Visual Tracking with Generative Template Model based on Riemannian Manifold of Covariances

Marcus Chen\textsuperscript{1⋆‡} Sze Kim Pang\textsuperscript{2†‡} Tat Jen Cham\textsuperscript{3⋆} Alvina Goh\textsuperscript{4†‡}
\textsuperscript{1}marcusch@pmail.ntu.edu.sg \textsuperscript{2}pszekim@dso.org.sg \textsuperscript{3}astjcham@ntu.edu.sg \textsuperscript{4}agoh@nus.edu.sg
\textsuperscript{*}Nanyang Technological University, Singapore \textsuperscript{†}National University of Singapore \textsuperscript{‡}DSO National Laboratories, Singapore

Abstract—Robust visual tracking is a research area that has many important applications. The main challenges include how the target image can be modeled and how this model can be updated. In this paper, we model the target using a covariance descriptor. This descriptor is robust to problems that commonly occur in visual tracking such as pixel-pixel misalignment, pose and illumination changes. We model the changes in the template using a generative process. We introduce a new dynamical model for the template update using a random walk on the Riemannian manifold where the covariance descriptors lie in. This enables us to jointly quantify the uncertainties relating to the kinematic states and the template in a principled way. The sequential inference of the posterior distribution of the kinematic states and the template is done using a particle filter. Our results show that this principled approach is robust to changes in illumination, pose and spatial affine transformation.

Keywords: Tracking, Particle filtering, Template update, Generative Template Model, Riemannian manifolds.

I. INTRODUCTION

Visual tracking has been an active research area for decades with a wide range of applications [1]. Despite the extensive research done in this area, many challenges still remain unsolved today. It is well-known that tracking over a long video sequence is an extremely difficult problem due to the temporal variation of poses, illuminations, occlusions. Figure 1 shows two examples of how a target can vary over a video sequence. There are several choices for the target model found in existing literature. For example, [3] uses the histogram of gradients while [4] uses the color histogram as the template. As illustrated by Figure 1, one can see that in order to track the target robustly, it is necessary to update the target model or template over time in an appropriate manner. However, as shown in [2], target model updating or template updating is an extremely difficult problem. If the template is not updated at all, the template would soon become out of date and not representative as the target appearance undergoes changes temporally [2]. On the other hand, if the template is updated every frame, it is observed that the target gradually drifts out of the template, eventually resulting in the loss of the target [2]. This phenomenon known as template drift. Many template updating methods have been proposed in current literature; however, these techniques are often very sensitive to pixel-wise alignment and small temporal variation in the target appearance. In addition, they are unable to recover the template if a bad decision is made earlier in time.

One common and intuitive method of target model updating is to update the template \( \overline{T_t} \) based on the posterior estimate on the position of the target. More precisely, this method uses new target model \( T_t \), and combines with the last updated target model \( \overline{T_{t-1}} \) linearly with some forgetting factor, to obtain the updated template

\[
\overline{T_t} = f \left( T_t, \overline{T_{t-1}} \right),
\]  

where \( f \) can be a linear function such as averaging. The Kalman filter or particle filter is commonly used in the calculation of the set of the possible target models \( \{ T_t \} \).

We will now review the state-of-art algorithms on template updating, that is, the calculation of \( \overline{T_t} \). There are three well-known template updating algorithms: template alignment [2], online Expectation and Maximization (EM) [5], and incremental subspace method [6]. We now briefly review these three algorithms.
In the template alignment algorithm [2], the authors proposed a heuristic but robust algorithm by using the first template to correct any template drift. The problem formulation is as follows. The deformation parameters $P$ at the $t^{th}$ frame are found via the following minimization problem,

$$P_t = \arg \min_P \sum_{x \in T_t} [I_t(W(x; P) - T_t(x))^2], \quad (2)$$

where $x = (x, y)^T$ are the coordinates of a pixel, $W(x; P)$ is the warping function, $I_t(\cdot)$ is the intensity value of the image and $T_t(\cdot)$ is the template at the $t^{th}$ frame. [2] shows how a gradient descent algorithm starting with $P = P_{t-1}$ is used to solve Eqn. (2) and that this equation can be rewritten as

$$P_t = \text{gd} \min_{P = P_{t-1}} \sum_{x \in T_t} [I_t(W(x; P) - T_t(x))^2], \quad (3)$$

where $\text{gd} \min$ means “perform a gradient descent minimization starting with $P = P_{t-1}$”. In order to correct for the template drift, an alternative update of the parameters is proposed below,

$$P_t^* = \text{gd} \min_{P = P_t} \sum_{x \in T_t} [I_t(W(x; P) - T_1(x))^2]. \quad (4)$$

Eqn. (4) shows the correction of drift is done against the first template $T_1$. The template is only updated if the solved optimal deformation parameters in Eqn. (4) do not differ much from that in Eqn. (2) by enforcing the following condition:

$$T_{t+1}(x) = \begin{cases} I_t(W(x; P_t^*)) & \text{if } \|P_t^* - P_t\| \leq \varepsilon, \\ T_t(x) & \text{otherwise.} \end{cases} \quad (5)$$

Figure 2 shows some of the frames used in an example video sequence used in [2]. The constraint imposed in Eqn. (5) restricts any large changes of the current template from both the previously updated template and first template. However, such a constraint, which is imposed on every pixel, is not valid in many scenarios. For example, it is unlikely that a single set of optimal deformation parameters can be found for a video sequence with a soccer player running on the field as there is a wide variation in poses as well as an issue with pixel-pixel misalignment.

The second algorithm, online EM, is proposed in [5]. More precisely, three templates are used to capture three types of variation in the target wavelet-based appearance of the target. The first template is known as the long term stable template. This template is learned over a long sequence, varies slowly temporally, and aims to capture stable appearance of the target. The second template is the interframe variational template. This template changes every frame and its aim is to capture the sudden changes in the target due to changes in illumination or pose. The last template is the outlier template. This template models the occlusion of the target, outliers or missing tracking.

Specifically, the target is modeled using mixture models. Each of the three templates consists of $N$ pixels and each pixel is modeled independently by a Gaussian distribution with a mean and variance. Therefore, there is a total of $3N$ Gaussian models, and one would need to estimate $6N$ Gaussian parameters and another $2N$ mixture ratios. The target model is updated via the incremental update of the $8N$ Gaussian parameters using online Expectation and Maximization method. Since each pixel is modeled independently of each other, pixels with small variation contribute more to the similarity measure. Thus, this would result in a tendency to include more and more background pixels and the templates would drift quickly to the more stable background.

The last algorithm is known as incremental subspace update, proposed in [6]. A subspace is used to represent the target and the subspace is updated via incremental Principal Component Analysis (PCA). PCA is performed on the most likely estimator of the target template together with the previous stored templates and the mean is estimated. The authors tested this algorithm on many datasets and showed that their algorithm is very robust. Figure 3 shows an example of the results.

Figure 3. The $1^{st}$ row shows an example frame. The $2^{nd}$ row images are the current sample mean, tracked region, reconstructed image, and the reconstruction error respectively. The $3^{rd}$ and $4^{th}$ rows are the top 10 principal eigenvectors.

There are a few issues with the incremental PCA approach. First of all, it is well-known that PCA seeks to maximize the variance, and hence, it is reasonable to assume the first eigenbasis which corresponds to the largest eigenvalue to be the most stable template, whereas the subsequent eigenbases corresponding to smaller eigenvalues to be the less stable templates. However, PCA assumes that the target templates are distributed in a Gaussian manner. While this assumption generally holds true for slow interframe variation, it is not true otherwise. This can be seen in Figure 4, taken from [6]. One can see that from frames 600 to 636, the eigenbases are not representative anymore and the tracker loses track of the target. During these frames, the difference between the target template and target region is more than the difference between target template and background region due to pixel-wise
Paper contributions. To the best of our knowledge, almost all the state-of-art algorithms use out-of-chain template updating methods. More precisely, updating of the template model is done after obtaining the posterior estimate of the target's position. In this paper, we propose a method to update the target model in tandem with the target kinematics. We model the target template as part of the state space. We choose the covariance descriptor for the target descriptor as it is more robust to problems such as pixel-pixel misalignment and changes in pose and illumination. Since positive definite covariance matrices form a Riemannian manifold, we model the target template model variation by a random walk on the covariance Riemannian manifold. Finally, we perform sequential inference using a particle filter due to the non-linear dynamical model and multi-modal posterior.

The paper is organized as follows. We review the covariance descriptor and the Riemannian manifold which this descriptor lies on in Section II. Section III formulates the template updating problem using a Bayesian framework. Section IV gives an analysis of our formulation using multi-dimensional scaling for visualization. Section V describes our experiments and results. Finally, Section VI gives the conclusion.

II. MODELING TARGET USING RIEMANNIAN MANIFOLD OF COVARIANCE MATRICES

In this section, we will briefly describe the feature descriptor we use in the modeling of the template. The covariance descriptor is proposed in [7] as a feature descriptor. The covariance descriptor has been used in existing literature for a wide variety of applications, ranging from image segmentation to diffusion tensor imaging [8], [9], [10]. Basically, this descriptor contains the covariance of features such as pixel coordinates, intensity, gradients etc. and is defined as

$$C = \frac{1}{N-1} \sum_{i=1}^{N} (f(i) - \bar{f})(f(i) - \bar{f})^T$$

where $f = [f_1, f_2, \ldots, f_d]$ is a feature vector, and

$$\bar{f} = \frac{1}{N} \sum_{i=1}^{N} (f(i))$$

is the mean of the feature vector over $N$ pixels in the target region. For example, the feature vector used in [11] for human detection is

$$f(i) = [x, y, |I_x|, |I_y|, \sqrt{I_x^2 + I_y^2}, |I_{xx}|, |I_{yy}|, \arctan(I_{xy}/I_{xx})],$$

dimension $d = 8$ where $x, y$ are the spatial coordinates, and $\{I_x, I_y\}$ and $\{I_{xx}, I_{yy}\}$ are the first and second image derivatives in the $x$ and $y$ directions. In this paper, we use

$$f(i) = [x_w, y_w, I(x_w, y_w), |I_x|, |I_y|, |I_x||I_y|]$$

where $x_w, y_w$ are the coordinates of the warped template, $I(x_w, y_w)$ is the pixel intensity and $I_x, I_y$ are the gradients in the $x$ and $y$ directions, and therefore, $d = 6$. The reasons behind choosing the covariance descriptor [7] to model the template include:

1) The distance between two covariance matrices is invariant to scaling/offset of any features $f_i$. As a result, the first two features $(x_w, y_w)$ do not require any normalization.
2) A single covariance matrix extracted from a region is usually sufficient to match the region in different views and poses. Large rotations and illumination changes are taken care of in the covariance matrices.
3) The covariance matrix of features has a much smaller dimensionality of $(d^2 + d)/2$, compared to using the raw values of the region.
4) The covariance matrix is a natural way of fusing multiple possibly correlated features.

Now, notice that by definition, the covariance matrix is a positive definite matrix. Now, it is well-known [12]–[14] that the covariance matrices do not lie on Euclidean space; rather they lie on a Riemannian space. We will now briefly review the Riemannian space of the covariance matrices.

Figure 5 shows an example of a manifold $\mathcal{M}$. The tangent space of $\mathcal{M}$ at a point $C_i \in \mathcal{M}$, denoted as $T_{C_i}\mathcal{M}$, is defined as the span of the tangent vectors for all the possible smooth curves $\gamma$, $\gamma(t): \mathbb{R} \rightarrow \mathcal{M}$, passing through $C_i$. A curve between two points $C_i$ and $C_j$ with minimum length is called a geodesic. We define the exponential map $\exp_{C_i} : T_{C_i}\mathcal{M} \rightarrow \mathcal{M}$, which maps each tangent vector $t_y \in T_{C_i}\mathcal{M}$ to the point $\gamma(t) \in \mathcal{M}$ obtained by following the geodesic $\gamma(t)$ (parametrized with arc-length) passing through $C_i$ with direction $y$ for a distance $t$. The logarithm map is defined as $\log_{C_i} = \exp_{C_i}^{-1}$. Table I shows the operations in Euclidean and Riemannian spaces.

The Riemannian space of covariance matrices has been extensively studied [14] and the Riemannian operations can be found in closed form. For two tangent vectors $y_k, y_l \in T_{C_i}\mathcal{M}$ at a point $C_i \in \mathcal{M}$, the Riemannian metric is given as

$$\langle y_k, y_l \rangle_{C_i} = \text{trace}(C_i^{-\frac{1}{2}}y_k C_i^{-\frac{1}{2}}y_l C_i^{-\frac{1}{2}}).$$
The exponential map is
\[ C_j = \exp_{C_i}(y) = C_i^{1/2} \exp \left( C_i^{-1/2} y C_i^{-1/2} \right) C_i^{1/2}, \]
and the logarithm map is
\[ y = \log_{C_i}(C_j) = C_i^{1/2} \log \left( C_i^{-1/2} C_j C_i^{-1/2} \right) C_i^{1/2}. \]
Note that \( \exp_{C_i}(\cdot) \) and \( \log_{C_i}(\cdot) \) are both \( d \times d \) matrices. In addition, \( \exp_{C_i}(\cdot) \) and \( \log_{C_i}(\cdot) \) are maps on the Riemannian manifold, whereas \( \exp(\cdot) \) and \( \log(\cdot) \) denote the normal matrix exponential and logarithmic operations. The distance between two covariance matrices \( C_i \) and \( C_j \) is given as
\[ d(C_i, C_j) = \sqrt{\sum_{k=1}^{d} \ln^2 \lambda_k(C_i, C_j)}, \]
where \( \lambda_k(C_i, C_j) \) are the generalized eigenvalues of \( C_i \) and \( C_j \). That is, \( \lambda_k C_i v_k = \lambda_k C_j v_k = 0 \), and \( d \) is the dimension of the covariance matrices.

### III. BAYESIAN FRAMEWORK

In this section, we will describe our proposed template updating algorithm. Unlike existing methods, we approach the difficult task of template updating by using a generative model. The joint template update and target tracking problem can be formulated using the standard Bayesian formulation [16]. This is summarized by the following equation
\[ P(C_t, s_t|z_{1:t}) \propto P(z_t|C_t, s_t) P(s_t|s_{t-1}, C_{t-1}) P(C_{t-1}, s_{t-1}|z_{1:t-1}) ds_{t-1} dC_{t-1}, \]
where \( z_t \) is the measurement of the likelihood of the target, \( s_t \) is the kinetic state variables for target motions and scaling, \( C_t \) is the covariance descriptor is defined in Eqn. (6) of the target template.

#### A. Dynamical Model

In our model, we treat both kinetic parameters \( s_t \) and target template \( C_t \) as state variables to be estimated. \( s_t \) and \( C_t \) are defined as follows
\[ s_t = [x_t, y_t, \dot{x}_t, \dot{y}_t, h_t, \theta_t], \]
\[ C_t = \text{cov} (x_w, y_w, I(x_w, y_w), |I_{x_w}|, |I_{y_w}|, |I_{x_w}|/|I_{y_w}|), \]
where \( x_t, y_t \) are the spatial coordinates of the target region at time \( t \), \( \dot{x}_t, \dot{y}_t \) are the velocities, \( h_t \) is the scaling factor, and \( \theta_t \) is the orientation. \( x_w, y_w \) are the coordinates of a pixel on the standard target patch warped from \( x_t, y_t, I(x_w, y_w) \) is the pixel intensity and \( \{I_x, I_y\} \) are the image gradients. We have chosen to model the dynamics as
\[ P(s_t, C_t|s_{t-1}, C_{t-1}) = P(s_t|s_{t-1})P(C_t|C_{t-1}), \]
with the following dynamical state equations,
\[ s_t = k(s_{t-1}) + u_t, \]
\[ C_t = \exp_{C_{t-1}}(n_t). \]
\( k \) is the kinematic model and we use a near constant velocity linear model \( k(s_{t-1}) = As_{t-1} \), where \( A = [1 0 0 0 0 0; 0 1 0 0 0 0; 0 0 1 0 0 0; 0 0 0 1 0 0; 0 0 0 0 0 1] \) is a constant matrix describing a linear model. \( u_t \) is a Gaussian dynamical noise with diagonal covariance. Here, \( n_t \) is the random vector on the tangent space at the point \( C_{t-1} \), such that the exponential map will project it back to a random point on the manifold close to \( C_{t-1} \). We know that \( C_{t-1} = UD^2U^\top \) where \( U \) is an orthonormal matrix and \( D \) is a diagonal matrix containing the square root of positive eigenvalues. Therefore, we obtain \( C_t \) by perturbing the square root eigenvalues \( D \), \( C_t = UD^2U^\top \), where \( \sqrt{i^{th}} \) eigenvalue \( D^2 \sim N(D_i, \sigma_i^2) \) is normally distributed. We will show in Section IV the reasons behind performing a random walk on the Riemannian manifold.

#### B. Observation Model

The observation model measures the likelihood of a target in a specific region of the image. This is given by
\[ P(z_t|C_t, s_t) \sim N(0, \sigma^2), \]
\[ z_t = d(C_t, C_t^*), \]
\[ C_t^* = g(s_t, I_{\text{Image}}). \]
Here, \( d(C_t, C_t^*) \) is given by Eqn. (11), \( g \) is the covariance computation operator; \( g \) takes the kinematic value \( s_t \) of each particle at time \( t \), warps the region to a standard rectangular region before computing covariance. Notice that such a model given by Eqn. (18) does not account for the density of the background regions of the image as it implicitly assumes that the background density is constant for all template variations in each time step.
C. Overall Framework

We use a standard particle filter to do sequential inference. The particle filter [15], [16], [17] represents the distribution of state variables by a collection of samples and their weights. The advantage of using a particle filter is that it can deal with a non-linear system and multi-modal posterior. The algorithm of a particle filter is as follows

1) **Initialization.** The particle filter is initialized with a known realization of target state variables. This includes the target initial state values. Covariance of the target $C_0$ is extracted for comparison later.

2) **Propagation.** Each particle is propagated according to the propagation model in Eqns. (16) and (17).

3) **Measure the likelihood.** At each particle $i$, the covariance descriptor $C_t(i)$ of the proposed region is compared to $C_t^*(i)$. The likelihood of the particle is then estimated as given in Eqn. (18).

4) **Posterior estimation.** The posterior estimate gives the estimate of the current target state, given all its previous information and measurements.

5) **Resampling.** To avoid any degeneracies, resampling is conducted to redistribute the weight of particles.

6) **Loop.** Repeat the process from step 2 to 5 as time progresses.

---

IV. Analysis of Random Walk on the Riemannian Manifold

In this section, we show that the covariance descriptor is a good representation of the target as well as the reasons behind performing a random walk as described in Section III-A.

We first show that the distribution of the covariance matrices of the target. To this end, we use multidimensional scaling [18] to construct a visualization of the distribution of the covariance matrices. The distance matrix is constructed based on the Riemannian distance given in Eqn. (11). The visualization shows the relative positions of targets (red) and backgrounds (blue). Figure 6 shows that the covariance matrices of the target tend to cluster together. There are some background “inside” the target cluster, illustrating the need to model the evolution of targets.

Note that we do not use the ground truth of the labels “target” and “background” in MDS; here we are simply using it to demonstrate that covariance descriptor of the target is a good representation and the covariance matrices of the target tend to cluster closely together. This observation motivates us to model the template variation by using a random walk on the Riemannian manifold. Based on Eqn. (17), Figure 7 illustrates a realization of the random walk. It shows that our template dynamic model can model the actual target variation.

V. Experiments and Results

We test our algorithm on some of popular tracking datasets. These include David Ross’s sequences, such as the David, Handsome Fellow, toy dog and car sequences, found at http://www.cs.toronto.edu/~dross/ivt/ , and the vehicle tracking sequences from PETS2001, and the soccer sequences from PETS2006. We compare our method with a similar implementation but without temporal update (denoted as PF+Cov). In PF+Cov, we use the same set of features as our method. Our results are shown in red and PF+Cov in green in Figures 8-13. Figure 8 shows results of the PETS2001 vehicle sequence, Figure 9 shows results of the toy dog sequence, Figure 10 shows results of David sequence, Figure 11 shows results of one soccer (PETS2006) sequence, Figure 12 shows results of another soccer (PETS2006) sequence and Figure 13 shows results of a car sequence.

In the vehicle sequence shown in Figure 8 and the toy dog sequence shown in Figure 9, we observe that PF+Cov locks partially onto the background whereas our algorithm does not exhibit this behavior. In the first soccer sequence shown in Figure 11, PF+Cov locks onto a wrong target nearby. This is explained by the variation of the template temporally as PF+Cov only keeps the first template for the likelihood estimate in Eqn. (18). In Figure 8, the zooming-in frames show that the target becomes bigger and appear in more details, it is required that the template be updated gradually as the words on the back of the truck appears. In Figures 9 and 11, the target changes quite a lot due to pose changes, and thus the original template cannot match well. This causes PF+Cov to lock on to a small portion of the target or lose track. In Figure 12, PF+Cov locks on the player’s legs. These results show that our method can indeed model the template variations temporally to a good extent.

In the David sequence, Figure 10, PF+Cov performs comparable with ours. This could be due to the target template undergoing no significant changes. Figure 13 shows an example of a car sequence where our method does not perform satisfactorily. Our method locks onto the background when PF+Cov drifts a bit. The possible explanation is the our template dynamics is unable to account for this dramatic and non-smooth transition of the template when the car goes into a shadowed region. PF+Cov manages to lock on because the
Figure 6. Covariance matrices of "track David" and "soccer" sequences visualized using Multidimensional scaling. Red: target patches, Blue: background patches.

Figure 8. Tracking results on PETS2001 vehicle sequence, frame #1, 21, 41, 61, 101, 141, Green: PF+Cov, Red: our results. PF+Cov locks partially on the background.

car appears to be the same as the starting frame after emerging from the shadow.

VI. CONCLUSIONS

In this paper, we present a new method to update target model in tandem with the target kinematics. More precisely,
we have developed a generative template model in a principled way within a Bayesian framework. By jointly quantifying the uncertainties of the target kinematics and template, we are able to achieve more robust visual tracking. We have chosen the covariance descriptor as the target representation. We have modeled the target template model dynamic using a random walk on the Riemannian manifold which the covariance matrices lie in. Our template dynamic model is an example of
a diffusion process on the covariance Riemannian manifold. Experimental results show that our framework is able to obtain reasonable performance on several sequences. Future work includes addressing a number of questions such as how should the diffusion speed be adjusted and can the diffusion process be better constrained.

ACKNOWLEDGEMENTS

We would like to thank DSO National Laboratories, Singapore for partially sponsoring this work, and David Ross for sharing the test sequences.

REFERENCES