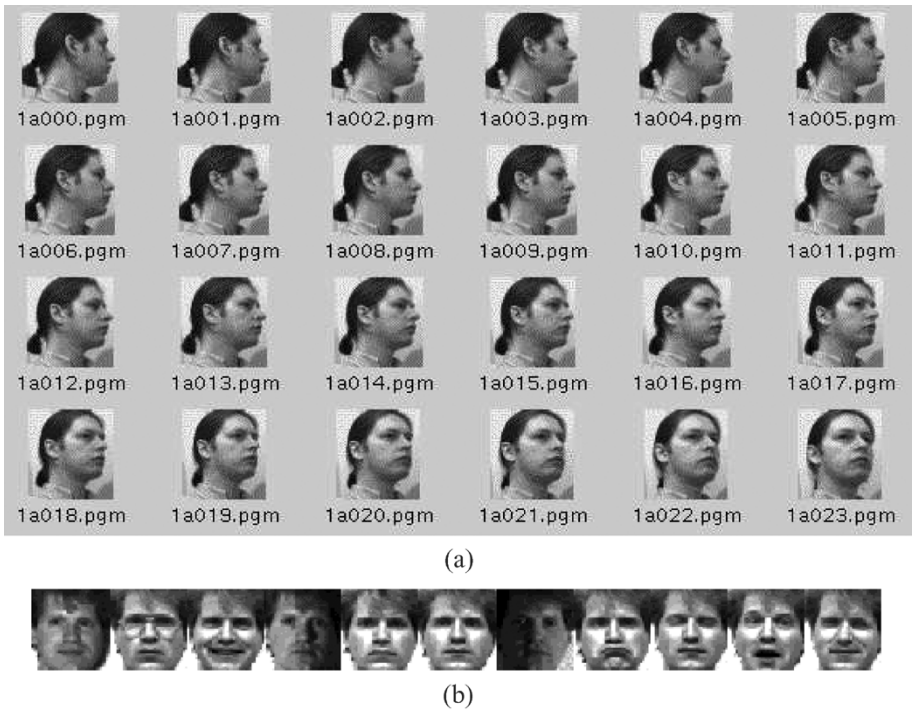



Fig. 5. Face recognition databases: (a) 23 different face poses used for each face from the UMIST face database; (b) ten facial expressions used for each face from the AT&T Cambridge face database (formerly, ORL face database).

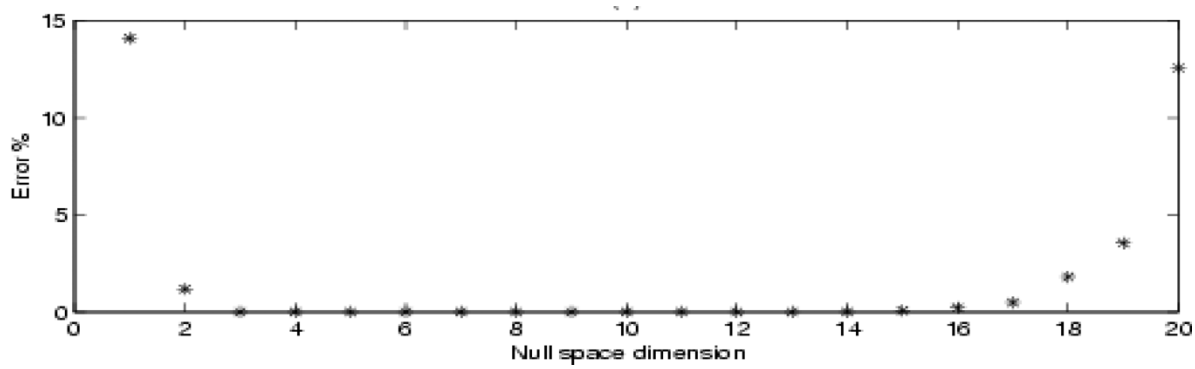



Fig. 6. Plot showing variation in error probability (shown as a percentage) with varying ANS dimension for the object recognition problem.

In Fig. 7, we show the ROC curves to compare new class detection ability of the algorithms without any kernel in Fig. 7(a), with the Cosine polynomial kernel with $d = 10$ in Fig. 7(b) and the Gaussian kernel in Fig. 7(c). Thirty-two trained classes and ten untrained ("new") classes were used for testing. As can be seen, prog-PCNSA has the best performance followed by PCNSA, SLDA, and PCA. The results with and without kernels are very similar.

### *B. Image Classification: Face Recognition*

Face recognition has been discussed as an example of an "apples from apples" type application where LDA and PCNSA have comparable performance. The algorithms were tested on two standard face databases: the UMIST face database [25], which consists of 22 images of each person, taken in different poses and the AT&T Cambridge database (formerly the ORL data-

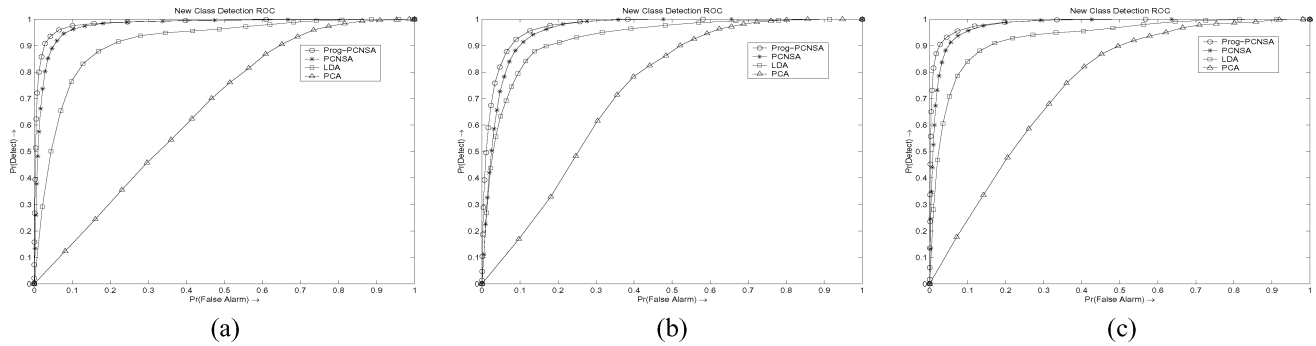



Fig. 7. ROC curves (object recognition): Plots of probability of new class detection versus probability of false alarm for varying new class detection thresholds. (a) No kernel. (b) Cosine polynomial: $\mathrm{d} = 10$. (c) Gaussian: $\sigma^2 = 4000$.

TABLE II
FACE RECOGNITION (UMIST DATABASE, 15 CLASSES): MISCLASSIFICATION ERROR PROBABILITY. WE HAVE HIGHLIGHTED THE BEST KERNEL CHOICE FOR EACH ALGORITHM BY UNDERLINING AND SHOWING THE ERROR IN BOLD

| Kernel Type | K-Prog-PCNSA | K-PCNSA | K-SLDA | K-PCA |
|---|---|---|---|---|
| $k(x,y) = x^T y$ (No Kernel) | **0.000** | **0.001** | **0.003** | 0.367 |
| Polynomial, d = 2 | **0.000** | 0.006 | 0.010 | 0.383 |
| Cosine Polynomial, d = 10 | 0.005 | 0.014 | 0.021 | 0.248 |
| Gaussian, $\sigma^2 = 4000$ | 0.001 | 0.009 | 0.005 | **0.230** |

TABLE III
FACE RECOGNITION (AT&T CAMBRIDGE (FORMERLY ORL) DATABASE, TEN CLASSES): MISCLASSIFICATION ERROR PROBABILITY. WE SHOW ONLY THE BEST KERNEL CHOICE FOR EACH ALGORITHM

| | K-Prog-PCNSA | K-PCNSA | K-SLDA | K-PCA |
|---|---|---|---|---|
| Least Error Kernel | $k(x,y) = x^T y$ (No Kernel) | $k(x,y) = x^T y$ (No Kernel) | Gaussian, $\sigma^2 = 4000$ | Polynomial, d = 2 |
| Error Probab. | 0.058 | 0.168 | 0.020 | 0.433 |

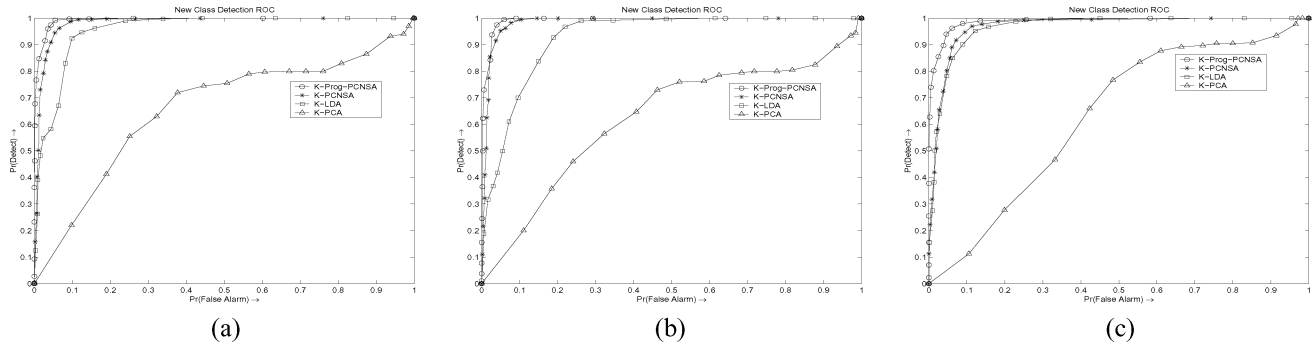



Fig. 8. ROC curves (face recognition): plots of probability of new class detection versus probability of false alarm for varying new class detection thresholds: (a) no kernel; (b) polynomial $\mathrm{d} = 2$; (c) Gaussian, $\sigma^2 = 4000$.

base) [26].[8] The face images can be downloaded from http://images.ee.umist.ac.uk/danny/database.html and http://www.uk.research.att.com/facedatabase.html respectively. We show a few samples of one face class from each database in Fig. 5. Also note that the UMIST database had large pose variation because of which the within class covariances of different classes were different (and, hence, PCNSA performs better for this database) while SLDA was better for the AT&T database. The "Leave two out" strategy was adopted for testing and 40 such iterations were run, each time choosing the two test samples from each class randomly. Five new classes and 12 trained classes were taken for UMIST, while five new and ten trained classes were taken for AT&T.

The misclassification error in the UMIST database is shown in Table II. The best kernel choice for all algorithms except PCA is the no kernel case, $k(x,y) = x^T y$. Prog-PCNSA and PCNSA have 0% error while SLDA has only a marginally higher error of 0.3%. For the AT&T database, we show in Table III results using only the best kernel type for each algorithm. Here SLDA outperforms prog-PCNSA and PCNSA. The fact that the within-class variation is very similar for all faces makes performance of LDA superior. We have also plotted the ROC curves for the UMIST database in Fig. 8. Here, prog-PCNSA and PCNSA have superior performance in all cases except in the Gaussian kernel case, where performance of SLDA is also comparable.

[8]The reason only these two databases were used is that they had enough training data per class to obtain reliable ANS representations for each class. When training data is small, performance of PCNSA deteriorates very fast.

TABLE IV
RETRIEVING ACTIONS USING PCNSA IN TANGENT TO SHAPE SPACE: THE DISTANCES OF THE QUERY SEQUENCES (TOP ROW) IN ANS SPACE OF TANGENT TO MEAN SHAPE ((27)) OF EACH OF THE 3 DATABASE SEQUENCES (IN LEFTMOST COLUMN) ARE SHOWN. THE BOLD AND UNDERLINED DISTANCE IN EACH COLUMN CORRESPONDS TO THE CLOSEST MATCH TO QUERY

| Action | bprowl-walk | broom1 | broom3 | crawl | jog1 | sit1 | sit2 | walk1 | walk2 | walk3 | walk-sad1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| walk | **1.43e-4** | 5.09e-4 | 4.58e-4 | **4.11e-4** | **2.36e-4** | 1.01e-3 | 2.08e-4 | 0.03e-6 | 1.60e-4 | **4.20e-5** | **2.20e-5** |
| broom | 5.65e-4 | **2.00e-6** | **2.90e-5** | 7.52e-4 | 4.36e-4 | 4.16e-3 | 1.95e-3 | 1.36e-4 | 2.05e-4 | 2.44e-4 | 3.31e-4 |
| sit | 9.66e-4 | 5.18e-4 | 3.53e-4 | 1.17e-3 | 6.33e-4 | **3.00e-6** | **4.70e-5** | 2.23e-4 | 3.69e-4 | 5.40e-4 | 3.67e-4 |

### C. Video Classification: Action Retrieval

We show here an application of PCNSA to action retrieval in a landmark shape dynamical framework proposed by us in [27]–[29]. We represented a stationary shape activity by a mean shape plus a linear dynamical model in the tangent space [30] at the mean shape. The dynamics in tangent space is modeled by a linear autoregressive (AR) model, $v_t = Av_{t-1} + n_t$, $n_t \sim \mathcal{N}(0, \Sigma_n)$. The sequence of operations can be summarized as follows:

$$\{Y_t\}_{t=1}^T \longrightarrow \{w_t\}_{t=1}^T \longrightarrow \{S_\mu, \{z_t\}_{t=1}^T\}$$
$$\{z_t\}_{t=1}^T \longrightarrow \{v_t = [I - S_\mu S_\mu^*]\, z_t\}_{t=1}^T \longrightarrow A, \Sigma_v, \Sigma_n \quad (26)$$

where $Y_t$ is the configuration vector (a complex vector containing the x and y coordinates of the landmarks as the real and imaginary parts) at time $t$, $w_t$ is the preshape obtained after translation and scale normalization of $Y_t$ and $S_\mu$ is the Procrustes mean shape [30] obtained after generalized Procrustes analysis [30] on the preshapes. $z_t$ is the shape obtained after aligning the preshapes, $w_t$, to $S_\mu$ [30] and $v_t$ is the tangent coordinate of $z_t$ in the tangent space at $S_\mu$. Also, $A$ is the autoregression matrix, $\Sigma_v$ and $\Sigma_n$ are the covariance matrices of $v_t$ and $n_t$ [27].

For representing actions, we used motion capture data (which provides locations of 53 human joints in a set of frames) to learn the shape dynamical models for three different actions—"walking," "brooming," and "sitting." Each joint location constituted a landmark. For each action, we learnt the mean shape and the $2*53-4 = 102$-dimensional tangent space at the mean shape. The PCA subspace of the tangent space of an action class was obtained by projecting the training data from all classes into the tangent space and evaluating the principal eigenvectors of the covariance matrix. The AR model, $v_t = Av_{t-1} + n_t$, $n_t \sim \mathcal{N}(0, \Sigma_n)$, was defined in this reduced dimensional PCA space and an ANS of the noise covariance matrix was learnt and used for classification. The entire algorithm is as follows.

1) For each class $i$, learn the mean shape and tangent space as summarized in (26).
2) For each class $i$:
   — project data from all classes into the tangent space of class $i$ and learn a $L = 20$-dimensional PCA space $W_i^{\mathrm{PCA}}$;
   — project the training data of class $i$ into this PCA space, to learn the Gauss-Markov model parameters, $A_i$, $\Sigma_{n,i}$, $\Sigma_{v,i}$ [27] in PCA space; project the autoregression matrix $A_i$ back into full tangent space to get $A_{\mathrm{full,i}} = W_i^{\mathrm{PCA}} A_i W_i^{\mathrm{PCA}T}$;
   — learn $W_i^{\mathrm{NSA}}$ by obtaining the trailing eigenvectors of $\Sigma_{n,i}$; combine both PCA and NSA projection matrices to obtain $W_i^{\mathrm{project}} = W_i^{\mathrm{PCA}} W_i^{\mathrm{NSA}}$.
3) Classification: Given a test sequence:
   — for each class $i$, project the sequence into its tangent space to obtain $\{v_{t,i}\}_{t=1}^T$;
   — choose the most likely class $c$ as $c = \arg\min_i d_i$ where

$$d_i = \sum_{\tau=1}^{T} \left\| W_i^{\mathrm{project}T} (v_{\tau,i} - A_{\mathrm{full,i}} v_{\tau-1,i}) \right\|^2. \quad (27)$$

Since stationarity is assumed, we were able to use a single training sequence of each of the actions to learn the mean shape, PCA space, AR model parameters, and ANS for each class. We then used different instances of "walking," "brooming" and "sitting" actions as queries and attempted to retrieve the closest action to the given action. We show the distances in Table IV. The query actions were prowl-walk, two brooming sequences, crawl, jog, two sitting sequences, three walking sequences, and a sad-walk sequence. We have underlined the distance of a query from its closest action. As can be seen from the table, for all the five walk sequences, the "walk" action sequence is correctly retrieved. Also for the two broom sequences and the two sit sequences, the correct action is retrieved. For crawl, which is a new class, the minimum distance (dmin) and second largest distance (dmin2) are quite close, so using the new class detection method given in (24) with $t = 0.5$, it gets classified as a new class.

### D. Video Classification: Abnormal Activity Detection

In [27], we have used a PCNSA based metric for abnormal activity detection in a shape dynamical model framework (discussed above in Section VII-C). A normal activity consisting of a group of people deplaning and moving toward the airport terminal was represented by a landmark shape dynamical model (with each person forming a landmark). The ANS matrix $W^{\mathrm{NSA}}$ was learnt for the noise covariance $\Sigma_n$ for a normal activity and the distance to activity metric was $d(t)^2 = \sum_{\tau=t-20}^{t} \|W^{\mathrm{NSA}T}(v_\tau - Av_{\tau-1})\|^2$. We observed in [27] that this detected abnormality faster than both full Euclidean distance and full Mahalonobis distance (log likelihood under the AR model). The data dimension was originally quite small (eight dimensional), and, hence, dimensionality reduction using PCA was not required for this application. Also, the "abnormal" class was not characterized, so one could not use PCA to increase between class variance. For the same reason LDA could not be used for this application.

A normal and an abnormal activity frame are shown in Fig. 9(a) and (b). Plots of the activity metric as a function of time for normal activity and two kinds of abnormalities are shown in Fig. 9(c). Note that this algorithm runs in realtime


(a) Normal Activity (b) An Abnormality (c) Detecting Abnormal Activities

Fig. 9. (a) A "normal activity" frame with four people, (b) shape distorted by one person deviating from path, (c) detecting abnormal activities. The blue solid and dotted plots are the activity metric for a normal activity as a function of time. The green—$*$ plot for the abnormality shown in (b) and the red—o plot is for abnormality introduced by one person stopping in the path. Both abnormalities were introduced at $t = 5$.

even with MATLAB code and, hence, can be used for real-time video surveillance applications.

### E. Other Applications

Another application to which we applied PCNSA was for feature matching for image registration [22]. Image registration is an important problem in many applications, one of which is three-dimensional model alignment. The first and most difficult step in image registration is obtaining feature correspondences between two or more frames. In [31], correlation matching has been used for obtaining correspondences. We replaced this correlation match by distance in PCA, SLDA or PCNSA space. Also new feature detection is very important here, since as the face moves, new (previously occluded) features can appear. Results showing superior performance of PCNSA for this application are shown in [22, Chapter 5].

Other possible applications of PCNSA are in content-based image retrieval and digit recognition. For large database image retrieval, a combination of PCNSA and SLDA can be used as a robust alternative to just LDA: Choose subsets of classes which have similar covariance matrices, use PCNSA to choose the class subset and SLDA to classify within the subset (an idea similar to MKL [13]).

## VIII. Conclusions and Future Directions

A new algorithm called PCNSA was presented for classification when different classes have unequal and nonwhite class covariance matrices. Error probability expressions were derived for PCNSA and its performance compared with subspace LDA which is another algorithm for classification in colored noise. Superior performance of PCNSA was shown for applications with vastly different within-class covariance matrices like object recognition, action retrieval or abnormal activity detection. Conditions when PCNSA would fail—no ANS space or small interclass distance in ANS space or small training data set (inaccurate ANS space estimation) were also discussed. Experimental comparison with SLDA, PCA, SVMs, and kernel methods is shown.

As part of future work, we plan to combine PCNSA and SLDA using the algorithm described in Section VI-D. For large number of classes, for example retrieval applications, one can use ideas similar to [13] as discussed in Section V. Also, an algorithm similar to Discriminant EM for LDA [32] can be used to increase the size of the training data set. Queries which have been reliably classified [i.e., have a low value of the ratio given in (25)] can be used as training data and improved ANS estimates can be obtained on the fly. There are many algorithms for online eigenvector estimation as new data comes in, without having to recalculate the covariance matrix. Since PCNSA uses a very small dimensional PCA space, the cost of re-estimating ANS would be small.

## Appendix
## Proofs for Section III

*Proof of Theorem 1*

The error event $E_1$ is defined in (6). Now since ANS is assumed 1-D, $W_1^{\mathrm{NSA}} = N_1$ and $d_1(\mathbf{X}) = |N_1^T(\mathbf{X} - \mu_1)|$ is a scalar. Using (2)

$$N_1^T(\mathbf{X} - \mu_1)|\{\mathbf{X} \in C_1\} \sim \mathcal{N}(0, \lambda_{\mathrm{ANS},1}). \tag{28}$$

$$\text{Now, } \left(d_1^2(\mathbf{X}) > \Delta_1^2 | \mathbf{X} \in C_1\right) = 2\left(1 - \Phi(k)\right) \triangleq g(k) \tag{29}$$

where $\Phi(.)$ is the cdf of an $\mathcal{N}(0,1)$ random variable and $\Delta_1$ is defined in (9). One can choose $k$ large enough so that $g(k)$ is small. For example, for $k = 10$, $g(k) = 10^{-23}$. Now, the error event $E_1$ (defined in (6)) can be split as,[9]

$$\begin{aligned} E_1 &= \left\{d_2^2(\mathbf{X}) \le d_1^2(\mathbf{X}), d_1^2(\mathbf{X}) \le \Delta_1^2\right\} \\ &\quad \cup \left\{d_2^2(\mathbf{X}) \le d_1^2(\mathbf{X}), d_1^2(\mathbf{X}) > \Delta_1^2\right\} \\ &\subseteq \left\{d_2^2(\mathbf{X}) \le \Delta_1^2\right\} \cup \left\{d_1^2(\mathbf{X}) > \Delta_1^2\right\}. \end{aligned} \tag{30}$$

Thus, $P(E_1) \le P(d_2^2(\mathbf{X}) \le \Delta_1^2) + g(k)$. Now $d_2(\mathbf{X}) = |N_2^T(\mathbf{X} - \mu_2)|$. Using (2), we get

$$\mathbf{Z} \triangleq \frac{N_2^T(\mathbf{X} - \mu_1)}{\sqrt{N_2^T \Sigma_1 N_2}} \sim \mathcal{N}(0,1) \tag{31}$$

[9]Assume $\mathbf{X} \in C_1$ everywhere.

and $d_2^2 = (\sigma_2^1\mathbf{Z} - N_2^T(\mu_2 - \mu_1))^2$. Using the definitions in (8), we get $P(d_2^2(\mathbf{X}) < \Delta_1^2) = P((\alpha_2^1 - \Delta_1)/\sigma_2^1) < \mathbf{Z} < (\alpha_2^1 + \Delta_1)/\sigma_2^1)$. Thus

$$P(E_1) \leq P\left(\frac{\alpha_2^1 - \Delta_1}{\sigma_2^1} < \mathbf{Z} < \frac{\alpha_2^1 + \Delta_1}{\sigma_2^1}\right) + g(k)$$
$$= \Phi\left(\frac{\alpha_2^1 + \Delta_1}{\sigma_2^1}\right) - \Phi\left(\frac{\alpha_2^1 - \Delta_1}{\sigma_2^1}\right) + g(k). \quad (32)$$

Substituting for $g(k)$ and $\Delta_1$ using (29) and (9) and taking a minimum over all value of $k > 0$ [(32) is valid for all $k > 0$, and, hence, a tighter bound is obtained by taking a minimum over $k$], we get the result.

*Proof of Theorem 2*

Error event $E_1$ is as defined in (6) with $\Delta_1$ now given by (10). It can be bounded using exactly the same logic as in (30). Thus, we have

$$P(E_1) \leq P\left(d_2^2(\mathbf{X}) < \Delta_1^2 | \mathbf{X} \in C_1\right) + P\left(d_1^2(\mathbf{X}) > \Delta_1^2 | \mathbf{X} \in C_1\right). \quad (33)$$

First consider $P(d_1^2(\mathbf{X}) > \Delta_1^2)$. Define

$$\mathbf{Z_{N_1}} \triangleq N_1^T(\mathbf{X} - \mu_1) \sim \mathcal{N}(0, \Lambda_{\text{ANS},1}) \quad (34)$$

where $\Lambda_{\text{ANS},1}$ is diagonal. Then $d_1^2(\mathbf{X}) = \|\mathbf{Z_{N_1}}\|^2$. Thus

$$\left\{d_1^2(\mathbf{X}) > \Delta_1^2\right\} \subseteq \{\cap_j A_j\}^c,$$
$$A_j = \left\{Z_{N_1,j}^2 < k^2 \lambda_{\text{ANS},1,j}^2\right\}. \quad (35)$$

This follows because $Z_{N_1,j}^2 < k^2\lambda_{\text{ANS},1,j}^2$, $\forall j$, implies that $d_1^2(\mathbf{X}) < \Delta_1^2$. Thus, $\{\cap_j A_j\} \subseteq \{d_1^2(\mathbf{X}) < \Delta_1^2\}$. Taking complements on both sides we get (35).

By (34), the components of the vector $\mathbf{Z_{N_1}}$ are independent, and, hence, the events $A_j$ are independent. Also, $P(A_j) = 1 - g(k)$ where $g(k)$ is defined in (29). Thus, using (35)

$$P\left(d_1^2(\mathbf{X}) > \Delta_1^2\right) \leq P\left(\{\cap_j A_j\}^c\right)$$
$$= 1 - \prod_{j=1}^{M_1} P(A_j) = 1 - (1 - g(k))^{M_1}$$
$$\triangleq g_{M_1}(k). \quad (36)$$

Now, consider $P(d_2^2(\mathbf{X}) \leq \Delta_1^2)$. Define

$$\mathbf{Z_{N_2}} \triangleq N_2^T(\mathbf{X} - \mu_1) \sim \mathcal{N}\left(0, \Sigma_2^1\right) \quad (37)$$

then $d_2^2(\mathbf{X}) = \|\mathbf{Z_{N_2}} - \beta_2^1\|^2$ ($\beta_2^1$, $\Sigma_2^1$ defined in Theorem 2). Using $U$ to diagonalize $\mathbf{Z_{N_2}}$, we get ($U$, $S_2^1$ defined in Theorem 2)

$$\mathbf{Z_{N_2}^{indep}} = U^T\mathbf{Z_{N_2}} \sim \mathcal{N}\left(0, S_2^1\right). \quad (38)$$

Since $U$ is orthonormal, $\|\mathbf{Z_{N_2}^{indep}} - \alpha_2^1\| = \|U^T(\mathbf{Z_{N_2}} - \beta_2^1)\| = \|\mathbf{Z_{N_2}} - \beta_2^1\|$ ($\alpha_2^1$ defined in Theorem 2) and so

$$P\left(d_2^2(\mathbf{X}) \leq \Delta_1^2\right) = P\left(\left\|\mathbf{Z_{N_2}^{indep}} - \alpha_2^1\right\|^2 < \Delta_1^2\right). \quad (39)$$

Using the fact that $\|\mathbf{Z_{N_2}^{indep}} - \alpha_2^1\|^2 < \Delta_1^2$ implies that $(\mathbf{Z_{N_{2,j}}^{indep}} - \alpha_{2,j}^1)^2 < \Delta_1^2$, $\forall j$, we get

$$\left\{\left\|\mathbf{Z_{N_2}^{indep}} - \alpha_2^1\right\|^2 < \Delta_1^2\right\} \subseteq \cap_j B_j$$
$$\text{where} \quad B_j = \left\{\left(\mathbf{Z_{N_{2,j}}^{indep}} - \alpha_{2,j}^1\right)^2 < \Delta_1^2\right\}.$$

The events $\{B_j\}$ are independent since elements of the vector $\mathbf{Z_{N_2}^{indep}}$ are independent. Using (7), $P(B_j) = P(\alpha_{2,j}^1 - \Delta_1 < \mathbf{Z_{N_{2,j}}^{indep}} < \alpha_{2,j}^1 + \Delta_1) = [\Phi((\alpha_{2,j}^1 + \Delta_1)/\sigma_{2,j}^1)) - \Phi((\alpha_{2,j}^1 - \Delta_1)/\sigma_{2,j}^1))]$ where $\alpha_{2,j}^1$ is the $j^{th}$ component of $\alpha_2^1$ and ${\sigma_{2,j}^1}^2$ is the $(j,j)^{th}$ element of $S_2^1$.

Thus,
$$P\left(d_2^2(\mathbf{X}) \leq \Delta_1^2\right) \leq P(\cap_j B_j)$$
$$= \prod_{j=1}^{M_2}\left[\Phi\left(\frac{\alpha_{2,j}^1 + \Delta_1}{\sigma_{2,j}^1}\right) - \Phi\left(\frac{\alpha_{2,j}^1 - \Delta_1}{\sigma_{2,j}^1}\right)\right]. \quad (40)$$

Finally, combining (33), (36), and (40) and substituting for $g_{M_1}(k)$ from (36), and taking a minimum over all $k > 0$, we get the result.

## REFERENCES


[1] N. Vaswani, "A linear classifier for gaussian class conditional distributions with unequal covariance matrices," presented at the Int. Conf. Pattern Recognition, 2002.

[2] N. Vaswani and R. Chellappa, "Classification probability analysis of principal component null space analysis," presented at the Int. Conf. Pattern Recognition, 2004.

[3] R. Chellappa, C. Wilson, and S. Sirohey, "Human and machine recognition of faces: A survey," *Proc. IEEE*, no. 5, pp. 705–740, May 1995.

[4] A. Samal and P. Iyengar, "Automatic recognition and analysis of human faces and facial expressions: A survey," *Pattern Recognit.*, pp. 65–77, 1992.

[5] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cogn. Neurosci.*, vol. 3, no. 1, pp. 72–86, 1991.

[6] W. Zhao, R. Chellappa, and P. J. Phillips, "Subspace linear discriminant analysis for face recognition," Center for Autom. Res., Univ. Maryland, College Park, Tech. Rep. CAR-TR-914, vol. 8, 1999.

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

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

[9] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neur. Comput.*, vol. 7, no. 6, pp. 1129–1159, 1995.

[10] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[11] M. Pontil and A. Verri, "Support vector machines for 3D object recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 6, pp. 637–646, Jun. 1998.

[12] X. S. Zhou and T. Huang, "Small sample learning during multimedia retrieval using biasmap," presented at the IEEE Conf. Computer Vision and Pattern Recognition, Dec. 2001.

[13] R. Cappelli, D. Maio, and D. Maltoni, "Multispace kl for pattern representation and classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 9, pp. 977–996, Sep. 2001.

[14] H. Murase and S. K. Nayar, "Visual learning and recognition of 3-D objects from appearance," *Int. J. Comput. Vis.*, pp. 5–24, 1995.

[15] J. Mao and A. Jain, "Artificial neural networks for feature extraction and multivariate data projection," *IEEE Trans. Neural Netw.*, vol. 6, no. 3, pp. 296–316, Mar. 1995.

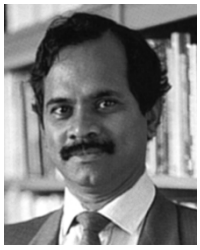
[16] S. Li and Q. Fu *et al.*, "Kernel machine based learning for multiview face detection and pose estimation," *Int. J. Comput. Vis.*, pp. 674–679, July 2001.

[17] V. Roth and V. Stainhage, "Nonlinear discriminant analysis using kernel functions," in *Neural Information Processing Systems 12*. Cambridge, MA: MIT Press, 2000, pp. 568–574.

[18] S. Mika, G. Ratsch, J. Weston, B. Scholkpof, A. Simola, and K. Miller, "Fisher discriminant analysis with kernels," presented at the IEEE Workshop on Neural Networks for Signal Processing, 1999.
[19] Q. Liu, H. Lu, and S. Ma, "Improving kernel discriminant analysis for face recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 1, pp. 42–49, Jan. 2004.
[20] G. Casella and R. Berger, *Statistical Inference*, 2nd ed. Belmont, CA: Duxbury, 2002.
[21] W. Zhao, R. Chellappa, and N. Nandhakumar, "Empirical performance analysis of linear discriminant classifiers," presented at the IEEE Conf. Computer Vision and Pattern Recognition, 1998.
[22] N. Vaswani, "Change Detection in Stochastic Shape Dynamical Models With Applications in Activity Modeling and Abnormality Detection," Ph.D. dissertation, Elect. Comput. Eng. Dept., Univ. Maryland, College Park, Aug. 2004.
[23] H. Vincent Poor, *An Introduction to Signal Detection and Estimation*, 2nd ed. New York: Springer.
[24] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia Object Image Library (COIL-100)," Tech. Rep. CUCS-006-96, Feb. 1996.
[25] D. Graham and N. Allinson, "Characterizing virtual eigensignatures for general purpose face recognition," in *Face Recognition: From Theory to Applications*. New York: Springer, 1998, pp. 446–456.
[26] F. Samaria and A. Harter, "Parameterization of a stochastic model for human face identification," presented at the 2nd IEEE Workshop on Applications of Computer Vision, Sarasota, FL, Dec. 1994.
[27] N. Vaswani, A. RoyChowdhury, and R. Chellappa, "Statistical shape theory for activity modeling," presented at the IEEE Int. Conf. Acoustics, Speech, Signal Processing, 2003.
[28] ——, "Activity recognition using the dynamics of the configuration of interacting objects," presented at the IEEE Conf. Computer Vision and Pattern Recognition, 2003.
[29] ——, ""Shape activity": A continuous state hmm for moving/deforming shapes with application to abnormal activity detection," *IEEE Trans. Image Processing*, vol. 14, no. 9, pp. 1603–1616, Oct. 2005.
[30] I. L. Dryden and K. V. Mardia, *Statistical Shape Analysis*. New York: Wiley, 1998.
[31] A. RoyChowdhury, R. Chellappa, and T. Keaton, "Wide baseline image registration with application to 3d face modeling," *IEEE Trans. Multimedia*, vol. 6, no. 3, pp. 423–434, Jun. 2004.
[32] Y. Wu, Q. Tian, and T. Huang, "Discriminant-em algorithm with application to image retrieval," presented at the *IEEE Conf. Computer Vision and Pattern Recognition*, Jun. 2000.

**Namrata Vaswani** (M'99) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology (I.I.T.), Delhi, in 1999, and the Ph.D. degree in electrical and computer engineering from the University of Maryland, College Park, in August 2004. Her Ph.D. thesis was on change detection in stochastic shape dynamical models and applications to activity modeling and abnormal activity detection.

She was a Postdoctoral Fellow with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, from 2004 to 2005, where she worked on particle filtering agorithms for level-set representations of continious curves and their applications to tracking deformable objects. She is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Iowa State University, Ames. Her research interests are in detection and estimation problems in signal and video processing, computer vision, and in biomedical image processing. In particular, she is interested in particle filtering theory and applications in tracking and change detection and in shape analysis and filtering. In the past, she has also worked on subspace methods for image classification.



**Rama Chellappa** (S'78–M'79–SM'83–F'92) received the B.E. (Hons.) degree from the University of Madras, Madras, India, in 1975 and the M.E. (Distinction) degree from the Indian Institute of Science, Bangalore, in 1977. He received the M.S.E.E. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, in 1978 and 1981, respectively.

Since 1991, he has been a Professor of electrical engineering and an affiliate Professor of Computer Science at the University of Maryland, College Park. Recently, he was named the Minta Martin Professor of Engineering. He is also affiliated with the Center for Automation Research (Director) and the Institute for Advanced Computer Studies (permanent member). Prior to joining the University of Maryland, he was an Assistant Professor (1981 to 1986) and an Associate Professor (1986 to 1991) and Director of the Signal and Image Processing Institute (1988 to 1990) with the University of Southern California (USC), Los Angeles. Over the last 24 years, he has published numerous book chapters and peer-reviewed journal and conference papers. He has edited a collection of Papers on Digital Image Processing (Los Alamitos, CA: IEEE Computer Society Press, 1992), coauthored a research monograph on *Artificial Neural Networks for Computer Vision* (with Y. T. Zhou) (New York: Springer-Verlag, 1990), and co-edited a book on *Markov Random Fields: Theory and Applications* (with A. K. Jain) (New York: Academic, 1993). His current research interests are face and gait analysis, 3-D modeling from video, automatic target recognition from stationary and moving platforms, surveillance and monitoring, hyperspectral processing, image understanding, and commercial applications of image processing and understanding.

Dr. Chellappa has served as an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON IMAGE PROCESSING, and IEEE TRANSACTIONS ON NEURAL NETWORKS. He was Co-Editor-in-Chief of *Graphical models and Image Processing*. He is now serving as the Editor-in-Chief of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. He served as a member of the IEEE Signal Processing Society Board of Governors from 1996 to 1999. Currently, he is serving as the Vice President of Awards and Membership for the IEEE Signal Processing Society. He has served as a General the Technical Program Chair for several IEEE international and national conferences and workshops. He received several awards, including the National Science Foundation (NSF) Presidential Young Investigator Award, an IBM Faculty Development Award, the 1990 Excellence in Teaching Award from School of Engineering at USC, the 1992 Best Industry Related Paper Award from the International Association of Pattern Recognition (with Q. Zheng), and the 2000 Technical Achievement Award from the IEEE Signal Processing Society. He was elected as a Distinguished Faculty Research Fellow (1996 to 1998) at the University of Maryland, he is a Fellow of the International Association for Pattern Recognition, and he received a Distinguished Scholar-Teacher award from the University of Maryland in 2003.