

## Research Article

# Covariance Tracking via Geometric Particle Filtering

Yunpeng Liu,<sup>1,2,3,4</sup> Guangwei Li,<sup>5</sup> and Zelin Shi<sup>1,3</sup>

<sup>1</sup>Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China

<sup>2</sup>Graduate School of Chinese Academy of Sciences, Beijing 100049, China

<sup>3</sup>Key Laboratory of Optical-Electronics Information Processing, Chinese Academy of Science, Shenyang 110016, China

<sup>4</sup>Key Laboratory of Image Understanding and Computer Vision, Liaoning Province 110016, China

<sup>5</sup>Management Science and Engineering Department, Qingdao University, Qingdao 266071, China

Correspondence should be addressed to Yunpeng Liu, ypliu@sia.cn

Received 30 November 2009; Accepted 24 June 2010

Academic Editor: Yingzi Du

Copyright © 2010 Yunpeng Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Region covariance descriptor recently proposed has been approved robust and elegant to describe a region of interest, which has been applied to visual tracking. We develop a geometric method for visual tracking, in which region covariance is used to model objects appearance; then tracking is led by implementing the particle filter with the constraint that the system state lies in a low dimensional manifold: affine Lie group. The sequential Bayesian updating consists of drawing state samples while moving on the manifold geodesics; the region covariance is updated using a novel approach in a Riemannian space. Our main contribution is developing a general particle filtering-based tracking algorithm that explicitly take the geometry of affine Lie groups into consideration in deriving the state equation on Lie groups. Theoretic analysis and experimental evaluations demonstrate the promise and effectiveness of the proposed tracking method.

## 1. Introduction

Visual tracking in an image sequence, which is now an active area of research in computer vision, is widely applied to vision guidance, surveillance, robotic navigation, human-computer interaction, and so forth. Dynamic deformation of object is a distinct problem in image-based tracking.

Conventional correlation-based trackers [1, 2] use either a region's gray information or edges and other features as the target signatures, but it is difficult to solve the problem of object region deformation in the tracking. Over the last 10 years, numerous approaches [3–10] have been proposed to address this problem. The main idea of them is molding geometric parameter models for the image motions of points within a target region. The parameter models including affine model, projective model, or other nonlinear models. The classic Lucas-Kanade tracker [3, 4] and Meanshift tracker [5] get the model parameters through gradient descent which minimizes the difference between the template and the current region of the image. These methods are computationally efficient. However, the methods may converge to a local maximum, they are

sensitive to background clutter, occlusion, and quick moving objects. These problems can be mitigated by stochastic methods which maintain multiple hypotheses in the state space and in this way, achieve more robustness to the local maximum. Among various stochastic methods, particle filters [5–10] are very successful. Particle filters provide a robust tracking framework as they are neither limited to linear systems nor require the noise to be Gaussian. Particle filters simultaneously track multiple hypotheses and recursively approximate the posterior probability density function in the state space with a set of random sampled particles.

Many papers, such as [5–10] utilize particle filter method to track deformable target. They use affine transform as parameter model, and the six affine parameters were treated as a vector. However, the affine parameters belong to spaces which are not vector spaces, but instead a curved Lie group. In general, the system state of the particle filter lies in a constrained subspace whose dimension is much lower than the whole space dimension. Only a few recent papers have tried to use the geometry of the manifold to design Bayesian filtering algorithms [11, 12]. However, there is little

discussion in the literature using the intrinsic geometry of manifold to develop particle filter-based tracking algorithms.

Object representation is one of major components for a typical visual tracker. Extensive researches have been done on this topic. Recently Tuzel et al. [13, 14] proposed an elegant and simple solution to integrate multiple features. In this method, covariance matrix was employed to represent the target (region covariance descriptor) has many advantages: (1) it embodies both spatial and statistical properties of the objects; (2) it provides an elegant solution to fuse multiple features and modalities; (3) it has a very low-dimensionality; (4) it is capable of comparing regions without being restricted to a constant window size; and (5) the estimation of the covariance matrix can be easily implemented.

In this paper, we integrate covariance descriptor into Monte Carlo technique for visual tracking, study the geometry structure of affine Lie groups, and propose a tracking algorithm through particle filtering on manifolds, which implement the particle filter with the constraint that the system state lies in a low dimensional manifold. The sequential Bayesian updating consists drawing state samples while moving on the manifold geodesics; this provides a smooth prior for the state space change. The regions covariance matrices are updated using a novel approach in a Riemannian space. Theoretic analysis and experimental results show the promise and effectiveness of the approach proposed.

The paper is organized as follows. In Section 2, The mathematical background is described. Section 3 shows the object regions descriptor and the new update solution for those descriptors. Section 4 describes the tracking algorithm using geometric particle filtering. Results on real image sequences for evaluating algorithm performance are discussed in Section 5. Section 6 concludes this paper.

## 2. Manifold and Lie Group

The tools used here come primarily from differential geometry. For more information on these subjects, the reader is referred to [15, 16].

A manifold is a topological space that is locally similar to an Euclidean space. Intuitively, we can think of a manifold as a continuous surface lying in a higher dimensional Euclidean space. Analytic manifolds satisfy some further conditions of smoothness [16]. From now onwards, we restrict ourselves to analytic manifolds and by manifold we mean analytic manifold.

The tangent space,  $T_x$  at  $x$ , is the plane tangent to the surface of the manifold at that point. The tangent space can be thought of as the set of allowable velocities for a point constrained to move on the manifold. For  $d$ -dimensional manifolds, the tangent space is a  $d$ -dimensional vector space. An example of a two-dimensional manifold embedded in  $R^3$  with the tangent space  $T_x$  is shown in Figure 1. The solid arrow  $\Delta$  is a tangent at  $x$ . The distance between two points on the manifold is given in terms of the lengths of curves between them. The length of any curve is defined

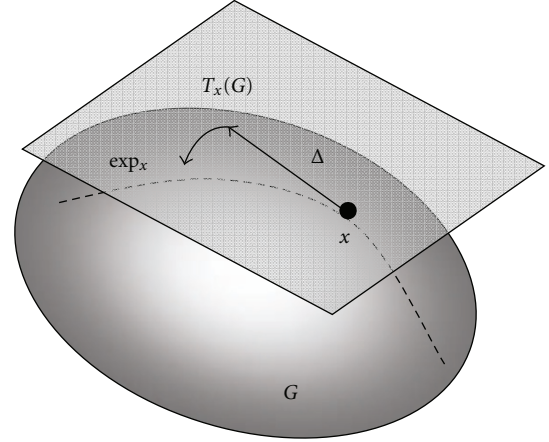


FIGURE 1: Riemannian Exponential Mapping.

by an integral over norms of tangents [17]. The curve with minimum length is known as the geodesic and the length of the geodesic is the intrinsic distance. Parameter spaces occurring in computer vision problems usually have well-studied geometries and closed form formulae for the intrinsic distance are available. Tangents and geodesics are closely related. For each tangent  $\Delta \in T_x$ , there is a unique geodesic starting at  $x$  with initial velocity  $\Delta$ . The exponential map,  $\exp_x$  maps  $\Delta$  to the point on the manifold reached by this geodesic.

A Lie group is a group with the structure of an analytic manifold such that the group operations are analytic, that is the maps

$$\begin{aligned} G \times G &\longrightarrow G & (X, Y) &\longmapsto XY, \\ G &\longrightarrow G & X &\longmapsto X^{-1}, \end{aligned} \quad (1)$$

are analytic [15]. The local neighborhood of any group element  $G$  can be adequately described by its tangent-space. The tangent-space at the identity element forms its Lie algebra.

The set of nonsingular  $n \times n$  square matrices forms a Lie group where the group product is modeled by matrix multiplication, usually denoted by  $GL(n, R)$  for the general linear group of the order  $n$ . Lie groups are differentiable manifolds on which we can do calculus.

In our task, we use affine transformation as parameter model. The set of all affine transformation forms a matrix Lie group.

## 3. Region Covariance Descriptor

Let  $I$  be the observed image with size of  $W \times H$ , and  $F$  be  $W \times H \times d$  dimensional feature image extracted from  $I$

$$F(x, y) = \phi(I, x, y), \quad (2)$$

where  $\phi$  can be any mapping such as color, gradients, filter responses, and so forth. Let  $\{z_k\}_{k=1 \dots n}$  be the  $d$ -dimensional feature points inside a given rectangular region. The region

is represented by the  $d \times d$  covariance matrix of the feature points

$$C_R = \frac{1}{n} \sum_{k=1}^n (z_k - \mu)(z_k - \mu)^T, \quad (3)$$

where  $n$  is the number of pixels in the region.  $\mu$  is the mean of the feature points.

In our task, we define  $\phi$  as

$$\begin{bmatrix} x & y & I & I_x & I_y & I_x I_y \end{bmatrix}, \quad (4)$$

where  $x$  and  $y$  are the pixel location in  $R$ ;  $I$  is the gray value;  $I_x$  and  $I_y$  are first derivatives of  $I$ ; In this way, the region  $R$  is mapped into a  $6 \times 6$  covariance matrix.

In a tracking process, the objects appearance changes over time. This dynamic behavior requires a robust temporal update of the region covariance descriptors and the definition of dissimilarity metric for the region covariance. The important question here is how to measure the dissimilarity between two region covariance matrices and how to update the regions covariance matrix in the next time slot. Note that the covariance matrices do not lie on Euclidean space. For example, the space is not closed under multiplication with negative scalars. So, it is necessary to get the dissimilarity between two covariance matrices in a different space. To overcome this problem a Riemannian Manifold is used.

**3.1. Dissimilarity Metric.** The dissimilarity between two regions covariance matrices can be given by the distance between two points of the manifold  $M$ , considering that those points are the two regions.

The covariance matrix, which is symmetric positive definite matrix, forms a Riemannian manifold. According to [14], we define a Riemannian metric like that

$$\langle y, z \rangle_X = \text{tr}(X^{-1/2} y X^{-1} z X^{-1/2}). \quad (5)$$

The exponential map associated to the above Riemannian metric is

$$\exp_X(y) = X^{1/2} \exp(X^{-1/2} y X^{-1/2}) X^{1/2}. \quad (6)$$

By (6), we can obtain the logarithm map

$$y = \log_X(Y) = X^{1/2} \log(X^{-1/2} Y X^{-1/2}) X^{1/2}, \quad (7)$$

Submit (7) to (5)

$$\begin{aligned} d^2(X, Y) &= \|y\|_X^2 = \langle y, y \rangle_X \\ &= \langle \log_X(Y), \log_X(Y) \rangle_X \\ &= \text{tr}(\log^2(X^{-1/2} Y X^{-1/2})). \end{aligned} \quad (8)$$

Furthermore, (8) is equivalent to

$$d(X, Y) = \sqrt{\sum_{k=1}^d \log^2 \lambda_k(X, Y)}, \quad (9)$$

where  $\lambda_k$  are the generalized eigenvalues of  $X$  and  $Y$ .

**3.2. Covariance Update.** A solution for the covariance matrices update was proposed in [14], that is based on the estimation of the points mean on a Riemannian Manifold, where each point corresponds to a covariance matrix. This mean estimation is obtained using a gradient descent approach. In this paper, we propose a novel solution for the covariance matrix update, that is based on the mean of the new covariance matrix and the last covariance updated. If  $y$  is the velocity that takes us from  $X$  to  $Y$ ,  $y/2$  will be the half distance to point  $\bar{C}$ . Using (6) and (7), we have

$$\begin{aligned} \bar{C} &= X^{1/2} \exp\left(X^{-1/2} \left(\frac{1}{2} y\right) X^{-1/2}\right) X^{1/2} \\ &= X^{1/2} \exp\left(\frac{1}{2} \log(X^{-1/2} Y X^{-1/2})\right) X^{1/2} \\ &= (X^{1/2} Y X^{1/2})^{1/2}, \end{aligned} \quad (10)$$

where  $\bar{C}$  is the average distance between two points on a Riemannian Manifold (the updated covariance matrix). This update means that the present covariance is more important than the previous covariances. Since we are tracking objects that can change over time, the last information about them is more reliable.

## 4. Tracking Model

The visual tracking problem is cast as an inference task in a Markov model with hidden state variables. The state variable  $S_t$  describes the affine parameters of the target at time  $t$ . Given a set of observed images  $I_{1:t} = \{I_1, \dots, I_t\}$ , we aim to estimate the value of the hidden state variable  $S_t$ . Using Bayesian theorem, we have the familiar result

$$p(S_t | I_{1:t-1}) = \int p(S_t | S_{t-1}) p(S_{t-1} | I_{1:t-1}) dS_{t-1}, \quad (11)$$

$$p(S_t | I_{1:t}) = \frac{p(I_t | S_t) p(S_t | I_{1:t-1})}{p(I_t | I_{1:t-1})}. \quad (12)$$

Equation (11) is called the prediction equation and (12) is called the update equation. The tracking process is governed by the observation model  $p(I_t | S_t)$ , where we estimate the likelihood of  $S_t$  observing  $I_t$ , and the dynamical model between two states  $p(S_t | S_{t-1})$ .

**4.1. Dynamical Model.** Dynamical model, also known as state transition model, can describe transition of object state in tracking process. In visual tracking problems, it is ideal to have an exact state transition model. In practice, however approximations models are used. The deformation and location of a target object in an image can be represented by affine transform. In this work, the state at time  $t$  consists of the six parameters of an affine transformation. 2-D affine transformation of the image can be written as

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} r_1 & r_2 \\ r_3 & r_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} r_5 \\ r_6 \end{bmatrix}, \quad (13)$$

where  $(x, y)$  and  $(x', y')$  denote the location of the corresponding points between two images,  $\begin{bmatrix} r_1 & r_2 \\ r_3 & r_4 \end{bmatrix}$  is a  $2 \times 2$  nonsingular matrix and translation vector  $\begin{bmatrix} r_5 \\ r_6 \end{bmatrix}$ ,  $(r_1, r_2, r_3, r_4, r_5, r_6)$  denotes affine transformation parameters. The transformation can be expressed in homogeneous coordinates as

$$A(r) = \begin{bmatrix} r_1 & r_2 & r_5 \\ r_3 & r_4 & r_6 \\ 0 & 0 & 1 \end{bmatrix}. \quad (14)$$

$A(r)$  specify the displacement between  $S_{t-1}$  and  $S_t$ , we define  $V_t$  as velocities between  $S_{t-1}$  and  $S_t$ , which specify the motion. These defines are analogous to the vector space case in that the velocities are determined by the tangent vectors along geodesics connecting the observed points ( $S_t$ ). Then the state transition model is of the following form:

$$S_t = S_{t-1} \exp(V_{t-1}), \quad (15)$$

$$V_t = V_{t-1} + \mu_{1:t}, \quad (16)$$

where  $S_{1:t}$  is a discrete-time trajectory on a six-dimensional affine Lie group,  $V_{1:t}$  is a velocity on the corresponding Lie algebra,  $\mu_{1:t}$  are Gaussian white zero-mean stochastic processes.

The tracking algorithm will not require the explicit functional form of the prior density; it will be dependent on the samples generated from the prior density. In a Markovian time-series analysis, often there is a standard characterization of a time-varying posterior density, in a convenient recursive form. This characterization relates an underlying Markov process to its observations at each observation time via a pair of state transition equations. The following algorithm specifies a procedure to sample from the conditional prior  $p(S_t | S_{t-1})$ :

*Algorithm 1.* For some  $t = 2, 3, \dots$ , we are given the values for  $S_{t-1}^i$  and  $V_{t-2}^i$ . For  $i = 1, 2, \dots, M$ :

- (1) Generate a sample of  $V_{t-1}^i$ , given  $V_{t-2}^i$ , according to (16).
- (2) For each sample of  $V_{t-1}^i$ , calculate  $S_t^i$  according to  $S_t^i = S_{t-1}^i \exp(V_{t-1}^i)$ .

The Algorithm 1 consists in drawing state samples while moving on the manifold geodesics. This geodesics sample give a dynamics-based smoothing prior on the state transition space. Figure 2 is an illustration of this geodesics sample process.

**4.2. Observation Model.** Next, we specify the probability model for the observed images.  $p(I_t | S_t)$  is the likelihood through the observation  $I_t$  under the state  $S_t$

$$p(I_t | S_t) \propto \exp(-\lambda \|d^2(C_*, C_{S_t})\|^2), \quad (17)$$

where  $C_*$  be covariance features of the template image, and  $C_{S_t}$  denote covariance features at the transformation  $S_t$ .

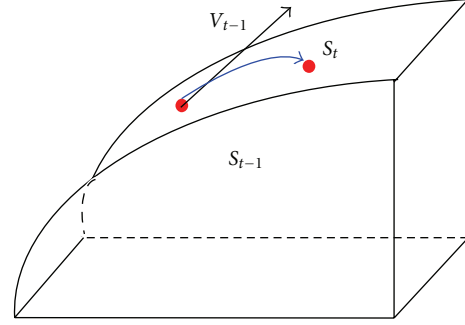


FIGURE 2: Drawing state samples moving on the geodesics.

**4.3. Sequential Monte Carlo Approach.** The Monte Carlo idea is to approximate the posterior density of  $S_t$  by a large number of samples drawn from it. Having obtained the samples, any estimate of  $S_t$  (MMSE, MAP, etc.) can be approximated using sample averages.

A recursive formulation, which takes samples from  $p(S_{t-1} | I_{1:t-1})$  and generates the samples from  $p(S_t | I_{1:t})$  in an efficient fashion, is desirable. We accomplish this task using ideas from sequential methods and importance sampling. Assume that, at the observation time  $t - 1$ , we have a set of  $M$  samples from the posterior,  $\{S_{t-1}^i : i = 1, 2, \dots, M\}$   $\propto p(S_{t-1} | I_{1:t-1})$ . Following are the steps to generate the set  $\{S_t^i : i = 1, 2, \dots, M\}$ .

**Prediction.** The first step is to sample from  $p(S_t | I_{1:t-1})$  given the samples from  $p(S_{t-1} | I_{1:t-1})$ . According to (11),  $p(S_t | I_{1:t-1})$  is the integral of the product of a marginal and a conditional density. This implies that, for each element  $S_{t-1}^i$ , by generating a sample from the conditional,  $p(S_t | S_{t-1}^i)$  we can generate a sample from  $p(S_t | I_{1:t-1})$ . In our case, this is accomplished using Algorithm 1. Now we have samples  $\{\tilde{S}_t^i\}$  from  $p(S_t | I_{1:t-1})$ ; these samples are called predictions, but we have used a geodesics prediction different to classic particle filter on vector space.

**Resampling.** Given these predictions, the next step is to generate samples from the posterior  $p(S_t | I_{1:t})$ . For this, we utilize importance sampling as follows. The samples from the prior  $p(S_t | I_{1:t-1})$  are resampled according to the probabilities that are proportional to the likelihoods  $p(I_t | \tilde{S}_t^i)$ . Form a discrete probability mass function on the set  $\{\tilde{S}_t^i : i = 1, 2, \dots, M\}$

$$\omega_t^i = \frac{p(I_t | \tilde{S}_t^i)}{\sum_{j=1}^M p(I_t | \tilde{S}_t^j)}. \quad (18)$$

Then, resample  $M$  values from the set  $\{\tilde{S}_t^1, \tilde{S}_t^2, \dots, \tilde{S}_t^M\}$  according to probability  $\omega_t^i$ . These values are desired samples from the posterior  $p(S_t | I_{1:t})$ . Denote the resampled set by  $\{S_t^i : i = 1, 2, \dots, M\}$ ,  $S_t^i \propto p(S_t | I_{1:t})$ .



*Averaging on the Lie Group.* Now that we have  $M$  samples from the posterior  $p(S_t | I_{1:t})$ , we can average them appropriately to approximate the posterior mean of  $S_t$ .

It may be recalled that for a vector space, the sample mean or average of a set  $\{S^1, S^2, \dots, S^M\}$  is given by  $\bar{S} = 1/M \sum_{i=1}^M S^i$ . However, such a notion cannot be applied directly to elements of a group manifold. There are at least two ways to define a mean value on a manifold: extrinsic means and intrinsic means. The extrinsic mean depends on the geometry of the ambient space and the embedding. The intrinsic mean is defined using only the intrinsic geometry of the manifold. In general, the intrinsic average is preferable over the extrinsic average but is often hard to compute due to the nonlinearity of the Riemannian distance function and the need to parameterize the group manifold. However, as we will see here, for matrix Lie groups the intrinsic average can be computed efficiently. In several applications, the Lie algebra is used for computing intrinsic means of points having Lie group structure [17–19]. We adopt the similar idea to obtain the intrinsic mean of the affine Lie group.

The “true” intrinsic sample mean is given by

$$\bar{S} = \arg \min \sum_{i=1}^M d^2(S^i, S) \quad S \in G. \quad (19)$$

It will be recalled that for matrix groups, the Riemannian distance is defined by the matrix logarithm operation, that is for matrix group elements  $X$  and  $Y$  we have

$$d(X, Y) = \|\log(YX^{-1})\|. \quad (20)$$

#### 4.4. Detail of Tracking Algorithm

*Algorithm 2.* (1) Initialize:

Generate samples  $\{S_0^i, i = 1, 2, \dots, M\}$  from the prior distribution  $p(S_0)$ . Set initial weights  $\omega_0^i = 1/M$ .

(2) Prediction:

Draw  $\{\tilde{S}_t^i, i = 1, 2, \dots, M\}$  from the conditional prior according to Algorithm 1.

(3) Importance Weights:

Compute the probability  $\omega_t^i, i = 1, 2, \dots, M$  according to (18).

(4) Resampling:

Generate  $M$  samples from the set  $\{\tilde{S}_t^i, i = 1, 2, \dots, M\}$  with the associated probabilities  $\{\omega_t^i, i = 1, 2, \dots, M\}$ . Denote these samples by  $\{S_t^i, i = 1, 2, \dots, M\}$ .

(5) MMSE Averaging:

Calculate the sample average according to (19) which is the target state. Set  $t = t + 1$  and go to step 2.

## 5. Experimental Results

In order to evaluate the performance of the proposed tracking algorithm based on geometric particle filtering and the new update method. We start by comparing proposed algorithm (referred as GPF) with the tracking algorithm based on Particle filtering on vector space (VPF) [5–10] with

the same real image sequences. After that, we evaluated the proposed update method with the one previously proposed in the literature. We also tested the proposed algorithm under varying illumination conditions. These algorithms are implemented in C++ running on an Intel Core-2 2.5 GHz processor with 2 GB memory.

*5.1. Compared with VPF.* Two typical image sequences where the objects undergo large changes in pose and scale were tested using GPF and VPF. Thus, the performance of the two algorithms has been compared with the same experimental setup.

The first sequence contains 150 frames of images, the size of each frame is  $768 \times 576$  and the size of template is  $51 \times 42$ . The target to track undergoes large scale change in the sequence. For the particle filtering in the visual tracking, the number of particles is set to 60. The Standard Deviations of the six affine parameters in 16 are assigned as (0.04, 0.003, 0.003, 0.04, 4, 4). The final tracking results of GPF and VPF are shown in Figure 3. For a better visualization, we just show the tracking results of four representative frames 52, 87, 135 and 148. The frame number is shown on the top left corner of each image. The value below each image is the likelihood of the matching, the smaller the matching error, the larger the likelihood. Figure 4(a) shows the likelihood curves.

From Figure 3, we see that the proposed tracking algorithm exhibits a robust tracking result and the tracking window adapts with the scale change of the target. While VPF tracker begins to drift away from the target from frame 135. This is due to the fact that the VPF treats the parameter space as a whole. There are not enough observations to provide a reliable estimate. While GPF considers the geometry of the parameter space, this prior of smooth changes of the parameter space. From Figure 4(a), we see that the likelihoods of GPF tracker are always larger than VPF tracker. The second sequence contains 370 frames of images, the size of each frame is  $352 \times 420$  and the size of template is  $60 \times 40$ . The target to track experiences large rotation change and shear change in the sequence. The number of particles is set to 60. The Standard Deviation of the six affine parameters in (16) are assigned as (0.04, 0.0003, 0.003, 0.04, 4, 4). The final tracking results of GPF and VPF are shown in Figure 5. Like sequence 1, we just show the tracking results of four representative frames 165, 281, 337 and 364. We see that the proposed tracking algorithm exhibits a robust tracking result and the tracking window adapts with the deformation of the target. While the tracking window of VPF cannot enclose the target well. So, the likelihoods are smaller than GPF. Figure 4(b) shows the likelihood curves. We see that in the first 150 frames the likelihoods of the two trackers are similar, but from the 150th frame the likelihood of GPF is always larger than VPF. This is due to the fact that the target does not experience rotation and shear changes before the 150th frame, just translation.

In summary, we observe that the GPF tracker outperforms VPF in the scenarios of scale, rotation, and shear changes of target.



FIGURE 3: Tracking results of sequence 1: (a) tracking using VPF; (b) tracking using GPF.

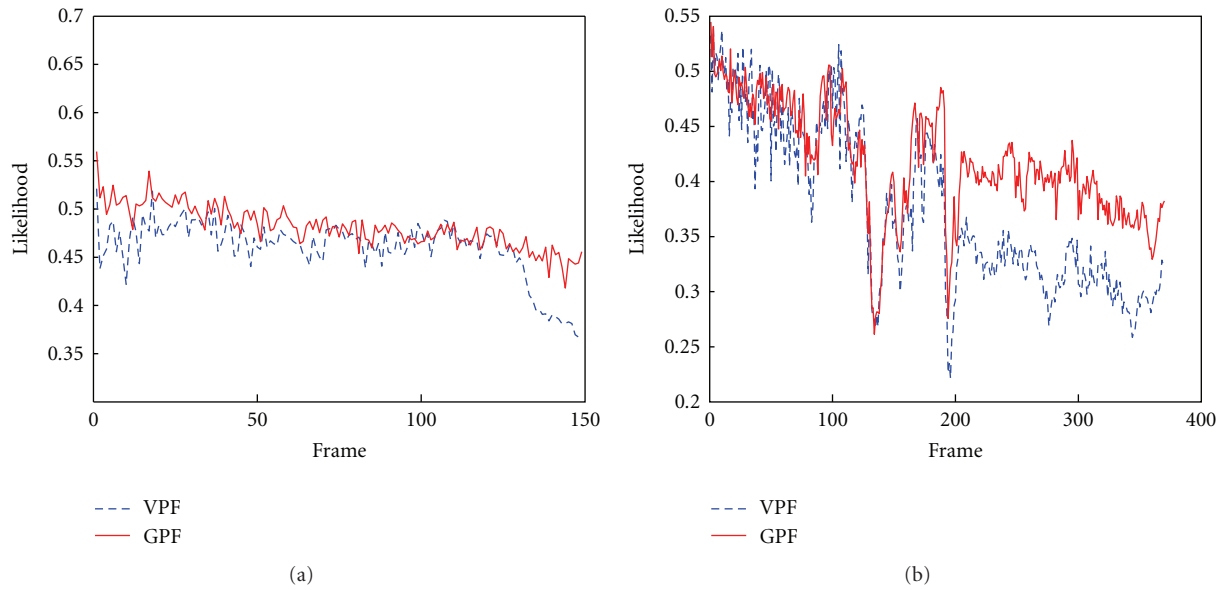


FIGURE 4: Performance comparison between VPF and GPF: (a) sequence 1; (b) sequence 2.

**5.2. Update Method.** To evaluate the effectiveness of the proposed update solution, we compare the result of it with the ones obtained by the Porikli update proposed in [14]. We compare the likelihood curves between above two image sequences; the results were obtained by just changing the update method.

Figure 6 shows the likelihood curves of the two update methods. From Figure 6, we see that likelihoods curves are similar; this means the two updates are equivalent.

However, the distinct advantage of this new update method is the time execution. In Table 1, we show the

results in milliseconds of the two updates methods. The Porikli update time execution was measured considering a stack of five regions covariance matrices. The new update is much faster than the one proposed in [14], with an average performance of 0.6 ms.

**5.3. Illumination Changes.** To analyze the robustness against the illumination changes using the covariance descriptor, we have used the algorithm on several sequences with illumination changes. One of which is a vehicle driving at

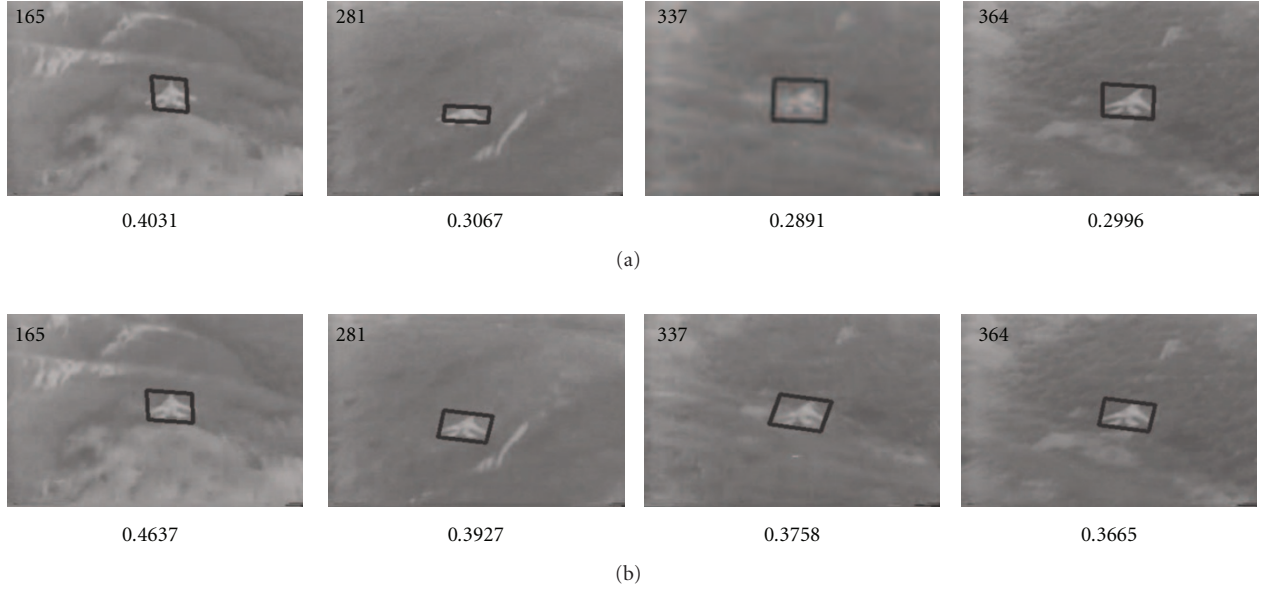


FIGURE 5: Tracking results of sequence 2: (a) tracking using VPF; (b) tracking using GPF.

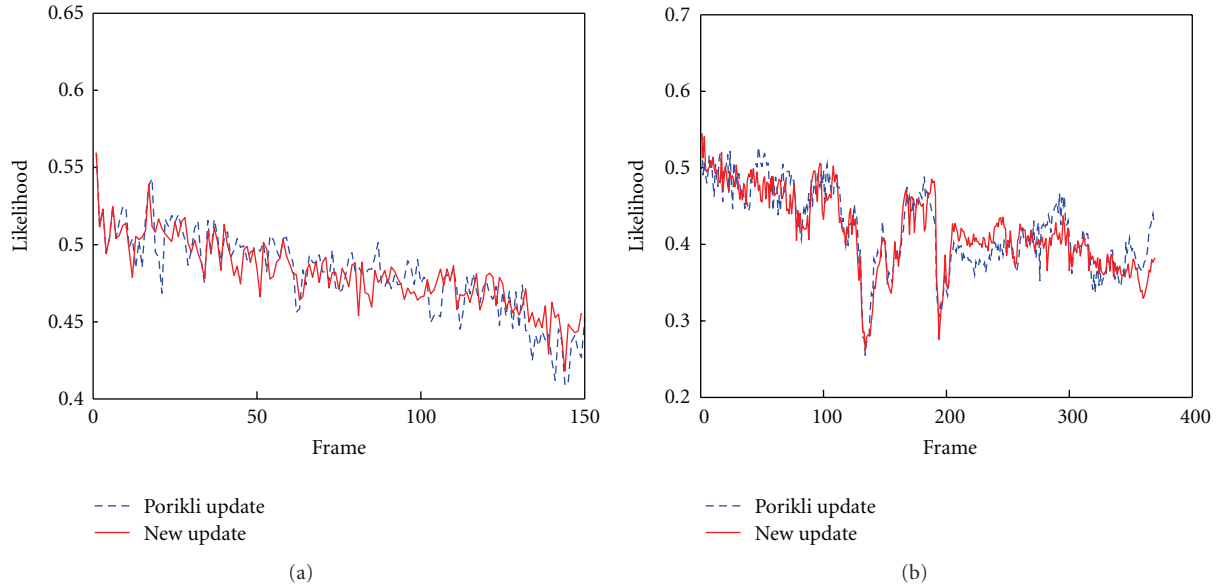


FIGURE 6: Performance comparison between two update methods: (a) sequence 1; (b) sequence 2.

TABLE 1: Execution time of two update methods size.

Method	Execution time (ms)
Porikli update	129.6
New update	0.6

night, shown in Figure 7(a). Despite the difficult illumination conditions, our algorithm is able to track the vehicle well. We also test the same image sequence using the image grayscale values. Tracking results are shown in Figure 7(b). we can see that from 280th frame, the tracking window drift away from the target, the red dashed window is the real

target. So the tracking algorithm using covariance descriptor outperformed gray-based tracking algorithm under illumination changes.

**5.4. Experimental Analyses.** The algorithm described in the paper consists of three components.

- (1) We develop a general particle filtering based tracking algorithm that explicitly take the geometry of affine Lie groups into consideration in deriving the state equation on Lie groups. This one is our main contribution and the dominating factor in improving the tracking performance.

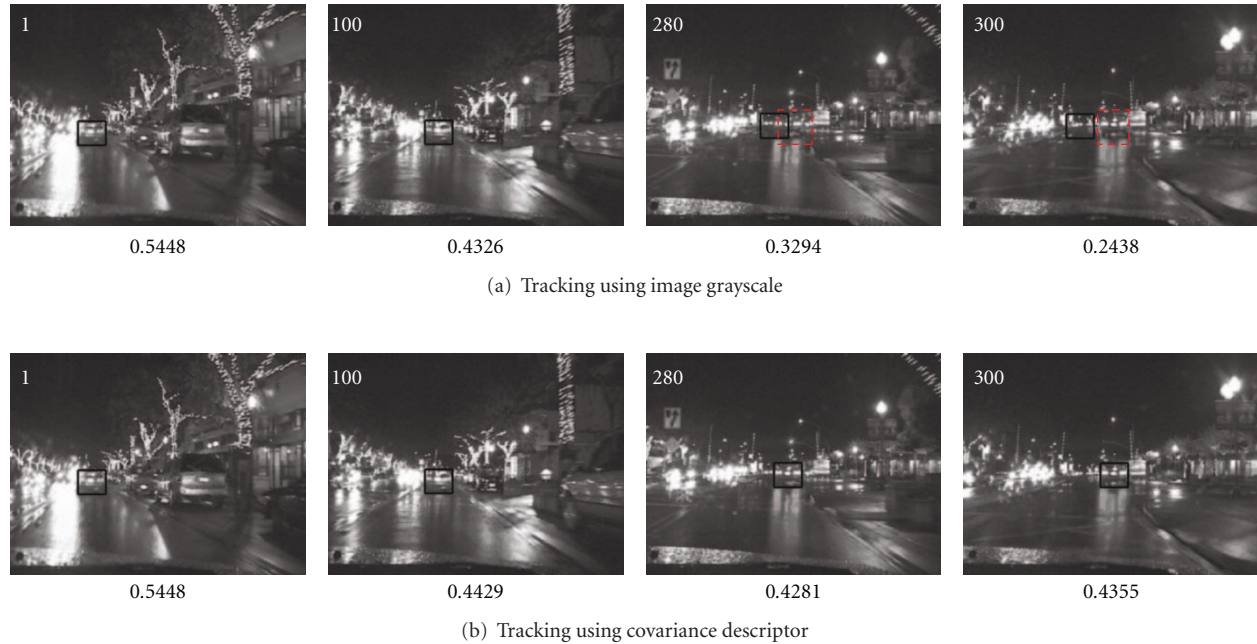


FIGURE 7: Vehicle moving in the night time with large illumination changes.

(2) We use region covariance descriptor to model objects appearance, the edge-like information more robust to the illumination changes than the image grayscale can be simultaneously considered with the image grayscale information and pixel spatial information, and the consequence is the quite robust tracking results as seen in Figure 7.

(3) We updated region covariance using a novel approach in a Riemannian space. The new update method has improved the real-time performance.

So the order of importance to the performance among these components is 1, 2, 3.

## 6. Conclusion

In this paper, we have proposed a visual tracking method, which integrate covariance descriptor into Mont Carlo tracking technique for visual tracking. The distinct advantage of this new approach is carrying Sequential Monte Carlo method over the affine Lie group, which consider the geometry prior of the parameter space. Theoretic analysis and experimental results shows the promise and effectiveness of the approach proposed.

This paper highlights the role of Monte Carlo methods in statistical inferences over affine lie group for visual tracking problem. There are several directions for extending the new idea. One is to consider more general differentiable manifolds beyond the affine lie group. In addition, we can deepen and broaden this research to other image processing problems.

## Acknowledgments

This work is partly supported by the National Natural Science Foundation of China (Grant no. 60603097) and the National Defense Innovation Foundation of Chinese Academy Sciences (CXJJ-65).

## References

- [1] D. A. Montera, S. K. Rogers, D. W. Ruck, and M. E. Oxley, "Object tracking through adaptive correlation," *Optical Engineering*, vol. 33, pp. 294–302, 1994.
- [2] H. S. Parry, A. D. Marshall, and K. C. Markham, "Tracking targets in FLIR images by region template correlation," in *Acquisition, Tracking, and Pointing XI*, vol. 3086 of *Proceedings of SPIE*, pp. 221–232, Orlando, Fla, USA, April 1997.
- [3] G. D. Hager and P. N. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 10, pp. 1025–1039, 1998.
- [4] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: a unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004.
- [5] H. Zhang, W. Huang, Z. Huang, and L. Li, "Affine object tracking with kernel-based spatial-color representation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 293–300, June 2005.
- [6] M. Isard and A. Blake, "Condensation-conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.
- [7] S. K. Zhou, R. Chellappa, and B. Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters," *IEEE Transactions on Image Processing*, vol. 13, no. 11, pp. 1491–1506, 2004.



- [8] Y. Rathi, N. Vaswani, A. Tannenbaum, and A. Yezzi, "Tracking deforming objects using particle filtering for geometric active contours," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 8, pp. 1470–1475, 2007.
- [9] J.-M. Odobez, D. Gatica-Perez, and S. O. Ba, "Embedding motion in model-based stochastic tracking," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3514–3530, 2006.
- [10] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1–3, pp. 125–141, 2008.
- [11] A. Srivastava and E. Klassen, "Monte Carlo extrinsic estimators of manifold-valued parameters," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 299–308, 2002.
- [12] H. Snoussi and A. Mohammad-Djafari, "Particle filtering on Riemannian manifold," in *Proceedings of the 27th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, vol. 872 of *AIP Conference Proceedings*, pp. 219–226, 2006.
- [13] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: a fast descriptor for detection and classification," in *Proceedings of the 9th European Conference on Computer Vision (ECCV '06)*, vol. 3952 of *Lecture Notes in Computer Science*, pp. 589–600, 2006.
- [14] F. Porikli, O. Tuzel, and P. Meer, "Covariance tracking using model update based on Lie algebra," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, vol. 1, pp. 728–735, New York, NY, USA, June 2006.
- [15] B. C. Hall, *Lie Algebras, and Representations: An Elementary Introduction*, Springer, New York, NY, USA, 2003.
- [16] M. Berger, *A Panoramic View of Riemannian Geometry*, Springer, Berlin, Germany, 2003.
- [17] E. Begelfor and M. Werman, "How to put probabilities on homographies," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1666–1670, 2005.
- [18] V. M. Govindu, "Lie-algebraic averaging for globally consistent motion estimation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 1, pp. 684–691, Washington, DC, USA, July 2004.
- [19] O. Tuzel, R. Subbarao, and P. Meer, "Simultaneous multiple 3D motion estimation via mode finding on lie groups," in *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV '05)*, vol. 1, pp. 18–25, Beijing, China, October 2005.