Statistical Computations on Grassmann and Stiefel Manifolds for Image and Video-Based Recognition

Pavan Turaga, *Member*, *IEEE*, Ashok Veeraraghavan, *Member*, *IEEE*, Anuj Srivastava, *Senior Member*, *IEEE*, and Rama Chellappa, *Fellow*, *IEEE*

Abstract—In this paper, we examine image and video-based recognition applications where the underlying models have a special structure—the linear subspace structure. We discuss how commonly used parametric models for videos and image sets can be described using the unified framework of Grassmann and Stiefel manifolds. We first show that the parameters of linear dynamic models are finite-dimensional linear subspaces of appropriate dimensions. Unordered image sets as samples from a finite-dimensional linear subspace naturally fall under this framework. We show that an inference over subspaces can be naturally cast as an inference problem on the Grassmann manifold. To perform recognition using subspace-based models, we need tools from the Riemannian geometry of the Grassmann manifold. This involves a study of the geometric properties of the space, appropriate definitions of Riemannian metrics, and definition of geodesics. Further, we derive statistical modeling of inter and intraclass variations that respect the geometry of the space. We apply techniques such as intrinsic and extrinsic statistics to enable maximum-likelihood classification. We also provide algorithms for unsupervised clustering derived from the geometry of the manifold. Finally, we demonstrate the improved performance of these methods in a wide variety of vision applications such as activity recognition, video-based face recognition, object recognition from image sets, and activity-based video clustering.

Index Terms—Image and video models, feature representation, statistical models, manifolds, Stiefel, Grassmann.

1 Introduction

Many applications in computer vision, such as dynamic textures [2], [3], human activity modeling and recognition [4], [5], video-based face recognition (FR) [6], and shape analysis [7], [8], involve learning and recognition of patterns from exemplars which obey certain constraints. To enable this study, we often make simplifying assumptions of the image-formation process such as a pin-hole camera model or the Lambertian reflectance model. These assumptions lead to constraints on the set of images thus obtained. A classic example of such a constraint is that images of a convex object under all possible illumination conditions form a "cone" in image-space [9]. Once the underlying assumptions and constraints are well understood, the next important step is to design inference algorithms that are consistent with the algebra and/or geometry of the constraint set. In this paper, we shall examine image and video-based recognition applications where the models have a special structure—the linear subspace structure.

In many of these applications, given a database of examples and a query, the following two questions are to be addressed—1) What is the "closest" example to the query in the database? 2) What is the "most probable" class to which the query belongs? A systematic solution to these problems involves a study of the underlying constraints that the data obeys. The answer to the first question involves a study of the geometric properties of the space, which then leads to appropriate definitions of Riemannian metrics and further to the definition of geodesics, etc. The answer to the second question involves statistical modeling of inter and intraclass variations. It is well known that the space of linear subspaces can be viewed as a Riemannian manifold [10], [11]. More formally, the space of d-dimensional subspaces in $\mathbb{R}^n$ is called the Grassmann manifold. On a related note, the Stiefel manifold is the space of d orthonormal vectors in $\mathbb{R}^n$. The study of these manifolds has important consequences for applications such as dynamic textures [2], [3], human activity modeling and recognition [4], [5], video-based face recognition [6], and shape analysis [7], [8], where data naturally lie either on the Stiefel or the Grassmann manifold. Estimating linear models of data is standard methodology in many applications and manifests in various forms such as linear regression, linear classification, linear subspace estimation, etc. However, comparatively less attention has been devoted to statistical inference on the space of linear subspaces.

- *P. Turaga is with the Center for Automation Research, University of Maryland, College Park, 4409 AV Williams Bldg., College Park, MD 20742. E-mail: pturaga@umiacs.umd.edu.*
- *A. Veeraraghavan is with Mitsubishi Electric Research Labs, 201 Broadway, Cambridge, MA 02139. E-mail: vashok@merl.com.*
- *A. Srivastava is with the Department of Statistics, 106D OSB, Florida State University, Tallahassee, FL 32306. E-mail: anuj@stat.fsu.edu.*
- *R. Chellappa is with the Department of Electrical and Computer Engineering, University of Maryland, College Park, 4411 AV Williams Bldg., College Park, MD 20742. E-mail: rama@umiacs.umd.edu.*

Manuscript received 20 Oct. 2009; revised 19 Oct. 2010; accepted 25 Nov. 2010; published online 15 Mar. 2011.
Recommended for acceptance by P. Perez.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2009-10-0700.
Digital Object Identifier no. 10.1109/TPAMI.2011.52.

1.1 Prior Work

The Grassmann manifold's geometric properties have been utilized in certain vision problems involving subspace constraints. Examples include [12], which deals with optimization over the Grassmann manifold for obtaining informative projections. The Grassmann manifold structure of the affine shape space is also exploited in [13] to perform affine invariant clustering of shapes. Hamm and Lee [14] perform discriminative classification over subspaces for object recognition tasks by using Mercer kernels on the Grassmann manifold. In [15], a face image and its perturbations due to registration errors are approximated as a linear subspace and hence are embedded as points on a Grassmann manifold. Most of these methods do not employ statistics on the Grassmann manifold or are tuned to specific domains lacking generality. Srivastava and Klassen [16] exploited the geometry of the Grassmann manifold for subspace tracking in array signal processing applications. On a related note, the geometry of the Stiefel manifold has been found to be useful in applications where, in addition to the subspace structure, the specific choice of basis vectors is also important [17]. The methods that we present in this paper form a comprehensive (not exhaustive) set of tools that draw upon the Riemannian geometry of the Grassmann manifold. Along with the mathematical formulations, we also present efficient algorithms to perform these computations.

The geometric properties of general Riemannian manifolds form a subject matter of differential geometry; a good introduction can be found in [18]. Statistical methods on manifolds have been studied for several years in the statistics community. Some of the landmark papers in this area include [19], [20], [21]; however, an exhaustive survey is beyond the scope of this paper. The geometric properties of the Stiefel and Grassmann manifolds have received significant attention. A good introduction to the geometry of the Stiefel and Grassmann manifolds can be found in [10], which introduced gradient methods on these manifolds in the context of eigenvalue problems. These problems mainly involved optimization of cost functions with orthogonality constraints. A compilation of techniques for solving optimization problems with such matrix manifolds is provided in [22]. Algorithmic computations of the geometric operations in such problems were discussed in [11]. A compilation of research results on statistical analysis on the Stiefel and Grassmann manifolds can be found in [23].

In addition to the Grassmann manifold, general Riemannian manifolds have found important applications in the vision community. A recently developed formulation of using the covariance of features in image patches has found several applications such as texture classification [24], pedestrian detection [25], and tracking [26]. The Riemannian geometry of covariance matrices was exploited effectively in all of these applications to design state-of-the-art algorithms. More recently, Subbarao and Meer [27] provided an extension of euclidean mean shift clustering to the case of Riemannian manifolds.

Shape analysis is another application area where statistics on Riemannian manifolds have found wide applicability. Theoretical foundations for manifolds-based shape analysis were described in [7], [8]. Statistical learning of shape classes using nonlinear shape manifolds was presented in [28], where statistics are learned on the manifold's tangent space. Using a similar formulation, the variations due to execution rate changes in human activities is modeled as a distribution over time-warp functions, which are considered as points on a spherical manifold in [29]. This was used for execution rate-invariant recognition of human activities.

A preliminary version of this paper was presented in [1], which used extrinsic methods for statistical modeling on the Grassmann manifold. This paper provides a mathematically well-grounded basis for these methods, where the specific choice of the method in [1] is interpreted as a special case of using a nonparametric density estimator with an extrinsic divergence measure. In this paper, we provide more detailed analysis and show how to exploit the geometry of the manifold to derive intrinsic statistical models. This provides a more consistent approach than the extrinsic methods of [1]. Further, the dimensionality of the manifold presents a significant road block for computer implementation of Riemannian computations. Straightforward implementation of formulas for geodesic distances, exponential and inverse-exponential maps given in earlier work such as [10], [11], [27] is computationally prohibitive for large dimensions. This is especially true of our applications where we deal with high-dimensional image and video data. Toward this end, we also employ numerically efficient versions of these computations.

Contributions. We first show how a large class of problems drawn from face, activity, and object recognition can be recast as statistical inference problems on the Stiefel and/or Grassmann manifolds. Then, we present methods to solve these problems using the Riemannian geometry of the manifolds. We also discuss some recently proposed extrinsic approaches to statistical modeling on the Grassmann manifold. We present a wide range of experimental evaluation to demonstrate the effectiveness of these approaches and provide a comprehensive comparison.

Organization of the paper. In Section 2, we discuss parametric subspace-based models of image sets and videos and show how the study of these models can be recast as a study of the Grassmann manifold. Section 3 introduces the special orthogonal group and its quotient spaces—the Stiefel and the Grassmann manifolds. Section 4 discusses statistical methods that follow from the quotient interpretation of these manifolds. In Section 5, we develop supervised and unsupervised learning algorithms. Complexity issues and numerically efficient algorithms for performing Riemannian computations are discussed in Section 6. In Section 7, we demonstrate the strength of the framework for several applications including activity recognition, video-based face recognition, object matching, and activity-based clustering. Finally, concluding remarks are presented in Section 8.

2 Models for Videos and Images

2.1 Spatio-Temporal Dynamical Models and the ARMA Model

A wide variety of spatio-temporal data have often been modeled as realizations of dynamical models. Examples include dynamic textures [2], human joint angle trajectories

[4], and silhouettes [5]. A well-known dynamical model for such time-series data is the autoregressive and moving average (ARMA) model. Linear dynamical systems represent a class of parametric models for time series. A wide variety of time-series data such as dynamic textures, human joint angle trajectories, shape sequences, video-based face recognition, etc., are frequently modeled as ARMA models [2], [4], [5], [6]. The ARMA model equations are given by

$$f(t) = Cz(t) + w(t) \quad w(t) \sim N(0, R), \tag{1}$$

$$z(t+1) = Az(t) + v(t) \quad v(t) \sim N(0, Q), \tag{2}$$

where $z \in \mathbb{R}^d$ is the hidden state vector, $A \in \mathbb{R}^{d \times d}$ the transition matrix, and $C \in \mathbb{R}^{p \times d}$ the measurement matrix. $f \in \mathbb{R}^p$ represents the observed features, while w and v are noise components modeled as normal with 0 mean and covariances $R \in \mathbb{R}^{p \times p}$ and $Q \in \mathbb{R}^{d \times d}$, respectively.

For the ARMA model, closed-form solutions for learning the model parameters have been proposed in [30], [2] and are widely used. For high-dimensional time-series data (dynamic textures, etc.), the most common approach is to first learn a lower-dimensional embedding of the observations via PCA, and learn the temporal dynamics in the lower-dimensional space. Let observations $f(1), f(2), \ldots f(\tau)$, represent the features for the time indices $1, 2, \ldots \tau$. Let $[f(1), f(2), \ldots f(\tau)] = U\Sigma V^T$ be the singular value decomposition of the data. Then,

$$\hat{C} = U, \hat{A} = \Sigma V^T D_1 V (V^T D_2 V)^{-1} \Sigma^{-1},$$

where

$$D_1 = \begin{bmatrix} 0 & 0 \\ I_{\tau-1} & 0 \end{bmatrix} \quad \text{and} \quad D_2 = \begin{bmatrix} I_{\tau-1} & 0 \\ 0 & 0 \end{bmatrix}.$$

The model parameters (A, C) do not lie in a vector space. The transition matrix A is constrained to be stable with eigenvalues inside the unit circle. The observation matrix C is constrained to be an orthonormal matrix. For comparison of models, the most commonly used distance metric is based on subspace angles between column spaces of the observability matrices [31]. For the ARMA model of (2), starting from an initial condition $z(0)$, it can be easily shown that the *expected* observation sequence is given by [32]

$$E\begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \cdot \\ \cdot \end{bmatrix} z(0) = O_\infty(M) z(0). \tag{3}$$

Thus, the expected observation sequence generated by a time-invariant model $M = (A, C)$ lies in the column space of the extended *observability* matrix given by $O_\infty^T = [C^T, (CA)^T, (CA^2)^T, \ldots (CA^n)^T \ldots]$. In experimental implementations, we approximate the extended observability matrix by the finite observability matrix, as is commonly done [33], $O_m^T = [C^T, (CA)^T, (CA^2)^T, \ldots (CA^{m-1})^T]$. The size of this matrix is $mp \times d$. The column space of this matrix is a d-dimensional subspace of $\mathbb{R}^{mp}$, where d is the dimension of the state-space z in (2). d is typically of the order of 5-10.

Thus, given a database of videos, we estimate the model parameters as described above for each video. The finite observability matrix is computed next. To represent the subspace spanned by the columns of this matrix, we store *an* orthonormal basis computed by Gram-Schmidt orthonormalization. Since a subspace is a point on a *Grassmann* manifold, a linear dynamical system can be alternately identified as a point on the Grassmann manifold corresponding to the column space of the observability matrix.

2.2 Image Sets as Collections of Subspaces

In image and object recognition, recent methods have focused on utilizing multiple images of the same object, taken under varying viewpoints or varying illumination conditions, for recognition [34], [14], [35], [36]. For example, it was shown by Jacobs et al. that the illumination cone of a convex Lambertian surface can be approximated by a 9D linear subspace [37]. Motivated by this, the set of face images of the same person under varying illumination conditions is frequently modeled as a linear subspace of 9D [38].

Given a large set of images indexed by, say, the pose or viewing angle of the camera, we estimate multiple subspaces—one for each view—as the model of object appearance. The subspaces can be estimated by straightforward principal component analysis. Given another set of images during testing, we would like to compute the likelihood of it coming from a specific class. In the training phase, given a set of these subspaces for a given class, we would like to compute their class conditional densities. During testing, we are given a set of images taken under approximately the same viewing angle, which allows us to model the set using a subspace. Then, the maximum likelihood classification can be performed for each test instance using these class conditional distributions. However, since subspaces are viewed as elements of a Grassmann manifold, the goal is to learn a probability distribution over the Grassmann manifold from the given image data.

2.3 Overall Approach

The set of all d-dimensional linear subspaces of $\mathbb{R}^n$ is called the Grassmann manifold, which will be denoted as $\mathcal{G}_{n,d}$. The set of all $n \times d$ orthonormal matrices is called the Stiefel manifold and shall be denoted as $\mathcal{S}_{n,d}$. As discussed in the applications above, we are interested in computing statistical models over the Grassmann manifold. Let $U_1, U_2, \ldots, U_k$ be some points on $\mathcal{S}_{n,d}$ and we seek their sample mean, an average, for defining a probability model on $\mathcal{S}_{n,d}$. Recall that these U_is are tall, orthogonal matrices. It is easy to see that the euclidean sample mean $\frac{1}{k}\sum_{i=1}^k U_i$ is not a valid operation because the resultant mean does not have the property of orthonormality. This is because $\mathcal{S}_{n,d}$ is not a vector space. Similarly, many of the standard tools in estimation and modeling theory do not directly apply to such spaces, but can be adapted by accounting for the underlying nonlinear geometry.

On a computer, a subspace is stored as an orthonormal matrix which forms a basis for the subspace. As mentioned earlier, orthonormal matrices are points on the Stiefel manifold. However, since the choice of basis for a subspace is not unique, any notion of distance and statistics should be invariant to this choice. This requires us to interpret each point on the Grassmann manifold as an equivalence of points on the Stiefel manifold, where all orthonormal matrices that span the same subspace are considered equivalent. This

interpretation is more formally described as a *quotient* interpretation, i.e., the Grassmann manifold is considered a quotient space of the Stiefel manifold. Quotient interpretations allow us to extend the results of the base manifold such as tangent spaces, geodesics, etc., to the quotient space. In our case, it turns out that the Stiefel manifold itself can be interpreted as a quotient of a more basic manifold—the special orthogonal group $SO(n)$. A quotient of Stiefel is thus a quotient of $SO(n)$ as well. Thus, we shall study the Grassmann as a quotient of $SO(n)$. Hence, first we recapitulate relevant results of $SO(n)$, then review the required concepts from differential geometry that enable us to derive distances and statistical models on the special manifolds.

3 Preliminaries: The Special Orthogonal Group $SO(n)$ and Its Quotients

Let $GL(n)$ be the generalized linear group of $n \times n$ nonsingular matrices. It is not a vector space but a differentiable manifold, i.e., it can be locally approximated by subsets of a euclidean space. The dual properties of being a group and a differentiable manifold make it a *Lie group*. If we consider the subset of all orthogonal matrices, and further restrict to the ones with determinant $+1$, we obtain a subgroup $SO(n)$, called the *special orthogonal group*. It can be shown that this is a submanifold of $GL(n)$ and is also a group by itself; it possesses the Lie group structure. Since it has n^2 elements and $n + n(n-1)/2$ constraints (unit length columns $\rightarrow n$ constraints and perpendicular columns $\rightarrow n(n-1)/2$ constraints), it is an $n(n-1)/2$-dimensional Lie group. To perform differential calculus on a manifold, one needs to specify its tangent spaces. For the $n \times n$ identity matrix I, an element of $SO(n)$, the tangent space $T_I(SO(n))$ is the set of all $n \times n$ skew-symmetric matrices ([18]). For an arbitrary point $O \in SO(n)$, the tangent space at that point is obtained by a simple rotation of $T_I(SO(n))$: $T_O(SO(n)) = \{OX | X \in T_I(SO(n))\}$. Define an inner product for any $Y, Z \in T_O(SO(n))$ by $\langle Y, Z \rangle = trace(YZ^T)$, where *trace* denotes the sum of diagonal elements. With this metric $SO(n)$ becomes a Riemannian manifold.

Using the Riemannian structure, it becomes possible to define lengths of paths on a manifold. Let $\alpha : [0,1] \rightarrow SO(n)$ be a parameterized path on $SO(n)$ that is differentiable everywhere on $[0,1]$. Then $\frac{d\alpha}{dt}$, the velocity vector at t, is an element of the tangent space $T_{\alpha(t)}(SO(n))$. For any two points $O_1, O_2 \in SO(n)$, one can define a distance between them as the infimum of the lengths of all smooth paths on $SO(n)$ which start at O_1 and end at O_2

$$d(O_1, O_2) = \inf_{\{\alpha:[0,1]\rightarrow SO(n) | \alpha(0)=O_1, \alpha(1)=O_2\}} \left(\int_0^1 \sqrt{\left\langle \frac{d\alpha(t)}{dt}, \frac{d\alpha(t)}{dt} \right\rangle} dt \right). \quad (4)$$

A path $\hat{\alpha}$ which achieves the above minimum, if it exists, is a **geodesic** between O_1 and O_2 on $SO(n)$. Geodesics on $SO(n)$ can be written explicitly using the matrix exponential [10]. For an $n \times n$ matrix A, define its matrix exponential by: $\exp(A) = I + \frac{A}{1!} + \frac{A^2}{2!} + \frac{A^3}{3!} + \cdots$. It is easy to show that given any skew-symmetric matrix X, $\exp(X) \in SO(n)$. Now we can define geodesics on $SO(n)$ as follows: For any $O \in SO(n)$ and any skew-symmetric matrix X, $\alpha(t) \equiv O\exp(tX)$, is the unique geodesic in $SO(n)$ passing through O with velocity vector OX at $t = 0$.[1]

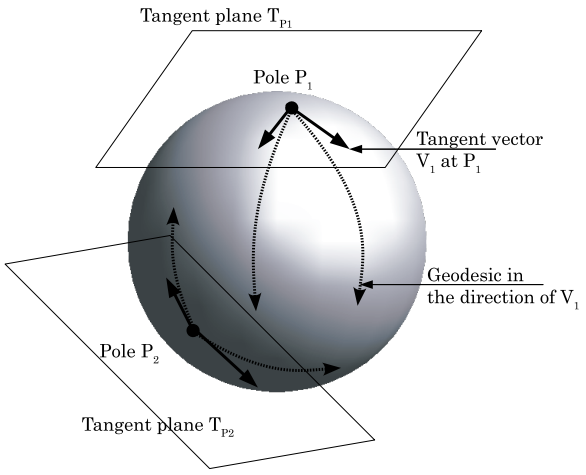


Fig. 1. Illustration of tangent spaces, tangent vectors, and geodesics. P_1 and P_2 are points on the manifold. T_{P_1} and T_{P_2} are the tangent spaces at these points. Note that there is a unique local mapping between the manifold and the tangent space and this local mapping depends upon the pole. Geodesics paths are constant velocity curves on the manifold. Tangent vectors correspond to velocities of curves on the manifold.

An important tool in statistics on a manifold is an exponential map. If M is a Riemannian manifold and $p \in M$, the **exponential map** $\exp_p : T_p(M) \rightarrow M$ is defined by $\exp_p(v) = \alpha_v(1)$, where α_v is a constant speed geodesic starting at p and with the initial velocity v. In case of $SO(n)$, the exponential map $\exp_O : T_O(SO(n)) \rightarrow SO(n)$ is given by $\exp_O(X) = O\exp(X)$, where the exponential on the right side is actually the matrix exponential. To help visualize these ideas, we illustrate the notions of tangent spaces, geodesics in Fig. 1. We illustrate the notions of the exponential map in Fig. 2.

3.1 Stiefel and Grassmann Manifolds as Quotients of $SO(n)$

A quotient of a group results from equivalence relations between points in the space. If one wants to identify certain elements of a set, using an equivalence relation, then the set of such equivalent classes forms a quotient space. This framework is very useful in understanding the geometry of $\mathcal{S}_{n,d}$ and $\mathcal{G}_{n,d}$ by viewing them as quotient spaces, using different equivalence relations, of $SO(n)$.

Stiefel manifold. A **Stiefel** manifold is the set of all d-dimensional orthonormal bases of $\mathbb{R}^n$ for $1 \leq d \leq n$. Since each orthonormal basis can be identified with an $n \times d$ matrix, a Stiefel manifold is also a set of $n \times d$ matrices with orthonormal columns. More interestingly, $\mathcal{S}_{n,d}$ can be viewed as a quotient space of $SO(n)$ as follows: Consider the subgroup of smaller rotations $SO(n-d)$ as a subgroup of $SO(n)$ using the embedding: $\phi_a : SO(n-d) \rightarrow SO(n)$, defined by

1. We note here the distinction between a *geodesic* and the *geodesic distance*. The *geodesic* passing through a point is simply a constant speed curve specified by its initial velocity, whereas the *geodesic distance* between two points is the length of the shortest constant speed curve passing through both points. For a point and a tangent vector on a Riemannian manifold, we can construct a geodesic path whose initial point and the velocity are same as the given pair.

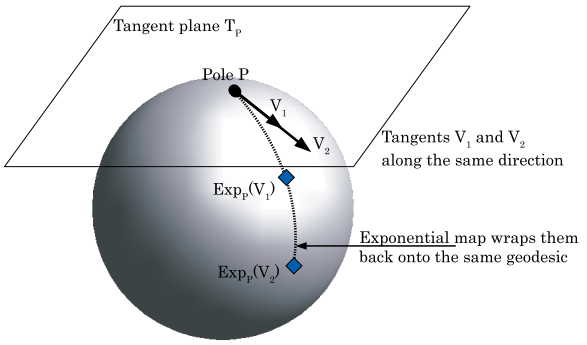


Fig. 2. Illustration of exponential maps. The exponential map is a "pull-back" map which takes points on the tangent space and pulls them onto the manifold in a manner that preserves distances. As an example, shown are two points V_1 and V_2 on the tangent space at pole P. Both points lie along the same tangent vector. The exponential map will map them onto the same geodesic. In a local neighborhood, the geodesic distance between the pole and the obtained points will be the same as the euclidean distance between the pole and the tangent vectors on the tangent space.

$$\phi_a(V) = \begin{bmatrix} I_d & 0 \\ 0 & V \end{bmatrix} \in SO(n)\,. \tag{5}$$

Now define two elements $O_1, O_2 \in SO(n)$ to be equivalent, i.e., $O_1 \sim_a O_2$, if $O_1 = O_2\phi_a(V)$ for some $V \in SO(n-d)$. (The subscript a is used to distinguish it from another equivalence relation used later for studying $\mathcal{G}_{n,d}$.) Note that $\phi_a(SO(n-d))$ consists of those rotations in $SO(n)$ that rotate only the last $(n-d)$ components in $\mathbb{R}^n$, leaving the first d unchanged. Hence, $O_1 \sim O_2$ if and only if their first d columns are identical, irrespective of the remaining columns. The resulting equivalence classes are: $[O]_a = \{O\phi_a(V) | V \in SO(n-d)\}$. Since all elements of $[O]_a$ have the same first d columns, we will use that submatrix $U \in \mathbb{R}^{n\times d}$ to represent $[O]_a$. $\mathcal{S}_{n,d}$ is now viewed as the set of all such equivalence classes and is denoted simply by $SO(n)/SO(n-d)$.

Grassmann manifold. A **Grassmann** manifold is the set of all d-dimensional subspaces of $\mathbb{R}^n$. Here we are interested in d-dimensional subspaces and not in a particular basis. In order to obtain a quotient space structure for $\mathcal{G}_{n,d}$, let $SO(d) \times SO(n-d)$ be a subgroup of $SO(n)$ using the embedding $\phi_b : (SO(d) \times SO(n-d)) \to SO(n)$

$$\phi_b(V_1, V_2) = \begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix} \in SO(n). \tag{6}$$

Define an equivalence relation on $SO(n)$ according to $O_1 \sim_b O_2$ if $O_1 = O_2\phi_b(V_1, V_2)$ for some $V_1 \in SO(d)$ and $V_2 \in SO(n-d)$. In other words, O_1 and O_2 are equivalent if the first d columns of O_1 are rotations of the first d columns of O_2 and the last $(n-d)$ columns of O_1 are rotations of the last $n-d$ columns of O_2. An equivalence class is given by

$$[O]_b = \{O\phi_b(V_1, V_2) | V_1 \in SO(d),\ V_2 \in SO(n-d)\},$$

and the set of all such equivalence classes is $\mathcal{G}_{n,d}$. Notationally, $\mathcal{G}_{n,d}$ can also be denoted as simply $SO(n)/(SO(d) \times SO(n-d))$. For efficiency, we often denote the set $[O]_b$ by $[U] = \{UV_1 \in \mathbb{R}^{n\times d} | V_1 \in SO(d)\}$, where U denotes the first d columns of O. Another way to express U is OJ where J is the matrix of the first d columns of I_n.

3.1.1 Tangent Structures via the Quotient Interpretation

As noted earlier, for any $O \in SO(n)$, a geodesic flow in a tangent direction, say, O^TA, is given by $\psi_O(A,t) = O^T\exp(tA)$ where exp is the matrix exponential. This is a one-parameter curve with t as the parameter. From this one can deduce that, in case of $\mathcal{S}_{n,d}$ and $\mathcal{G}_{n,d}$ a geodesic flow starting from a point $U = O^TJ \in \mathcal{S}_{n,d}$ is of the type

$$t \mapsto O^T\exp(tA)J. \tag{7}$$

Here, the skew-symmetric matrix A is either of the type

$$\begin{bmatrix} C & -B \\ B^T & 0 \end{bmatrix}$$

for $\mathcal{S}_{n,d}$, of the type

$$\begin{bmatrix} 0 & -B \\ B^T & 0 \end{bmatrix}$$

for $\mathcal{G}_{n,d}$. In general, the tangent vectors on $\mathcal{S}_{n,d}$ or $\mathcal{G}_{n,d}$ can be written as O^TAJ.

Tangent structure of $\mathcal{S}_{n,d}$. It can be shown that the tangent structure of $\mathcal{S}_{n,d}$ is given as

$$T_J(\mathcal{S}_{n,d}) = \left\{ \begin{bmatrix} C \\ B^T \end{bmatrix} | C \in \mathbb{R}^{d\times d} \text{ skew-symm}, B \in \mathbb{R}^{d\times(n-d)} \right\}. \tag{8}$$

For any other point $U \in \mathcal{S}_{n,d}$, let $O \in SO(n)$ be a matrix that rotates the columns of U to align with the columns of J, i.e., let $U = O^TJ$. Note that the choice of O is not unique. It follows that the tangent space at U is given by: $T_U(\mathcal{S}_{n,d}) = \{O^TG | G \in T_J(\mathcal{S}_{n,d})\}$.

Tangent structure of $\mathcal{G}_{n,d}$. The tangent space at $[J] \in \mathcal{G}_{n,d}$ is

$$T_{[J]}(\mathcal{G}_{n,d}) = \left\{ \begin{bmatrix} 0 \\ B^T \end{bmatrix} |\ B \in \mathbb{R}^{d\times(n-d)} \right\}. \tag{9}$$

For any other point $[U] \in \mathcal{G}_{n,d}$, let $O \in SO(n)$ be a matrix such that $U = O^TJ$. Then, the tangent space at $[U]$ is given by $T_{[U]}(\mathcal{G}_{n,d}) = \{O^TG | G \in T_{[J]}(\mathcal{G}_{n,d})\}$.

On $\mathcal{S}_{n,d}$ and $\mathcal{G}_{n,d}$, the exponential map is given by

$$O^T \begin{bmatrix} C \\ B^T \end{bmatrix} \equiv O^TAJ \mapsto O^T\exp(A)J,$$

where A takes an appropriate structure for each case. The expression for inverse exponential map is not available analytically for these manifolds and is computed numerically as described later in Section 6.

4 Using Geometry to Compute Sample Statistics on the Grassmann Manifold

The first question that we consider is: What is a suitable notion of a mean on the Riemannian manifold $\mathcal{M}$? A popular method for defining a mean on a manifold was proposed by Karcher [39] who used the centroid of a density as its mean.

Karcher mean [39]. The Karcher mean μ of a probability density function f on $\mathcal{M}$ is defined as a local minimizer of the cost function: $\rho : \mathcal{M} \to \mathbb{R}_{\geq 0}$, where

$$\rho(p) = \int_{\mathcal{M}} d(p,q)^2 f(q)\, dq, \tag{10}$$

dq denotes the reference measure used in defining the probability density f on $\mathcal{M}$. The value of the function ρ at the Karcher mean is called the **Karcher variance**. How does the definition of the Karcher mean adapt to a sample set, i.e., a finite set of points drawn from an underlying probability distribution? Let $q_1, q_2, \ldots, q_k$ be independent random samples from the density f. Then, the sample Karcher mean of these points is defined to be the local minimizer of the function: $\rho_k(p) = \frac{1}{k}\sum_{i=1}^{k} d(p, q_i)^2$.

An iterative algorithm is employed for computing the sample Karcher mean, as summarized in Algorithm 1. It can be shown that this algorithm converges to a local minimum of the cost function given in the definition of μ [40]. Depending upon the initial value μ_0 and the step size ϵ, the algorithm converges to a local minimum.

Algorithm 1. Algorithm for computing the sample Karcher mean

1. Given a set of k points $\{q_i\}$ on the manifold.
2. Let μ_0 be an initial estimate of the Karcher mean, usually obtained by picking one element of $\{q_i\}$ at random. Set $j = 0$.
3. For each $i = 1, \ldots, k$, compute the inverse exponential map v_i of q_i about the current estimate of the mean, i.e., $v_i = \exp_{\mu_j}^{-1}(q_i)$.
4. Compute the average tangent vector $\bar{v} = \frac{1}{k}\sum_{i=1}^{k} v_i$.
5. If $\|\bar{v}\|$ is small, then stop. Else, move μ_j in the average tangent direction using $\mu_{j+1} = \exp_{\mu_j}(\epsilon\bar{v})$, where $\epsilon > 0$ is small step size, typically 0.5.
6. Set $j = j + 1$ and return to Step 3. Continue till μ_j does not change anymore or till maximum iterations are exceeded.

5 Supervised and Unsupervised Learning Algorithms for the Grassmannian

Many of the image and video-based analysis tasks involve one of two tasks: 1) recognition of an input video as one of several classes or 2) finding underlying structural similarities in a large collection of videos. For example, given videos of activities, the ARMA model parameters $M = (A, C)$ are estimated using the methods described in Section 2. Subsequently, the finite observability matrix $O_m(M)$ is computed. Then for each observability matrix, an orthonormal basis is computed using standard SVD-based algorithms. So, we now have a set of subspaces or, in other words, a point cloud on the Grassmann manifold. In recognition problems, we also have corresponding class labels provided in the training set. In this section, we shall provide methods that follow from the theory described above to solve the supervised and unsupervised learning problems.

5.1 Learning with Parametric Class Conditional Densities

In addition to sample statistics such as the mean and covariance, it is possible to define probability density functions (pdfs) on manifolds for use in modeling random quantities. Similarly to the euclidean spaces, we have a choice between parametric and nonparametric probability models. While parametric models are typically more efficient, the nonparametric models often require fewer assumptions. For nonlinear manifolds, one can also have a choice between extrinsic and intrinsic probability models. The extrinsic models result from embedding nonlinear manifolds in higher dimensional euclidean spaces and defining models in those larger spaces. In contrast, the intrinsic models are completely restricted to the manifolds themselves and do not rely on any euclidean embedding. In view of the efficient nature of parametric models and the independence of intrinsic models from a need for euclidean embedding, we will pursue intrinsic parametric models. The general idea here is to define a pdf on the tangent space of the manifold, and then "wrap" the distribution back onto the manifold. This allows us to draw upon the wealth of methods available from classical multivariate statistics for the problem at hand.

Suppose we have n sample points, given by $q_1, q_2, \ldots q_n$, from a manifold $\mathcal{M}$. Then, we first compute their Karcher mean $\bar{q}$ as discussed before. The next step is to define and compute a sample covariance for the observed q_is. The key idea here is to use the fact that the tangent space $T_{\bar{q}}(\mathcal{M})$ is a vector space. For a d-dimensional manifold, the tangent space at a point is also d dimensional. Using a finite-dimensional approximation, say $V \subset T_{\bar{q}}(\mathcal{M})$, we can use classical multivariate statistics for this purpose. We can estimate the parameters of a family of pdfs, such as Gaussian or mixtures of Gaussian, and then use the exponential map to wrap these parameters back onto the manifold.

Truncation of domains. The exponential map $\exp_{\bar{q}} : T_{\bar{q}}(\mathcal{M}) \to \mathcal{M}$ proves useful to map estimated pdfs back to the manifold $\mathcal{M}$, giving rise to wrapped densities [40], [28]. In general, one can define arbitrary pdfs on the tangent space, such as mixtures of Gaussian, Laplace, etc., and wrap it back to the manifold via the exponential map. However, for manifolds of interest in this paper, the exponential map is a bijection only if its domain is restricted. Otherwise, any tangent line, being of infinite length, can be wrapped around these compact manifolds infinitely many times. Consequently, if one is interested in deriving an explicit expression for a wrapped density on $\mathcal{M}$, the resulting expression will have infinite sums and will complicate the derivations. Truncating the domain of density functions in the space $T_{\bar{q}}(\mathcal{M})$ such that $\exp_{\bar{q}}$ is a bijection is one solution. This would require truncation beyond a radius of π in $T_{\bar{q}}(\mathcal{M})$. The main modification required is that, for the multivariate density in $T_{\bar{q}}(\mathcal{M})$, the normalization constant changes. It gets scaled down depending on how much of the probability mass is left out of the truncation region. This can be evaluated empirically by drawing a large number of samples N from the estimated density and counting the number, N_π, of them that are within a radius of π from the origin in $T_{\bar{q}}(\mathcal{M})$. Then, the normalization constant needs to be multiplied by the effective fraction of samples within this radius, i.e., $N_{eff} = N_\pi/N$.

In experiments, we employ wrapped Gaussians in two ways which we denote as common-pole and class-specific pole wrapped Gaussians. In the common-pole case, given points on the manifold with class labels, we compute the mean of the entire data set without regard to class labels. This data set mean is referred to as the common pole. Then, class conditional densities are estimated in this tangent space. In

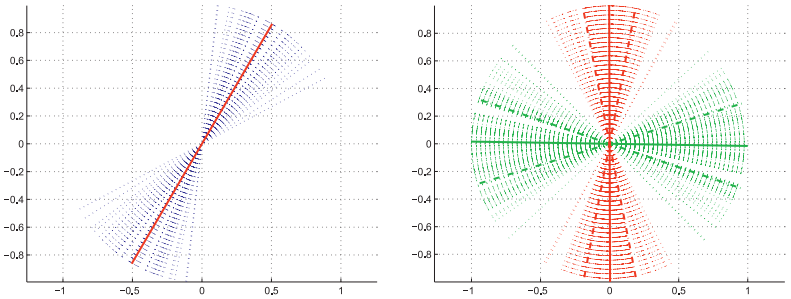


Fig. 3. In $\mathbb{R}^2$, the set of all axes (lines passing through the origin) is the Grassmann manifold with $n = 2$ and $d = 1$. (a) Karcher mean illustration: Blue dotted lines represent individual points on the Grassmann manifold. The bold red line is the Karcher mean of this set. The Karcher mean corresponds to the notion of a mean axis. (b) Illustration of wrapped Gaussian: Wrapped normal class conditional densities of two classes on the Grassmann manifold. Each class is shown in a different color. The mean of each class is shown in bold lines. The wrapped standard-deviation lines are shown in dashed lines for each class.

the class-specific pole case, we compute the Karcher mean for each class. Separate tangent spaces are considered for each class at the class-mean. The class conditionals are estimated in these individual tangent spaces. Algorithms for estimating class conditionals for both these cases are shown in Algorithms 2 and 3.

Algorithm 2. Truncated Wrapped Gaussian using common pole

1. Given a set of points with class labels $D = \{(U_i, l_i)\}$ on the manifold, and number of classes K.
2. Compute the Karcher mean μ of the entire data set without regards to class labels.
3. For each point U_i, compute the inverse exponential map about the data set mean $v_i = \exp_{\mu}^{-1}(U_i)$ and associate with the corresponding class label l_i, giving rise to a set of tuples $V = \{(v_i, l_i)\}$.
4. For each class fit a Gaussian distribution in the tangent space $T_{\mu}(\mathcal{M})$.
5. For each class, sample a large number N of points from the estimated Gaussian distribution.
6. Count the number of points N_{π} which lie within a distance π from the origin of $T_{\mu}(\mathcal{M})$ (origin here corresponds to $exp_{\mu}^{-1}(\mu)$). Compute multiplication factor $N_{eff} = N_{\pi}/N$ and adjust the normalization factor.

Algorithm 3. Truncated Wrapped Gaussian using class-specific pole

1. Given a set of points with class labels $D = \{(U_i, l_i)\}$ on the manifold, and number of classes K.

for $i = 1, \ldots K$ **do**

Compute the Karcher mean μ_i of the ith class using algorithm 1.

For all points $\{U_j\}$ of the current class, compute the inverse exponential map about the class mean $v_j = \exp_{\mu_i}^{-1}(U_j)$.

Fit a Gaussian distribution for the ith class in the tangent space $T_{\mu_i}(\mathcal{M})$.

Sample a large number N of points from the estimated Gaussian distribution.

Count the number of points N_{π} which lie within a distance π from the origin of $T_{\mu_i}(\mathcal{M})$ (origin here corresponds to $exp_{\mu_i}^{-1}(\mu_i)$). Compute multiplication factor $N_{eff} = N_{\pi}/N$ and adjust the normalization factor for the ith class conditional density.

end for

5.1.1 Synthetic Examples

In this section, we illustrate the concepts of sample Karcher mean and wrapped densities on a Grassmann manifold. To help visualization, we choose $\mathcal{G}_{n,d}$ with $n = 2$ and $d = 1$, i.e., one-dimensional subspaces of $\mathbb{R}^2$. This is the set of all lines passing through of the origin on the X-Y plane. Lines on a plane can be parametrized by their principal angle with the X-axis. Using this parameterization, in the first experiment we randomly sample directions centered around $\theta = \pi/3$ with variance in θ set to 0.2. A set of such samples in shown in Fig. 3a with dotted blue lines. The Karcher mean of this set is shown as a red line in Fig. 3a. As can be seen, the Karcher mean corresponds well to the notion of a "mean-axis" in this case. In Fig. 3b, we illustrate the concept of estimating the wrapped normal distribution. In this experiment, we generated samples from two classes—one centered at $\theta = 0$ and the other centered at $\theta = \pi/2$. Points from each class are shown in different colors. The Karcher mean of the whole data set was taken as the pole to compute the tangent vectors for the points. Each of the classes was parameterized by a mean μ and standard deviation σ on the tangent plane. The points corresponding to μ and and $\mu \pm \sigma$ were then wrapped back onto the manifold. The mean and standard-deviation axes for each of the classes are shown as bold and dashed lines, respectively.

An earlier paper [1] used extrinsic nonparametric models for similar purposes and in this paper we will compare them with our current approach. Recall that the Karcher mean computation is an iterative procedure. In recent years, the Procrustes methods proposed by [23] have become popular for noniterative density estimation as an alternative. However, it requires a choice of parameters (kernel-width) whose optimal value is not known in advance. Given several examples from a class $(U_1, U_2, \ldots, U_n)$ on the Grassmann manifold, the class conditional density is given by [23] as

$$\hat{f}(U; M) = \frac{1}{n} C(M) \sum_{i=1}^{n} K[M^{-1/2}(I_k - U_i^T U U^T U_i) M^{-1/2}], \tag{11}$$

where $K(T)$ is the kernel function, M is a $d \times d$ positive definite matrix which plays the role of the kernel width or a smoothing parameter. $C(M)$ is a normalizing factor chosen so that the estimated density integrates to unity. The matrix valued kernel function $K(T)$ can be chosen in several ways. We have used $K(T) = exp(-tr(T))$ in all of the experiments reported in this paper.

5.2 Unsupervised Clustering

The statistical tools that have been described in the previous sections can be used for unsupervised learning tasks such as clustering of data. Using them, it is possible to estimate clusters in an intrinsic manner. Let us assume that we have a set of points $D = (U_1, U_2, \ldots, U_n)$ on the Grassmann manifold. We seek to estimate k clusters $\mathbb{C} = (C_1, C_2, \ldots, C_k)$ with cluster centers $(\mu_1, \mu_2, \ldots, \mu_k)$ so that the sum of geodesic-distance squares,

$$\sum_{i=1}^{k} \sum_{U_j \in C_i} d^2(U_j, \mu_i),$$

is minimized. Here $d^2(U_j, \mu_i) = |\exp_{\mu_i}^{-1}(U_j)|^2$. As is the case with standard k-means, we can solve this problem using an EM-based approach. We initialize the algorithm with a random selection of k points as the cluster centers. In the E-step, we assign each of the points of the data set D to the nearest cluster center. Then in the M-step, we recompute the cluster centers using the Karcher mean computation algorithm described in Section 4. The procedure is summarized in Algorithm 4.

Algorithm 4. Intrinsic K-means clustering algorithm on Riemannian manifolds

1. Given set of points $D = (U_1, U_2, \ldots, U_n)$ on the Grassmann manifold, number of clusters K, maximum iteration N_{max}.
2. Initialize cluster centers $(\mu_1^{(0)}, \mu_2^{(0)}, \ldots, \mu_k^{(0)})$ randomly.
   while $(i \leq N_{max})$ **do**
   Assign each point to nearest cluster center by
   computing $d^2(U_j, \mu_k) = \left|\exp_{\mu_k}^{-1}(U_j)\right|^2$.
   Recompute cluster centers $(\mu_1^{(i)}, \mu_2^{(i)}, \ldots, \mu_k^{(i)})$ using algorithm 1.
   $i = i + 1$

end while

6 Size of Problems and Methods for Efficient Riemannian Computations

As described in Section 2, the finite observability matrix is given by $O_m^T = [C^T, (CA)^T, (CA^2)^T, \ldots (CA^{m-1})^T]$. The size of this matrix is $mp \times d$. The column space of this matrix is a d-dimensional subspace of $\mathbb{R}^{mp}$. d is typically of the order of 5-10, and we choose m to be the same as d. However, p is the dimension of the feature vectors, and this in general can be quite large. Typical image sequences used for, say, video-based face recognition result in images of size 100×100 resulting in $p = 10^4$. Similarly, in the case of modeling image sets, the PCA basis vectors are stored as $p \times d$ matrices, where p is the size of raw images and d is the subspace dimension (typically small). Due to the large size of these matrices, straightforward implementation of Riemannian computations is nontrivial. The computation of the geodesic $O^T exp(tA)J$ in the direct form implies a complexity of $O(n^3)$, where $n = mp$ for the observability matrix and $n = p$ for the case of PCA basis vectors. By exploiting the special structure of the matrix A, it is possible to reduce the complexity of these operations to no more than $O(nd^2)$ and $O(d^3)$, which represents a significant reduction. These efficient methods were first proposed by Gallivan et al. [41]. For a self-contained treatment, here we summarize the key results that will be used in this paper in the Appendix, which can be found in the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2011.52.

7 Applications and Experiments

In this section, we show the utility of the methods discussed so far on several image and video-based recognition tasks. We shall show four different applications:

1. Activity recognition on INRIA iXMAS data [42].
2. Video-based Face recognition on NIST-MBGC data [43].
3. Face Recognition from Image Sets on CMU-PIE data [44].
4. Video Clustering on SFU figure skating data [45].

In all of these applications, we show that subspace matching arises naturally. We compare with other related methods that involve subspace matching, and show that statistical modeling of class conditionals using Riemannian geometry demonstrates better performance over other simpler methods.

Implementation details. For parametric class conditional densities as described in Section 5.1, we consider two versions of wrapped Gaussians—common-pole and class-specific poles. In the common-pole case, the tangent space is constructed at the Karcher mean of the entire training data set (Algorithm 2). In the class-specific pole case, we construct a class-specific tangent space at the Karcher mean of each of the classes (Algorithm 3). The class conditional for the ith class is completely specified by the tuple $C_i = \{p_i, \bar{v}_i, \Sigma_i\}$, where p_i is the pole about whose tangent space the density is defined, $\bar{v}_i$ is the mean in $T_{p_i}(\mathcal{M})$, and Σ_i the covariance matrix in $T_{p_i}(\mathcal{M})$. In the common-pole case, all p_is are set to the data set mean. In the class-specific pole case, the p_is are set to individual class means. To evaluate the ith class conditional density at a test point, one merely evaluates the truncated Gaussian by mapping the test point to the tangent space at p_i. Then, the point is classified into the class that has the highest likelihood. In our experiments, we have restricted Σ_i to be a diagonal matrix instead of a full covariance matrix. As mentioned in Section 5.1, to evaluate the class conditional probability using truncated wrapped Gaussians we also need to adjust the normalizing constant of each Gaussian. It is our experience that the appearance/activity models on Stiefel and Grassmann manifolds are rather clustered around their class mean and rarely are some points so far away from the mean to necessitate truncation. So, we ignore this minor adjustment.

TABLE 1
Comparison of View-Invariant Recognition of Activities in the INRIA Data Set Using 1) Best Dim. Red. [42] on $16 \times 16 \times 16$ Features, 2) Best Dim. Red. [42] on $64 \times 64 \times 64$ Features, 3) Nearest Neighbor Using ARMA Model Distance, 4) Procrustes Distance (Reported in [1])

Activity	**Dim. Red.** [42] 16^3 volume	**Best Dim. Red.** [42] 64^3 volume	**Subspace Angles** 16^3 volume	**NN-Procrust** 16^3 volume [1]
Check Watch	76.67	86.66	93.33	90
Cross Arms	100	100	100	96.67
Scratch Head	80	93.33	76.67	90
Sit Down	96.67	93.33	93.33	93.33
Get Up	93.33	93.33	86.67	80
Turn Around	96.67	96.67	100	100
Walk	100	100	100	100
Wave Hand	73.33	80	93.33	90
Punch	83.33	96.66	93.33	83.33
Kick	90	96.66	100	100
Pick Up	86.67	90	96.67	96.67
Average	88.78	93.33	93.93	92.72

7.1 Activity Recognition

We performed a recognition experiment on the publicly available INRIA data set [42]. The data set consists of 10 actors performing 11 actions, each action executed three times at varying rates while freely changing orientation. We used the view-invariant representation and features as proposed in [42]. Specifically, we used the $16 \times 16 \times 16$ circular FFT features proposed by [42]. Instead of modeling each segment of activity as a single motion history volume as in [42], we build a time series of motion history volumes using small sliding windows. This allows us to build a dynamic model for each segment. We use the segmentation results used in [42]. Using these features, we first performed a recognition experiment on the provided data.

To perform recognition, first each activity was modeled as an ARMA model given in (2). The state-space dimension d was chosen to be five. Model fitting was performed as described in Section 2. After this, the finite observability matrix $O_m(M)$ is computed, and an orthonormal basis corresponding to its column space is stored. Testing was performed using a round-robin (leave-one-person-out) experiment where activity models were learned using nine actors and tested on one actor. For fitting the ARMA model we used $16 \times 16 \times 16 = 4{,}096$ dimensional features, chose state-space dimension $d = 5$, and truncated the observability matrix at $m = d = 5$. Thus, in this case, the Grassmann manifold $\mathcal{G}_{n,d}$ corresponds to $n = mp = 20{,}480$ and $d = 5$.

In Table 1, we show the recognition results obtained using four baseline methods that do not require any statistical modeling. The first column shows the results obtained using dimensionality reduction approaches of [42] on $16 \times 16 \times 16$ features. Reference [42] reports recognition results using a variety of dimensionality reduction techniques (PCA, LDA, Mahalanobis) and here we choose the row-wise best performance from their experiments (denoted "Best Dim. Red.") which were obtained using $64 \times 64 \times 64$ circular FFT features. The third column corresponds to the method of using subspace angles-based distance between dynamical models [31]. This is based on computing the angles between subspaces θ_i and measuring the distance using $\sum sin^2(\theta_i)$. Column 4 shows the nearest-neighbor classifier performance using Procrustes distance measure ($16 \times 16 \times 16$ features). We see that the manifold Procrustes distance performs as well as ARMA model distance [31].

In Table 2, we show results of statistical modeling using parametric and nonparametric methods. As can be seen in the results in Table 2, statistical modeling of class conditional densities leads to a significant improvement in recognition performance over simpler methods shown in Table 1. We also present the results of nonparametric kernel density estimator reported in [1]. Note that even though the manifold approaches presented here use only $16 \times 16 \times 16$ features they outperform other approaches that use higher resolution ($64 \times 64 \times 64$ features) as shown in Table 1.

As mentioned before, for the nonparametric case, an appropriate choice of the kernel width M has to be made. In general, cross validation is suggested to estimate the optimal kernel width. Different classes may have a different optimal kernel width. Hence, cross validation requires a lengthy training phase. A suboptimal choice can often lead to poor performance. This is one of the significant drawbacks of nonparametric methods. However, addressing this formally is beyond the scope of the current paper.

7.2 Video-Based Face Recognition

Video-based face recognition by modeling the "cropped video" either as dynamical models ([6]) or as a collection of PCA subspaces [46] have recently gained popularity because of their ability to recognize faces from low-resolution videos. Given a video, we estimate the a low-dimensional subspace from the sequence of frames using standard PCA. The subspace is then considered as a point on the Grassmann manifold.

We performed a recognition experiment on NIST's Multiple Biometric Grand Challenge (MBGC) data set.

TABLE 2
Statistical Modeling for Recognition of Activities in the INRIA Data Set Using 1) Common-Pole Wrapped Normal, 2) Class-Specific Pole Wrapped Normal, 3) Kernel Density (First Reported in [1])

Activity	Wrapped Normal: Common-Pole (Algorithm 2)	Wrapped Normal: Class-specific Pole (Algorithm 3)	Procrustes Kernel $M = I$ [1]
Check Watch	96.67	100	100
Cross Arms	93.33	100	100
Scratch Head	93.33	90	96.67
Sit Down	90	96.67	93.33
Get Up	100	96.67	96.67
Turn Around	96.67	100	100
Walk	93.33	90	100
Wave Hand	86.67	93.33	100
Punch	90	100	100
Kick	93.33	100	100
Pick Up	93.33	100	100
Average	93.33	96.06	98.78

TABLE 3
Comparison Recognition Accuracies of Video-Based Face Recognition Using Subspace-Based Approaches 1) Subspace Angles + Arc-Length Metric, 2) Procrustes Distance, 3) Kernel Density, 4) Wrapped Normal Using a Common Pole for All Classes (Algorithm 2)

Subset	**Distinct Subjects**	**Total Sequences**	**Arc-length Metric**	**Procrustes Metric**	**Kernel density**	**Wrapped Gaussian Common Pole**
S_2	143	395	38.48	43.79	39.74	**63.79**
S_3	55	219	48.85	53.88	50.22	**74.88**
S_4	54	216	48.61	53.70	50.46	**75**
Avg.			45.31%	50.45%	46.80%	**71.22%**

TABLE 4
CMU-PIE Database: Face Identification Using Various Grassmann Statistical Methods

Subspace Dimension	m=2	m=3	m=4	m=5	m=6	m=7	m=8	m=9
GDA (Proj) [14]	74.8	**89.8**	87.2	91.7	92.5	93.8	93.6	95.3
GDA (BC) [14]	71.4	82.5	64.8	58.6	47.5	43.1	39.9	36.3
MSM [48]	67.0	65.0	64.6	64.2	64.0	64.6	64.6	64.6
cMSM [49]	71.2	67.6	68.2	69.7	69.9	70.2	72.7	72.5
DCC [34]	**78.9**	66.5	63.8	64.6	67.6	67.6	67.6	65
Wrapped Normal: Algorithm 2	69.95	76.89	69.74	77.73	79.83	79.20	80.46	76.26
Wrapped Normal: Algorithm 3	69.95	76.89	70.16	77.31	82.56	84.66	85.50	86.97
Grassmann Kernel Density: $M = I$	78.36	88.44	**89.91**	**93.69**	**95.79**	**97.26**	**96.84**	**97.26**

Performance of various methods is compared as the subspace dimension is varied.

The MBGC Video Challenge data set consists of a large number of subjects walking toward a camera in a variety of illumination conditions. Face regions are manually tracked and a sequence of cropped images is obtained. There were a total of 143 subjects with the number of videos per subject ranging from 1 to 5. In our experiments, we took subsets of the data set which contained at least two sequences per person denoted as S_2, at least three sequences per person denoted as S_3, etc. Each of the face images was first preprocessed to zero-mean and unity variance and scaled to

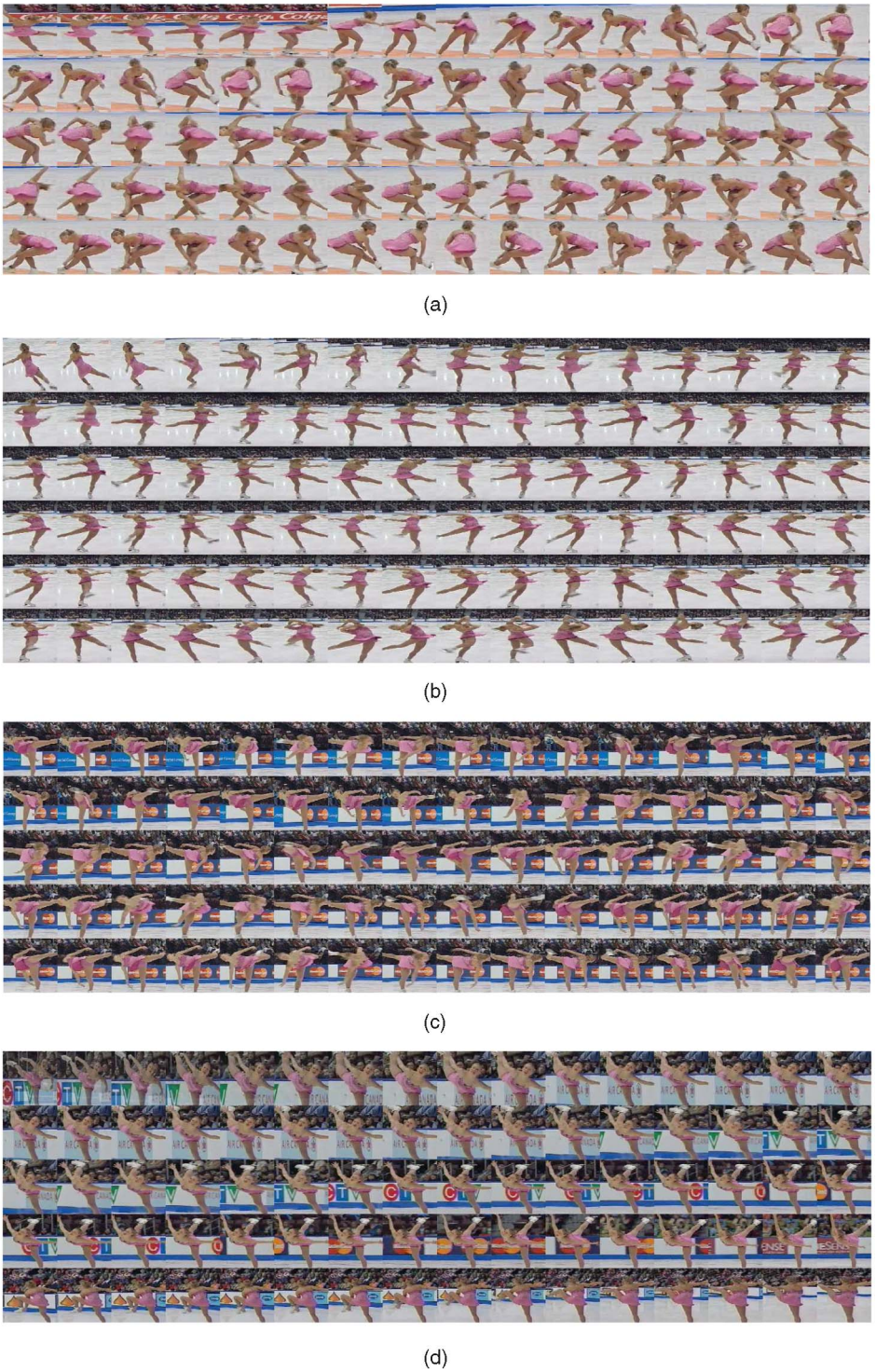


Fig. 4. Shown here are a few sequences from each obtained cluster. Each row in a cluster shows contiguous frames of a sequence. (a) Cluster 1: Sit-spins. (b) Cluster 2: Stand-spins. (c) Cluster 3: Camel-spins. (d) Cluster 4: Spirals.

100×100. For each subject, a PCA basis is estimated of dimension $d = 5$. Thus, in this case $\mathcal{G}_{n,d}$ corresponds to $n = 10{,}000$, $d = 5$. In each of these subsets, we performed a leave-one-out testing. The results of the leave-one-out testing are shown in Table 3. In the comparisons, we show results using the "arc-length" metric between subspaces [10]. This metric computes the subspace angles between two subspaces and takes the L-2 norm of the angles as a distance measure [10]. We also show comparisons with the Procrustes measure, the Kernel density estimate with $M = I$, and a wrapped normal density with the Karcher mean of the entire data set as the pole given in Algorithm 2.

As can be seen, statistical methods outperform nearest-neighbor-based approaches. As one would expect, the results improve when more examples per class are available. Since the optimal kernel-width is not known in advance, this might explain the relatively poor performance of the kernel density method.

7.3 Face Recognition from Image Sets

We consider the CMU-PIE face data set which contains images of 68 persons under varying poses, illumination, and expressions. For comparison, we use the methods proposed in [14]. The methods proposed in [14] involve discriminative approaches on the Grassmann manifold using Mercer-kernels. In this approach, a Mercer-kernel is defined on the Grassmann manifold which then enables using kernel versions of SVMs, Fisher Discriminant Analysis, etc., for classification. In this experiment, we use the experimental protocol suggested in [47]. For each of the 68 subjects, seven near frontal poses are used in the experiment. For each person under a fixed pose, we approximate the variations due to expressions and illumination as a linear subspace. Thus, for each person we have a set of subspaces corresponding to each pose. This allows us to build a statistical model on the Grassmann manifold for each person. A round-robin (leave-one-pose-out) experiment is performed in which six poses are used for training and the remaining pose is used for testing. The results are shown in Table 4. The results using the other methods were reported in [47].

As can be seen, the proposed statistical approaches compare well with the state-of-the-art. In particular, the kernel density method outperforms all of the other methods. The discriminative approaches of [14] outperforms the wrapped normal approach. However, the variability of the performance is high depending on what Mercer kernel is chosen. The wrapped normal provides consistent performance and beats most other methods.

7.4 Video Clustering

We performed a clustering experiment on the figure skating data set of [45]. These videos are unconstrained and involve rapid motion of both the skater and the camera. As reported in [50], color models of the foreground and background are used to segment the background and foreground pixels. Median filtering followed by connected component analysis is performed to reject small isolated blobs. From the segmented results, we fit a bounding box to the foreground pixels by estimating the 2D mean and second order moments along x and y directions. We perform temporal smoothing of the bounding box parameters to remove jitter effects. The final feature is a rescaled binary image of size 100×100 of the pixels inside the bounding box. We build ARMA models for fixed length subsequences using sliding windows as done in [50]. State-space dimension $d = 5$, and the observability matrix is truncated at $m = 5$. Thus, we have $\mathcal{G}_{n,d}$ with $n = mp = 50{,}000$, $d = 5$. Then, we used the intrinsic K-means clustering on the Grassmann manifold using Algorithm 4. In [50], the segments were treated as nodes in a graph and normalized cuts (N-cuts) was used for clustering. The cited reason was that the space of ARMA models is not a vector space and it is not apparent how to perform k-means clustering and thereby N-cuts is used as an alternative. The approach that we use here, while achieving similar results, is a principled method to solve the video-clustering problem using ARMA models. As is the case with standard k-means, it enjoys lower computational load compared to the spectral clustering algorithms, especially for long videos. We show some sample sequences in the obtained clusters in Fig. 4. We observe that the clusters predominantly correspond to "Sitting Spins," "Standing Spins," "Camel Spins," and "Spirals." There is a fifth cluster which corresponds mainly to "Glides" and has been omitted due to space constraints.

8 Conclusion

We have shown that the Grassmann manifold arises naturally in many image and video-based classification problems. We have presented statistical modeling methods that are derived from the Riemannian geometry of the manifold. We have shown the utility of the methods on several applications such as activity recognition, video-based face recognition, and recognition from image sets. In addition to definitions of distances and statistics on manifolds, many interesting problems such as interpolation, smoothing, and time-series modeling on these manifolds of interest are potential directions of future work. These techniques can prove useful in applications such as adapting appearance models for active vision applications, or modeling time-varying dynamic models for human activities [32].

Acknowledgments

A preliminary version of this paper appeared in [1]. This work was partially supported by US Office of Naval Research Grant N00014-09-1-0664.

References

[1] P. Turaga, A. Veeraraghavan, and R. Chellappa, "Statistical Analysis on Stiefel and Grassmann Manifolds with Applications in Computer Vision," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1-8, June 2008.

[2] G. Doretto, A. Chiuso, Y.N. Wu, and S. Soatto, "Dynamic Textures," *Int'l J. Computer Vision,* vol. 51, no. 2, pp. 91-109, Feb. 2003.

[3] A.B. Chan and N. Vasconcelos, "Modeling, Clustering, and Segmenting Video with Mixtures of Dynamic Textures," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 5, pp. 909-926, May 2008.

[4] A. Bissacco, A. Chiuso, Y. Ma, and S. Soatto, "Recognition of Human Gaits," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 52-57, Dec. 2001.

[5] A. Veeraraghavan, A. Roy-Chowdhury, and R. Chellappa, "Matching Shape Sequences in Video with an Application to Human Movement Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 12, pp. 1896-1909, Dec. 2005.

[6] G. Aggarwal, A. Roy-Chowdhury, and R. Chellappa, "A System Identification Approach for Video-Based Face Recognition," *Proc. Int'l Conf. Pattern Recognition,* Aug. 2004.

[7] C.R. Goodall and K.V. Mardia, "Projective Shape Analysis," *J. Computational and Graphical Statistics,* vol. 8, no. 2, pp. 143-168, June 1999.

[8] V. Patrangenaru and K.V. Mardia, "Affine Shape Analysis and Image Analysis," *Proc. 22nd Leeds Ann. Statistics Research Workshop,* July 2003.

[9] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman, "From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 6, pp. 643-660, June 2001.

[10] A. Edelman, T.A. Arias, and S.T. Smith, "The Geometry of Algorithms with Orthogonality Constraints," *SIAM J. Matrix Analysis and Application,* vol. 20, no. 2, pp. 303-353, Apr. 1999.

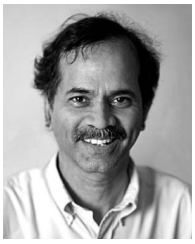
[11] P.-A. Absil, R. Mahony, and R. Sepulchre, "Riemannian Geometry of Grassmann Manifolds with a View on Algorithmic Computation," *Acta Applicandae Mathematicae,* vol. 80, no. 2, pp. 199-220, Jan. 2004.

[12] D. Lin, S. Yan, and X. Tang, "Pursuing Informative Projection on Grassmann Manifold," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 1727-1734, June 2006.
[13] E. Begelfor and M. Werman, "Affine Invariance Revisited," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 2087-2094, June 2006.
[14] J. Hamm and D.D. Lee, "Grassmann Discriminant Analysis: A Unifying View on Subspace-Based Learning," *Proc. Int'l Conf. Machine Learning,* pp. 376-383, June 2008.
[15] Y.M. Lui and J.R. Beveridge, "Grassmann Registration Manifolds for Face Recognition," *Proc. European Conf. Computer Vision,* pp. 44-57, Oct. 2008.
[16] A. Srivastava and E. Klassen, "Bayesian and Geometric Subspace Tracking," *Advances in Applied Probability,* vol. 36, no. 1, pp. 43-56, Mar. 2004.
[17] Y.M. Lui, J.R. Beveridge, and M. Kirby, "Canonical Stiefel Quotient and Its Application to Generic Face Recognition in Illumination Spaces," *Proc. IEEE Third Int'l Conf. Biometrics: Theory, Applications, and Systems,* Aug. 2009.
[18] W.M. Boothby, *An Introduction to Differentiable Manifolds and Riemannian Geometry.* Academic Press, 1975.
[19] R. Bhattacharya and V. Patrangenaru, "Large Sample Theory of Intrinsic and Extrinsic Sample Means on Manifolds-I," *Annals of Statistics,* vol. 31, no. 1, pp. 1-29, 2003.
[20] B. Pelletier, "Kernel Density Estimation on Riemannian Manifolds," *Statistics and Probability Letters,* vol. 73, no. 3, pp. 297-304, July 2005.
[21] X. Pennec, "Intrinsic Statistics on Riemannian Manifolds: Basic Tools for Geometric Measurements," *J. Math. Imaging and Vision,* vol. 25, no. 1, pp. 127-154, July 2006.
[22] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds.* Princeton Univ. Press, 2008.
[23] Y. Chikuse, *Statistics on Special Manifolds: Lecture Notes in Statistics.* Springer, 2003.
[24] O. Tuzel, F. Porikli, and P. Meer, "Region Covariance: A Fast Descriptor for Detection and Classification," *Proc. European Conf. Computer Vision,* pp. 589-600, May 2006.
[25] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian Detection via Classification on Riemannian Manifolds," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 10, pp. 1713-1727, Oct. 2008.
[26] F. Porikli, O. Tuzel, and P. Meer, "Covariance Tracking Using Model Update Based on Lie Algebra," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 728-735, June 2006.
[27] R. Subbarao and P. Meer, "Nonlinear Mean Shift for Clustering over Analytic Manifolds," *Int'l J. Computer Vision,* vol. 84, no. 1, pp. 1-20, Aug. 2009.
[28] A. Srivastava, S.H. Joshi, W. Mio, and X. Liu, "Statistical Shape Analysis: Clustering, Learning, and Testing," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 4, pp. 590-602, Apr. 2005.
[29] A. Veeraraghavan, A. Srivastava, A.K. Roy Chowdhury, and R. Chellappa, "Rate-Invariant Recognition of Humans and Their Activities," *IEEE Trans. Image Processing,* vol. 18, no. 6, pp. 1326-1339, June 2009.
[30] P.V. Overschee and B.D. Moor, "Subspace Algorithms for the Stochastic Identification Problem," *Automatica,* vol. 29, no. 3, pp. 649-660, May 1993.
[31] K.D. Cock and B.D. Moor, "Subspace Angles between ARMA Models," *Systems and Control Letters,* vol. 46, pp. 265-270, July 2002.
[32] P. Turaga and R. Chellappa, "Locally Time-Invariant Models of Human Activities Using Trajectories on the Grassmannian," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 2435-2441, June 2009.
[33] P. Saisan, G. Doretto, Y.N. Wu, and S. Soatto, "Dynamic Texture Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 58-63, Dec. 2001.
[34] T.K. Kim, J. Kittler, and R. Cipolla, "Discriminative Learning and Recognition of Image Set Classes Using Canonical Correlations," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 6, pp. 1005-1018, June 2007.
[35] O. Arandjelovic, G. Shakhnarovich, J. Fisher, R. Cipolla, and T. Darrell, "Face Recognition with Image Sets Using Manifold Density Divergence," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 581-588, June 2005.
[36] S.K. Zhou and R. Chellappa, "From Sample Similarity to Ensemble Similarity: Probabilistic Distance Measures in Reproducing Kernel Hilbert Space," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 6, pp. 917-929, June 2006.
[37] R. Basri and D.W. Jacobs, "Lambertian Reflectance and Linear Subspaces," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 2, pp. 218-233, Feb. 2003.
[38] K.-C. Lee, J. Ho, and D.J. Kriegman, "Acquiring Linear Subspaces for Face Recognition under Variable Lighting," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 5, pp. 684-698, May 2005.
[39] H. Karcher, "Riemannian Center of Mass and Mollifier Smoothing," *Comm. Pure and Applied Math.,* vol. 30, pp. 509-541, 1977.
[40] X. Pennec, "Statistical Computing on Manifolds: From Riemannian Geometry to Computational Anatomy," *Proc. Emerging Trends in Visual Computing,* pp. 347-386, 2008.
[41] K. Gallivan, A. Srivastava, X. Liu, and P. VanDooren, "Efficient Algorithms for Inferences on Grassmann Manifolds," *Proc. IEEE 12th Workshop Statistical Signal Processing,* Oct. 2003.
[42] D. Weinland, R. Ronfard, and E. Boyer, "Free Viewpoint Action Recognition Using Motion History Volumes," *Computer Vision and Image Understanding,* vol. 104, no. 2, pp. 249-257, Nov. 2006.
[43] "NIST Multiple Biometric Grand Challenge," http://face.nist.gov/mbgc/, 2011.
[44] T. Sim, S. Baker, and M. Bsat, "The CMU Pose, Illumination, and Expression Database," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 12, pp. 1615-1618, Dec. 2003.
[45] Y. Wang, H. Jiang, M.S. Drew, Z.N. Li, and G. Mori, "Unsupervised Discovery of Action Classes," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1654-1661, 2006.
[46] K.C. Lee, J. Ho, M.H. Yang, and D. Kriegman, "Video-Based Face Recognition Using Probabilistic Appearance Manifolds," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 313-320, June 2003.
[47] J. Hamm, "Subspace-Based Learning with Grassmann Kernels," PhD thesis, Univ. of Pennsylvania, 2008.
[48] K. Fukui and O. Yamaguchi, "Face Recognition Using Multi-Viewpoint Patterns for Robot Vision," *Proc. Int'l Symp. Robotics Research,* pp. 192-201, 2003.
[49] O. Yamaguchi, K. Fukui, and K. Maeda, "Face Recognition Using Temporal Image Sequence," *Proc. Third Int'l Conf. Face and Gesture Recognition,* pp. 318-323, Apr. 1998.
[50] P. Turaga, A. Veeraraghavan, and R. Chellappa, "Unsupervised View and Rate Invariant Clustering of Video Sequences," *Computer Vision and Image Understanding,* vol. 113, no. 3, pp. 353-371, Mar. 2009.

Pavan Turaga received the BTech degree in electronics and communication engineering from the Indian Institute of Technology, Guwahati, India, in 2004 and the MS and PhD degrees in electrical engineering from the University of Maryland, College Park, in 2008 and 2009, respectively. Currently, he is a research associate at the Center for Automation Research, University of Maryland, College Park. His published works span human activity analysis from videos, video summarization, dynamic scene analysis, and statistical inference on manifolds for these applications. He was awarded the Distinguished Dissertation Fellowship in 2009. He was selected to participate in the Emerging Leaders in Multimedia Workshop by IBM, New York, in 2008. His research interests include computer vision, pattern analysis, and multimedia content analysis. He is a member of the IEEE.

Ashok Veeraraghavan received the bachelor's degree in electrical engineering from the Indian Institute of Technology, Madras in 2002 and the MS and PhD degrees from the Department of Electrical and Computer Engineering at the University of Maryland, College Park, in 2004 and 2008, respectively. Currently, he is a research scientist at Mitsubishi Electric Research Labs in Cambridge, Massachusetts. His thesis received the Doctoral Dissertation award from the Department of Electrical and Computer Engineering at the University of Maryland. His research interests are broadly in the areas of computational imaging, computer vision, and robotics. He is a member of the IEEE.

Anuj Srivastava received the MS and PhD degrees in electrical engineering from Washington University in St. Louis in 1993 and 1996, respectively. After spending the year 1996-1997 at Brown University as a visiting researcher, he joined Florida State University (FSU) as an assistant professor in 1997where he is currently a professor of statistics. He has received the Developing Scholar and the Graduate Faculty Mentor Awards at FSU. He has held visiting professor appointments at INRIA, Sophia Antipolis, France, and the University of Lille, Lille, France. His research has been supported by grants from the US National Science Foundation, the US Army Research Office, the US Office of Naval Research, the US Air Force Office of Scientific Research, and Northrop-Grumman Company. He has developed computational tools for performing statistical inferences on certain nonlinear manifolds, in particular the shape spaces of curves and surfaces. He has published more than 120 journal and conference articles in these areas. His research interests include pattern theoretic approaches to problems in image analysis, computer vision, and signal processing. He is a senior member of the IEEE.



Rama Chellappa received the BE (Hons.) degree from the University of Madras, India, in 1975 and the ME (Distinction) degree from the Indian Institute of Science, Bengaluru, in 1977. He received the MSEE and PhD degrees in electrical engineering from Purdue University, West Lafayette, Indiana, 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. He is also affiliated with the Center for Automation Research (director) and the Institute for Advanced Computer Studies (permanent member). In 2005, he was named the Minta Martin Professor of Engineering. Prior to joining the University of Maryland, he was an assistant (1981-1986) and associate professor (1986-1991) and director of the Signal and Image Processing Institute (1988-1990) at the University of Southern California (USC), Los Angeles. Over the last 29 years, he has published numerous book chapters, peer-reviewed journal, and conference papers. He has coauthored and coedited books on MRFs, face and gait recognition, and collected works on image processing and analysis. He has received several awards, including a US National Science Foundation (NSF) Presidential Young Investigator Award, four IBM Faculty Development Awards, an Excellence in Teaching Award from the School of Engineering at USC, and two paper awards from the International Association of Pattern Recognition. He received the Society, Technical Achievement, and Meritorious Service Awards from the IEEE Signal Processing Society. He also received the Technical Achievement and Meritorious Service Awards from the IEEE Computer Society. At the University of Maryland, he was elected a distinguished faculty research fellow, a distinguished scholar-teacher, received the Outstanding Faculty Research Award from the College of Engineering, an Outstanding Innovator Award from the Office of Technology Commercialization, and an Outstanding GEMSTONE Mentor Award. In 2010, he was recognized as an Outstanding ECE by Purdue University. He served as an associate editor of four IEEE transactions, as a co-editor-in-chief of *Graphical Models and Image Processing*, and 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 and as its vice president of Awards and Membership. He has served as a general and technical program chair for several IEEE international and national conferences and workshops. He is a Golden Core Member of the IEEE Computer Society and served a two-year term as a distinguished lecturer of the IEEE Signal Processing Society. He is serving a two-year term as the president of the IEEE Biometrics Council. His current research interests include face and gait analysis, markerless motion capture, 3D modeling from video, image and video-based recognition and exploitation, compressive sensing, and hyper spectral processing. He is a fellow of the IEEE and the International Association for Pattern Recognition and the Optical Society of America.