Published in final edited form as:

SIAM J Imaging Sci. ; 6(16): 592–615. doi:10.1137/110853376.

A NOVEL DYNAMIC SYSTEM IN THE SPACE OF SPD MATRICES WITH APPLICATIONS TO APPEARANCE TRACKING*

GUANG CHENG[†] and BABA C. VEMURI[†]

[†]Department of Computer & Information Science & Engineering, University of Florida

Keywords

Intrinsic recursive filter; Riemannian geometry; Space of symmetric positive definite matrices; Covariance descriptor

In this paper, we address the problem of video tracking using covariance descriptors constructed from simple features extracted from the given image sequence. Theoretically, this can be posed as a tracking problem in the space of (n,n) symmetric positive definite (SPD) matrices denoted by P_n . A novel probabilistic dynamic model in P_n based on Riemannian geometry and probability theory is presented in conjunction with a geometric (intrinsic) recursive filter for tracking a time sequence of SPD matrix measurements in a Bayesian framework. This newly developed filtering method can be used for the covariance descriptor updating problem in covariance tracking, leading to new and efficient video tracking algorithms. To show the accuracy and efficiency of our tracker in comparison to the state-of-the-art, we present synthetic experiments on P_n and several real data experiments for tracking in video sequences.

1. Introduction

The space of Symmetric Positive Definite (SPD) matrices denoted as $P_n = \{\mathbf{X} = (x_{ij})_{1 \leq i,j \leq n} | \mathbf{X} = \mathbf{X}^T, \mathbf{v} \in \mathbb{R}^n, \mathbf{v} \neq 0, \mathbf{v}^T \mathbf{X} \mathbf{v} > 0\}$ is very important as a feature space in the area of computer vision and its applications. Several kinds of features lie in P_n , such as covariance matrices, diffusion tensors in medical imaging, Cauchy deformation tensors in mechanics etc. Unlike Euclidean space, P_n is a Riemannian manifold, but not a vector space. Many operations and algorithms in Euclidean space cannot be applied directly to P_n and this has led to a flurry of research activity in the recent past. Several operations in Euclidean space have been extended to Riemannian manifolds. For example, the extension of the arithmetic mean to a Riemannian manifold is called the Karcher Mean [14]; The extension of Principal Component Analysis (PCA) is called the Principle Geodesic Analysis [10, 11]; the mean-shift algorithm [8] has also been extended to Riemannian manifolds [31]. However, for filtering operations in dynamic scenes such as the popular Kalman filter [29], an intrinsic extension does not exist in literature to date.

Recursive filtering is a technique to reduce the noise in the measurements by using theory of recursion applied to filtering. It is often used in time sequence data analysis especially in the tracking problem where the model of the target needs to be updated based on the measurement and previous tracking results. Many recursive filtering techniques have been developed in Euclidean space, such as Kalman filter, extended Kalman filter etc, where the inputs and outputs of the filter are all vectors [29]. However, several tracking problems are

*This research was in part funded by the NIH grant NS066340 to BCV.

naturally set in P_n , a Riemannian symmetric space [13]. Recent work reported in [26] on covariance tracking uses a covariance matrix (constructed from pixel-wise features inside the object region) that belongs to P_n in order to describe the appearance of the target being tracked. This covariance descriptor has proved to be robust in both video detection [35, 33] and tracking [26, 24, 39, 18, 36, 15, 19, 5]. The covariance descriptor is a compact feature representation of the object with relatively low dimension compared to other appearance models such as the histogram model in [9]. In [34] an efficient algorithm for generating covariance descriptors from feature vectors is reported based on the integral image technique, which makes it possible to use covariance descriptors in real time video tracking and surveillance.

One major challenge in covariance tracking is how to recursively estimate the covariance template (a covariance descriptor that serves as the target appearance template) based on the input video frames. In [26] and also in [24, 19] the Karcher mean of sample covariance descriptors from a fixed number of video frames is used as the covariance template. This method is based on the natural Riemannian distance – the GL -invariant distance (Sec. 2.1) in P_n . Currently, this Karcher mean can not be computed in closed form, and the computation is achieved using a gradient based optimization technique which is inefficient especially when the input contains a large number of samples. To overcome this efficiency problem, a Log-Euclidean metric was used in [18, 15], an arithmetic mean like method is used in [39], and a recursive filter for linear systems in P_n was developed in [36]. However, none of these are intrinsic because they adopt methods which are extrinsic to P_n .

Recently, some methods were reported addressing the recursive filtering problem on Riemannian manifolds other than P_n . For example, the geometric particle filter for handling 2D affine motions (2-by-2 non-singular matrix) was reported in [25, 17, 15], and an extension to Riemannian manifolds was developed in [28]. However, since the covariance descriptor is usually a high dimensional descriptor, e.g. the degrees of freedom of a 5×5 covariance matrix are 15, the number of samples required for the particle filter would be quite large in this case. Additionally, computing the intrinsic (Karcher) mean on P_n is computationally expensive for large sample sizes. Thus, using an intrinsic particle filter to update covariance descriptor would be computationally expensive for the tracking problem. There are also existing tracking methods on Grassmann manifolds [30, 7] however, it is non-trivial to extend these to P_n since Grassmann manifolds and P_n have very different geometric properties, e.g. Grassmann manifolds are compact and have a non-negative sectional curvature when using an invariant Riemannian metric [38], while P_n is non-compact and has non-positive sectional curvature when using an invariant (to the general linear group (GL)) Riemannian metric [13].

In this paper, we focus on the problem of developing an intrinsic recursive filter – abbreviated IRF for the rest of this paper – on P_n . A novel probabilistic dynamic model on P_n based on Riemannian geometry and probability theory is presented. Here, the noisy state and observations are described by matrix-variate random variables whose distribution is a generalized normal distribution on P_n based on the GL -invariant measure. In [23, 16] authors provide a linear approximation of this distribution for cases when the variance of the distribution is very small. In contrast, in this paper, we explore several properties of this distribution for the arbitrary variance case. We then develop the IRF based on this novel dynamic model and the Bayesian framework with a moving window approximation presented in [7] which tracks modes of the distribution (for details, see section 3). By applying this recursive filter – to achieve covariance tracking – in conjunction with an existing particle position tracker [3], we obtain a new efficient real time video tracking algorithm described in Section 3.2. We present experiments with comparisons to existing

state-of-the-art methods and quantitative analysis that support the effectiveness and efficiency of the proposed algorithm.

The remainder of this paper is organized as follows: In section 2 we introduce the probabilistic dynamic model on P_n after presenting some background Riemannian geometry and an invariant probability measure. Then the IRF and the tracking algorithms are presented in section 3, followed by the experiments in section 4. Finally we draw conclusions in section 5.

2. IRF: A New Dynamic Tracking Model on P_n

2.1. Riemannian Geometry on P_n

In this section, we briefly introduce the basic tools of Riemannian geometry for P_n and then motivate the use of the GL -invariant metric on P_n for developing our new dynamic model. We refer the reader to [21, 13, 32] for details. Following this, we contrast the popularly used Log-Euclidean framework against the intrinsic framework for developing the dynamic recursive filter proposed in this paper. This provides the necessary motivation for an intrinsic recursive filter.

P_n is the space of $n \times n$ symmetric positive definite (SPD) matrices, which is a Riemannian manifold. It can be identified with the quotient space $GL(n)/O(n)$ [32], where $GL(n)$ denotes the General Linear group – the group of $(n \times n)$ non-singular matrices, and $O(n)$ is the orthogonal group – the group of $(n \times n)$ orthogonal matrices. This makes P_n a homogeneous space with $GL(n)$ as the group that acts on it and the group action defined for any $\mathbf{X} \in P_n$ by $\mathbf{X}[\mathbf{g}] = \mathbf{g}\mathbf{X}\mathbf{g}^t$. One can now define GL -invariant quantities such as the GL -invariant inner product based on the group action defined above. We will now begin with inner product in the tangent space of P_n . For tangent vectors $\mathbf{U}$ and $\mathbf{V} \in T_{\mathbf{X}}P_n$ (the tangent space at point $\mathbf{X}$, which is the space of symmetric matrices of dimension $(n+1)n/2$ and a vector space) the GL invariant inner product is defined as $\langle \mathbf{U}, \mathbf{V} \rangle_{\mathbf{X}} = \langle \mathbf{g}\mathbf{U}\mathbf{g}^t, \mathbf{g}\mathbf{V}\mathbf{g}^t \rangle_{\mathbf{g}\mathbf{X}\mathbf{g}^t}$. On P_n this GL invariant inner product takes the form,

$$\langle \mathbf{U}, \mathbf{V} \rangle_{\mathbf{X}} = \text{tr}(\mathbf{X}^{-1/2} \mathbf{U} \mathbf{X}^{-1} \mathbf{V} \mathbf{X}^{-1/2}). \quad (2.1)$$

With metric/inner product defined on the manifold, the length of any curve in $P_n : [0, 1]$

P_n is defined as $(\gamma)^2 = \int_0^1 \langle \dot{\gamma}, \dot{\gamma} \rangle_{\gamma(t)} dt$. The distance between any $\mathbf{X}, \mathbf{Y} \in P_n$ is defined as the length of the shortest curve between $\mathbf{X}$ and $\mathbf{Y}$ (Geodesic distance). With the GL -invariant metric, the distance between $\mathbf{X}, \mathbf{Y} \in P_n$ is given by (see [32])

$$\text{dist}(\mathbf{X}, \mathbf{Y})^2 = \text{tr}(\log^2(\mathbf{X}^{-1} \mathbf{Y})) \quad (2.2)$$

where $\log$ is the matrix log operator. Since this distance is induced from the GL -invariant metric in Eq. 2.1, it is naturally GL -invariant i.e. $\text{dist}^2(\mathbf{X}, \mathbf{Y}) = \text{dist}^2(\mathbf{g}\mathbf{X}\mathbf{g}^t, \mathbf{g}\mathbf{Y}\mathbf{g}^t)$.

With GL -invariant metric defined on P_n , the intrinsic or Karcher mean of a set of elements $\mathbf{X}_i \in P_n$ can be computed by performing the following minimization:

$$\mu^* = \underset{\mu}{\text{argmin}} \sum_i \text{dist}^2(\mathbf{X}_i, \mu) \quad (2.3)$$

using a gradient based technique.

The Log and Exponential maps [13] are very useful tools on the Riemannian manifold. The Exponential map denoted as $\text{Exp}_{\mathbf{X}}(\cdot)$, where $\mathbf{X} \in P_n$, maps a vector rooted at the origin of the tangent space $T_{\mathbf{X}}P_n$ to a geodesic emanating from $\mathbf{X}$. The Log map ($\text{Log}_{\mathbf{X}}(\cdot)$) is the inverse of the Exponential map. The Exponential and Log map on P_n are given by:

$$\begin{aligned}\text{Exp}_{\mathbf{X}}(\mathbf{V}) &= \mathbf{X}^{1/2} \exp(\mathbf{X}^{-1/2} \mathbf{V} \mathbf{X}^{-1/2}) \mathbf{X}^{1/2} \\ \text{Log}_{\mathbf{X}}(\mathbf{Y}) &= \mathbf{X}^{1/2} \log(\mathbf{X}^{-1/2} \mathbf{Y} \mathbf{X}^{-1/2}) \mathbf{X}^{1/2}\end{aligned}\quad (2.4)$$

where $\mathbf{X}, \mathbf{Y} \in P_n$, $\mathbf{V} \in T_{\mathbf{X}}P_n$, and $\log$ and $\exp$ denote the matrix exp and log operators.

2.1.1. Geodesic, Euclidean and Log-Euclidean distances—To illustrate the differences between geodesic, Euclidean and Log-Euclidean distances on a Riemannian manifold, we have a simple example on S^2 – the unit 2-sphere depicted in Figure 2.1. Given two points $\mathbf{X}, \mathbf{Y} \in S^2$, the geodesic distance is the length of the shorter arc of the great circle between $\mathbf{X}$ and $\mathbf{Y}$ – the solid black line. The Euclidean distance between them is the length of the straight line between $\mathbf{X}$ and $\mathbf{Y}$ in the embedded 3D Euclidean space – the dashed grey line. Given another point $\mathbf{B} \in S^2$, we could project $\mathbf{X}, \mathbf{Y}$ to the tangent space at $\mathbf{B}$, $T_{\mathbf{B}}S^2$ by the Log map. In Figure 2.1, the points after projection are $\mathbf{U}, \mathbf{V}$. The length of the line $\mathbf{UV}$ (the dashed black line) can then be used as the distance between $\mathbf{X}, \mathbf{Y}$, which is called Log-Euclidean distance. Note that if $\mathbf{B} = \mathbf{X}$ or $\mathbf{B} = \mathbf{Y}$, the Log-Euclidean and geodesic distances are the same. Otherwise, they are different in Riemannian manifolds except for certain manifolds like Euclidean space.

The Euclidean and Log-Euclidean distances are extrinsic to the manifold i.e., depend on the embedding Euclidean space, and a predefined base point $\mathbf{B}$ for the Log-Euclidean distance. Therefore, they are often called extrinsic distances, while the geodesic distance which only depends on the manifold is called intrinsic distance. In [2], the base point $\mathbf{B}$ is fixed at the identity element of the space. However, so long as there is a predefined base point $\mathbf{B} = \mathbf{X}$ or $\mathbf{Y}$ used to compute the distance between $\mathbf{X}$ and $\mathbf{Y}$ as shown in Figure 2.1, the framework can still be viewed as Log-Euclidean or extrinsic, especially when this distance serves the purpose of a cost function being optimized. In [36], the estimation error is measured using a arbitrarily chosen base point $\mathbf{B}$, which makes it a Log-Euclidean method, even though the based point is not fixed.

In P_n , the intrinsic/geodesic distance is based on the GL -invariant metric as shown in Section 2.1. Euclidean and Log-Euclidean distances can also be viewed as being based on corresponding metrics which are invariant to $O(n)$. So, which metric should we choose for the covariance tracking problem? There are two primary reasons for the choice of a GL -invariant metric over the conventional Euclidean metric.

First of all, P_n is an open subset of the corresponding Euclidean space $\mathbb{R}^{(n+1)n/2}$, which implies that P_n would be incomplete with a Euclidean metric since it's possible to find a Cauchy sequence which might not converge for this case. This implies that for some of the optimization problems set in P_n , the optimum cannot be achieved inside P_n . This in turn means that the covariance updates could lead to matrices that are not covariance matrices, an unacceptable situation in practice. This problem will not arise when using the GL -invariant metric, since the space of P_n is geodesically complete with a GL -invariant metric [32].

Secondly, the feature vectors in general might contain components of different scales and from disparate sources, e.g. object position, color etc. In this case, a normalization of the (in general) unknown scales of different components would be necessary when using Euclidean distance, which is non trivial and may lead to use of ad hoc methods. However, with a GL invariant metric, this scaling issue does not arise since, the presence of different scales for

the elements of a feature vector from which the covariance matrix is constructed, is equivalent to multiplication of the covariance matrix with a positive definite diagonal matrix. This operation is a GL group operation and since GL -invariance implies invariance to GL group operations, this scaling issue is a non issue when using a GL -invariant metric.

The Log-Euclidean metric can be viewed as a linear approximation of the GL -invariant metric. This approximation has lower error when in a small neighbourhood of the predefined base point. But when computed in a large region around the base/anchor point, it will suffer from high approximation error as shown in our variance computation in Section 2.2.3 and also the result in the synthetic experiment in Section 4.1, which would accumulate with each frame and affect the tracking result especially for noisy data. Also, the Log-Euclidean metric as an approximation doesn't preserve some of the good properties of GL -invariant metric such as the scale invariance as mentioned above.

GL -invariant metric on the other hand is more difficult to compute. Many GL -invariant computations do not have closed form solutions and thus less efficient to compute. However, the proposed filter in this paper could be computed with the similar computational effort as the filter in [36] but is more accurate for larger amounts of noise.

2.2. Invariant Measure & Generalized Normal Distribution on P_n

2.2.1. Probability Measures on P_n —To define a probability distribution on a manifold, first we need to define a measure on the manifold. In this paper, we use the GL -invariant measure $[d\mathbf{X}]$ on P_n . GL invariance here implies $\mathbf{g} \in GL(n)$ and $\mathbf{X} \in P_n$, $[d\mathbf{g}\mathbf{X}\mathbf{g}^t] = [d\mathbf{X}]$. From [32], we know that $[d\mathbf{X}] = |\mathbf{X}|^{-(n+1)/2} \prod_{i,j=1}^n dx_{ij}$, where x_{ij} is the element in the i -th row and j -th column of the SPD matrix $\mathbf{X}$. Also, this measure is consistent with the GL invariant metric defined on P_n defined earlier and also presented in [23].

Similar to the Karcher mean, the Karcher Expectation for the random variable $\mathbf{X}$ on any Riemannian manifold M can be defined as the result of the following minimization problem,

$$\bar{\mathbf{X}} = E(\mathbf{X}) = \operatorname{argmin}_{\mathbf{Y} \in M} \int_M \operatorname{dist}^2(\mathbf{X}, \mathbf{Y}) d\mu(\mathbf{X}) \quad (2.5)$$

where $\mathbf{X}$ denotes the expectation of random variable $\mathbf{X}$ and $\mu(\mathbf{X})$ is the probability measure defined on M . Similarly, the variance can be defined based on this expectation by,

$$\operatorname{Var}(\mathbf{X}) = \int_M \operatorname{dist}^2(\mathbf{X}, \bar{\mathbf{X}}) d\mu(\mathbf{X}) \quad (2.6)$$

Note that, in Euclidean space R^m which is also a Riemannian manifold, the Karcher Expectation is equivalent to the traditional definition of expectation, and the variance in Eq. 2.6 is the trace of the covariance matrix. In P_n , by taking gradient of the energy function in Eq. 2.5 and setting it to zero, we find that the expectation of the random variable will satisfy the following equation.

$$\int_{P_n} \log(\bar{\mathbf{X}}^{-1/2} \mathbf{X} \bar{\mathbf{X}}^{-1/2}) p(\mathbf{X}) [d\mathbf{X}] = 0 \quad (2.7)$$

2.2.2. Generalized Normal Distribution on P_n —The generalization of the normal distribution to P_n used in this paper is defined as follows:

$$dP(\mathbf{X}; \mathbf{M}, \omega^2) = p(\mathbf{X}; \mathbf{M}, \omega^2) [d\mathbf{X}] = \frac{1}{Z} \exp\left(-\frac{\text{dist}(\mathbf{X}, \mathbf{M})^2}{2\omega^2}\right) [d\mathbf{X}] \quad (2.8)$$

where $P(\cdot)$ and $p(\cdot)$ are the probability distribution and density respectively of the matrix-variate random variable $\mathbf{X} \in P_n$, with two parameters $\mathbf{M} \in P_n$ and $\omega^2 \in \mathbb{R}^+$, and Z is the scalar normalization factor. $\text{dist}(\cdot)$ is defined in Eq. 2.2. As shown in [23], this distribution has minimum information given the Karcher mean and variance. That is, in the absence of any information this distribution would be the best possible assumption from an information theoretic view point. Also, this distribution is different from the Log-normal distribution which was used in [36, 27]. Actually, the two distributions have very similar densities, but the density used in this paper is based on GL -invariant measure while Log-normal density is based on the Lebesgue measure in Euclidean space.

A very important property of the above generalized normal distribution is summarized in the following theorem, the proof for which is given in Appendix A.

Theorem 2.1: The normalization factor Z in Eq. 2.8 is a finite constant with respect to parameter $\mathbf{M} \in P_n$

The consequence of Theorem 2.1 is that, if the prior and the likelihood are both based on the generalized normal distribution defined using the GL -invariant measure, computing the mode of the posterior density can be achieved by minimizing the sum of squared GL -invariant distances from the unknown expectation of the given samples.

One direct consequence of Theorem 2.1 is the following corollary, whose proof is given in Appendix A.

Corollary 2.2: Given a set of i.i.d samples $\{\mathbf{X}_i\}$ drawn from the distribution $dP(\mathbf{X}; \mathbf{M}, \omega^2)$, the MLE of the parameter $\mathbf{M}$ is the Karcher mean of all samples.

From theorem 2.1 we know that the normalization factor Z in Eq. 2.8 is a function of ω^2 . The integral in Eq A.2 is non-trivial, and currently no exact solution is available for arbitrary n . For $n = 2$ we have,

$$\begin{aligned} Z_2(\omega) &= 2c_2 \int_{-\infty}^{\infty} \int_{-\infty}^{y_1} \exp\left(-\sum_{i=1}^2 \left(\frac{1}{2\omega^2} y_i^2 + \frac{1}{2} y_i\right)\right) (\exp(y_1) - \exp(y_2)) dy_2 dy_1 \\ &= \sqrt{2\pi} c_2 \omega \exp\left(\frac{1}{4}\omega^2\right) \int_{-\infty}^{\infty} \left(\exp\left(-\frac{(y_1 - 0.5\omega^2)^2}{2\omega^2}\right) (1 + \text{erf}\left(\frac{y_1 + 0.5\omega^2}{\sqrt{2\omega^2}}\right)) \right. \\ &\quad \left. - \exp\left(-\frac{(y_1 + 0.5\omega^2)^2}{2\omega^2}\right) (1 + \text{erf}\left(\frac{y_1 - 0.5\omega^2}{\sqrt{2\omega^2}}\right))\right) dy_1 \\ &= 4\pi c_2 \omega^2 \exp\left(\frac{1}{4}\omega^2\right) \text{erf}\left(\frac{\omega}{2}\right) \end{aligned} \quad (2.9)$$

where $\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2) dt$ is the error function.

2.2.3. Mean and the Variance of the Generalized Normal Distribution—Similar to the normal distribution in Euclidean space, the mean and variance of the generalized normal distribution on P_n in Eq. 2.8 are controlled by the parameters $\mathbf{M}$ and ω^2 respectively. The relation between $\mathbf{M}$ and $dP(\mathbf{X}; \mathbf{M}, \omega^2)$ is given by the following theorem, the proof for which is given in Appendix B.

Theorem 2.3: $\mathbf{M}$ is the Karcher Expectation of the generalized normal distribution $dP(\mathbf{X}; \mathbf{M}, \omega^2)$.

The variance of $dP(\mathbf{X}; \mathbf{M}, \omega^2)$ is controlled by the parameter ω^2 . Unlike the multivariate normal distribution in Euclidean space, where the Karcher variance (Eq. 2.6) is equal to $n \omega^2$, the relation between the variance and ω^2 of the generalized normal distribution is much more complex. Without loss of generality we assume $\mathbf{X} \in P_n$ is a matrix variate random

variable from $dP(\mathbf{X}; \mathbf{I}, \omega^2)$. The variance $Var(\mathbf{X}) = \frac{1}{Z} \int_{P_n} \|\log(\mathbf{X})\|^2 \exp(-\frac{\|\log(\mathbf{X})\|^2}{2\omega^2}) [d\mathbf{X}]$. As in Eq. A.2, by using the Polar coordinates and taking log of the eigen values, we can get

$$Var(\mathbf{X}) = \omega^2 Var_q(\mathbf{y}) \quad (2.10)$$

where $\mathbf{y}$ is a random vector in R^n having a distribution with density function,

$$q(\mathbf{y}) = \frac{1}{z(\omega)} \exp(-\frac{1}{2} \sum_i y_i^2) \prod_{1 \leq i < j \leq n} 2 |\sinh(\frac{\omega(y_i - y_j)}{2})|, \quad (2.11)$$

where $z(\omega)$ is the normalization factor. Currently, there are no analytic solutions for $Var_q(\mathbf{y})$ for arbitrary n . When $n = 2$ we can compute $Var(\mathbf{y})$ using the similar technique as in Eq. 2.9

$$Var_q(\mathbf{y}) = \frac{\omega}{\sqrt{\pi} \exp(\frac{1}{4}\omega^2) \operatorname{erf}(\frac{\omega}{2})} + 2(1 + \frac{\omega^2}{4}) \quad (2.12)$$

From Eq. 2.12 we can find that, in P_2 when ω is close to zero, $Var(\mathbf{X}) \approx 3 \omega^2$, and when ω is large $Var(\mathbf{X}) \approx \frac{\omega^2}{2}$. This is because P_n can be locally approximated by an Euclidean space. When ω is close to zero, the major portion of the distribution would be in a small region in P_n , where Euclidean approximation is relatively accurate. Hence, $Var(\mathbf{X})$ is proportional to ω^2 which is similar to the normal distribution in Euclidean space. When ω^2 is not close to zero, Euclidean approximation is no longer accurate, and the $Var(\mathbf{X})$ becomes a complicated function of ω^2 . This property has been used to get the approximation of the generalized normal distribution with small ω^2 in [23, 16].

The following two theorems show that the above stated approximations will still be satisfied for $n > 2$, the proof of which is given in Appendix C and Appendix D respectively.

Theorem 2.4

$$\lim_{\omega \rightarrow 0} \frac{Var(\mathbf{X})}{\omega^2} = \frac{n(n+1)}{2} \quad (2.13)$$

Note that this theorem can also be obtained using the approximation of the generalized normal distribution with small ω^2 in [23, 16]. Furthermore, from the proof in Appendix C, we can deduce that since the Log-Normal is a projection of a normal distribution from the tangent space (can be identified with $Sym(n)$) to P_n , and here the random vector $\mathbf{y}$ is the normalized log of the eigen values of $\mathbf{X}$, we can see that *when ω is close to zero the generalized normal distribution can be approximated by a Log-Normal distribution.*

Theorem 2.5

$$\lim_{\omega \rightarrow \infty} \frac{Var(\mathbf{X})}{\omega^4} = \frac{(n^3 - n)}{12} \quad (2.14)$$

2.3. The Probabilistic Dynamic Model on P_n

To perform tracking on P_n , obviously the observation $\mathbf{Y}_k$ at frame k lies on P_n . Also, we can define the state to lie on P_n i.e. $\mathbf{X}_k \in P_n$. The state transition model and the observation model can then be defined as

$$p(\mathbf{X}_k | \mathbf{X}_{k-1}) = \frac{1}{Z_s} \exp\left(-\frac{dist^2(\mathbf{X}_k, \mathbf{g}\mathbf{X}_{k-1}\mathbf{g}^t)}{2\omega^2}\right) \quad (2.15)$$

$$p(\mathbf{Y}_k | \mathbf{X}_k) = \frac{1}{Z_o} \exp\left(-\frac{dist^2(\mathbf{Y}_k, \mathbf{h}\mathbf{X}_k\mathbf{h}^t)}{2\varphi^2}\right) \quad (2.16)$$

where $\mathbf{g}, \mathbf{h} \in GL(n)$, $\omega^2, \varphi^2 > 0$ are the parameters that control the variance of the state transition and the observation noise. The above two densities are both based on the GL -invariant measure on P_n unlike in [36, 27] where they are based on the Lebesgue measure. The key implication of this is that the normalization factor in the densities is a constant for the GL -invariant measure and not so for the Lebesgue measure case. If the normalization factor is not a constant, one does not have a valid density.

3. IRF-based Tracking Algorithm on P_n **3.1. The Bayesian Tracking Framework**

For simplicity, we use the Bayesian tracking framework described in [7] here. The tracking problem can be viewed as, given a time sequence of observations $\underline{\mathbf{Y}}_s = \{\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_s\}$ from time 1 to time s , how can one compute the state $\mathbf{X}_s$ at time s ? To solve this problem, first we make two assumptions: (1) The state transition is Markovian, i.e., the state $\mathbf{X}_s$ depends only on $\mathbf{X}_{s-1}$, or say

$$p(\mathbf{X}_s | \underline{\mathbf{X}}_{s-1}, \underline{\mathbf{Y}}_{s-1}) = p(\mathbf{X}_s | \mathbf{X}_{s-1}) \quad (3.1)$$

(2) The observation $\mathbf{Y}_s$ is dependent only on the state $\mathbf{X}_s$ at the current time point s , in other words,

$$p(\mathbf{Y}_s | \underline{\mathbf{X}}_s, \underline{\mathbf{Y}}_{s-1}) = p(\mathbf{Y}_s | \mathbf{X}_s) \quad (3.2)$$

And hence, $p(\mathbf{X}_s | \mathbf{X}_{s-1})$ is called the state transition model and $p(\mathbf{Y}_s | \mathbf{X}_s)$ is called the observation model.

The goal of tracking can thus be viewed as computing the posterior $p(\underline{\mathbf{X}}_s | \underline{\mathbf{Y}}_s)$. First we have,

$$p(\underline{\mathbf{X}}_s | \underline{\mathbf{Y}}_s) = p(\underline{\mathbf{X}}_s, \underline{\mathbf{Y}}_s) / p(\underline{\mathbf{Y}}_s) \propto p(\underline{\mathbf{X}}_s, \underline{\mathbf{Y}}_s). \quad (3.3)$$

And also,

$$\begin{aligned}
p(\underline{\mathbf{X}}_s, \underline{\mathbf{Y}}_s) &= p(\mathbf{Y}_s | \mathbf{X}_s) p(\mathbf{X}_s | \mathbf{X}_{s-1}) p(\mathbf{X}_{s-1}, \underline{\mathbf{Y}}_{s-1}) \\
&= \prod_{k=1}^s p(\mathbf{Y}_k | \mathbf{X}_k) p(\mathbf{X}_k | \mathbf{X}_{k-1}).
\end{aligned}$$

By using a moving window approximation and set the window width to be 2, we can then compute $\mathbf{X}_k, \mathbf{X}_{k-1}$ by solving the following optimization problem:

$$\begin{aligned}
\hat{\mathbf{X}}_k, \hat{\mathbf{X}}_{k-1} &= \underset{\mathbf{X}_k, \mathbf{X}_{k-1}}{\operatorname{argmax}} \prod_{j=k-1}^k p(\mathbf{Y}_j | \mathbf{X}_j) p(\mathbf{X}_j | \mathbf{X}_{j-1}) \\
&= \underset{\mathbf{X}_k, \mathbf{X}_{k-1}}{\operatorname{argmin}} E_k(\mathbf{X}_k, \mathbf{X}_{k-1})
\end{aligned}$$

where

$$\begin{aligned}
E_k(\mathbf{X}_k, \mathbf{X}_{k-1}) &= \varphi^{-2} \operatorname{dist}^2(\mathbf{h}^{-1} \mathbf{Y}_k \mathbf{h}^{-t}, \mathbf{X}_k) + \omega^{-2} \operatorname{dist}^2(\mathbf{g} \mathbf{X}_{k-1} \mathbf{g}^t, \mathbf{X}_k) + \\
&\quad \varphi^{-2} \operatorname{dist}^2(\mathbf{h}^{-1} \mathbf{Y}_{k-1} \mathbf{h}^{-t}, \mathbf{X}_{k-1}) + \omega^{-2} \operatorname{dist}^2(\mathbf{X}_{k-1}, \mathbf{g} \mathbf{X}_{k-2} \mathbf{g}^t)
\end{aligned} \quad (3.4)$$

Thus E_k is the energy function we would like to optimize at each frame k . Upon a closer look, we get the following theorem on the geodesic convexity of E_k and whose proof is given in Appendix E.

Theorem 3.1—The energy function E_k in Eq. 3.4 is geodesically convex on the Riemannian manifold $P_n \times P_n$ where $\times$ denotes the Cartesian product.

A function $f: P_n \mapsto R$ is geodesically convex implies, for all geodesics on $P_n: [0, 1] \mapsto P_n$ the composition $f \circ \gamma$ is convex [37]. Geodesic convexity is an extension of the standard notion of convexity to the Riemannian manifold. It is not hard to show both convexities share most of the properties such as local minimum is equivalent to global minimum etc. More details on the properties can be found in [37].

By taking the gradient of E_k with respect to $\mathbf{X}_k$, we can find that at the local minimum Eq. 3.5 will be satisfied.

$$\varphi^{-2} \operatorname{Log}_{\mathbf{X}_k}(\mathbf{h}^{-1} \mathbf{Y}_k \mathbf{h}^{-t}) + \omega^{-2} \operatorname{Log}_{\mathbf{X}_k}(\mathbf{g} \mathbf{X}_{k-1} \mathbf{g}^t) = 0 \quad (3.5)$$

This means when reaching the optimum, $\mathbf{X}_k$ will be on the geodesic of $\mathbf{h}^{-1} \mathbf{Y}_k \mathbf{h}^{-t}$ and $\mathbf{g} \mathbf{X}_{k-1} \mathbf{g}^t$, and $\frac{\operatorname{dist}(\mathbf{h}^{-1} \mathbf{Y}_k \mathbf{h}^{-t}, \mathbf{X}_k)}{\operatorname{dist}(\mathbf{g} \mathbf{X}_{k-1} \mathbf{g}^t, \mathbf{X}_k)} = \frac{\varphi^2}{\omega^2}$. Thus at the optimum point we will have

$$\begin{aligned}
&\varphi^{-2} \operatorname{dist}^2(\mathbf{h}^{-1} \mathbf{Y}_k \mathbf{h}^{-t}, \mathbf{X}_k) + \omega^{-2} \operatorname{dist}^2(\mathbf{g} \mathbf{X}_{k-1} \mathbf{g}^t, \mathbf{X}_k) \\
&= \frac{1}{\varphi^2 + \omega^2} \operatorname{dist}^2(\mathbf{h}^{-1} \mathbf{Y}_k \mathbf{h}^{-t}, \mathbf{g} \mathbf{X}_{k-1} \mathbf{g}^t)
\end{aligned} \quad (3.6)$$

Combining Eq. 3.6 and Eq. 3.4, we can get

$$\begin{aligned}
E'_k(\mathbf{X}_{k-1}) &= \frac{1}{\varphi^2 + \omega^2} \operatorname{dist}^2(\mathbf{h}^{-1} \mathbf{Y}_k \mathbf{h}^{-t}, \mathbf{g} \mathbf{X}_{k-1} \mathbf{g}^t) \\
&+ \varphi^2 \operatorname{dist}^2(\mathbf{h}^{-1} \mathbf{Y}_{k-1} \mathbf{h}^{-t}, \mathbf{X}_{k-1}) + \omega^{-2} \operatorname{dist}^2(\mathbf{X}_{k-1}, \mathbf{g} \mathbf{X}_{k-2} \mathbf{g}^t)
\end{aligned} \quad (3.7)$$

It is obvious that E'_k and E_k have the same optimum. Instead of minimizing E_k we can minimize E'_k which is a weighted Karcher mean of 3 points on P_n . The classical gradient decent algorithm on P_n [21] can efficiently solve this problem.

After getting the optimal $\mathbf{X}_{k-1}^*$, $\mathbf{X}_k^*$ can be computed in a closed form.

$$\mathbf{X}_k^* = (\mathbf{h}^{-1} \mathbf{Y}_k \mathbf{h}^{-t})^{1/2} [(\mathbf{h}^{-1} \mathbf{Y}_k \mathbf{h}^{-t})^{-1/2} \mathbf{g} \mathbf{X}_{k-1}^* \mathbf{g}^t (\mathbf{h}^{-1} \mathbf{Y}_k \mathbf{h}^{-t})^{-1/2}] \frac{\varphi^2}{\varphi^2 + \omega^2} (\mathbf{h}^{-1} \mathbf{Y}_k \mathbf{h}^{-t})^{1/2} \quad (3.8)$$

It is easy to show that, the state update here is an estimation of the mode of the posterior $p(\mathbf{X}_s | \mathbf{Y}_s)$, which is different from the usual Kalman filter and particle filter methods, where the state update is the mean of the posterior $p(\mathbf{X}_s | \mathbf{Y}_s)$. In the proposed update process, the covariance of the posterior is not necessary for updating the state. We do not provide an update of the covariance here, partly because the covariance update is hard to compute for this distribution on P_n . Actually, there is no existing closed form solution for the covariance

matrices even for the distribution $p(\mathbf{X}_k | \mathbf{X}_{k-1}) = \frac{1}{Z_s} \exp(-\frac{\text{dist}^2(\mathbf{X}_k, \mathbf{g} \mathbf{X}_{k-1} \mathbf{g}^t)}{2\omega^2})$. In our future work, we will focus on developing an efficient and convergent covariance updating mechanism in this framework.

3.2. The Tracking Algorithm

The intrinsic recursive filter (IRF) for covariance matrices (descriptors) on P_n presented above can be used in combination with many existing tracking techniques. Many algorithms based on covariance descriptors like those in [26, 36] can use our IRF as the model updating method for covariance descriptors. In this paper we combine the IRF with an existing particle position tracker [3] and get a real-time video tracking algorithm.

3.2.1. Feature Extraction—Assume we have an rectangular region R with width W and height H which represents the target object in a certain image I in the video sequence. The feature vector $\mathbf{f}(x, y)$, where $(x, y) \in R$, can be extracted to include the information of appearance, position, etc. to describe information at the point (x, y) . In [26], the feature vector was chosen to be $\mathbf{f} = [x, y, I(x, y), |I_x(x, y)|, |I_y(x, y)|]$ where I_x and I_y are the components of the gradient ∇I . For color images, $\mathbf{f}(x, y) = [R, G, B]$ is a vector. With the feature vectors at each point in the region of the object, the covariance matrix can be

computed as $C_R = \frac{1}{WH} \sum_{k \in R} (\mathbf{f}_k - \mu_R) (\mathbf{f}_k - \mu_R)^t$. This covariance matrix can be computed in constant time with respect to the size of the region R by using the technique called the integral image as was done in [34]. We can also add the mean μ_R in to the covariance descriptor and still obtain a symmetric positive definite matrix in the following manner,

$$\widehat{C}_R = \begin{bmatrix} C_R + \lambda^2 \mu \mu^t & \lambda \mu \\ \lambda \mu^t & 1 \end{bmatrix} \quad (3.9)$$

where λ is a parameter used to balance the affect of the mean and variance in the descriptor (in this paper $\lambda = 0.001$).

As in [34], we use several covariance descriptors for each object in the scene. Very briefly, each region enclosing an object is divided into 5 regions and in each of these, a covariance descriptor is computed and tracked individually. A matching score (likelihood) is computed using 4 of them with relatively small distance to the corresponding template in the template

matching stage described below. This approach is used in order to increase the robustness of our algorithm.

3.2.2. Tracking and Template Matching—We use a sampling importance re-sampling (SIR) particle filter [3] as a position and velocity tracker. The state vector of the particle filter at the k -th frame is now given by, $\mathbf{u}_k = (x_k, y_k, v_{x,k}, v_{y,k}, \log(s_k))t$, where $x_k, y_k, v_{x,k}, v_{y,k}$ denote the position and velocity of the object in the 2D image, and $\log(s_k)$ is the log of the scale. The state transition is defined by the equation below:

$$\mathbf{u}_k = \mathbf{F}\mathbf{u}_{k-1} + \mathbf{n} \quad (3.10)$$

where the state transition matrix $\mathbf{F}$ is defined based on Newton's first law

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 0.01 & 0 & 0 \\ 0 & 1 & 0 & 0.01 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.11)$$

The additive noise $\mathbf{n}$ is normally distributed with the covariance matrix assumed to be a diagonal matrix and in our work reported here, it was set to, $(4^2, 4^2, 20^2, 20^2, 0.015^2)$. These state transition parameters are dependent on the videos being tracked. They could also be learned from the manually labelled training sets.

At the k -th frame, the likelihood for the particle filter is based on the generalized normal distribution as discussed in Section 2.2.

$$p(I_k | \mathbf{u}_k) = \frac{1}{Z} \exp\left(-\frac{\text{dist}(\mathbf{Y}_k, \mathbf{h}\mathbf{X}_k\mathbf{h}^t)^2}{2\varphi^2}\right) \quad (3.12)$$

where $\mathbf{Y}_k$ is the covariance descriptor extracted from the image I_k based on the state vector $\mathbf{u}_k$ as described in Section 3.2.1. $\mathbf{X}_k$ is the covariance template updated using our IRF.

The classical particle filter updates the weight for the sample using the following equation:

$$w_k^i \propto w_{k-1}^i \frac{p(I_k | \mathbf{u}_k^i) p(\mathbf{u}_k^i | \mathbf{u}_{k-1}^i)}{q(\mathbf{u}_k^i | \mathbf{u}_{k-1}^i, I_k)} \quad (3.13)$$

where w_k^i is the weight for the i -th sample at frame k , q is the importance sampling density. Since we are using sampling importance re-sampling (SIR), $q(\mathbf{u}_k | \mathbf{u}_{k-1}, I_k) = p(\mathbf{u}_k | \mathbf{u}_{k-1})$, and the weight update equation becomes

$$w_k^i \propto w_{k-1}^i p(I_k | \mathbf{u}_k^i) \quad (3.14)$$

At step k , we first compute the prediction of the object covariance template using $\hat{\mathbf{Y}}_k = \mathbf{h}\mathbf{g}\mathbf{X}_{k-1}\mathbf{g}^t\mathbf{h}^t$, and the prediction of the position & scale of the object represented in the set of particles based on the state transition matrix 3.11. Covariance descriptors then are computed for each of the predicted particle state at the corresponding object regions. The likelihood for each covariance descriptor is computed based on the generalized normal distribution centred at the predicted corresponding covariance template. And the likelihood for each particle's

state is computed as multiplication of the likelihoods of covariance descriptors that are closer to their corresponding template as mentioned above. After multiplying the likelihood with the weight of each particle, the mean of the sample set is computed. This is followed by computation of covariance descriptors at the location of the mean of the particle set. This covariance descriptor then forms the input to our IRF. The algorithm for each frame is given in the text box below.

Tracking Algorithm

- Step 1. Resample and draw samples following Eq. 3.10.
- Step 2. Predict the covariance template at the current step using Eq. 2.15.
- Step 3. Extract covariance descriptor using methods in Section 3.2.1 for each particle and then update each particle weight using Eq. 3.14.
- Step 4. Compute the weighted mean $\mathbf{u}_k$ of all the particles.
- Step 5. Extract the covariance descriptor at $\mathbf{u}_k$ and use as observation $\mathbf{Y}_k$ to optimize the energy function in Eq. 3.7, and then update the covariance template $\mathbf{X}_k$ using Eq. 3.8.

In our paper, we use 300 particles for the particle set. Our tracking algorithm runs at around 15Hz for videos with a frame size of 352×288 , on a desktop with a 2.8GHz CPU.

4. Experiments

In this section, we present the results of applying our intrinsic recursive filter to both synthetic and real data sets. The real data sets were taken from standard video sequences used in the computer vision community for testing tracking algorithms. First we present the synthetic data experiments and then the real data.

4.1. Synthetic Data Experiments

To validate the proposed filtering technique, we first performed synthetic data experiments on P_3 , the space of 3×3 SPD matrices. A time sequence of i.i.d samples of SPD matrices were randomly drawn from the Log-normal distribution [27] centred at the identity matrix. This was done by first drawing samples $\{\mathbf{v}_i\}$ in $\mathbb{R}^6$ (isomorphic to $Sym(3)$) from the Normal distribution $\mathcal{N}(0, \sqrt{2}\mathbf{I}_6)$. Then, these samples are projected to P_3 (denoted by $\{\mathbf{X}_i\}$) using the exponential map at the point $\mathbf{I}_3$ (identity matrix). Thus, $\{\mathbf{X}_i\}$ can be viewed as a time sequence of random measurements of the identity matrix. Our recursive filter can then be used as an estimator of this random process. The estimation error at time point k can be computed as the Riemannian distance by Eq. 2.2 between the estimate $\mathbf{X}_k$ and the ground truth (the identity matrix).

We tested our recursive filter (IRF) and evaluated by comparing its performance with the recursive filter for linear systems on P_n (denoted by LRF) reported in [36]. The parameters of LRF were set to be exactly the same as was presented in [36] except for the initial base point $\mathbf{X}_b$, where all the samples are projected to the tangent space $T_{\mathbf{X}_b}P_n$ and then processed with LRF. We set $\mathbf{X}_b$ to be the observation in the first step. In this problem, setting $\mathbf{X}_b$ to be the ground truth would give the best result for LRF, because in this case LRF would reduce to the Kalman filter on the tangent space. Since in the practical case, the ground truth is unknown, here we set $\mathbf{X}_b$ as the observation at the first step which is the best information we know about the data sequence before tracking. We also tried to randomly set $\mathbf{X}_b$, and this did not lead to any observable differences. For the proposed method, the GL actions $\mathbf{g}, \mathbf{h}$ were both set to be the identity, and $\sqrt{2}/\sqrt{2} = 200$. We performed experiments with three different noise levels, $\sigma^2 = 0.1, 1$ and 2 . At each noise level we execute the whole process 20 times and computed the mean error for the corresponding time step.

The results are summarized in Figure 4.1. From the figure, we can see that the LRF performs better when $\sigma^2 = 0.1$, and our method (IRF) performs better when the data is more noisy ($\sigma^2 = 1, 2$). The reason is that LRF uses several Log-Euclidean operations, which is in fact an approximation. For low noise level data, the data points are in a relatively small region around the ground truth (Identity), in which case, the Log-Euclidean approximation is quite accurate. But for higher noise levels in the data, the region becomes larger and the approximation becomes inaccurate which leads to large estimation errors. In contrast, our filtering method is fully based on the Riemannian geometry without any Log-Euclidean approximation, so it performs consistently, and correctly converges for all the three noise levels. In conclusion, although our recursive filter might converge a little bit slower than the LRF, it is more robust to larger amounts noise which is common in real tracking situations.

In the synthetic experiments, IRF takes on average 0.86 seconds for each sequence (1000 samples) which is slower than LRF (0.52 seconds per sequence). This is not significant since, in the video tracking program, most time consuming part is the likelihood computation which usually takes 10 to 100 times more than the state update time.

4.2. Real Data Experiments

For the real data experiment, we applied our IRF to more than 3000 frames in different video sequences. Two other covariance descriptor updating methods were also applied to these sequences for comparison namely, (1) the LRF reported in [36]; (2) The updating method using Karcher mean of tracking results in previous frames (KM) reported in [26]. The image feature vectors for the target region were computed as reported in [26]. The buffer size T in the KM method were set to 20 which means the Karcher mean of covariance descriptors in 20 previous frames were used for the prediction of the covariance descriptor in current frame. The parameters for the LRF were set to values given in [36]. The parameter controlling the state transition and observation noise in our IRF are set to $\sigma^2 = 0.0001$ and $\sigma^2 = 0.01$. Since our IRF is combined with a particle filter as a position tracker, for the purpose of comparisons, the KM and LRF are also combined with exactly the same particle filter-based position tracker.

Firstly, we used three video sequences from the dataset CAVIAR [1]: 1. ThreePastShop1cor(C3ps1); 2. ThreePastShop2cor(C3ps2); 3. OneShopOneWait2cor(Cosow2). All three sequences are from a fixed camera and a frame size of 384×288 . 7 “objects” were tracked separately. Given ground truth was used to quantitatively evaluate the tracking results. To measure the error for the tracking results, we used the distance between the center of the estimated region and the ground truth. With all the three methods having the same initialization, the tracking results are shown in the Table 4.1, where all the errors shown are the average errors over all the tracked frames. From the table we can see that LRF is more accurate than KM based methods in most of the results, and our IRF outperforms both these methods. The KM drifts away from the target because it is based on a sliding window approach. If the number of consecutive noisy frames is close to the window size, the tracker will tend to track the noisy features. For LRF, since it is a non-intrinsic approach, the approximation of the GL -invariant metric would introduce errors that accumulate over time across the frames causing it to drift away. Since IRF is an intrinsic recursive filter, which uses the GL -invariant metric, there is less error introduced in the covariance tracker updates. This in turn leads to higher accuracy in the experiments above.

In the second experiment, we performed head tracking in video sequences with a moving camera. Two video sequences were used: (i) Seq_mb sequence (tracking face) and (ii) Seq_sb. Each of the sequences have 500 frames with frame size 96×128 . Both sequences are challenging because of complex background, fast appearance changes and occlusions. The results are summarized in Figure 4.2.

In Seq_mb, KM fails at frame 450 where the occlusion occurs, while LRF and IRF do not lose track. Both KM and LRF produce relatively large errors in capturing the position of the girl's face after the girl turns around the first time between frames 100 to 150 due to the complete change in appearance of the target (girl's face). LRF produces relatively larger error in estimating the scale (compared to the initialization) of the face between frames 400 to 500, which can be found in the snap shots included in figure 4.2. The result of our method (IRF) has relatively larger error at around frame 100 and 180, because at this time, the camera is tracking the hair of the girl where no feature can be used to locate the position of the face. However, for other frames, IRF tracks the face quite accurately.

In Seq_sb both KM and LRF fail at frame 200, and IRF however successfully tracked the whole sequence with relatively high accuracy even with fast appearance changes and occlusions, as shown in the quantitative analysis figure in 4.2. These experiments thus demonstrate the accuracy of our method in both moving camera and fixed camera cases.

For the speed of the tracking algorithm, since we only extract features from a window around the target object, the resolution of the video doesn't really affect the tracking speed while the size of the target object does. Here we compute the average processing time per frame for all sequences. IRF on the average takes 0.027 seconds to process each frame which is faster than KM (0.035 seconds), and LRF is the fastest of all three methods which takes 0.0067 seconds. One main reason is, LRF is not a particle filter, so it needs much less time in computing the likelihood. The state update only takes 0.0012 seconds in IRF. Also, IRF is still a real time filter (on average more than 30 fps), and yields more accurate results especially in high noise cases.

5. Discussion and Conclusion

In this paper, we presented a novel intrinsic recursive filter (IRF) for covariance tracking, which proved to be more robust to noise than existing methods reported in literature. IRF is based on the intrinsic geometry of the space of covariance matrices and a GL -invariant metric that are used in developing the dynamic model and the recursive filter.

We presented a generalization of the normal distribution to P_n and used it to model the system and the observation noise in the IRF. Several properties of this distribution in P_n were also presented in this paper, which to the best of our knowledge have never been addressed in the literature. Note that our generalization of the normal distribution to P_n is rotationally invariant, and the variance of the distribution is controlled by a scalar (σ^2 in Eq. 2.8) rather than a variance control tensor which is a more general form. One main reason for using this specific form is that the scalar variance control parameter is GL -invariant, while the variance control tensor is not as shown through the following simple calculation. Suppose $\mathbf{V} \in T_{\mathbf{X}}P_n$ is a tangent vector (which is a symmetric matrix) at point $\mathbf{X} \in P_n$, and Σ is a variance control tensor. The value of the density function on $\exp_{\mathbf{X}}(\mathbf{V})$ would depend upon the quadratic form $\text{vec}(\mathbf{V})^t \Sigma^{-1} \text{vec}(\mathbf{V})$ where $\text{vec}(\cdot)$ is the vectorization operation on the matrix argument and Σ is a second order tensor. In practice $\mathbf{X}$ would be the Karcher expectation of the aforementioned distribution and $\mathbf{V}$ would be the tangent vector corresponding to the geodesic from $\mathbf{X}$ to a sample point from the distribution. If we change the coordinates by using a GL operation $\mathbf{g}$, the Karcher expectation becomes $\mathbf{g}\mathbf{X}\mathbf{g}^t$, the vector becomes $\mathbf{g}\mathbf{V}\mathbf{g}^t$, and the quadratic form becomes $\text{vec}(\mathbf{g}\mathbf{V}\mathbf{g}^t)^t \Sigma^{-1} \text{vec}(\mathbf{g}\mathbf{V}\mathbf{g}^t)$. If we want to keep the value of the density unchanged, we need to change Σ according to $\mathbf{g}$ which means that Σ is not GL -invariant. However, in contrast, it is easy to show that σ^2 in Eq. 2.8 is GL -invariant.

Further, the IRF is quite different from the Kalman filter which is known to be an optimal linear filter (in a vector space) based on an additive Gaussian noise assumption. One reason for the Kalman filter to be optimal is that it actually tracks the distribution of the object state (posterior) based on a Bayesian tracking framework. If a filter doesn't track the whole distribution, usually it would explicitly or implicitly approximate the posterior based on the state variables it has tracked. However, the approximation error might accumulate in the system. From a geometric point of view, Kalman filter is highly dependent on geometric properties of the Euclidean space. This is because Kalman filter is based on the fact that the convolution of two Gaussians is a Gaussian. And this property of the Gaussian stems from the fact that the Gaussian is the limit distribution in the central limit theorem. One key problem in extending the Kalman filter intrinsically to P_n is finding two densities $p_A(\mathbf{X}; \theta_A)$, $p_B(\mathbf{X}|\mathbf{Y})$ with following properties

$$p_A(\mathbf{X}; \theta'_A) = \int_{P_n} p_B(\mathbf{X}|\mathbf{Y}) p_A(\mathbf{Y}; \theta) [d\mathbf{Y}] \quad (5.1)$$

where θ_A is the parameter of density p_A . p_A here is usually the posterior and p_B is the state transition noise distribution. The equation above means that after the state transition the form of the posterior remains the same. Without this property, even if the whole distribution is tracked, the filter is implicitly approximating the true distribution after the state transition by using the same form as the posterior from the last step which still would lead to errors being accumulated in the system. However, it is non-trivial to find such distributions on P_n . In [32, 12], a central limit theorem was presented in P_n for rotationally invariant probability measures based on the Helgason-Fourier transform [13]. However, currently the probability measure in the limit does not have a closed form in the space domain. Thus, intrinsically extending the Kalman filter to P_n is still an open problem. IRF instead only tracks the mode of the distribution. It is not an optimal filter, but is *intrinsic and mathematically consistent* with respect to the noise model used, unlike the LRF in [36]. We also presented a real time covariance tracking algorithm based on this filter which is combined with an existing particle position tracker from literature [3]. Finally, experiments on synthetic and real data favourably demonstrated the accuracy of our method over rival methods.

References

1. CAVIAR: Context aware vision using image-based active recognition. <http://homepages.inf.ed.ac.uk/rbf/CAVIAR/>
2. Arsigny V, Fillard P, Pennec X, Ayache N. Fast and simple calculus on tensors in the log-euclidean framework. MICCAI. 2005; 3749:115–122. [PubMed: 16685836]
3. Arulampalam MS, Maskell S, Gordon N, Clapp T. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. Signal Processing, IEEE Transactions on. 2002; 50:174–188.
4. Ballmann, Werner. Lectures on Spaces of Nonpositive Curvature. Oberwolfach Seminars; Birkhuser Basel. 1995.
5. Cherian, A.; Sra, S.; Banerjee, A.; Papanikolopoulos, N. Efficient similarity search for covariance matrices via the jensen-bregman logdet divergence. ICCV; 2011.
6. Chikuse, Y. Statistics On Special Manifolds. Springer; 2002.
7. Chikuse Y. State space models on special manifolds. Journal of Multivariate Analysis. 2006; 97:1284–1294.
8. Comaniciu D, Meer P. Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2002; 24:603–619.
9. Comaniciu, D.; Ramesh, V.; Meer, P. Real-time tracking of non-rigid objects using mean shift. CVPR; 2000.
10. Fletcher, P.; Joshi, S. Principal geodesic analysis on symmetric spaces: Statistics of diffusion tensors. In: Sonka, Milan; Kakadiaris, Ioannis; Kybic, Jan, editors. Computer Vision and

Mathematical Methods in Medical and Biomedical Image Analysis. 2004. p. 87-98. 3117 of Lecture Notes in Computer Science

11. Fletcher PT, Lu C, Pizer SM, Joshi S. Principal geodesic analysis for the study of nonlinear statistics of shape. *Medical Imaging, IEEE Transactions on*. 2004; 23:995–1005.
12. Graczyk P. A central limit theorem on the space of positive definite symmetric matrices. *Ann Inst Fourier*. 1992; 42:857–874.
13. Helgason, S. *Differential Geometry, Lie Groups, and Symmetric Spaces*. New York: Academic Press; 2001.
14. Karcher H. Riemannian center of mass and mollifier smoothing. *Comm Pure Appl Math*. 1977; 30:509–541.
15. Kwon J, Park FC. Visual tracking via particle filtering on the affine group. *The International Journal of Robotics Research*. 2010; 29:198–217.
16. Lenglet C, Rousson M, Deriche R, Faugeras O. Statistics on the manifold of multivariate normal distributions: Theory and application to diffusion tensor mri processing. *Journal of Mathematical Imaging and Vision*. 2006; 25:423–444. 10.1007/s10851-006-6897-z
17. Li, M.; Chen, W.; Huang, K.; Tan, T. Visual tracking via incremental selftuning particle filtering on the affine group. *CVPR*; 2010.
18. Li, X.; Hu, W.; Zhang, Z.; Zhang, X.; Zhu, M.; Cheng, J. Visual tracking via incremental log-euclidean riemannian subspace learning. *CVPR*; 2008. p. 1-8.
19. Liu Y, Li G, Shi Z. Covariance tracking via geometric particle filtering. *EURASIP J Adv Signal Process*. 2010; 2010:22:1–22:9.
20. Metha, ML. *Random Matrices and the statistics theory of Energy levels*. Academic Press; New York and London: 1967.
21. Moakher M. A differential geometric approach to the geometric mean of symmetric positive-definite matrices. *SIAM J MATRIX ANAL APPL*. 2005; 26:735–747.
22. Mostow, GD. *Strong Rigidity of Locally Symmetric Spaces*. Annals of Mathematics Studies, Princeton University Press; Princeton: 1973.
23. Pennec X. Intrinsic statistics on riemannian manifolds basic tools for geometric measurements. *Journal of Mathematical Imaging and Vision*. 2006; 25:127–154.
24. Porikli, F. Learning on manifolds. *Proceedings of the 2010 joint IAPR international conference on Structural, syntactic, and statistical pattern recognition, SSPR&SPR'10*; Berlin, Heidelberg. Springer-Verlag; 2010. p. 20-39.
25. Porikli, F.; Pan, P. Regressed importance sampling on manifolds for efficient object tracking. *IEEE International Conference on Advanced Video and Signal Based Surveillance*; 2009.
26. Porikli F, Tuzel O, Meer P. Covariance tracking using model update based on lie algebra. *CVPR*. 2006; 1:728–735.
27. Schwartzman, A. PhD thesis. Stanford University; 2006. Random ellipsoids and false discovery rates: Statistics for diffusion tensor imaging data.
28. Snoussi, H.; Richard, C. Monte carlo tracking on the riemannian manifold of multivariate normal distributions. *DSPSPE*; 2009.
29. Sorenson, HW., editor. *Kalman Filtering: Theory and Application*. IEEE Press; 1985.
30. Srivastava A, Klassen E. Bayesian and geometric subspace tracking. *Advances in Applied Probability*. 2004; 36:43–56.
31. Subbarao R, Meer P. Nonlinear mean shift over riemannian manifolds. *International Journal of Computer Vision*. 2009; 84:1–20. 10.1007/s11263-008-0195-8
32. Terras, A. *Harmonic analysis on symmetric spaces and applications*. Springer-Verlag; 1988.
33. Tosato D, Farenzena M, Cistani M, Murino V. A re-evaluation of pedestrian detection on riemannian manifolds. *Pattern Recognition, International Conference on*. 2010; 0:3308–3311.
34. Tuzel, O.; Porikli, F.; Meer, P. Region covariance: A fast descriptor for detection and classification. *ECCV*; 2006.
35. Tuzel O, Porikli F, Meer P. Pedestrian detection via classification on riemannian manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2008; 30:1713–1727. [PubMed: 18703826]

36. Tyagi, A.; Davis, JW. A recursive filter for linear systems on riemannian manifolds. CVPR; 2008. p. 1-8.
37. Udriste, C. Convex Functions and Optimization Methods on Riemannian Manifolds. Kluwer Academic Publishers; 1994.
38. Wong Y. Sectional curvatures of grassmann manifolds. Proc Nati Acad Sci USA. 1968; 60:75–79.
39. Wu, Y.; Cheng, J.; Wang, J.; Lu, H. Real-time visual tracking via incremental covariance tensor learning. ICCV; 2009.
40. Xie, Yuchen; Vemuri, Baba; Ho, Jeffrey. Statistical analysis of tensor fields. Int. Conf. on Medical Image Computing and Computer Assisted Intervention (MICCAI); 2010.

Appendix A. Proof of Theorem 2.1 and Corollary 2.2

Theorem 2.1

The normalization factor Z in Eq. 2.8 is a finite constant with respect to parameter $\mathbf{M}$ P_n

To prove this theorem, we need to first prove the following lemma:

Lemma A.1

$$W = \int_{P_n} \exp\left(-\frac{\text{Tr}(\log \mathbf{X} \log \mathbf{X}^t)}{2\omega^2}\right) [d\mathbf{X}] < \infty$$

Proof

This lemma indicates that the normalization factor Z is constant, and hence $p(\mathbf{X}; \mathbf{M}, \omega^2)$ is a probability density function on P_n . To prove this lemma, we first represent $\mathbf{X}$ in polar coordinates $\{\lambda_i\}$, $\mathbf{R}$ based on the eigen decomposition, $\mathbf{X} = \mathbf{R} \mathbf{\Lambda} \mathbf{R}^t$, where $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_n)$, $\mathbf{R} \mathbf{R}^t = \mathbf{I}_{n \times n}$. From [32] we know that

$$[d\mathbf{X}] = c_n \prod_{j=1}^n \lambda_j^{-(n-1)/2} \prod_{1 \leq i < j \leq n} |\lambda_i - \lambda_j| \prod_{i=1}^n \frac{d\lambda_i}{\lambda_i} d\mathbf{R} \quad (\text{A.1})$$

where $d\mathbf{R}$ is the invariant measure on the orthogonal group $O(n)$ [6] with $\int_{O(n)} d\mathbf{R} = 1$ (Since the orthogonal group is compact, we can easily normalize the measure), c_n is a constant depending on n , and $d\lambda_i$ is the Lebesgue measure in $\mathbb{R}$. With the following change of variables, $y_i = \log(\lambda_i)$, we get

$$W = c_n \int_{\mathbb{R}^n} \exp\left(-\sum_{i=1}^n \left(\frac{1}{2\omega^2} y_i^2 + \frac{n-1}{2} y_i\right)\right) \prod_{1 \leq i < j \leq n} |\exp(y_i) - \exp(y_j)| dy \quad (\text{A.2})$$

$$= c_n \int_{\mathbb{R}^n} \left| \sum_{\gamma \in S_n} \text{sgn}(\gamma) \exp\left(-\frac{1}{2} \sum_{i=1}^n (y_i^2 / \omega^2 + (n-1-2\gamma(i)) y_i)\right) \right| dy \quad (\text{A.3})$$

$$\leq c_n (2\pi\omega^2)^{\frac{n}{2}} \sum_{\gamma \in S_n} \exp\left(\frac{\omega^2 \sum_{i=1}^n \gamma(i)^2 - \omega^2 n(n-1)^2 / 4}{2}\right) < \infty \quad (\text{A.4})$$

where σ is an element of S_n which is the set of all permutations of $0, 1, \dots, n-1$, and $\text{sgn}(\sigma)$ denotes the signature of σ which is 1 or -1 depending on the permutation. The derivation from Eq. A.2 to A.3 is based on the fact that $\sum_{i=1}^n \exp(y_i) - \exp(y_j)$ is actually a Vandermonde determinant. By expansion using Leibniz formula and putting the Gaussian term in, we can directly get the Eq. A.3. The inequality in Eq. A.4 is based on the convexity of the absolute-value function.

We are now ready to present the proof of Theorem 2.1.

Proof

Assume Z is a function of $\mathbf{M}$, $\mathbf{M} \in P_n$ denoted by $Z(\mathbf{M})$.

$$Z(\mathbf{M}) = \int_{P_n} \exp\left(-\frac{\text{dist}^2(\mathbf{X}, \mathbf{M})}{2\omega^2}\right) [d\mathbf{X}] \quad (\text{A.5})$$

Since the GL group action is transitive on P_n , $\mathbf{N} \in P_n$, $\mathbf{g} \in GL(n)$ such that $\mathbf{N} = \mathbf{g}\mathbf{M}\mathbf{g}^t$. Thus,

$$Z(\mathbf{N}) = \int_{P_n} \exp\left(-\frac{\text{dist}^2(\mathbf{X}, \mathbf{g}\mathbf{M}\mathbf{g}^t)}{2\omega^2}\right) [d\mathbf{X}] = \int_{P_n} \exp\left(-\frac{\text{dist}^2(\mathbf{g}^{-1}\mathbf{X}\mathbf{g}^{-t}\mathbf{M})}{2\omega^2}\right) [d\mathbf{X}]$$

Let $\mathbf{Y} = \mathbf{g}^{-1}\mathbf{X}\mathbf{g}^{-t}$, so $\mathbf{X} = \mathbf{g}\mathbf{Y}\mathbf{g}^t$. Substituting this into the above equation we get

$$Z(\mathbf{N}) = \int_{P_n} \exp\left(-\frac{\text{dist}^2(\mathbf{Y}, \mathbf{M})}{2\omega^2}\right) [d\mathbf{g}\mathbf{Y}\mathbf{g}^t] = \int_{P_n} \exp\left(-\frac{\text{dist}^2(\mathbf{Y}, \mathbf{M})}{2\omega^2}\right) [d\mathbf{Y}] = Z(\mathbf{M})$$

Thus, $\mathbf{M}, \mathbf{N} \in P_n$, $Z(\mathbf{M}) = Z(\mathbf{N})$. From A.1 we know that $Z(\mathbf{I}) < \infty$, by substitution as in the above, we obtain the result, Z is finite and constant with respect to $\mathbf{M}$.

Following is the proof of corollary 2.2.

Corollary 2.2

Given a set of i.i.d samples $\{\mathbf{X}_i\}$ drawn from the distribution $dP(\mathbf{X}; \mathbf{M}, \omega^2)$, the MLE of the parameter $\mathbf{M}$ is the Karcher mean of all samples.

Proof

$$\begin{aligned} -\log(p(\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_m; \mathbf{M}, \omega^2)) &= -\sum_i \log(p(\mathbf{X}_i; \mathbf{M}, \omega^2)) \\ &= n\log Z + \sum_i \frac{\text{dist}^2(\mathbf{X}_i, \mathbf{M})}{2\omega^2} \end{aligned}$$

Since Z is constant with respect to $\mathbf{M}$ as proved the Theorem 2.1, we have

$$\text{argmax}_{\mathbf{M}} p(\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_m; \mathbf{M}, \omega^2) = \text{argmin}_{\mathbf{M}} \sum_i \text{dist}^2(\mathbf{X}_i, \mathbf{M})$$

Thus, MLE of the parameter $\mathbf{M}$ of the distribution $dP(\mathbf{X}; \mathbf{M})$ equals to the Karcher mean of samples.

Appendix B. Proof of Theorem 2.3

Theorem 2.3

$\mathbf{M}$ is the Karcher Expectation of the generalized normal distribution $dP(\mathbf{X}; \mathbf{M}, \omega^2)$.

Proof

To prove this, we need to show that $dP(\mathbf{X}; \mathbf{M}, \omega^2)$ satisfies Eq. 2.7. Let

$$\Psi = \int_{P_n} \log(\mathbf{M}^{-1/2} \mathbf{X} \mathbf{M}^{-1/2}) dP(\mathbf{X}; \mathbf{M}, \omega^2)$$

then in the integral, using a change of variable, $\mathbf{X}$ to $\mathbf{Y} = \mathbf{M} \mathbf{X}^{-1} \mathbf{M}$ ($\mathbf{X} = \mathbf{M} \mathbf{Y}^{-1} \mathbf{M}$). Since P_n is a symmetric space and the metric/measure is GL -invariant we know that $[d\mathbf{X}] = [d\mathbf{Y}]$, and $\text{dist}(\mathbf{X}, \mathbf{M}) = \text{dist}(\mathbf{Y}, \mathbf{M})$. Thus we have,

$$\begin{aligned} \Psi &= \int_{P_n} \log(\mathbf{M}^{-1/2} \mathbf{X} \mathbf{M}^{-1/2}) \frac{1}{Z} \exp\left(-\frac{\text{dist}^2(\mathbf{X}, \mathbf{M})}{2\omega^2}\right) [d\mathbf{X}] \\ &= \int_{P_n} \log(\mathbf{M}^{1/2} \mathbf{Y}^{-1} \mathbf{M}^{1/2}) \frac{1}{Z} \exp\left(-\frac{\text{dist}^2(\mathbf{Y}, \mathbf{M})}{2\omega^2}\right) [d\mathbf{Y}] = -\Psi = 0. \end{aligned}$$

Since P_n has non-positive curvature, the solution of Eq. 2.5 is unique [14]. Thus $\mathbf{M}$ is the Karcher Expectation of $dP(\mathbf{X}; \mathbf{M}, \omega^2)$.

Appendix C. Proof of Theorem 2.4

Theorem 2.4

$$\lim_{\omega \rightarrow 0} \frac{\text{Var}(\mathbf{X})}{\omega^2} = \frac{n(n+1)}{2}$$

Proof

Let

$$v(\mathbf{y}, \omega) = \sum_{\gamma \in S_n} \text{sgn}(\gamma) \exp\left(-\frac{1}{2} \sum_{i=1}^n (y_i^2 + \omega(n-1-2\gamma(i))y_i)\right) \quad (\text{C.1})$$

where S_n and $\text{sgn}(\gamma)$ are related to the permutation of $0, 1, \dots, n-1$ which is defined to be the same as in Eq. A.3. Also we can find that $q(\mathbf{y}) = \frac{|v(\mathbf{y}, \omega)|}{z(\omega)}$ and $Z(\omega) = \int_{\mathbb{R}^n} |v(\mathbf{y}, \omega)| d\mathbf{y}$. The Taylor expansion of $v(\mathbf{y}, \omega)$ up to $\frac{n(n-1)}{2}$ -th order with respect to ω around zero is

$$\begin{aligned}
v(\mathbf{y}, \omega) &= \sum_{\gamma \in S_n} \operatorname{sgn}(\gamma) \sum_{k=0}^{\frac{n(n-1)}{2}} \frac{(-\omega)^k}{k!} \exp\left(-\sum_{i=1}^n \frac{y_i^2}{2}\right) \left(\sum_{i=1}^n \left(\frac{n-1}{2} - \gamma(i)\right) y_i\right)^k + O(\omega^{\frac{n^2-n+2}{2}}) \\
&= C(-\omega)^{\frac{n(n-1)}{2}} \exp\left(-\sum_{i=1}^n \frac{y_i^2}{2}\right) \prod_{1 \leq i < j \leq n} (y_i - y_j) + O(\omega^{\frac{n^2-n+2}{2}})
\end{aligned} \quad (\text{C.2})$$

where C is a constant. The Eq. C.2 above used the fact that given n non-negative integers k_i and $\sum_{i=1}^n k_i \leq \frac{n(n-1)}{2}$,

$$\sum_{\gamma} \operatorname{sgn}(\gamma) \prod_{i=1}^n \gamma(i)^{k_i} = 0 \quad \text{if } \{\kappa_i\} \notin S_n \quad (\text{C.3})$$

So, in the Taylor expansion all the terms with degree less than $\frac{n(n-1)}{2}$ are zeros. In the $\frac{n(n-1)}{2}$ -th order terms, only terms with powers in S_n will be non-zero.

Let the density $\hat{q}(\mathbf{y}) = \frac{1}{z} \exp\left(-\sum_{i=1}^n \frac{y_i^2}{2}\right) \prod_{1 \leq i < j \leq n} |y_i - y_j|$, which is exactly the joint distribution of the eigen values of a Gaussian Orthogonal Ensemble [20] which is a symmetric random matrix with each of its elements being independent random variables drawn from a zero mean Gaussian. In this case, the variance of the diagonal elements in the random matrix is 1 and that of the off diagonal elements is $\frac{1}{2}$. Recall that we are now in polar coordinates. By transforming q to the Cartesian coordinates of the space of symmetric matrices we get

$$\begin{aligned}
\operatorname{Var}_{\hat{q}}(\mathbf{y}) &= \frac{1}{z} \int_{\mathbb{R}^n} \mathbf{y}^t \mathbf{y} \hat{q}(\mathbf{y}) d\mathbf{y} \\
&= (2\pi)^{-\frac{n(n-1)}{4}} \int_{\mathbf{V} \in \operatorname{Sym}(n)} \operatorname{tr}(\mathbf{V}^2) \exp\left(-\frac{\operatorname{tr}(\mathbf{V}^2)}{2}\right) d\mathbf{V} = \frac{n(n+1)}{2} \quad (\text{C.4})
\end{aligned}$$

where $\operatorname{Sym}(n)$ is the space of $n \times n$ symmetric matrices, and $d\mathbf{V}$ is the Lebesgue measure in $\operatorname{Sym}(n)$.

From above we know that,

$$\lim_{\omega \rightarrow 0} \frac{\operatorname{Var}(\mathbf{X})}{\omega^2} = \lim_{\omega \rightarrow 0} \operatorname{Var}_q(\mathbf{y}) = \operatorname{Var}_{\hat{q}}(\mathbf{y}) = \frac{n(n+1)}{2} \quad (\text{C.5})$$

Appendix D. Proof of Theorem 2.5

Theorem 2.5

$$\lim_{\omega \rightarrow \infty} \frac{\operatorname{Var}(\mathbf{X})}{\omega^4} = \frac{(n^3 - n)}{12}$$

Proof

We first define the upper bound and lower bound on $q(\mathbf{y})$.

$$\begin{aligned}
 q_u(\mathbf{y}) &= \frac{1}{z_u(\omega)} \sum_{\gamma} \exp\left(-\frac{1}{2} \sum_{i=1}^n (y_i^2 + \omega(n-1-2\gamma(i))y_i)\right) \\
 q_l(\mathbf{y}) &= \frac{1}{z_l(\omega)} \exp\left(-\frac{1}{2} \sum_i y_i^2\right) \prod_{1 \leq i < j \leq n} 2(\cosh(\frac{\omega(y_i - y_j)}{2}) - 1)
 \end{aligned} \quad (D.1)$$

with z_u and z being the normalization factors respectively. Note that both q_u and q are both Gaussian mixtures. In q_u all mixing weights are positive, while in q there are negative weights. After expansion we have,

$$\begin{aligned}
 q_l(\mathbf{y}) &= \frac{1}{z_l(\omega)} \sum_{\beta \in B_n} w_{\beta} \exp\left(-\frac{1}{2} \sum_{i=1}^n (y_i + \omega(n-1-2\beta(i))/2)^2\right) \\
 w_{\beta} &= \alpha_{\beta} \exp\left(\frac{\omega^2 \sum_{i=1}^n \beta(i)^2 - \omega^2 n(n-1)^2/4}{2}\right)
 \end{aligned} \quad (D.2)$$

where B_n is the set of all possible power combinations of polynomial

$$\sum_{\beta \in B_n} \alpha_{\beta} \prod_{i=1}^n x_i^{\beta(i)} = \prod_{1 \leq i < j \leq n} (x_i + x_j - 2\sqrt{x_i x_j}) \quad (D.3)$$

and α_{β} are the coefficients. We can prove that

$$\max_{\beta \in B_n} \sum_{i=1}^n \beta(i)^2 = \sum_{i=1}^n \gamma(i)^2 = \frac{(2n-1)(n^2-n)}{6} \quad (D.4)$$

The maximum can be achieved only at S_n and S_{n-1} , S_{n-2} , ..., S_1 .

From the definition we can compute the normalization constants and the variances of q and q_u in a closed form.

$$\begin{aligned}
 z_u &= (2\pi)^{n/2} \sum_{\gamma} \exp\left(\frac{\omega^2 \sum_{i=1}^n \gamma(i)^2 - \omega^2 n(n-1)^2/4}{2}\right) \\
 &= (2\pi)^{n/2} n! \exp\left(\frac{\omega^2 n(n^2-1)}{24}\right) \\
 z_l &= (2\pi)^{n/2} \sum_{\beta \in B_n} \alpha_{\beta} \exp\left(\frac{\omega^2 \sum_{i=1}^n \beta(i)^2 - \omega^2 n(n-1)^2/4}{2}\right) \\
 &= z_u + (2\pi)^{n/2} \sum_{\beta \in B_n \setminus S_n} \alpha_{\beta} \exp\left(\frac{\omega^2 \sum_{i=1}^n \beta(i)^2 - \omega^2 n(n-1)^2/4}{2}\right) \\
 \text{Var}_{q_u}(\mathbf{y}) &= n + \frac{n^3-n}{12} \omega^2 \\
 \text{Var}_{q_l}(\mathbf{y}) &= n - \frac{n(n-1)^2}{4} \omega^2 + \frac{(2\pi)^{n/2}}{z_l} \omega^2 \sum_{\beta \in B_n} \exp\left(\frac{\omega^2 \sum_{i=1}^n \beta(i)^2 - \omega^2 n(n-1)^2/4}{2}\right) \left(\sum_{i=1}^n \beta(i)^2\right)
 \end{aligned} \quad (D.5)$$

Since $0 < \cosh(x) - 1 < |\sinh(x)|$ and $|x| < \sqrt{x^2 + 1}$, we have $\mathbf{y} \in \mathbb{R}^n$, $z < q(\mathbf{y}) < z q(\mathbf{y}) < z_u q_u(\mathbf{y})$ and also $z = z < z_u$. We can then get the following bounds for $\text{Var}_q(\mathbf{y})$.

$$\frac{z_l}{z_u} \text{Var}_{q_l}(\mathbf{y}) \leq \text{Var}_q(\mathbf{y}) \leq \frac{z_u}{z_l} \text{Var}_{q_u}(\mathbf{y}) \quad (D.6)$$

From Eq. D.5 we can show that,

$$\lim_{\omega \rightarrow \infty} \frac{z_t}{z_u} = \lim_{\omega \rightarrow \infty} (1 + \sum_{\beta \in B_n \setminus S_n} \alpha_\beta \exp(\frac{\omega^2}{2} (\sum_{i=1}^n \beta(i)^2 - \frac{(2n-1)(n^2-n)}{6}))) = 1, \quad (D.7)$$

because $B_n \setminus S_n, \sum_i \beta(i)^2 < \frac{(2n-1)(n^2-n)}{6}$. Similarly,

$$\begin{aligned} \lim_{\omega \rightarrow \infty} \frac{Var_{q_t}(\mathbf{y})}{\omega^2} &= -\frac{n(n-1)^2}{4} \\ + \lim_{\omega \rightarrow \infty} \sum_{\beta \in B_n} \alpha_\beta \exp(\frac{\omega^2}{2} (\sum_{i=1}^n \beta(i)^2 - \frac{(2n-1)(n^2-n)}{6})) & (\sum_{i=1}^n \beta(i)^2) \quad (D.8) \\ &= \frac{n^3-n}{12} = \lim_{\omega \rightarrow \infty} \frac{Var_{q_u}(\mathbf{y})}{\omega^2} = \lim_{\omega \rightarrow \infty} \frac{Var_q(\mathbf{y})}{\omega^2} = \lim_{\omega \rightarrow \infty} \frac{Var(\mathbf{X})}{\omega^4} \end{aligned}$$

Appendix E. Proof of Theorem 3.1

Theorem 3.1

The energy function E_k in Eq. 3.4 is geodesically convex on the Riemannian manifold $P_n \times P_n$ where $\times$ denotes the Cartesian product.

Proof

Recall that E_k in Eq. 3.4 is a sum of squared distance function with positive weights. Here the strategy is to prove that each item inside the sum is geodesically convex.

It is known that $C \rightarrow P_n$ function $d_C(\mathbf{X}) = \text{dist}(\mathbf{X}, C)$ is geodesically convex [22]. Also, it is known that if $\gamma_1, \gamma_2 : [0, 1] \rightarrow P_n$ are two geodesics on P_n , $(t) = (\gamma_1(t), \gamma_2(t))$ is a geodesic on $P_n \times P_n$ [40]. Combining these two facts with the non-negativity of the distance function, we can see that in $P_n \times P_n$ the function $f_C((\mathbf{X}_1, \mathbf{X}_2)) = \text{dist}(\mathbf{X}_1, C)^2$ is geodesically convex.

Now we want to prove that the function $g((\mathbf{X}_1, \mathbf{X}_2)) = \text{dist}(\mathbf{X}_1, \mathbf{X}_2)^2$ is geodesically convex. As shown in [4], for any two geodesics $\gamma_1, \gamma_2 : [0, 1] \rightarrow P_n$,

$$\text{dist}(\gamma_1(\frac{1}{2}), \gamma_2(\frac{1}{2})) \leq \frac{1}{2} \text{dist}(\gamma_1(0), \gamma_2(0)) + \frac{1}{2} \text{dist}(\gamma_1(1), \gamma_2(1)) \quad (E.1)$$

With the definition of the geodesic convexity, we can see that $\sqrt{g}$ and g are both geodesically convex.

Since the distance function is GL invariant, E_k can be written as a weighted sum of f s and g with positive weights, and then we know that E_k is also geodesically convex.

Fig. 2.1.
An example of different distances on S^2 .

Fig. 4.1.

Mean estimation error from 20 trials for the synthetic data experiment. The x-axis denotes the time step. The y-axis denotes the estimation error measured using the Riemannian distance between the estimates and the ground truth. In all three sub-figures the black curves denote the estimation error for our IRF, the grey curves for LRF with $\mathbf{X}_b$ set as the observation in the first step.

Fig. 4.2.

Head tracking result for video sequences with moving camera. Top and bottom rows depict snapshots and quantitative evaluations of the results from the Seq mb and Seq sb respectively (<http://www.ces.clemson.edu/stb/research/headtracker/seq/>). The tracking error is measured by the distance between the estimated object center and the ground truth. Tracking results from the three methods are shown by using different coloured boxes superposed on the images and different coloured lines in the plots. Results from our method (IRF) are in black, LRF in dark grey and KM in white (box) or light grey (error curve).

Table 4.1

Tracking result for the real data experiment.

Seq.	Obj.	Start	End	Err(IRF)	Err(LRF)	Err(KM)
C3ps1	1	200	700	4.3213	10.8467	10.7666
C3ps1	7	200	700	4.1998	7.3524	4.7929
C3ps1	8	200	700	2.6258	5.5844	6.2928
C3ps2	7	500	800	2.7605	10.6478	12.4289
C3ps2	8	500	800	3.4451	7.2113	11.6432
Cosow2	1	500	900	3.7612	4.613	6.0196
Cosow2	3	500	900	4.9871	5.8552	7.9788