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

Learning on Manifolds

Fatih Porikli

TR2010-079 October 2010

Abstract

Mathematical formulation of certain natural phenomena exhibits group structure on topological spaces that resemble the Euclidean space only on a small enough scale, which prevents incorporation of conventional inference methods that require global vector norms. More specifically in computer vision, such underlying notions emerge in differentiable parameter spaces. Here, two Riemannian manifolds including the set of affine transformations and covariance matrices are elaborated and their favorable applications in distance computation, motion estimation, object detection and recognition problems are demonstrated after reviewing some of the fundamental preliminaries.

IAPRS + SSPR Conference

Learning on Manifolds*

Fatih Porikli

Mitsubishi Electric Research Labs,
Cambridge, MA, USA

Abstract. Mathematical formulation of certain natural phenomena exhibits group structure on topological spaces that resemble the Euclidean space only on a small enough scale, which prevents incorporation of conventional inference methods that require global vector norms. More specifically in computer vision, such underlying notions emerge in differentiable parameter spaces. Here, two Riemannian manifolds including the set of affine transformations and covariance matrices are elaborated and their favorable applications in distance computation, motion estimation, object detection and recognition problems are demonstrated after reviewing some of the fundamental preliminaries.

Key words: Region Covariance, Riemannian Geometry, Detection, Tracking, Regression, Classification

1 Topological Spaces

A group $\mathcal{G}$ is a set that is endowed with a binary operation and satisfies the closure, associativity, identity, and invertibility properties. A simple example of a group is the set of integers $\mathbb{Z}$ under addition where the identity is 0 and the inverse of any integer is its negative, which is still in $\mathbb{Z}$. Note that, if the binary operation is chosen to be multiplication the set of integers is no longer a group because the inverse may not be an integer. A subset of $\mathcal{G}$ is called as a subgroup if it satisfies all the group properties of being a group under the same binary operation. For example the set of positive rational numbers $\mathbb{Q}^+$ forms a subgroup of rational numbers under multiplication. Yet, the set of negative rational numbers $\mathbb{Q}^-$ is not a subgroup since it does not contain the identity and it is not closed under multiplication.

A topological space is a set $\mathcal{S}$ together with a family of subsets $\mathcal{T}$ if the empty set $\emptyset \in \mathcal{T}$ and $S \in \mathcal{T}$, the union of any family of sets in $\mathcal{T}$ also lies in $\mathcal{T}$, and the intersection of any finite number of sets in $\mathcal{T}$ belongs to $\mathcal{T}$. The family $\mathcal{T}$ is said to be the a topology of $\mathcal{S}$ and the sets in $\mathcal{T}$ are called open sets of the topological space. A given set may have many different topologies. Any open set $\mathcal{U} \in \mathcal{T}$ which contains point $X \in \mathcal{S}$ is called the neighborhood of the point. A Hausdorff space is a topological space in which distinct points have disjoint

* Throughout this paper, learning on manifolds refers to the family of supervised and unsupervised methods to search, cluster, classify, and recognize given observations on smooth manifolds without flattening, charting, or dimensionality reducing them.

neighborhoods, such that, $X, Y \in \mathcal{S}$ and there exists $\mathcal{U}_X, \mathcal{U}_Y \in \mathcal{T}$, $X \in \mathcal{U}_X$, $Y \in \mathcal{U}_Y$ and $\mathcal{U}_X \cap \mathcal{U}_Y = \emptyset$. For instance, the real numbers constitute a Hausdorff space.

For functions defined on Hausdorff spaces it is possible to introduce notions such as continuity by saying that as we move towards a point X, the value of the function gets closer to the value of the function at the point. The idea of being 'close' to a particular point is captured by its neighborhood and the continuity of a function is defined by how it maps open sets of the topology. A mapping between two topological spaces is called continuous if the inverse image of any open set with respect to the mapping is again an open set. A bijective (one-to-one and onto) mapping that is continuous in both directions is called a homeomorphism. Such mappings preserve the topological properties of a given space. Two spaces with a homeomorphism between them are called homeomorphic, and from a topological viewpoint, they are the same, e.g. a square and a circle are homeomorphic to each other, but a sphere and a torus are not.

A manifold $\mathcal{M}$ of dimension d is a connected Hausdorff space for which every point has a neighborhood that is homeomorphic to an open subset $\mathcal{U}$ of $\mathbb{R}^d$. In other words, a manifold corresponds to a topological space which is locally similar to an Euclidean space. For any point $X \in \mathcal{M}$, there exists an open neighborhood $\mathcal{U} \subset \mathcal{M}$ containing the point and homeomorphism ϕ mapping the neighborhood to an open set $\mathcal{V} \subset \mathbb{R}^d$, such that $\phi : \mathcal{U} \mapsto \mathcal{V}$. The pair $(\mathcal{U}, \phi)$ is called as a coordinate chart. An atlas is a family of charts for which the open sets constitute an open covering of the manifold. Every topological manifold has an atlas.

Let $(\mathcal{U}_X, \phi_X)$ and $(\mathcal{U}_Y, \phi_Y)$ be two coordinate charts, such that, $\mathcal{U}_X \cap \mathcal{U}_Y$ is nonempty. The transition map $\phi_X \circ \phi_Y^{-1}$ is a mapping between two open sets $\phi_X(\mathcal{U}_X \cap \mathcal{U}_Y)$ and $\phi_Y(\mathcal{U}_X \cap \mathcal{U}_Y)$. In other words, the transition maps relate the coordinates defined by the various charts to one another. A differentiable manifold C^k is a topological manifold equipped with an equivalence class of atlas whose transition maps are k-times continuously differentiable. In case all the transition maps of a differentiable manifold are smooth, i.e. all its partial derivatives exist, then it is a smooth manifold C^∞.

It is possible to define the derivatives of the curves on a differentiable manifold and attach to every point X a tangent space T_X, a real vector space that intuitively contains the possible directions in which one can tangentially pass through X. Suppose two curves with $\gamma_1(0) = \gamma_2(0) = X$ are equivalent, that is the ordinary derivatives of $\phi \circ \gamma_1$ and $\phi \circ \gamma_2$ at 0 coincide for all charts $(\mathcal{U}, \phi)$ where $X \in \mathcal{U}$. A tangent vector at X is defined by the equivalence class of the smooth curves $\gamma(0) = X$. Tangent vectors are the tangents to the smooth curves lying on the manifold. The tangent space T_X is the set of all tangent vectors at X. The tangent space is a vector space, thereby it is closed under addition and scalar multiplication.

A Riemannian manifold $(\mathcal{M}, g)$ is a differentiable manifold in which each tangent space has an inner product g metric, which varies smoothly from point to point. It is possible to define different metrics on the same manifold to obtain

different Riemannian manifolds. In practice this metric is chosen by requiring it to be invariant to some class of geometric transformations. The inner product g induces a norm for the tangent vectors on the tangent space $\|X\|^2 =< X, X >= g(X)$. A detailed description of these concepts can be found in [1].

A Lie group is a group $\mathcal{G}$ with the structure of a differentiable manifold such that the group operations, multiplication and inverse, are differentiable maps. The tangent space to the identity element of the group forms a Lie algebra. The group operation provides Lie groups with additional algebraic structure. Let $X \in \mathcal{G}$. Left multiplication by the inverse of the group element $X^{-1} : \mathcal{G} \to \mathcal{G}$ maps the neighborhood of X to neighborhood of identity. The inverse mapping is defined by left multiplication by X.

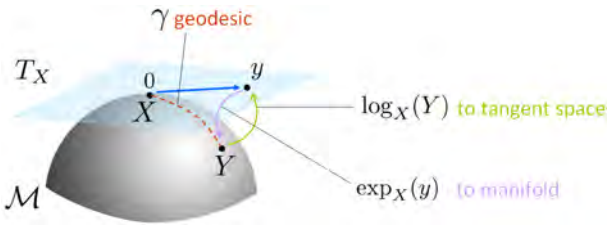

Fig. 1. Manifold and tangent space.

2 Distance on Riemannian Manifolds

A geodesic is a smooth curve that locally joins their points along the shortest path. Suppose $\gamma(r) : [r_0, r_1] \mapsto \mathcal{M}$ be a smooth curve on $\mathcal{M}$. The length of the curve $L(\gamma)$ is defined as

$$L(\gamma) = \int_{r_0}^{r_1} \|\gamma'(r)\| dr. \tag{1}$$

A smooth curve is called geodesic if and only if its velocity vector is constant along the curve $\|\gamma'(r)\| = const$. Suppose X and Y be two points on $\mathcal{M}$. The distance between the points $d(X, Y)$, is the infimum of the length of the curves, such that, $\gamma(r_0) = X$ and $\gamma(r_1) = Y$. All the shortest length curves between the points are geodesics but not vice-versa. However, for nearby points the definition of geodesic and the shortest length curve coincide. For each tangent vector $x \in T_X$, there exists a unique geodesic γ starting at $\gamma(0) = X$ having initial velocity $\gamma'(0) = x$.

The exponential map, $\exp_X : T_X \mapsto \mathcal{M}$, maps the vector y in the tangent space to the point reached by the geodesic after unit time $\exp_X(y) = 1$. Since the velocity along the geodesic is constant, the length of the geodesic is given by the norm of the initial velocity $d(X, \exp_X(y)) = \|y\|_X$. An illustration is shown in Figure 1. Under the exponential map, the image of the zero tangent vector is the

point itself $\exp_X(0) = X$. For each point on the manifold, the exponential map is a diffeomorphism (one-to-one, onto and continuously differentiable mapping in both directions) from a neighborhood of the origin of the tangent space T_X onto a neighborhood of the point X. In general, the exponential map $\exp_X$ is onto but only one-to-one in a neighborhood of X. Therefore, the inverse mapping $\log_X : \mathcal{M} \mapsto T_X$ is uniquely defined only around the neighborhood of the point X. If for any $Y \in \mathcal{M}$, there exists several $y \in T_X$ such that $Y = \exp_X(y)$, then $\log_X(Y)$ is given by the tangent vector with the smallest norm. Notice that both operators are point dependent. For certain manifolds the neighborhoods can be extended to the whole tangent space and manifold hence the exponential map is a global diffeomorphism. From the definition of geodesic and the exponential map, the distance between the points on manifold can be computed by

$$d(X,Y) = d(X, \exp_X(y)) =< \log_X(Y), \log_X(Y) >_X = \| \log_X(Y) \|_X = \|y\|_X. \tag{2}$$

For Riemannian manifolds endowing an inverse mapping, the geodesic distance between two group elements can be written as

$$d(X,Y) = \| \log(X^{-1}Y) \|. \tag{3}$$

The exponential identity $\exp(X)\exp(Y) = \exp(X+Y)$ does not hold for noncommutative matrix Lie groups. The identity is expressed through Baker-Campbell-Hausdorff formula [2] $\exp(X)\exp(Y) = \exp(\mathrm{BCH}(X,Y))$ as

$$\mathrm{BCH}(X,Y) = X + Y + \frac{1}{2}[X,Y] + O(|(X,Y)|^3). \tag{4}$$

where $[X,Y] = XY - YX$ is the Lie bracket operation for nonsingular matrix group.

2.1 Space of Nonsingular Covariance Matrices

The $d \times d$ dimensional symmetric positive definite matrices $\mathbb{S}_d^+$, can be formulated as a Riemannian manifold. Let points on this manifold are covariance matrices X, Y. An invariant Riemannian metric on the tangent space of $\mathbb{S}_d^+$ is given by [4]

$$< y, z >_{\mathrm{X}} = \mathrm{tr}\left(\mathrm{X}^{-\frac{1}{2}} y \mathrm{X}^{-1} z \mathrm{X}^{-\frac{1}{2}}\right). \tag{5}$$

The exponential map associated to the Riemannian metric

$$\exp_{\mathrm{X}}(y) = \mathrm{X}^{\frac{1}{2}} \exp\left(\mathrm{X}^{-\frac{1}{2}} y \mathrm{X}^{-\frac{1}{2}}\right) \mathrm{X}^{\frac{1}{2}} \tag{6}$$

is a global diffeomorphism. Therefore, the logarithm is uniquely defined at all the points on the manifold

$$\log_{\mathrm{X}}(\mathrm{Y}) = \mathrm{X}^{\frac{1}{2}} \log\left(\mathrm{X}^{-\frac{1}{2}} \mathrm{Y} \mathrm{X}^{-\frac{1}{2}}\right) \mathrm{X}^{\frac{1}{2}}. \tag{7}$$

Above,the exp and log are the ordinary matrix exponential and logarithm operators. Not to be confused, $\exp_{\mathrm{X}}$ and $\log_{\mathrm{X}}$ are manifold specific operators which are also point dependent, $\mathrm{X} \in \mathbb{S}_d^+$. The tangent space of $\mathbb{S}_d^+$ is the space of $d \times d$ symmetric matrices and both the manifold and the tangent spaces are $d(d+1)/2$ dimensional.

For symmetric matrices, the ordinary matrix exponential and logarithm operators can be computed easily. Let $\Sigma = \mathrm{U}\mathrm{D}\mathrm{U}^T$ be the eigenvalue decomposition of a symmetric matrix. The exponential series is

$$\exp(\Sigma) = \sum_{k=0}^{\infty} \frac{\Sigma^k}{k!} = \mathrm{U} \exp(\mathrm{D})\mathrm{U}^T \tag{8}$$

where exp(D) is the diagonal matrix of the eigenvalue exponentials. Similarly, the logarithm is given by

$$\log(\Sigma) = \sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k}(\Sigma - \mathrm{I})^k = \mathrm{U} \log(\mathrm{D})\mathrm{U}^T. \tag{9}$$

The exponential operator is always defined, whereas the logarithms only exist for symmetric matrices with positive eigenvalues, $\mathbb{S}_d^+$. From the definition of the geodesic given in the previous section, the distance between two points on $\mathbb{S}_d^+$ is measured by substituting (7) into (5)

$$\begin{aligned} d^2(\mathrm{X}, \mathrm{Y}) &= < \log_{\mathrm{X}}(\mathrm{Y}), \log_{\mathrm{X}}(\mathrm{Y}) >_{\mathrm{X}} \\ &= \mathrm{tr}\left(\log^2(\mathrm{X}^{-\frac{1}{2}}\mathrm{Y}\mathrm{X}^{-\frac{1}{2}})\right). \end{aligned} \tag{10}$$

An equivalent form of the affine invariant distance metric was first given in [3], in terms of joint eigenvalues of X and Y as

$$d(\mathrm{X}, \mathrm{Y}) = \left(\sum_{k=1}^{d} (\ln \lambda_k(\mathrm{X}, \mathrm{Y}))^2\right)^{\frac{1}{2}} \tag{11}$$

where $\lambda_k(\mathrm{X}, \mathrm{Y})$ are the generalized eigenvalues of X and Y, computed from

$$\lambda_k \mathrm{X}\mathbf{v}_k - \mathrm{Y}\mathbf{v}_k = 0 \quad k = 1 \ldots d \tag{12}$$

and $\mathbf{v}_k$ are the generalized eigenvectors. This distance measure satisfies the metric axioms, positivity, symmetry, triangle inequality, for positive definite symmetric matrices.

An orthogonal coordinate system on the tangent space can be defined by the vector operation. The orthogonal coordinates of a vector y on the tangent space at point X is given by

$$\mathrm{vec}_{\mathrm{X}}(y) = \mathrm{upper}(\mathrm{X}^{-\frac{1}{2}} y \mathrm{X}^{-\frac{1}{2}}) \tag{13}$$

where upper refers to the vector form of the upper triangular part of the matrix. The mapping $\mathrm{vec}_{\mathrm{X}}$, relates the Riemannian metric (5) on the tangent space to the canonical metric defined in $\mathbb{R}^d$.

2.2 Region Covariance Descriptor and Pattern Search

Suppose θ be a feature map extracted from a given image I comprising pixel coordinates, color values, pixel-wise derivatives, oriented gradients, filter responses, etc. of appearance and spatial attributes $\theta_{m,n} = [m, n, I, I_m, ...]^T_{m,n}$. Different functions of coordinates enables imposing of different spatial structures e.g. rotational invariance, symmetry, etc.

A region covariance matrix X for any image region is defined as

$$\mathrm{X} = \frac{1}{N} \sum_{m,n \in R}^{N} (\theta_{m,n} - \bar{\theta})(\theta_{m,n} - \bar{\theta})^T \tag{14}$$

where N is the number of pixels and $\bar{\theta}$ is the mean vector of the corresponding features within the region R. Note that, this is not the computation of the covariance of two image regions, but the covariance of image features of a region. Refer to [5] for more details. Such a descriptor provides a natural way of fusing multiple features without a weighted average. Instead of evaluating the first order statistics of feature distributions through histograms, it embodies the second order characteristics. The noise corrupting individual samples are largely filtered out by the multitude of pixels. It endows spatial scale and feature shift invariance. It is possible to compute covariance matrix from feature images in a very fast way using integral image representation [6]. After constructing $d(d+1)/2$ tensors of integral images corresponding to each feature dimension and multiplication of any two feature dimensions, the covariance matrix of any arbitrary rectangular region can be computed in $\mathcal{O}(d^2)$ time independent of the region size.

The space of region covariance descriptors is not a vector space. For example, it is not closed under multiplication with negative scalars. They constitute the space of positive semi-definite matrices $\mathbb{S}_d^{0,+}$. By adding a small diagonal matrix (or guaranteeing no features in the feature vectors would be exactly identical), they can be transformed into $\mathbb{S}_d^{+}$, which is a Riemannian manifold, in order to apply the Riemannian metrics (10, 11).

Fig. 2. Object representation by multiple covariance matrices of subregions.

A first example using the covariance region descriptor is pattern search to locate a given object of interest in an arbitrary image. To find the most similar region in the image, distances between the descriptors of the object and candidate regions are computed. Each pixel of the image is converted to a 9-dimensional

feature vector $\theta_{m,n} = \left[m, n, I^r, I^g, I^b, |I_m|, |I_n|, |I_{mm}|, |I_{nn}|\right]^T_{m,n}$ where $I^{r,g,b}$ are the RGB color values, and $I_{m,n}$ are spatial derivatives. An object is represented by a collection of partial region covariance matrices as shown in Figure 2.

At the first phase, only the covariance matrix of the whole region from the source image is computed. The target image is searched for a region having similar covariance matrix at all the locations and different scales. A brute force search can be performed since the covariance of an arbitrary region can be obtained efficiently. Instead of scaling the target image, the size of the search window is changes. Keeping the best matching locations and scales, the search for initial detections is repeated using the covariance matrices of N_R partially occluded subregions at the second phase. The distance of the object model O and a candidate region R is computed as

$$|O - R| = min_j \left[\sum_{i=0}^{N_R} d(\mathrm{X}_i^R, \mathrm{X}_i^O) - d(\mathrm{X}_j^R, \mathrm{X}_j^O) \right] \tag{15}$$

where the worst match is dismissed to provide robustness towards possible occlusions and changes. The region with the smallest distance is selected as the matching region. Sample matching results are presented in Figure 3 where the manifold search using the Riemannian metrics is compared to the histogram features using the Bhattacharyya distance.

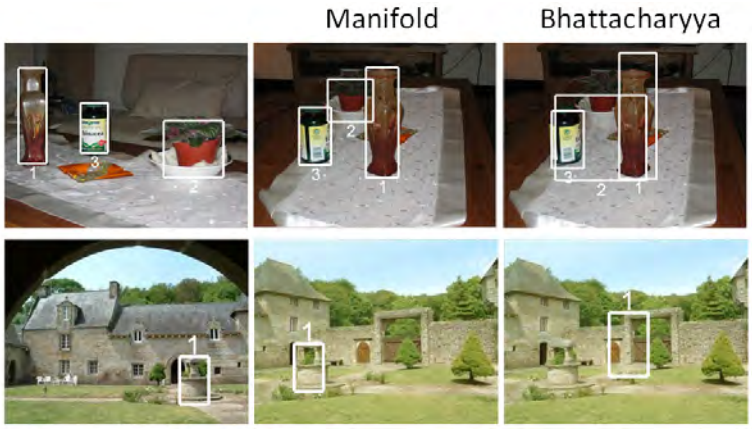

Fig. 3. Regions found via region covariance descriptor and feature histograms.

Region covariance descriptor can be used for texture recognition within a k-NN framework. Each texture class in the training dataset is represented by a bag of region covariance descriptors of the randomly sampled subregions with random sizes between 16×16 and 128×128. Given a test image, a certain number of subregions are extracted and their descriptors are computed. For each covariance matrix, the distances from matrices in the training set are calculated. The label

is predicted according to the majority voting among the k nearest neighbors. Votes are accumulated for all images in the dataset and the class having the maximum vote is assigned as the matching class.

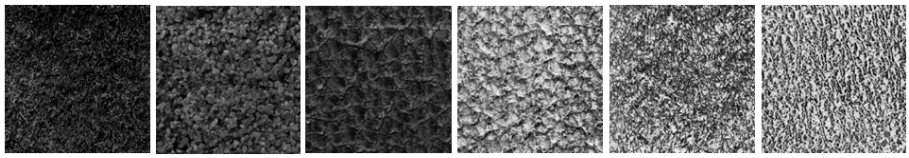

Fig. 4. Samples from 6 classes that are all correctly classified 109 classes out of 112.

A multi-class classifier on Brodatz texture database that consists of 112 gray scale textures (Figure 4) is also tested. Image intensities and norms of first and second order derivatives in both x and y direction are incorporated into the pixel feature vector. Each pixel is mapped to a $d = 5$ dimensional feature space (only 15 independent coefficients). Due to the nonhomogeneous nature, recognition on this dataset is a challenging task. Each 640×640 texture image is divided into four 320×320 subimages and half of the images are used for training and half for testing. The k-NN on manifold is compared with the results reported in [7]. Even though the best performing conventional approach utilizes computationally very expensive texton histograms of 560 coefficients, its performance is limited to 97.32%. Experiments with 100 random covariances from each texture image, $k = 5$ for the k-NN algorithm shows 97.77% recognition with a fraction of the load.

3 Computing Mean on Riemannian Manifolds

Similar to Euclidean spaces, the Karcher mean [8] of points on Riemannian manifold, is the point on $\mathcal{M}$ which minimizes the sum of squared distances

$$\bar{X} = \arg \min_{X \in \mathcal{M}} \sum_{k=1}^{K} d^2(X_k, X) \tag{16}$$

where the distance metric is defined by (10,11).

Differentiating the error function with respect to X and setting it equal to zero gives the following gradient descent procedure [4]

$$\bar{X}^{j+1} = \exp_{\bar{X}^j} \left[\frac{1}{K} \sum_{k=1}^{K} \log_{\bar{X}^j}(X_k) \right] \tag{17}$$

which finds a local minimum of the error function. The method iterates by computing the first order approximations to the mean on the tangent space. The weighted mean computation is similar to arithmetic mean. Replacing the

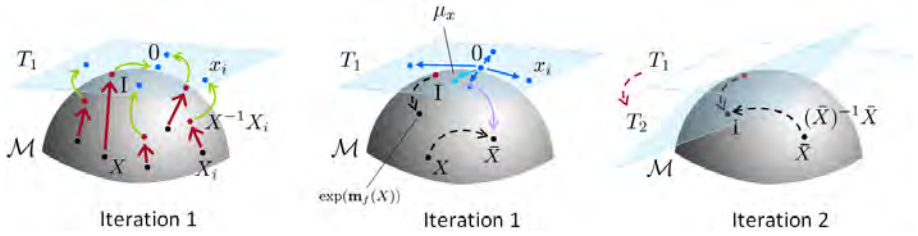

Fig. 5. Mean computation is achieved by transforming points on manifold to the neighborhood of I on the manifold by $X^{-1}X_i$, mapping them to the tangent space of X, finding the mean in the tangents space, back projecting the tangent space mean onto the manifold, and repeating these steps until the dislocation between the successive iterations becomes negligible.

inside of the exponential, the mean of the tangent vectors with the weighted mean can be obtained as

$$\bar{X}^{j+1} = \exp_{\bar{X}^j}\left[\frac{1}{\sum w_k}\sum_{k=1}^{K} w_k \log_{\bar{X}^j}(X_k)\right]. \tag{18}$$

3.1 Object Model Update

Finding the correspondences of the previously detected objects in the current frame, called as tracking, is an essential task in many computer vision applications.

For a given object region, the covariance matrix of the features can be computed as the model of the object. within all possible locations of the current frame, the region that has the minimum covariance distance from the model can be searched and assigned as the estimated location. Note that such an exhaustive search is performed to highlight the discriminant power of the region descriptor and the distance metric on manifold. Often search is constrained by a predictive prior. In order to adapt to variations in object appearance, a set of previous covariance matrices are stored and a mean covariance matrix is computed on the manifold as the object representative. Sample tracking results are shown in Figure 6 below.

4 Computing Kernel Density

The mean-shift is a nonparametric clustering technique which does not require prior knowledge of the number of clusters, and does not constrain the shape of the clusters. Data points are assumed to be originated from an unknown distribution which is approximated by kernel density estimation in vector spaces

$$f(x) = \frac{1}{N}\sum_{i=1}^{N} H\left(x - x_i\right) = \frac{\kappa}{N}\sum_{i=1}^{N} h\left(\|x - x_i\|^2\right) \tag{19}$$

Fig. 6. Montages of the detection results (middle) without model update: detection rate is 47.7%, (right) with weighted mean based update mechanism on manifold: detection rate is 100%.

where $H(x) = \kappa h(\|x\|^2)$ is a radially symmetric kernel with unit radius. The cluster centers are located by the mean-shift procedure and the data points associated with the same modes produce a partitioning of the feature space. By taking the gradient of the above equation, the stationary points of the density function can be found iteratively via

$$\bar{x} = \frac{\sum_i x_i \cdot k\left(\|x - x_i\|^2\right)}{\sum_i k\left(\|x - x_i\|^2\right)} \tag{20}$$

where $k(x) = -h'(x)$. At each step, a local weighted mean is computed, and the computation is repeated centered on the previous estimate. The difference between the current and the previous location estimates is called the mean-shift vector

$$m(x) = \bar{x} - x. \tag{21}$$

Starting at each data point, mean-shift iterations convergence to a local mode of the distribution, i.e. a basin of attraction.

A generalization of the mean-shift procedure for parameter spaces having matrix Lie group structure where the mean-shift algorithm runs on a Lie group by iteratively transforming points between the Lie group (on Riemannian manifold) and Lie algebra (on tangent space). Using the intrinsic distance, the multivariate kernel density estimate at X is given by

$$f(X) = \frac{\kappa}{N} \sum_{i=1}^{N} h\left(\|\log(X^{-1}X_i)\|^2\right) \tag{22}$$

where $x_i = \log(X^{-1}X_i)$.

The group operation maps the neighborhood of X to the neighborhood of I and the tangent space at X to the Lie algebra g. The approximation error can be expressed in terms of the higher order terms in BCH formula (4). The error is minimal around I and the mapping assures that the error is minimized. The

point X is mapped to 0, thus the second term in the mean-shift vector does not exists. The mean-shift vector on the tangent space can be transferred to the Lie group as

$$m(X) = \exp\left(\frac{\sum_i \log(X^{-1}X_i) \cdot k\left(\|\log(X^{-1}X_i)\|^2\right)}{\sum_i k\left(\|\log(X^{-1}X_i)\|^2\right)}\right) \tag{23}$$

and the location of X can be updated as

$$\bar{X} = X\exp(m(X)). \tag{24}$$

An invariant estimator on the linear group of non-singular matrices with positive determinant can be found in [11].

4.1 Motion Detection

Several parameter spaces which commonly occur in computer vision problems do not form a vector space. For instance, the set of all affine transformations forms a matrix Lie group. Two-dimensional affine transformation $A(2)$ is given by the set of matrices in the following form

$$\mathrm{X} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ 0 & 1 \end{bmatrix}_{3\times 3} \tag{25}$$

where $\mathbf{A}$ is a nonsingular 2×2 matrix. By selecting each of the entries as an orthonormal basis, X constitutes a $d = 6$ dimensional manifold.

One application of the mean-shift on manifolds is multiple rigid motion estimation from noisy point correspondences in presence of large amount of outliers [9]. Given two images, local feature points such as corner points are found. These points are paired via a descriptor matching algorithm. Due to occlusions and errors in the point matching process most of the point correspondences are outliers. For each set of randomly selected 3-point correspondences a 2D rigid affine transformation $(\mathbf{A}, \mathbf{b})$ is estimated. These transformations constitute the set of X. Then the above mean-shift procedure is applied to find the local modes that represent rigid objects having distinct affine motions. A sample result is given in Figure 7.

5 Linear Regression on Riemannian Manifolds

Regression refers to understand the relationship between multiple variables. Linear regression assumes the relationship depends linearly on a model in which the conditional mean of a scalar variable given the other variables is an affine function of those variables. Numerous procedures have been developed for parameter estimation and inference in linear regression. Here a least squares estimator is described.

Fig. 7. (Left) 2D images with 83 points are detected via corner detection algorithm. Less than 50% of the point correspondences are accurate. (Right) The boundaries of the bodies and transformed boundaries with the estimated motion parameters. The estimation is almost perfect. *Courtesy O. Tuzel*

Suppose (α_i, X_i) are the pairs of observed data $\alpha \in \mathbb{R}^d$ in vector space and the corresponding points on the manifold $X \in \mathcal{M}$. The regression function φ maps the vector space data onto the manifold $\varphi : \mathbb{R}^d \mapsto \mathcal{M}$. An objective function is defined as the sum of the squared geodesic distances between the estimations $\varphi(\alpha_i)$ and the points X_i

$$J = \sum_i d^2 \left[\varphi(\alpha_i), X_i\right]. \tag{26}$$

Assuming a Lie algebra on the manifold can be defined, the objective function can be written using the Baker-Campbell-Hausdorff approximation (4) as

$$J = \sum_i \left\|\log\left[\varphi^{-1}(\alpha_i) X_i\right]\right\|^2 \approx \sum_i \left\|\log\left[\varphi(\alpha_i)\right] - \log\left[X_i\right]\right\|^2 \tag{27}$$

up to the first order terms. The regression function φ can be written as

$$\varphi(\alpha_i) = \exp\left(\alpha_i^T \Omega\right) \tag{28}$$

to learn the function $\Omega : \mathbb{R}^d \mapsto \mathbb{R}^n$ which estimates the tangent vectors $\log\left(X_i\right)$ on the Lie algebra where Ω is the $d \times n$ matrix of regression coefficients. Thus, the objective function (27) becomes

$$J = \sum_i \left\|\alpha_i^T \Omega - \log\left[X_i\right]\right\|^2 \tag{29}$$

Let $\mathbf{X}$ be the $k \times d$ matrix of initial observations and $\mathbf{Y}$ be the $k \times n$ matrix of mappings to the Lie algebra

$$\mathbf{X} = \begin{bmatrix} [\alpha_1]^T \\ \vdots \\ [\alpha_k]^T \end{bmatrix} \quad \mathbf{Y} = \begin{bmatrix} [\log(X_1)]^T \\ \vdots \\ [\log(X_k)]^T \end{bmatrix} \tag{30}$$

Substituting (30) into (29), one can obtain

$$J = tr[(\mathbf{X}\Omega - \mathbf{Y})^T (\mathbf{X}\Omega - \mathbf{Y})] \tag{31}$$

where the trace replaces the summation in (27). Differentiating the objective function J with respect to Ω, the minimum is achieved at $\Omega = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$. To avoid overfitting, additional constraints on the size of the regression coefficients can be introduced

$$J = tr[(\mathbf{X}\Omega - \mathbf{Y})^T(\mathbf{X}\Omega - \mathbf{Y})] + \beta\|\Omega\|^2 \tag{32}$$

which is called the ridge regression [10]. The minimizer of the cost function J is given by $\Omega = (\mathbf{X}^T\mathbf{X} + \beta\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y}$ where $\mathbf{I}$ is an $d \times d$ identity matrix. The regularization coefficient β determines the degree of shrinkage on the regression coefficients.

5.1 Affine Motion Tracking

At the initialization of the object, the affine motion tracker estimates a regression function that maps the region feature vectors to the hypothesized affine motion vectors by first hypothesizing a set of random motion vectors within the given bounds, determining the transformed regions for these motions, and then computing the corresponding features within each warped region. In the tracking time, it extracts the feature vector only for the previous object region location and applies the learned regression function.

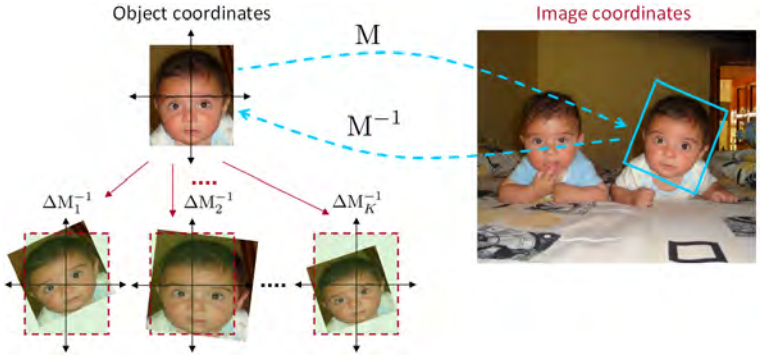

Fig. 8. Random transformations are applied in object coordinates to generate the training features for regression function estimation.

Let M transforms a unit square at the origin to the affine region enclosing the target object $[x \;\; y \;\; 1]_I^T = \mathrm{M}[x \;\; y \;\; 1]_O^T$ where the subscripts indicate the image and object coordinates respectively. The inverse M^{-1} is an affine motion matrix and transforms the image coordinates to the object coordinates. The aim of tracking is to estimate the transformation matrix M_i, given the previous images and the initial transformation M_0. The transformations are modeled incrementally

$$\mathrm{M}_i = \mathrm{M}_{i-1}.\Delta\mathrm{M}_i \tag{33}$$

and estimate the increments $\Delta \mathrm{M}_i$ at each time. The transformation $\Delta \mathrm{M}_i$ corresponds to motion of target from time $i-1$ to i in the object coordinates.

Suppose the target region is represented with orientation histograms computed at a regular grid inside the unit square in object coordinates, i.e with $\alpha(I(\mathrm{M}_i^{-1})) \in \mathbb{R}^d$ where d is the dimension of the descriptor. Given the previous location of the object M_{i-1} and the current observation I_i, the new transformation $\Delta \mathrm{M}_i$ by the regression function is estimated as

$$\Delta \mathrm{M}_i = \varphi(\alpha_i(\mathrm{M}_{i-1}^{-1})). \tag{34}$$

The problem reduces to learning and updating the regression function φ.

During the learning step, a training set of K random affine transformation matrices $\{\Delta \mathrm{M}_j\}_{j=1\ldots K}$ are generated around the identity matrix (Figure 8). The approximation is good enough since the transformations are in a small neighborhood of the identity. The object coordinates are transformed by multiplying on the left with $\Delta \mathrm{M}_j^{-1}$ and the descriptor α_j is computed at $\Delta \mathrm{M}_j^{-1}.\mathrm{M}_i^{-1}$. The transformation M_i^{-1} moves the object back to the unit square. The training set consists of samples $\{\alpha_j, \Delta \mathrm{M}_j\}_{j=1\ldots K}$. The size of the training set is kept relatively small $K = 200$. Since number of samples is smaller than the dimension of the feature space, $K < d$, the system is underdetermined. To relieve this, the ridge regression is applied to estimate the regression coefficients.

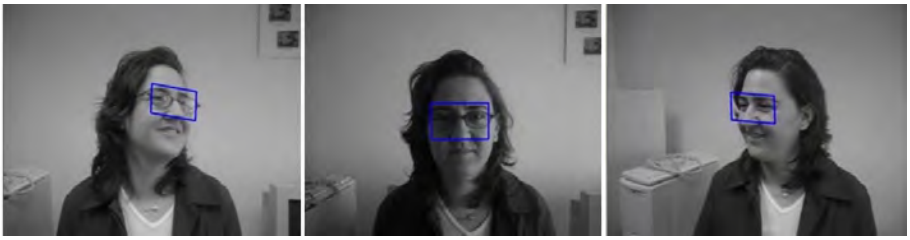

Fig. 9. Regression tracking on manifold for a given region. Note that the tracking is still valid even the region undergoes out-of-plane rotations.

Since objects can undergo appearance changes in time, it is necessary to adapt to these variations. The model update achieves reestimating the regression function. During tracking, a set of random observations are generated at each frame with the same method described above. The observations stored for most recent frames constitute the update training set. More details and an importance sampling based adaptation can be found in [12].

5.2 Pose Invariant Detection

Above method can be used to build an affine invariant object detection algorithm by incorporating a class specific regression function to an existing pose dependent

detector. Instead of learning a tracking function of the specific target object, a regression function of the object class is trained. The learning is performed on the training set generated by applying a total of K random affine transformations to multiple samples from the same class, e.g. face images. The training stage is an offline process and a more complicated model can be learned compared to tracking applications. However, the learned function should be evaluated fast at runtime, since the tracker is initiated at several locations for each test image.

On a sparse grid on the test image a sparse scan of the image is performed. At each grid point the class specific regression function is applied and the region it converges is determined. This scan finds all the locations in the motion space (e.g. affine) which resemble the object model. The object detector is then evaluated only at these locations. The benefits of the approach is two-fold. First, the size of the search space drastically reduces. Secondly, it performs continuous estimation of the target pose in contrast to the existing techniques perform search on a quantized space. Utilizing a pose dependent object detection algorithm (e.g., frontal in upright position), the method enables to detect objects in arbitrary poses.

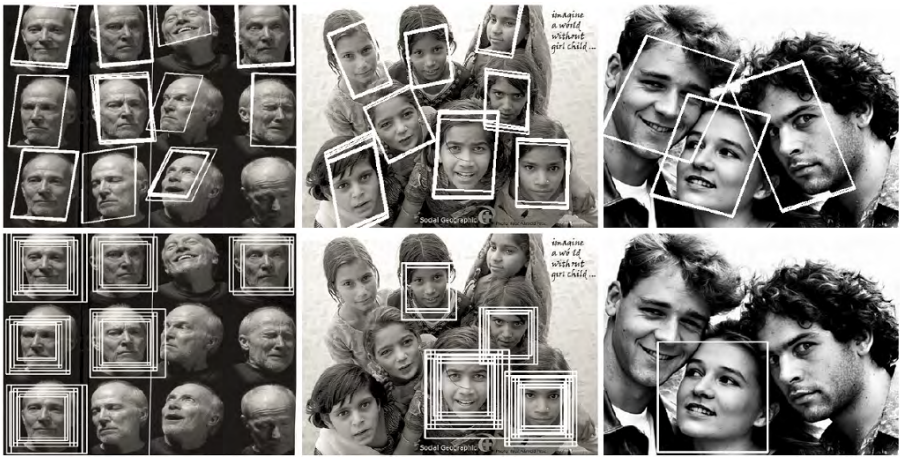

Fig. 10. (Top) Class specific affine invariant face detection. (Bottom) VJ multi-pose face detector results for sample images containing non-frontal faces.

In experiments on a face dataset which consists of 803 face images from CMU, MIT and MERL datasets, the Viola and Jones (VJ) face detector [13] evaluated at the affine warped face images could detect only 5% of the faces that are norm 0.5 distant. The Lie algebra based estimation is significantly superior by achieving 95.6% detection for the same images. Sample detection results for affine invariant detection of faces are given in Figure 10.

6 Classifiers on Riemannian Manifolds

Let $\{(X_i, y_i)\}_{i=1...N}$ be the training set with respect to class labels, where $X_i \in \mathcal{M}$, $y_i \in \{0, 1\}$. Our task is to find a classifier $Z(X) : \mathcal{M} \mapsto \{0, 1\}$, which divides the manifold into two sets based on the training samples of labeled points. A function to divide a manifold is an intricate notion compared to Euclidean spaces. A linear classifier that is represented by a point and a direction vector on $\mathbb{R}^2$ separates the space into two. However, such lines on the 2-torus cannot divide the manifold. A straightforward approach for classification would be to map the manifold to a higher dimensional Euclidean space, which can be considered as flattening or charting the manifold. However, there is no such mapping that globally preserves the distances between the points on the manifold in general.

6.1 Local Maps and Boosting

One can design an incremental approach by training several weak classifiers on the tangent space and combining them through boosting. Since the mappings from neighborhoods on the manifold to the Euclidean space are homeomorphisms around the neighborhood of the points, the structure of the manifold is preserved locally in tangent spaces, thus, the classifiers can be trained on the tangent space at any point on the manifold. The mean of the points (16) minimizes the sum of squared distances on the manifold, therefore it is a good approximation up to the first order.

At each iteration, the weighted mean of the points where the weights are adjusted through boosting are computed. The points to the tangent space are mapped at the mean and a weak classifier on this vector space is learned. Since the weights of the samples which are misclassified during earlier stages of boosting increase, the weighted mean moves towards these points producing more accurate classifiers for these points (Figure 11). The approach minimizes the approximation error through averaging over several weak classifiers.

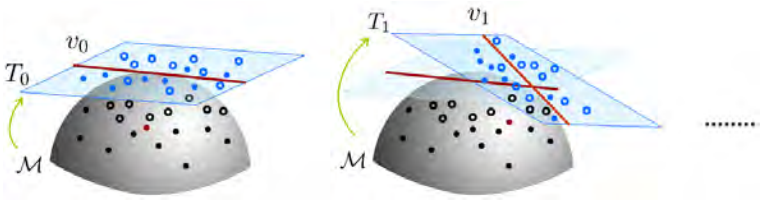

Fig. 11. Illustration of successive learning of weak classifiers on tangent spaces.

6.2 LogitBoost on Riemannian Manifolds

The probability of X being in class 1 is represented by

$$p(X) = \frac{e^{Z(X)}}{e^{Z(X)} + e^{-Z(X)}} \qquad Z(X) = \frac{1}{2}\sum_{l=1}^{L} z_l(X). \tag{35}$$

The LogitBoost algorithm learns the set of regression functions $\{z_l(X)\}_{l=1...L}$ (weak learners) by minimizing the negative binomial log-likelihood of the data

$$-\sum_{i=1}^{N} [y_i \log(p(X_i)) + (1 - y_i) \log(1 - p(X_i))] \tag{36}$$

through Newton iterations. At the core of the algorithm, LogitBoost fits a weighted least square regression, $z_l(X)$ of training points $X_i \in \mathbb{R}^d$ to response values $\beta_i \in \mathbb{R}$ with weights w_i.

Input: Training set $\{(X_i, y_i)\}_{i=1...N}$, $X_i \in \mathcal{M}$, $y_i \in \{0, 1\}$

- Start with weights $w_i = 1/N$, $i = 1...N$,
  $Z(X) = 0$ and $p(X_i) = \frac{1}{2}$
- Repeat for $l = 1...L$
  - Compute the response values and weights
    $\beta_i = \frac{y_i - p(X_i)}{p(X_i)(1-p(X_i))}$
    $w_i = p(X_i)(1 - p(X_i))$
  - Compute weighted mean of the points
    $\bar{X}_l = \arg\min_{Y \in \mathcal{M}} \sum_{i=1}^{N} w_i d^2(X_i, Y)$ (17)
  - Map the data points to the tangent space at X_l
    $x_i = \mathrm{vec}_{\bar{X}_l}(\log_{\bar{X}_l}(X_i))$
  - Fit the function $v_l(x)$ by weighted least-square regression of β_i to x_i using weights w_i
  - Update $Z(X) \leftarrow Z(X) + \frac{1}{2} z_l(X)$ where z_l is defined in (37) and $p(X) \leftarrow \frac{e^{Z(X)}}{e^{Z(X)} + e^{-Z(X)}}$
- Output the classifier sign
  $[Z(X)] = \mathrm{sign}\,[\sum_{l=1}^{L} z_l(X)]$

Fig. 12. LogitBoost on Riemannian Manifolds.

The LogitBoost algorithm on Riemannian manifolds is similar to original LogitBoost, except differences at the level of weak learners. In our case, the domain of the weak learners are in $\mathcal{M}$ such that $z_l(X) : \mathcal{M} \mapsto \mathbb{R}$. Following the discussion of the previous section, the regression functions are learned in the tangent space at the weighted mean of the points on the manifold. The weak learners are defined as

$$z_l(X) = v_l(\mathrm{vec}_{\bar{X}_l}(\log_{\bar{X}_l}(X))) \tag{37}$$

and learn the functions $v_l(x) : \mathbb{R}^d \mapsto \mathbb{R}$ and the weighted mean of the points $\bar{X}_l \in \mathcal{M}$. Notice that, the mapping vec (13), gives the orthogonal coordinates

of the tangent vectors. For functions $\{v_l\}_{l=1...L}$, it is possible to use any form of weighted least squares regression such as linear functions, regression stumps, etc., since the domain of the functions are in $\mathbb{R}^d$.

6.3 Object Detection

Due to the articulated structure and variable appearance of the human body, illumination and pose variations, human detection in still images presents a challenge. For this task, $K = 30$ LogitBoost classifiers on $\mathbb{S}_8^+$ are combined with rejection cascade, as shown in Figure 13. Weak classifiers $\{v_l\}_{l=1...L}$ are linear regression functions learned on the tangent space of $\mathbb{S}_8^+$

$$\left[m \;\; n \;\; |I_m| \;\; |I_n| \;\; \sqrt{I_m^2 + I_n^2} \;\; |I_{mm}| \;\; |I_{nn}| \;\; \arctan \frac{|I_m|}{|I_n|}\right]^T \tag{38}$$

The covariance descriptor of a region is an 8×8 matrix and due to symmetry only upper triangular part is stored, which has only 36 different values. The tangent space is $d = 36$ dimensional vector space as well.

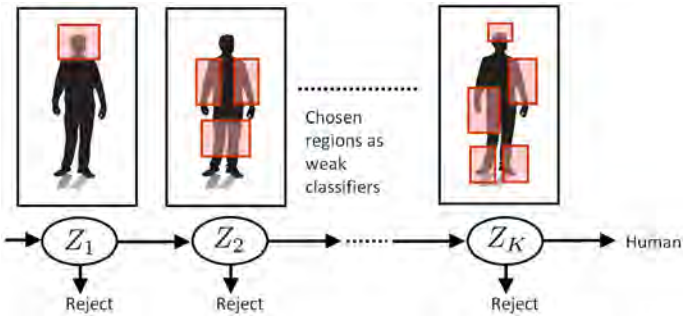

Fig. 13. Cascade of LogitBoost classifiers. The kth classifier selects normalized region covariance descriptors of the corresponding subregions.

Let N_p and N_n be the number of positive and negative images in the training set. Since any detection window sampled from a negative image is a negative sample, it is possible to generate much more negative examples than the number of negative images. Suppose that the kth cascade level is being trained. All the possible detection windows on the negative training images are classified with the cascade of the previous $(k-1)$ LogitBoost classifiers. The samples which are misclassified form the possible negative set. Since the cardinality of the possible negative set is very large, examples from this set are sampled as the negative examples at cascade level k. At every cascade level, all the positive training images are considered as the positive training set.

A very large number of covariance descriptors can be computed from a single detection window and it is computationally intractable to test all of them. At

Fig. 14. Detection examples using cascade of LogitBoost classifiers on manifold. White dots show all the detection results. Black dots are the modes generated and the ellipses are average detection window sizes. There are extremely few false detections and misses.

each boosting iteration of kth LogitBoost level, subwindows are sampled, and normalized region covariance descriptors are constructed. The weak classifiers representing each subwindow are learned, and the best classifier which minimizes negative binomial log-likelihood (36) is added to the cascade level k.

Each level of cascade detector is optimized to correctly detect at least 99.8% of the positive examples, while rejecting at least 35% of the negative examples. In addition, a margin constraint between the positive samples and the decision boundary is enforced. Let $p_k(X)$ be the probability of a sample being positive at cascade level k, evaluated through (35). Let X_p be the positive example that has the $(0.998N_p)$th largest probability among all the positive examples. Let X_n be the negative example that has the $(0.35N_n)$th smallest probability among all the negative examples. Weak classifiers are added to cascade level k until $p_k(X_p) - p_k(X_n) > \tau$, where $\tau = 0.2$. When the constraint is satisfied, a new sample is classified as positive by cascade level k if $p_k(X) > p_k(X_p) - \tau > p_k(X_n)$ or equivalently $Z_k(X) > Z_k(X_n)$.

Since the sizes of the pedestrians in novel scenes are not known a priori, the images are searched at multiple scales. Utilizing the classifier trained on the INRIA dataset, sample detection examples for crowded scenes with pedestrians having variable illumination, appearance, pose and partial occlusion are shown in Figure 14.

7 Conclusions

Several parameter spaces that commonly occur in computer vision problems have Riemannian manifold structure including invertible affine transformations, non-zero quaternions with multiplication, general linear group (invertible square real matrices), real matrices with unit determinant, orientation-preserving isometries, real orthogonal matrices, and symplectic matrices.

Manifold based methods provide major improvements over the existing Euclidean techniques as demonstrated in this paper.

Acknowledgments

Thanks to Oncel Tuzel, Peter Meer and Pan Pan for their contributions and inspiring discussions.

References

1. W. M. Boothby, "An Introduction to Differentiable Manifolds and Riemannian Geometry," Academic Press, second edition, (1986)
2. W. Rossmann, "Lie Groups: An Introduction Through Linear Groups," Oxford Press, (2002)
3. W. Forstner and B.Moonen, "A metric for covariance matrices," Technical report, Dept. of Geodesy and Geoinformatics, Stuttgart University, (1999)
4. X. Pennec, P. Fillard, and N. Ayache, "A Riemannian framework for tensor computing," International Journal of Computer Vision, vol. 66, no. 1, pp. 4166, (2006)
5. O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification, " in Proc. 9th European Conference on Computer Vision, Gratz, Austria, Vol. 2, pp. 589600, (2006)
6. F. Porikli, "Integral histogram: A fast way to extract histograms in Cartesian spaces," in Proc. IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA, Vol. 1, pp. 829836, (2005)
7. B. Georgescu, I. Shimshoni, and P. Meer, "Mean shift based clustering in high dimensions: A texture classification example," in Proc. 9th International Conference on Computer Vision, Nice, France, pp. 456463, (2003)
8. H. Karcher, "Riemannian center of mass and mollifier smoothing," Commun. Pure Appl. Math., vol. 30, pp. 509541, (1977)
9. O. Tuzel, R. Subbarao, and P. Meer, "Simultaneous multiple 3D motion estimation via mode finding on Lie groups," in Proc. 10th International Conference on Computer Vision, Beijing, China, volume 1, pp. 1825, (2005)
10. T. Hastie, R. Tibshirani, and J. Freidman, "The Elements of Statistical Learning," Springer, (2001)
11. E. Miller and C. Chefd'hotel, "Practical non-parametric density estimation on a transformation group for vision," In Proc. IEEE Conference on Computer Vision and Pattern Recognition, (2003)
12. F. Porikli and P. Pan, "Regressed importance sampling on manifolds for efficient object tracking," in Proc. 6th IEEE Advanced Video and Signal based Surveillance Conference, (2009)
13. P. Viola and M. Jones, "Robust real-time object detection," International Journal of Computer Vision, (2001)