Free Form Face Recognition using Kernel Sparse Representation

Anoop K.R.*
Dept. of Electrical Engineering
Indian Institute of Science
Bangalore, India
anoopkr@ee.iisc.ernet.in

Narasimhan R.S.
Dept. of Electrical Engineering
Indian Institute of Science
Bangalore, India
narasimhanrs@gmail.com

K.R.Ramakrishnan
Dept. of Electrical Engineering
Indian Institute of Science
Bangalore, India
krr@ee.iisc.ernet.in

Chiranjib Bhattacharya
Computer Science and
Automation
Indian Institute of Science
Bangalore, India
chiru@csa.iisc.ernet.in

ABSTRACT

Recognizing faces from face detector outputs is a hard problem. While existing face recognition (FR) techniques essentially work on pre-processed (cropped and aligned) data, we employ Gabor-based covariance descriptors for recognition from *free-form* faces (raw face detector outputs). Our recognition algorithm employs a Principal Geodesic Analysis (PGA) of Covariance Descriptors, followed by a transformation on to tangent space where faces are sparsely represented. Employing the *kernel trick* on this sparse feature space enables upto 10% improvement in recognition accuracy.

Keywords

Covariance descriptor, Principal Geodesic Analysis, Kernel trick, Sparse representation

1. INTRODUCTION

Face recognition (FR) is an actively researched area over the past few decades. Extensive literature exists where a high-dimensional test image is projected onto lower dimensions like Eigenfaces [12], Fisherfaces [6], Laplacianfaces [4] and many other variants. Use of Gabor features is also explored in [5]. All these algorithms represent faces in a vectored form; also, test faces need to be properly cropped, aligned and of the same scale as the training faces. However, typical face detector outputs are neither aligned nor cropped and also vary in scale. Classification of such data is very challenging.

Recently, Olshausen *et al.* [9] have shown that human perception of vision is sparsely modeled. Wright *et al.* [15] have also come up with a sparse representation of faces. Due to compact representation of the test data as a linear combination of the training set, the sparsest solution is shown to be indeed discriminative for FR. Sparse representation is achieved efficiently using ℓ_1 minimization or Basis pursuit introduced in the context of compressive sensing [1].

Instead of representing the face as a vector, Tuzel *et al.* [13] propose Region Covariance matrices (RCM). Fletcher *et al.* [2] propose a dimensionality reduction for these RCM descriptors called Principal Geodesic Analysis (PGA). These PGA-RCM descriptors are robust to alignment and scale variations. However, these descriptors belong to a symmetric space which is not a vector space. As a result, sparse representation for PGA-RCM is not feasible. This can be overcome by transforming PGA-RCM descriptors onto a tangent space using logarithmic mapping [2].

In this paper, we propose to use PGA-RCM descriptors for representing *free-form* faces and also, employ the sparse representation of these descriptors for FR. Also, to overcome the limitations of linear modeling, we propose kernelization [16] of the sparse feature space to perform classification in a sparse, non-linear space. On the raw AR [8] and YaleB databases [3], which contain faces obtained from face detector outputs without any preprocessing, we demonstrate higher recognition accuracy using the proposed features.

The paper is organized as follows. We introduce PGA of covariance descriptors in Section 2. In Section 3, we describe sparse modeling using PGA features, and detail the steps for kernelization of this sparse model in section 4. Experimental results are discussed in Section 5, while Section 6 outlines the conclusions.

2. PGA OF COVARIANCE DESCRIPTORS (PGA-RCM)

Covariance descriptors are a natural way of fusing multiple correlated features. They are also of low dimension

*Corresponding author

ICVGIP '10, December 12-15, 2010, Chennai, India

compared to other region descriptors. These descriptors belong to a positive definite symmetric space, which is not a vector space. Therefore, these features are transformed to a vector space from a fixed base point. The transformation spaces used for Covariance matrix is known as Lie groups, which form a smooth Riemannian manifold, having closed form solution to distance computation and hence suitable for establishing statistics.

For a curve on the manifold, an instantaneous speed vector and the norm can be computed at each point on the curve, and its integration along the curve gives the length of the curve. The distance between two points of a connected Riemannian manifold is the minimum length along the curve joining the two points. The curves realizing this minimum are called as *Geodesics*. This is an intrinsic way of measuring the length. The extrinsic way is to embed the manifold in a vector space, and the length of the curve will be the distance between the two points in the vector space. The Riemannian metric is the inner product on the tangent space, which is a vector space, at each point on the manifold. Thus, Riemannian metric on a manifold, $\mathcal{M}$ smoothly assigns to each point $x \in \mathcal{M}$ a continuous collection of inner product $\langle , \rangle_x$ on $T_x\mathcal{M}$, tangent space to $\mathcal{M}$ at x . An important property of the positive definite symmetric space is that they are geodesically *complete*, i.e. the manifold has no boundary nor will reach any singular point in finite time. As a result, Hopf-Rinow-De Rham theorem states that there always exist atleast one geodesic between any two points on the manifold.

2.1 Exponential and Logarithmic maps

From the theory of differential equations, there exists a unique geodesic going through the point $x \in \mathcal{M}$, with the tangent vector $\vec{v} \in T_x\mathcal{M}$. The geodesics through this reference point x are transformed into straight lines on the tangent space, preserving the distance along the curve. The function that maps this vector $\vec{v}$ to the point on the manifold that a geodesic reaches in unit time starting at x, is called the *exponential map*. Mathematically,

$$\begin{aligned} Exp_x &: T_x\mathcal{M} \to \mathcal{M} \\ &: \vec{v} \to Exp_x(\vec{v}) = \gamma(1) \end{aligned} \tag{1}$$

where $\gamma(t)$ is the geodesic.

The origin of $T_x\mathcal{M}$ is mapped to the point itself, i.e. $Exp_x(0) = x$. For each point $x \in \mathcal{M}$, there exists a diffeomorphism from neighbourhood of the origin in $T_x\mathcal{M}$ to the neighbourhood of $x \in \mathcal{M}$. Thus there exists an inverse of the exponential map known as the logarithmic map, $Log_x = Exp_x^{-1}$. The algorithms for calculating these are as given in [2]. The above operations can be visualized as shown in Fig. 1

2.2 Principal Geodesic Analysis (PGA)

Principal Geodesic Analysis [2] on manifolds is a generalization and extension of the Principal Component Analysis on Euclidean space. This requires the computation of the following statistics:

- **Intrinsic Mean**: The mean μ of set of points $\{x_i\}_{i=1}^n \in \mathcal{M}$, is defined as the point that minimizes the sum of the squared distance function.

$$\mu = \underset{\bar{\mu} \in \mathcal{M}}{\arg\min} \sum_{i=1}^{n} d(\bar{\mu}, x_i) \tag{2}$$

where $d(x, y) = \|Log_x(y)\|$ denotes the Riemannian metric. For a Riemannian manifold the existence and uniqueness of the mean is guaranteed. In [11] a gradient descent algorithm to calculate such a mean is described, which is given by

$$\mu_{k+1} = Exp_{\mu_k}\left[\frac{1}{N}\sum_{i=1}^{N} Log_{\mu_k}(x_i)\right] \tag{3}$$

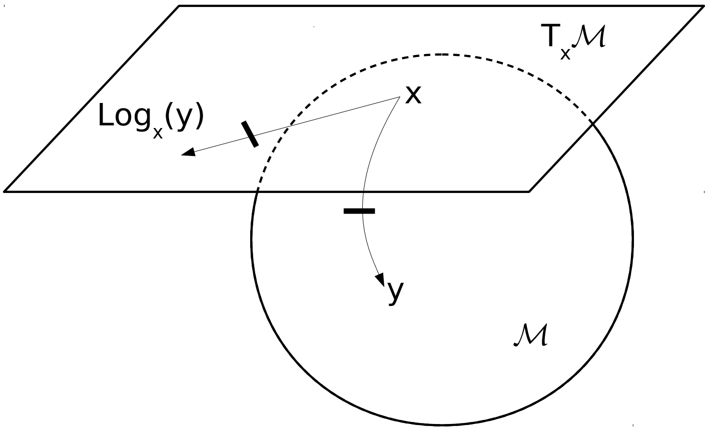

Figure 1: Figure Depicting the mapping of a point $y = \gamma(1) \in \mathcal{M}$ to a vector $\vec{v} = Log_x(y) \in T_x\mathcal{M}$. Length of the vector is geodesic distance between x and y

- **Variance**: If x is the random variable and μ is its mean, the sample variance is given by

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} d(\mu, x_i)^2 = \frac{1}{n}\sum_{i=1}^{n} \|Log_\mu(x_i)\|^2 \tag{4}$$

- **Geodesic submanifold**: A geodesic curve in a manifold is the generalization of a straight line in a linear space. A submanifold H is said to be geodesic at $x \in H$, if all geodesics of H, passing through x are also geodesics of $\mathcal{M}$. Submanifold geodesics at x preserve the distances to x.

- **Projection**: The projection of a point x onto a geodesic submanifold H is the point on the submanifold that is nearest to x in Riemannian metric, given by

$$\pi_H(x) = \underset{y \in H}{\arg\min}\ d(x, y)^2 \tag{5}$$

For the symmetric space of covariance matrices this projection exists and is unique.

Given a set of data points $x_1, x_2, x_3, \ldots, x_n \in \mathcal{M}$, the goal is to find a geodesic submanifold such that the projected variance of the data is maximized. These submanifolds are referred to as the Principal Geodesic Submanifolds [2]. These Principal Geodesic Submanifolds are constructed by obtaining an orthonormal basis $\zeta_1, \zeta_2, ..., \zeta_d$ of tangent vectors that span the tangent space $T_\mu\mathcal{M}$. These tangent vectors form a subspace V. The nested subspaces are represented by $V_k = span(\zeta_1 ..., \zeta_k)$. The image of the nested subspaces under the exponential map are the Principal Geodesic submanifolds. The first principal component chosen to maximize the projected variance is given by

$$\zeta_1 = \underset{\|\zeta\|=1}{\arg\max} \sum_{i=1}^{N} \|Log_\mu(\pi_H(x_i))\|^2 \tag{6}$$

where $H = Exp_\mu(span(\zeta))$
The projection operator is approximated as [2]

$$Log_\mu(\pi_H(x)) \approx \sum_{i=1}^{k} \langle \zeta_i, Log_\mu(x) \rangle \tag{7}$$

The remaining principal directions are defined as

$$\zeta_k = \underset{\|\zeta\|=1}{\arg\max} \sum_{i=1}^{N} \|Log_\mu(\pi_H(x_i))\|^2 \tag{8}$$

where $H = Exp_\mu(span(\zeta_1, \zeta_2, \ldots, \zeta_{k-1}, \zeta))$
Substituting for the projection operator, we get

$$\zeta_1 \approx \underset{\|\zeta\|=1}{\arg\max} \sum_{i=1}^{N} \langle \zeta, Log_\mu(x_i) \rangle^2 \tag{9}$$

$$\zeta_k \approx \underset{\|\zeta\|=1}{\arg\max} \sum_{i=1}^{N} \sum_{j=1}^{k-1} \langle \zeta_j, Log_\mu(x_i) \rangle^2 + \langle \zeta, Log_\mu(x_i) \rangle^2 \tag{10}$$

The above minimization problem is simply the PCA in $T_\mu \mathcal{M}$ of the vectors $Log_\mu(x_i)$

Following are the properties of the PGA:

- It preserves positive definiteness of Covariance matrix after projection.
- Determinant and orientation of the Covariance matrix is preserved
- Any matrix generated by Principal geodesic components ζ_i's are also positive definite.

In Eigen faces, it is assumed that all the faces lie in a subspace that maximizes the projected variances of the training samples. On similar lines we assume that the all the faces represented by covariance matrices lie on a geodesic submanifold as explained above.

Let covariance descriptor of training set be represented as $x_1^{tr}, x_2^{tr}, \ldots, x_n^{tr} \in \mathcal{M}$.

Calculate μ mean of the points $x_1^{tr}, x_2^{tr}, \ldots, x_n^{tr}$.

The features x_i^{tr} are mapped onto the tangent space to obtain

$$f_i = Log_\mu(x_i^{tr})$$

The principal geodesic components are calculated to obtain $\zeta_1, \zeta_2, \zeta_3, \ldots, \zeta_d$. The new projected principal features (PGA-RCM) are now generated as

$$p_i^{tr} = Exp_\mu \left(\sum_{k=1}^{d} \lambda_{i,k} \zeta_k \right) \tag{11}$$

where $\lambda_{i,k}$ are the coefficients obtained by

$$\lambda_{i,k} = \zeta_k^T f_i \tag{12}$$

and we define the PGA feature v_i of subject i as:

$$v_i = \sum_{k=1}^{d} \lambda_{i,k} \zeta_k \tag{13}$$

For classification, test data is also projected on to this submanifold to obtain the PGA-RCM, p^t, and NN classification for face recognition is employed using the measure

$$d(p_i^{tr}, p^t) = \|Log_{p_i^{tr}}(p^t)\| \tag{14}$$

As in [13], the above measure can be given in terms of generalized eigen values κ_i of the covariance matrices p_i^{tr} and p^t. i.e.,

$$d(p_i^{tr}, p^t) = \sqrt{\sum_{i=1}^{d} (log(\kappa_i))^2} \tag{15}$$

In the next section we see sparse modeling of face recognition using these PGA features.

3. FACE RECOGNITION USING SPARSE MODELING

It has been shown by Olshausen [9] that the human perception of vision is sparsely modeled. This concept is explored by Wright *et al.* [15]. It is shown that the sparsest solution is indeed discriminative for the classification of the face as each face is compactly represented as a linear combination of its training set. Such compact representation is extremely useful if the training set is large. The sparse representation problem is solved efficiently using ℓ_1 minimization or Basis pursuit introduced in the context of compressive sensing by Donoho *et al.* [1]. We employ the model followed by Wright *et al.* [15] explained in next paragraph using the principal geodesic features.

Let c be the number of classes of different subjects. Let $f \in T_\mu \mathcal{M}$ be the covariance feature on tangent space of a given subject as explained in the previous section. Let n_i be the number of training images available for each subject. Let this be denoted by $\{f_{1,i}, f_{2,i}, \ldots, f_{n_i,i}\} \in T_\mu \mathcal{M}$, where n_i is the number of training samples of subject i. Assume these vectors span a geodesic submanifold of the subject i. Any test image whose tangent space features is $l \in T_\mu \mathcal{M}$ can now be represented as linear combination of the training set.

$$l = \psi \alpha \tag{16}$$

where $\psi = [f_{1,1}, \ldots, f_{n_1,1}, \ldots, f_{1,c}, \ldots, f_{n_c,c}]$. If the test image l belongs to subject i, then its representation in ψ basis can be assumed to be sparse with non zero coefficients at locations corresponding to the vectors of the i^{th} subject. If the dimension of f's is larger than the number of training images, then the (16) becomes an overdetermined system. This high dimensionality problem can be addressed using the Principal Geodesic analysis. If R is the dimensionality reduction matrix in the tangent space $T_\mu \mathcal{M}$ (note that the elements of the matrix ψ are in the tangent space) then (16) can be modified as below

$$y = Rl = R\psi\alpha \tag{17}$$

Now, the system of equation in (17) is ensured to be underdetermined and a test image can be represented as a linear combination of the training images of subject only to which it belongs to. Hence a sparse solution of α can be determined by solving the following problem

$$\widehat{\alpha} = \arg\min \|\alpha\|_{\ell_0} \text{ subject to } y = R\psi\alpha \tag{18}$$

But solving (18) is an NP hard problem. Recent development in compressive sensing by Donoho [1], show that if the solution α is "*sparse enough*" then the solution of the ℓ_0 minimization of (18) is equivalent to solving the ℓ_1 minimization problem as below

$$\widehat{\alpha} = \arg\min \|\alpha\|_{\ell_1} \text{ subject to } y = R\psi\alpha \tag{19}$$

But since the real data is noisy it may not be possible to represent the test image as sparse linear combination of the training images, in which case the constraint in (18) and (19) will not be appropriate. Hence the modified optimization problem can be expressed as below

$$\widehat{\alpha} = \arg\min \|\alpha\|_{\ell_1} \text{ subject to } \|y - R\psi\alpha\|_2 \leq \epsilon \tag{20}$$

where ϵ is a very small quantity greater than zero.

3.1 Classification

In the ideal case the non-zero entries of the estimate α will be coefficients of the columns in $R\psi$ which belong to the same subject. But in practice is not true because of the inherent noise in the data. As a result the non-zero entries may be associated with multiple classes. Taking advantage of the subspace structure, we classify the test vector l based on how well the coefficients associated with all the training vectors of each subject reproduce l. The residue $r(i)$ of test vector l, with respect to subject i is calculated by defining the characteristic function $\delta_i : R^n \to R^n$ for subject i, i.e. for $\alpha \in R^n$, $\delta_i(\alpha)$ is a vector whose only nonzero entries are the entries in α that are associated with subject i.

$$\mathrm{r}(i) = \|y - R\psi\delta_i(\widehat{\alpha})\|_{\ell_2} \tag{21}$$

Here $\widehat{\alpha}$ is the solution of optimization problem (20).

4. FACE RECOGNITION WITH SPARSE MODELING IN KERNEL SPACE

It is shown that linear models are inaccurate if we need to recognize faces against slight changes in pose (but still close to frontal pose), severe expression changes and scale variations. Contrarily the nonlinear models captures higher order statistics beyond second order there by offering rich feature representation and exploiting this could be crucial for classification. A non-linear modeling problem can be posed as a linear modeling problem in a higher dimensional space, thanks to Cover's theorem for linear separability of patterns. To enhance the performance of the sparse linear modeling for **free form** face recognition we propose sparse representation using non-linear models. This is accomplished by defining a non-linear map ϕ which maps training examples in tangent space to higher dimensional reproducing kernel Hilbert space. In this approach instead of expressing the test example in tangent space as linear combination of training examples, see (16), the data vectors in tangent space is transformed to a higher dimensional feature space through a nonlinear mapping. The transformed test vector is expressed as a linear combination of transformed training vectors. Recent advances in kernel methods demonstrate a way to efficiently perform computations in feature space using kernel trick. We employ Kernel based feature extraction techniques such as Kernel PCA or Kernel FDA [16].

4.1 Kernel functions

Reproducing kernels are functions of form $k : \mathcal{X}^2 \to \Re$, where $\mathcal{X}^2$ is a Cartesian product, and are those functions which, for all finite pattern sets

$$\{x_1, x_2, \cdots, x_\ell \subset \mathcal{X}\}$$

give rise to positive semi definite matrices K with $K_{i,j} := k(x_i, x_j)$. Here $\mathcal{X}$ denotes compact set in which data points live. In most pattern recognition task, the set $\mathcal{X}$ is N- dimensional Euclidean space $\Re^N$. The properties that a kernel function must satisfy are:

1. The kernel function must be symmetric in its arguments.

2. Must be positive semidefinite obeying the following condition
$$\sum_{i=1}^{n}\sum_{j=1}^{n} k(x_i, x_j)\gamma_i\gamma_j \geq 0$$
for some $n \in \mathbb{N}$ and $\gamma_i \neq 0 \quad \forall\ i$. This property is also stated as: For any finite n set of data points $\{x_i\}_{i=1}^n \in \mathcal{X}$ the matrix K with $K_{i,j} = k(x_i, x_j)$ is positive semidefinite.

Kernel functions handle computations of nonlinear models in input space $\Re^N$ by reducing them to linear models in higher dimensional feature space $\mathcal{F}$. An algorithm which involves only inner-products of data points in input space $\Re^N$ can be easily converted to nonlinear algorithm by substituting the all dot product terms by kernel function k. This corresponds to mapping the data points in $\Re^N$ to higher dimensional feature space $\mathcal{F}$ with a map $\phi : \Re^N \to \mathcal{F}$, and taking inner product in feature space $\mathcal{F}$, i.e, $k(x_i, x_j) = \langle\phi(x_i), \phi(x_j)\rangle$. This approach is commonly referred to as the "Kernel Trick" in machine learning literature. One major drawback of such approach is that the solutions to nonlinear problem posed in feature space with kernel trick can only be obtained as linear combinations of input data points.

4.2 Kernels for sparse model

Let ϕ be any nonlinear mapping from input space to feature space.

$$\phi\ : T_\mu\mathcal{M}\ \to\ \mathcal{F} \tag{22}$$

The notations used in this section are:

1. c is the number of classes.

2. $v_{j,k}$ is j^{th} PGA features of training image of k^{th} subject.

3. n_i is the number of images for subject i.

4. $n = n_1 + n_2 + n_3 + ... + n_c$ be total number of images.

5. l be PGA features of test image

Let $\Psi = [\phi(v_{1,1})\ \phi(v_{2,1})\ \phi(v_{n_1,1}) \ldots\ \phi(v_{n_c,c})]$ be a matrix with its columns as transformed PGA features of training images and l be the PGA feature of test data. We can now express the transformed test vector $\phi(l)$ as

$$\phi(l) \;=\; \Psi\alpha \tag{23}$$

As in Section 3, we can expect the representation of $\phi(l)$ to be sparse in basis Ψ. Thus recognizing the class of test pattern is solved by finding this sparse representation. But since (23) is overdetermined and the problem is of very high dimensionality, we make use of suitable dimensionality reduction technique like KPCA to convert the problem (23) to underdetermined problem. This facilitates the use of kernel trick to perform higher dimensional computations efficiently. Hence we solve the following problem:

$$R^T\phi(l) \;=\; R^T\Psi\alpha \tag{24}$$

where R is the dimensionality reduction matrix. This problem is reduced to an ℓ_1 minimization the derivation of which is explained next.

4.2.1 Sparse modeling in feature space with Kernel PCA

Let

$$C^{\phi} = (1/n)\sum_{i=1}^{n}\phi(v_i)\phi(v_i)^T \tag{25}$$

be the covariance matrix in feature space, where v_i are the PGA features. For simplicity here it is assumed that the data is centered in feature space. It is shown that the projection directions in KPCA corresponds to eigenvectors of covariance matrix with leading d eigenvalues denoted as $u_1, u_2, ...u_d$. Here d indicates dimensionality reduction parameter.

Since C^{ϕ} is symmetric PSD matrix, all the eigenvectors corresponding to nonzero eigenvalue must live in range space of C^{ϕ} and hence we can express these vectors as linear combination of transformed training examples.

$$u_k = \sum_{i=1}^{n}\beta_{k,i}\phi(v_i) \tag{26}$$

Here $[\beta_{k,1}.....\beta_{k,n}]^T$ is representation for k^{th} eigenvector. Now we need to solve for leading d eigenvectors

$$C^{\phi}u = \lambda u \tag{27}$$

This can be written as (To find k^{th} eigenvector):

$$\frac{1}{n}\sum_{i=1}^{n}\phi(v_i)\phi(v_i)^T\sum_{j=1}^{n}\beta_{k,j}\phi(v_j) = \lambda\sum_{j=1}^{n}\beta_{k,j}\phi(v_j) \tag{28}$$

Multiply $\phi(v_r)^T$ both sides, for $\forall\ r \;=\; 1\ldots n$.

$$\frac{1}{n}\phi(v_r)^T\sum_{i=1}^{n}\phi(v_i)\phi(v_i)^T\sum_{j=1}^{n}\beta_{k,j}\phi(v_j) = \lambda\sum_{j=1}^{n}\beta_{k,j}\phi(v_r)^T\phi(v_j) \tag{29}$$

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\beta_{k,j}\phi(v_r)^T\phi(v_i)\phi(v_i)^T\phi(v_j) = \lambda\sum_{j=1}^{n}\beta_{k,j}\phi(v_r)^T\phi(v_j) \tag{30}$$

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\beta_{k,j}K_{r,i}K_{i,j} = \lambda\sum_{j=1}^{n}\beta_{k,j}K_{r,j}\quad\forall\ r\ \in\ \{1,2,3....n\} \tag{31}$$

This set of equations can be reduced to matrix form as:

$$(1/n)K^2\beta = \lambda K\beta \tag{32}$$

where $\beta \;=\; [\beta_{k,1}\ \beta_{k,2}\ldots\ \beta_{k,n}]^T$ is the column vector. The above problem can also be solved as:

$$K\beta = n\lambda\beta \tag{33}$$

The KPCA projection matrix is

$$R = [u_1\ u_2\ldots\ u_d] \tag{34}$$

where k^{th} eigenvector is $u_k \;=\; \sum_{i=1}^{n}\beta_{k,i}\phi(v_i)$. Thus, (24) becomes:

$$[u_1\ u_2\ldots\ u_d]^T\phi(l) \;=\; [u_1\ u_2\ldots\ u_n]^T\Psi\alpha \tag{35}$$

Which is

$$\underbrace{\left[\sum_{i=1}^{n}\beta_{r,i}\phi(v_i)^T\phi(l)\right]}_{(d\,\times\,1)\text{ vector}} \;=\; \underbrace{\left[\sum_{i=1}^{n}\beta_{r,i}\phi(v_i)^T\phi(v_c)\right]}_{(d\,\times\,n)\text{ matrix}}\alpha \tag{36}$$

Where $r = 1 \to d$ and $c = 1 \to n$

This can be further written as:

$$\underbrace{[\beta_{r,c}]_{d\times n}}_{\text{KPCA matrix}}\ \underbrace{\left[\phi(v_c)^T\phi(l)\right]_{n\times 1}}_{w\text{ vector}} \;=\; [\beta_{r,c}]\ \underbrace{\left[\phi(v_r)^T\phi(v_c)\right]_{d\times n}}_{\text{Kernel gram Matrix}}\alpha \tag{37}$$

The simplified matrix version is thus given as:

$$R_{\beta}w = R_{\beta}K\alpha \tag{38}$$

where $w \;=\; [\phi(v_1)^T\phi(l).......\phi(v_n)^T\phi(l)]^T$ and K is kernel gram matrix with $K_{i,j} \;=\; \phi(v_i)^T\phi(v_j)$ and R_{β} is the Kernel PCA matrix.

We solve the following optimization problem:

$$\widehat{\alpha} \;=\; \min_{\alpha}\|\alpha\|_{\ell_1}\quad\text{subject to}\quad R_{\beta}K\alpha \;=\; R_{\beta}w \tag{39}$$

The residue is then calculated for each of the test face as in the previous section.

5. EXPERIMENTS AND RESULTS

5.1 Gabor features based Covariance matrix

It is known that for the task of face recognition using Covariance descriptors, Gabor features have been very effective [10].They are orientation and scale tunable filters. A two dimensional Gabor transform $g(x, y)$ is given as in [7]

$$g(x,y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right) exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right) + j2\pi Wx\right]$$

Here W is the upper frequency. Self-similar functions, referred as Gabor Wavelets are obtained by appropriate dilations and rotations of the mother wavelet $g(x, y)$, and let it be represented as $g_{s,r}(x, y)$ where s and r denote the scale and rotation respectively. Vectorized Gabor feature is given by

$$F(x, y) = [f_{0,0}(x, y) \ \ldots f_{S-1,Z-1}(x, y)]^T \tag{40}$$

where Z is the total number of rotations and S is the number of scales and $f_{s,r}(x, y) = I(x, y) * g_{s,r}(x, y)$, $*$ being the convolution operator. For a face containing $W \times L$ pixels, the covariance descriptors are now calculated as

$$C_R = \frac{1}{W \times L}\sum_{x=1}^{W}\sum_{y=1}^{L}(F(x, y) - M)(F(x, y) - M)^T \tag{41}$$

where M is the mean, calculated by

$$M = \frac{1}{W \times L}\sum_{x=1}^{W}\sum_{y=1}^{L}F(x, y) \tag{42}$$

We use 5 region covariance descriptors [13] of gabor features for each image [10], one for the entire image, two for the left and right halves and two more for the top and bottom halves. This is illustrated in Fig 2

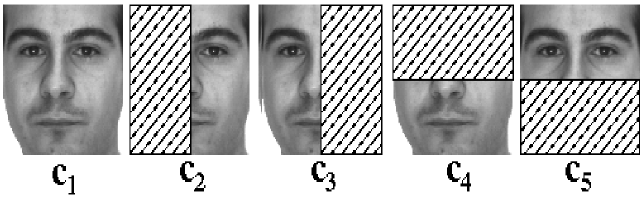

Figure 2: Image regions corresponding to the five covariance descriptors.

Hence, each image datapoint is represented as $D_t = (C_{t1}, C_{t2}, C_{t3}, C_{t4}, C_{t5})$ where t is the face. For each of the regions we obtain the principal geodesic components and obtain a new covariance matrix (PGA-RCM) represented as $P^{tr} = (P_1^{tr}, P_2^{tr}, P_3^{tr}, P_4^{tr}, P_5^{tr})$. Similarly, we obtain the covariance matrix projected on the geodesic submanifold for the test image P^t. The distance between P^{tr} and P^t is calculated as

$$\rho(P^{tr}, P^t) = \sum_{i=1}^{5} d(P_i^{tr}, P_i^t) - \max_i \ d(P_i^{tr}, P_i^t) \tag{43}$$

where d is the distance given in (14)

For the sparse model, we calculate the residue $r(i, k)$ as in (21) for each region k and Class label of t is given by

$$Class(t) = \underset{i=1\rightarrow c}{\arg\min}\left(\sum_{k=1}^{5} r(i, k) - \max_k \ r(i, k)\right) \tag{44}$$

Of the five covariance matrices the least matching matrix is ignored. This gives robustness to possible illumination changes, noise and slight changes in the pose.

The proposed methods using PGA-RCM (PGA of Covariance matrix in Section 2) and PGA features (PGA of RCM mapped on the tangent space) with sparse modeling (Sections 3,4) for free form face recognition was tested on two benchmark databases viz, AR and YaleB Database. In our experiments, we run a Viola-Jones Face detector [14] on the uncropped images and use these outputs for evaluating the performance of the proposed method without employing any alignment and cropping. In Fig. 3, we show the comparison between a properly cropped and aligned face with the face detector output (free-form faces).

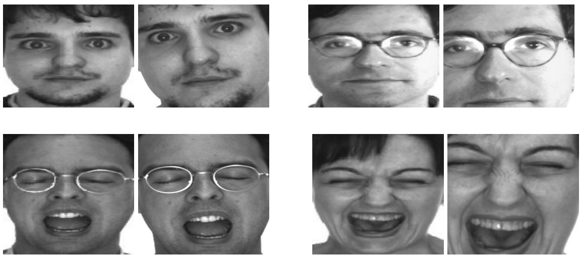

Figure 3: Four face pairs with properly cropped and aligned faces placed alongside face detector outputs. Inconsistencies in scale and cropping can severely impede recognition accuracy.

In [15], it is shown that sparse modeling of faces outperforms many of the state-of-the-art algorithms. We compare the effectiveness of the PGA-RCM and PGA features by comparing with other features like PCA with sparse modeling [15]. We also kernelize the sparse representation and show results of KPCA and KFDA on image intensities and PGA features with sparse modeling. PGA-sparse is the sparse modeling using PGA features *i.e.* features on the tangent space. KPGA-sparse is kernelized PCA on PGA features with sparse modeling, and KFPGA-sparse is Kernel fisher analysis on PGA features with sparse representation. The choice of kernels were empirically selected to obtain better recognition accuracy.

5.2 Uncropped AR database

We have considered a total of 110 subjects, 57 male and 53 females. Each subject has 26 images with varying expression, occlusion, illumination and little pose variation. We consider only faces which are not occluded for experimental evaluation.The images of individuals are captured in two sessions with 13 images in each session. We consider 7 images of first session for training and remaining 7 images of session two for testing. The Figure 4 shows a few examples of the free form faces for AR database. In Table 1, we show the performance of the sparse methods and PGA-RCM for 100 dimensions.

From Table 1, it is seen that PGA-RCM gives better accuracies. In Table 2, we show the results using Kernelized

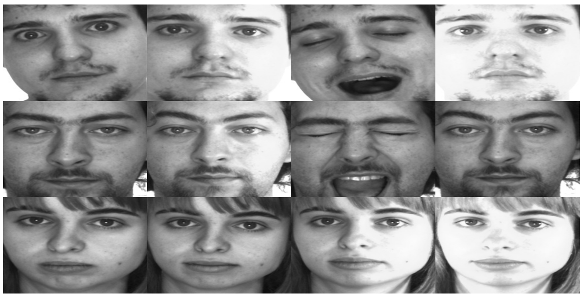

Figure 4: Free form faces of AR Database

Method	Accuracy
PCA-sparse	61
PGA-sparse	55
PGA-RCM	67

Table 1: Accuracies with sparse representation using PCA and PGA-RCM features with 100 dimensions for AR database

sparse models for image intensities as in [15] and PGA features (PGA dimension 101) modeled on the tangent space. For KPCA and KPGA we used RBF kernels with variance set to 100 and dimension was chosen to be 100. For KFDA and KFPGA polynomial kernel were used with maximum allowed dimensions(in this case one less than total number of classes).

Method	Accuracy
KPCA-sparse	63
KPGA-sparse	53
KFDA-sparse	77
KFPGA-sparse	85

Table 2: Accuracies with Kernelized sparse models using features intensities and PGA features(101 dimensions) for AR database.

From Table 2 we clearly see the improvement of accuracies using the PGA features for kernel sparse representation.

In the Fig 5 we show the plot of dimension versus accuracy for PGA-RCM and sparse-pca model and show the superior accuracy of the PGA-RCM.

The Fig 6 shows the plot of using PGA features of different dimension in the tangent space and then performing Kernel Fisher Analysis (Fisher dimension was chosen as 110) on each of these features using the sparse representation. From the plot, we infer that there is a flexibility in choosing PGA features of appropriate dimension. Such a behaviour is not seen with intensity-based features.

5.3 Uncropped YaleB database

This database contains 28 subjects, both male and female faces. There are 64 images per subject, with varying illumination, expression and little pose. We make random selection of 10 training examples and another 10 remaining unseen patterns for testing. We evaluate the performance of proposed algorithm with exactly the same setup as explained for AR database using the Detector outputs. The Fig 7 shows a few examples of the free form faces for YaleB database.

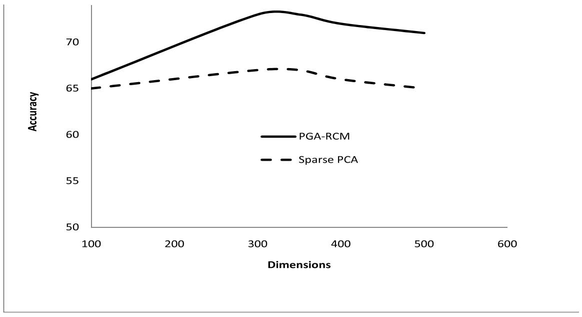

Figure 5: Plot of Accuracies of PGA-RCM and sparse representation v/s Dimensions for AR database

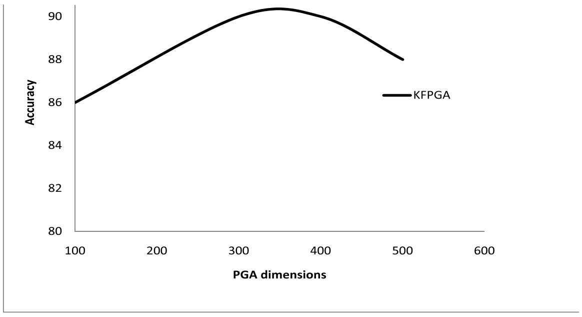

Figure 6: Plot of PGA features of different dimensions vs Accuracy in Kernel Fisher Analysis for AR database

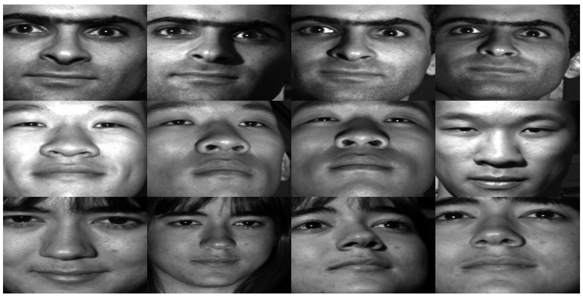

Figure 7: Free form faces of YaleB Database

In Table 3, we compare the results for PCA-sparse and PGA-sparse (tangent space modeling) with PGA-RCM features with 100 dimensions. We see that PGA-RCM clearly outperforms the other two. We don't see much difference in the performance of both the sparse representations.

The Table 4 shows the results for kernel versions. We see that KFPGA outperforms the other methods

The Fig 8 shows the variation of KPCA sparse and KPGA sparse with varying dimensions for the RBF kernel. The Fig 9 shows the variation of KFDA sparse and KFPGA sparse with varying Fisher dimensions.

Method	Accuracy
PCA-sparse	65
PGA-sparse	63
PGA-RCM	82

Table 3: Accuracies with sparse PCA and PGA representation, and PGA-RCM for the YaleB database

Method	Accuracy
KPCA-sparse	67
KPGA-sparse	66
KFDA-sparse	72
KFPGA-sparse	83

Table 4: Accuracies with Kernelized sparse models using intensity features and PGA features (100 dimensions) on YaleB database

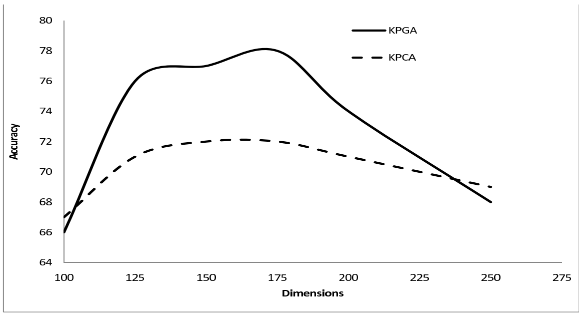

Figure 8: Figure depicting Accuracies of KPGA-sparse and KPCA-sparse representation vs Dimensions for RBF kernel for YaleB database

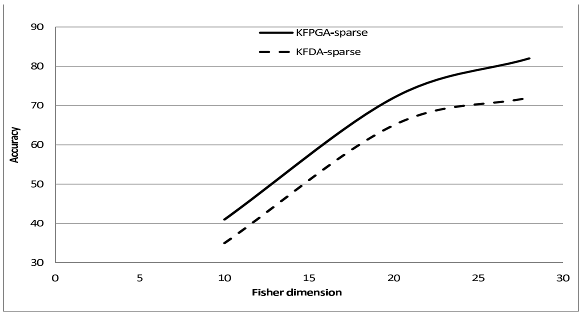

Figure 9: Figure depicting Accuracies of KFPGA-sparse and KFDA-sparse representation v/s Dimensions for Kernel Fisher Analysis for YaleB database

From the results we can infer that for linear models PGA-RCM gives better accuracy and for kernel methods PGA features are better features than the image intensity features.

6. CONCLUSION

In this paper, we have proposed free-form face recognition using PGA for Covariance descriptors. We show that these descriptors are efficient for representing free-from faces by showing superior results over intensity-based sparse models for representing faces. Also, kernelized sparse representation is found to achieve higher recognition accuracy than its linear counterpart. Investigating FR performance with sparse representation on the original manifold, instead of the tangent space, is an interesting direction for future work.

7. REFERENCES

[1] D. L. Donoho. Compressed sensing. *IEEE Trans. Inform. Theory*, 52:1289–1306, 2006.

[2] P. T. Fletcher and S. Joshi. Riemannian geometry for the statistical analysis of diffusion tensor data. *Signal Processing*, 87(2):250–262, 2007.

[3] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001.

[4] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang. Face recognition using laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):328–340, 2005.

[5] C. Liu. Gabor-based kernel pca with fractional power polynomial models for face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):572–581, 2004.

[6] J. Lu, N. Plataniotis Kostantinos, and V. Anastasios. Face recognition using lda-based algorithms. *IEEE Transactions on Neural Networks*, 14(1):195–201, 2003.

[7] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(8):837–842, 1996.

[8] A. Martinez and R. Benavente. The AR face database. Technical report, CVC Technical report, 1998.

[9] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 37(23):3311 – 3325, 1997.

[10] Y. Pang, Y. Yuan, and X. Li. Gabor-based region covariance matrices for face recognition. *IEEE Trans. Circuits and Systems for Video Technology*, 18:989–993, 2008.

[11] X. Pennec, P. Fillard, and N. Ayache. A riemannian framework for tensor computing. *International Journal of Computer Vision*, 66(1):41–66, 2006.

[12] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[13] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In *ECCV06*, pages II: 589–600, 2006.

[14] P. Viola and M. J. Jones. Robust real-time object detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[15] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. volume 31, pages 210–227, 2009.

[16] M. Yang. Kernel eigenfaces vs. kernel fisherfaces:face recognition using kernel methods. In *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 205–211, 2002.