# Real-time Probabilistic Covariance Tracking with Efficient Model Update

Yi Wu, Jian Cheng, *Member, IEEE,* Jinqiao Wang, *Member, IEEE,* Hanqing Lu, *Senior Member, IEEE,* Jun Wang, Haibin Ling, *Member, IEEE,* Erik Blasch, *Senior Member, IEEE,* and Li Bai, *Senior Member, IEEE*

**Abstract**—The recently proposed covariance region descriptor has been proven robust and versatile for a modest computational cost. The covariance matrix enables efficient fusion of different types of features, where the spatial and statistical properties as well as their correlation are characterized. The similarity between two covariance descriptors is measured on Riemannian manifolds. Based on the same metric, but with a probabilistic framework, we propose a novel tracking approach on Riemannian manifolds with a novel incremental covariance tensor learning (ICTL). To address the appearance variations, ICTL incrementally learns a low-dimensional covariance tensor representation and efficiently adapts online to appearance changes of the target with only $\mathcal{O}(1)$ computational complexity, resulting in a real-time performance. The covariance-based representation and ICTL are then combined with the particle filter framework to allow better handling of background clutter as well as the temporary occlusions. We test the proposed probabilistic ICTL tracker on numerous benchmark sequences involving different types of challenges including occlusions and variations in illumination, scale, and pose. The proposed approach demonstrates excellent real-time performance, both qualitatively and quantitatively, in comparison with several previously proposed trackers.

**Index Terms**—Visual tracking, particle filter, covariance descriptor, Riemannian manifolds, incremental learning, model update.

## 1 Introduction

Visual tracking is a challenging problem, which can be attributed to the difficulty in handling the appearance variability of a target. In general, appearance variations can be divided into two types: intrinsic and extrinsic. The intrinsic appearance variations include pose change and shape deformation, whereas the extrinsic variations include changes in illumination and camera viewpoint, and occlusions. Consequently, it is imperative for a robust tracking algorithm to model such appearance variations to ensure real-time and accurate performance.

Appearance models in visual tracking approaches are often sensitive to the variations in illumination, view, and pose. Such sensitivity results from a lack of a competent object description criterion that captures both statistical and spatial properties of the object appearance. Recently, the covariance region descriptor (CRD) is proposed in [39] to address these sensitivities by capturing the correlations among extracted features inside an object region.

Using the CRD as the appearance model, we propose a novel probabilistic tracking approach via Incremental Covariance Tensor Learning (ICTL). In contrast to the covariance tracking algorithm [33], with the tensor analysis, we simplify the complex model update process on the Riemannian manifold by computing the weighted sample covariance, which can be updated incrementally during the object tracking process. Thus our appearance model can update more efficiently, adapt to extrinsic variations, and afford object identification with intrinsic variations - which is the main contribution of our work. Further, our ICTL method uses a particle filter [13] for motion parameter estimation rather than the exhaustive search-based method [33] which is very time-consuming and often distracted by outliers. Moreover, the integral image data structure [32] is adopted to accelerate the tracker.

In summary, our proposed tracking framework includes two stages: (a) probabilistic Bayesian inference for covariance tracking; and (b) incremental covariance tensor learning for model update. In the first stage, the object state is obtained by a maximum a posterior (MAP) estimation within the Bayesian state inference framework in which a particle filter is applied to propagate sample distributions over time. In the second stage, a low dimensional covariance model is learned online. The model uses the proposed ICTL algorithm to find the compact covariance representation in the multi-modes. After the MAP estimation of the Bayesian inference, we use the covariance matrices of image features associated with the estimated target state to update the compact covariance tensor model for each mode. The two stage architecture is executed repeatedly as time progresses as shown in Fig. 1. Moreover, with the use of tensors of integral images, our tracker achieves real-time

*This work was supported in part by 973 Program (Project No. 2010CB327905), the National Natural Science Foundation of China (Grant No. 61005027, 60835002, 60833006, 60905008) and NSF (Grants No. IIS-0916624 and IIS-1049032).*

- *Yi Wu is with the School of Information and Control Engineering, Nanjing University of Information Science and Technology, Nanjing, China, 210044.*
- *Jinqiao Wang is the corresponding author. He is with the Institute of Automation, Chinese Academy of Sciences, Beijing, China, 100190. E-mail: jqwang@nlpr.ia.ac.cn*
- *Jian Cheng, Hanqing Lu are with the Institute of Automation, Chinese Academy of Sciences, Beijing, China, 100190.*
- *Jun Wang is with the Network Center, Nanjing University of Information Science and Technology, Nanjing, China, 210044.*
- *Haibin Ling is with the Department of Computer and Information Science, Temple University, Philadelphia, PA 19122 USA.*
- *Erik Blasch is with the US Air Force Research Laboratory (AFRL), AFRL/RYAA, 2241 Avionics Cir, WPAFB, OH 45433.*
- *Li Bai is with the Department of Electrical and Computer Engineering, Temple University, Philadelphia, PA 19122 USA.*

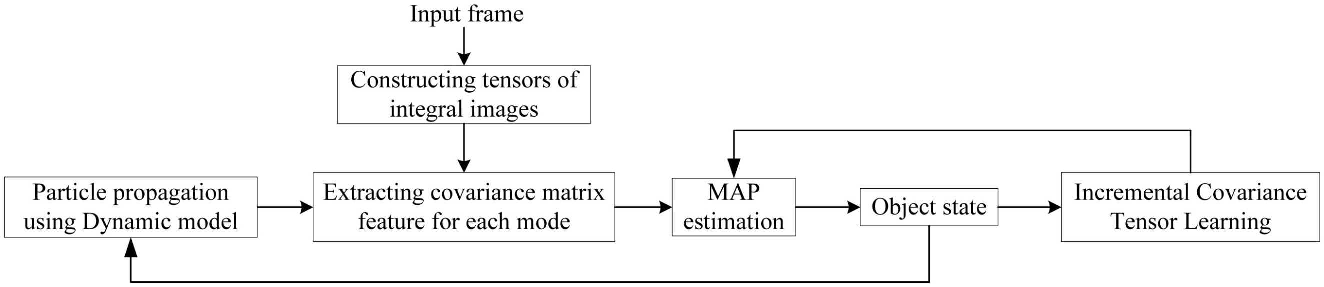

Input frame
Constructing tensors of integral images
Particle propagation using Dynamic model
Extracting covariance matrix feature for each mode
MAP estimation
Object state
Incremental Covariance Tensor Learning

Fig. 1. Overview of the proposed tracking approach.

performance.

## 2 Related work

There is a rich literature in visual tracking and a thorough discussion on this topic is beyond the scope of this paper. There are many uses of covariance information in target tracking such as covariance intersection for measurement-based tracking from multiple sensors [14], covariance control for sensor scheduling and management [16], [41], etc. Given the widespread use of covariance analysis in target tracking, in this section we review only the most relevant visual tracking work that motivated our approach, focusing on target representation and model update.

### 2.1 Target representation

Target representation is one of major components in typical visual trackers and extensive studies have been presented. Histograms prove to be a powerful representation for an image region. Discarding the spatial information, the color histogram is robust to the change of object pose and shape. Several successful tracking approaches utilize color histograms [8], [26]. Recently, Stanley *et al.* [6] proposed a novel histogram, named spatiogram, to capture not only the values of the pixels but their spatial relationships as well. To calculate the histogram efficiently, Porikli *et al.* [32] proposed a fast way to extract histograms called the integral histogram. Recently, sparse representation has been introduced for visual tracking via the $\ell_1$-minimization [23] and been further extended in [25], [44], [17], [22], [42], [21].

The covariance region descriptor (CRD) proposed in [39] has been proved to be robust and versatile for a modest computational cost. The CRD has been applied to many computer vision tasks, such as object classification [12], [38], [36], human detection [40], [28], face recognition [29], action recognition [10] and tracking [33], [46], [45], [43]. The covariance matrix enables efficient fusion of different types of features and its dimensionality is small. An object window is represented as the covariance matrix of features, where the spatial and statistical feature properties as well as their correlations are characterized within the same representation. The similarity of two covariance descriptors is measured on Riemannian manifolds which we call the Manifold Covariance Similarity (MCS) metric. Porikli *et al.* [33] generalized the covariance descriptor to a tracking problem by exhaustively searching the whole image for the region that best matches the model descriptor (i.e. maximum likelihood estimation - MLE). Using the MLE covariance descriptor is time consuming, computationally inefficient, easily affected by background clutter, and ineffective over occlusions.

Improvement for such situations is one of the benefits of our proposed probabilistic ICTL tracking approach. Relying on the same MCS metric to compare two covariance descriptors, we embed it within a sequential Monte Carlo framework. To utilize the MCS requires building Riemannian manifold local likelihoods, coupling the manifold observation model with a dynamical state space model, and sequentially approximating the posterior distribution with a particle filter. Using the sample-based filtering technique enables tracking multiple posterior modes, which is the key to mitigate background distractions and to recover after temporary occlusions.

### 2.2 Appearance variations modeling

To model the appearance variations of a target, there have been many visual tracking approaches reported in the last decades. Zhou *et al.* [48] embedded appearance adaptive models into a particle filter to achieve a robust visual tracking. In [34], Ross *et al.* proposed a generalized visual tracking framework based on the incremental image-as-vector subspace learning methods with a sample mean update. The sparse representation of target [24], [25] is updated by introducing importance weights for the templates and identifying rarely used templates for replacement. To handle appearance changes, SVT [3] integrates an offline trained support vector machine (SVM) classifier into an optic-flow-based tracker. In [7], the most discriminative RGB color combination is learned online to build a confidence map in each frame. In [4], an ensemble of online learned weak classifiers is used to label a pixel as belonging to either the object or the background. To encode the object appearance variations, Yu *et al.* [47] proposed to use co-training to combine generative and discriminative models to learn an appearance model on-the-fly. In [15], Kalal *et al.* proposed a learning process guided by positive and negative constraints to distinguish the target from background.

For visual target tracking with a changing appearance, it is likely that recent observations will be more indicative of its appearance than more distant ones. One way to balance old and new observations is to allow newer images to have a larger influence on the estimation of the current appearance model than the older ones. To do this, a forgetting factor is incorporated in the incremental eigenbasis updates in [19]. Further, Ross *et al.* [34] provided an analysis of its effect on the resulting eigenbasis. Skocaj and Leonardis [37] presented an incremental method, which sequentially updates the principal subspace considering weighted influence of individual images as well as individual pixels in an image.

However, appearance models adopted in the above mentioned trackers are usually sensitive to the variations in illumination, view and pose. These tracking approaches lack a competent object description criterion that captures both statistical and spatial properties of the object appearance. The covariance region descriptor (CRD) [39] is proposed to characterize the object appearance, which is capable of capturing the correlations among extracted features inside an object region and is robust to some appearance variations. In the recently proposed covariance tracking approach [33], the Riemannian mean under the affine-invariant metric is used to update the target model. Nevertheless, the computational cost for the Riemannian mean grows rapidly as time progresses and is very time-consuming for long-term tracking. Based on the Log-Euclidean Riemannian metric [2], Li *et al.* [20] presented an online subspace learning algorithm which models the appearance changes by incrementally learning an eigenspace representation for each mode of the target through adaptively updating the sample mean and eigenbasis.

Our work is motivated in part by the prowess of covariance descriptor as appearance models [39], the effectiveness of particle filters [13], and the adaptability of on-line update schemes [34]. In contrast to the covariance tracking algorithm [33], our algorithm does not require a complex model update process on Riemannian manifold but learns the compact covariance tensor representation incrementally during the object tracking process. Thus our appearance model can update more efficiently. Further, our method uses a particle filter for motion parameter estimation rather than the exhaustive search-based method [33] which is very time-consuming and often distracted by outliers. Moreover, with the help of integral images [32], our tracker achieves real-time performance. A preliminary conference version of this paper appears in [43].

# 3 Probabilistic covariance tracking

In this section, we first review the covariance descriptor [39] and particle filter [13], then the probabilistic covariance tracking approach is introduced.

## 3.1 Covariance descriptor

Let $I$ be the observed image, and $F$ be the $W \times H \times d$ dimensional feature image extracted from $I$, $F(x, y) = \Phi(I, x, y)$, where $\Phi$ can be any mapping such as color, gradients, filter responses, etc. Let $\{f_i\}_{i=1}^{N}$ be the $d$-dimensional feature points inside a given rectangular region $R$ of $F$. The region $R$ is represented by the $d \times d$ covariance matrix of the feature points

$$C = \frac{1}{N-1} \sum_{i=1}^{N} (f_i - \mu)(f_i - \mu)^T,$$

where $N$ is the number of pixels in the region $R$ and $\mu$ is the mean of the feature points.

The element $(i, j)$ of $C$ represents the correlation between feature $i$ and feature $j$. When the extracted $d$-dimensional feature includes the pixel's coordinate, the covariance descriptor encodes the spatial information of features.

With the help of integral images, the covariance descriptor can be calculated efficiently [39]. Specifically, $d(d+1)/2$ integral images are used such that the covariance descriptor of any rectangular region can be computed independent of the region size.

### 3.1.1 Metric on Riemannian manifolds

Supposing no features in the feature vector would be exactly identical, the covariance matrix is positive definite. Thus the nonsingular covariance matrix can be formulated as a connected Riemannian manifold, which is locally similar to a Euclidean space. For differentiable manifolds, the derivative at a point X lies in its tangent space denoted as $\mathrm{T_X}$. Each tangent space has an inner product $\langle \cdot, \cdot \rangle_{\mathrm{X}}$ and the norm for a tangent vector is defined by $\|\mathrm{y}\|_{\mathrm{X}}^2 = \langle \mathrm{y}, \mathrm{y} \rangle_{\mathrm{X}}$.

An invariant Riemannian metric on the tangent space is defined as $\langle y, z \rangle_{\mathrm{X}} = tr\left(\mathrm{X}^{-\frac{1}{2}} y \mathrm{X}^{-1} z \mathrm{X}^{-\frac{1}{2}}\right)$. The exponential map associated to the Riemannian metric is given by $\exp_{\mathrm{X}}(y) = \mathrm{X}^{\frac{1}{2}} \exp\left(\mathrm{X}^{-\frac{1}{2}} y \mathrm{X}^{-\frac{1}{2}}\right) \mathrm{X}^{\frac{1}{2}}$. The logarithm uniquely defined at all the points on the manifold is $\log_{\mathrm{X}}(y) = \mathrm{X}^{\frac{1}{2}} \log\left(\mathrm{X}^{-\frac{1}{2}} y \mathrm{X}^{-\frac{1}{2}}\right) \mathrm{X}^{\frac{1}{2}}$.

For a symmetric matrix, its exponential and logarithm are given respectively by $\exp(\Sigma) = U \exp(D) U^T$, and $\log(\Sigma) = U \log(D) U^T$, where $\Sigma = UDU^T$ is the eigenvalue decomposition of the symmetric matrix $\Sigma$. $\exp(D)$ and $\log(D)$ are the diagonal matrix of the eigenvalue exponentials and logarithms respectively.

The distance between symmetric positive definite matrices is measured by

$$d^2(\mathrm{X}, \mathrm{Y}) = \langle \log_{\mathrm{X}}(\mathrm{Y}), \log_{\mathrm{X}}(\mathrm{Y}) \rangle_{\mathrm{X}} = tr\left(\log^2(\mathrm{X}^{-\frac{1}{2}} \mathrm{Y} \mathrm{X}^{-\frac{1}{2}})\right) .$$

## 3.2 Sequential Inference Model

In the Bayesian perspective, object tracking can be viewed as a state estimation problem. At time $t$, denote the state of a target and its corresponding observation as $x_t$ and $y_t$, respectively. The state set from beginning to time $t$ is $x_{0:t}$, where $x_0$ is the initial state, and the corresponding observation set is $y_{0:t}$.

The purpose of tracking is to predict the future location and estimate the current state given all previous observations or equivalently to construct the filtering distribution $p(x_t|y_{0:t})$. Using the conditional independence properties, we can formulate the density propagation for the tracker as follows:

$$p(x_t|y_{0:t}) \propto p(y_t|x_t) \int p(x_t|x_{t-1}) p(x_{t-1}|y_{0:t-1}) dx_{t-1}.$$

For visual tracking problems, the recursion can be accomplished within a sequential Monte Carlo framework where the posterior $p(x_t|y_{0:t})$ is approximated by a weighted sample set $\{x_t^n, w_t^n\}_{n=1}^{N_s}$, where $\sum_{n=1}^{N_s} w_t^n = 1$. All the particles are sampled from a proposal density $q(x_t^n|x_{t-1}^n, y_t)$. The weight associated with each particle is formulated as follows:

$$w_t^n \propto \frac{p(y_t|x_t^n) p(x_t^n|x_{t-1}^n)}{q(x_t^n|x_{T-1}^n, y_t)} w_{T-1}^n \ .$$

To avoid weight degeneracy, the particles are resampled so that all of them have equal weights after resampling.

The common choice of proposal density is by taking $q(x_t|x_{t-1}, y_t) = p(x_t|x_{t-1})$. As a result, the weights become the local likelihood associated with each state $w_t^n \propto p(y_t|x_t^n)$. The Monte Carlo approximation of the expectation $\hat{x}_t = \frac{1}{N_s}\sum_{n=1}^{N_s} x_t^n \approx E(x_t|y_{0:t})$ is used for state estimation at time $t$.

### 3.3 Probabilistic covariance tracking

Based on the same Manifold Covariance Similarity (MCS) metric to compare two covariance descriptors on the Riemannian manifolds, the probabilistic covariance tracking approach embeds the MCS metric within a sequential Monte Carlo framework. To develop the manifold covariance approach requires the building of a local likelihood on Riemannian manifolds, the coupling of the observation model with a dynamical state space model, and the sequential approximation of the posterior distribution with a particle filter. The sample-based filtering technique enables tracking the multiple posterior modes, which is the key to mitigate the effects of background distractors and to recover from temporary occlusions.

Specifically, to measure the similarity between covariance matrices corresponding to the target model $C^*$ and the candidate $C(x_t^n)$, we use the Manifold Covariance Similarity metric on Riemannian manifolds. An exponential function of the distance is adopted as the local likelihood in the particle filter: $p(y_t|x_t^n) \propto \exp\{-\lambda d^2(C^*, C(x_t^n))\}$.

## 4 Incremental Covariance Tensor Learning for Model Update

The main challenge of visual tracking can be attributed to the difficulty in handling the appearance variability of a candidate object. To address the model update problem, we present a model update scheme to incrementally learn a low-dimensional covariance tensor representation and consequently adapts online the appearance changes with a constant computational complexity. Moreover, a weighting scheme is adopted to ensure less modeling power is expended to fit older observations with existing models. Both of these features significantly contribute to improve overall real-time tracking performance. In the following, we provide a detailed discussion of our proposed Incremental Covariance Tensor Learning (ICTL) algorithm for model update.

### 4.1 Object representation

In our tracking framework, an object is represented by multiple covariance matrices of the image features inside the object region, as shown in Fig.2. These covariance matrices correspond to the multiple modes of the object appearance. Without loss of generality, we only discuss one mode in the following.

As time progresses from $t = 1, \ldots, T$, all the object appearances form object appearance tensor $\mathcal{A} = \{A_t \in R^{m\times n}\}_{t=1}^T$, and $d$-dimensional feature vector is extracted for each element of $A_t$ forming a 4th-order object feature tensor $\mathcal{F} \in R^{m\times n\times d\times T}$. Flattening $\mathcal{F}$, we can obtain the matrix comprising its mode-3 vector (i.e., each column is a $d$-dimensional feature vector):

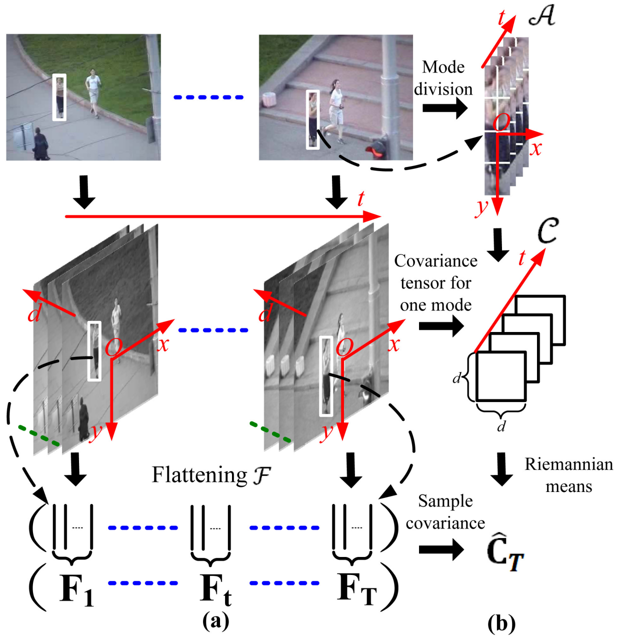


Fig. 2. Illustration of object representation, the flattening of $\mathcal{F}$ and two different formulations for $\hat{C}_T$. The input sequence is shown in the upper part of (a) while the fourth order object feature tensor $\mathcal{F}$ is displayed in the middle of (a). The result of flattening $\mathcal{F}$ is exhibited in the bottom of (a). The appearance tensor $\mathcal{A}$ with mode division is shown in the top of (b) while the covariance tensor for one mode in the middle of (b). The bottom of (b) displays two different formulations for $\hat{C}_T$.

$$F = (f_{1,1,1} f_{1,1,2} \cdots f_{1,2,1} \cdots f_{2,1,1} \cdots f_{t,y,x} \cdots f_{T,m,n}),$$

where $f_{t,y,x}$ denotes a $d$-dimensional feature vector at location $(x, y)$ at time $t$. Reforming $x$ and $y$ into one index $i$, $F$ can be represented neatly by

$$F = (f_{1,1} \cdots f_{1,N} \cdots f_{t,i} \cdots f_{T,N}) = (F_1 \cdots F_t \cdots F_T),$$

where $N = m \times n$, $F_t = (f_{t,1} \cdots f_{t,i} \cdots f_{t,N}) \in R^{d\times(m\cdot n)}$. The column covariance of $F_t$ can be represented as:

$$C_t = \frac{1}{N-1}\sum_{i=1}^{N} (f_{t,i} - \mu_t)(f_{t,i} - \mu_t)^T,$$

where $\mu_t$ is the column mean of $F_t$. This covariance can be viewed as an informative region descriptor for an object [39]. All the covariance matrices up to time $T$, $\{C_t \in R^{d\times d}\}_{t=1}^T$, constitute a covariance tensor $\mathcal{C} \in R^{d\times d\times T}$. We need to track the changes of $\mathcal{C}$ and as new data arrives, update the compact representation of $\mathcal{C}$.

A straightforward compact representation of $\mathcal{C}$ is the mean of $\{C_t \in R^{d\times d}\}_{t=1}^T$. Porikli *et al.* [33] calculated the mean of several covariance matrices through Riemannian geometry. The metric they used is the affine-invariant Riemannian metric. The distance between two covariance matrices X and Y under this Riemannian metric is computed by $\|\log(\mathrm{X}^{-\frac{1}{2}}\mathrm{Y}\mathrm{X}^{-\frac{1}{2}})\|$.

An equivalent form is given in [9]

$$\rho\left(\mathrm{X},\mathrm{Y}\right)=\sqrt{\sum\nolimits_{k=1}^{d}\ln^2\lambda_k\left(\mathrm{X},\mathrm{Y}\right)}\ , \tag{1}$$

where $\lambda_k(X,Y)$ are the generalized eigenvalues of X and Y. Under this metric, an iterative numerical procedure [30] is applied to compute the Riemannian mean. The computational cost for this Riemannian mean grows linearly as time progresses. In the following, we propose a novel compact representation of $\mathcal{C}$, which can be updated in constant time by avoiding the computation of the Riemannian mean.

### 4.2 Incremental Covariance Tensor Learning

From a generative perspective, $\mu_t$ and $C_t$ are generated from $F_t$ and the covariance tensor $\mathcal{C}$ is generated from the feature tensor $\mathcal{F}$. Therefore, the compact tensor representation can be obtained directly from $\mathcal{F}$. We get the compact representation by computing the column covariance of $F$:

$$\hat{C}_T=\frac{1}{NT-1}\sum_{t=1}^{T}\sum_{i=1}^{N}\left(f_{t,i}-\hat{\mu}_T\right)\left(f_{t,i}-\hat{\mu}_T\right)^T\ ,$$

where $\hat{\mu}_T$ is the column mean of $F$. Although (4.2) is arguably straightforward, it is computationally expensive and needs a large amount of memory to store all the previous observations. Here, we propose a novel formulation that could be computed efficiently with only $\mathcal{O}(d^2)$ arithmetic operations.

We treat (4.2) as a sample covariance estimation problem by considering each column $f_{t,i}$ of $F$ as a sample. As time progresses, the sample set $F$ grows and our aim is to incrementally update the sample covariance. In order to moderate the balance between old and new observations, each sample $f_{t,i}$ is associated with a weight, allowing newer samples to have a larger influence on the estimation of the current covariance tensor representation than the older ones. As a result, the covariance estimation problem can be reformulated as estimating the weighted sample covariance of $F$. Furthermore, under formulation (4.2), it is unnecessary to normalize the object appearance to the same size as [20]. In the following, we use $N_t$ to denote the size of the object at time $t$.

One of the critical issues for our formulation is the design of the sample weight. Four issues are considered to chose the sample weight: 1) the weight of each sample should vary over time $T$; 2) the samples from current time $T$ should have the higher weights than previous samples; 3) the weight should not affect the fast covariance computation using integral images; and 4) the weight should not affect the ability to incremental obtain the covariance tensor representation. Therefore, when the current time is $T$, the sample weight at time $t$ is set as $w^{T-t}$, where $w\in[0,1], t\in[1,T]$. With this weight setting, the samples at the same time share the same weight and the weighted sample covariance of $F$ can be incrementally updated.

To obtain an efficient algorithm to update the covariance tensor representation, we put forward the following definitions and theorem.

**Definition 1.** *Denote the weighted samples up to current time $T$ as*

$$\hat{F}_T=\{f_{t,i},w_{T,t,i}\}_{t=1,\ldots,T;i=1,\ldots,N_t},$$

*where $w_{T,t,i}$ is the weight of sample $f_{t,i}$. Let the number of samples in $\hat{F}_T$ be $\hat{N}_T$ and the sum of weights in $\hat{F}_T$ be $\hat{w}_T$, namely $\hat{N}_T=\sum_{t=1}^{T}N_t$ and $\hat{w}_T=\sum_{t=1}^{T}\sum_{i=1}^{N_t}w_{T,t,i}$.*

**Definition 2.** *Let $C_t$, $\mu_t$ be the weighted covariance and the weighted sample mean at time t, respectively. Denote the weighted covariance and the weighted sample mean of $\hat{F}_T$ as $\hat{C}_T$ and $\hat{\mu}_T$, respectively. The formulation of $\hat{C}_T$ and $\hat{\mu}_T$ are as follows:*

$$\hat{C}_T=\frac{1}{1-\bar{w}_T^2}\sum_{t=1}^{T}\sum_{i=1}^{N}\frac{w_{T,t,i}}{\hat{w}_T}\left(f_{t,i}-\hat{\mu}_T\right)\left(f_{t,i}-\hat{\mu}_T\right)^T, \tag{2}$$

*where*

$$\bar{w}_T^2=\sum_{t=1}^{T}\sum_{i=1}^{N_t}\left(\frac{w_{T,t,i}}{\hat{w}_T}\right)^2, \hat{\mu}_T=\frac{1}{\hat{w}_T}\sum_{t=1}^{T}\sum_{i=1}^{N_t}w_{T,t,i}f_{t,i}.$$

*Let weights of all samples at time t be equal, the formulation of $C_t$, $\mu_t$ are as follows:*

$$C_t=\frac{1}{N_t-1}\sum_{i=1}^{N_t}(f_{t,i}-\mu_t)(f_{t,i}-\mu_t)^T, \mu_t=\frac{1}{N_t}\sum_{i=1}^{N_t}f_{t,i}.$$

**Theorem 1.** *Given $C_T$, $\mu_T$, $\hat{C}_{T-1}$, $\hat{\mu}_{T-1}$, $\hat{w}_{T-1}, \bar{w}_{T-1}^2$, if $w_{T,t,i}=w^{T-t}, w\in[0,1]$, it can be shown that:*

$$\begin{aligned}\hat{C}_T=&\frac{1}{\hat{w}_T(1-\bar{w}_T^2)}\{w\hat{w}_{T-1}(1-\bar{w}_{T-1}^2)\hat{C}_{T-1}+(N_T-1)C_T\\&+\frac{w\hat{w}_{T-1}N_T}{\hat{w}_T}(\mu_T-\hat{\mu}_{T-1})(\mu_T-\hat{\mu}_{T-1})^T\}\end{aligned} \tag{3}$$

*where $\hat{w}_T=w\hat{w}_{T-1}+N_T$, $\hat{\mu}_T=\frac{w\hat{w}_{T-1}}{\hat{w}_T}\hat{\mu}_{T-1}+\frac{N_T}{\hat{w}_T}\mu_T$, $\bar{w}_T^2=\frac{(\hat{w}_{T-1}^2\bar{w}_{T-1}^2-N_{T-1})w^2+N_T}{\hat{w}_T^2}$. The initial conditions are $\hat{C}_1=C_1$, $\hat{\mu}_1=\mu_1$, $\hat{w}_1=N_1$, and $\bar{w}_1^2=1/N_1$.*

To make the proof of Theorem 1 concise, we give some lemmas first. The proof of all the lemmas appears in the Appendix.

**Lemma 1.** *If $w_{T,t,i}=w^{T-t}, w\in[0,1]$, we have $\hat{w}_T=w\hat{w}_{T-1}+N_T$, and $\bar{w}_T^2=\frac{(\hat{w}_{T-1}^2\bar{w}_{T-1}^2-N_{T-1})w^2+N_T}{\bar{w}_T^2}$.*

**Lemma 2.** *$\sum_{t=1}^{T}\sum_{t=1}^{N_t}w_{T,t,i}(f_{t,i}-\hat{\mu}_T)=0$ and $\sum_{t=1}^{T}\sum_{t=1}^{N_t}w_{T,t,i}(f_{t,i}-\hat{\mu}_T)^T=0$.*

**Lemma 3.** *If weights of all the samples at time $T$ are equal, then $\sum_{i=1}^{N_T}(f_{T,i}-\hat{\mu}_T)(f_{T,i}-\hat{\mu}_T)^T=(N_T-1)C_T+N_T(\mu_T-\hat{\mu}_T)(\mu_T-\hat{\mu}_T)^T$.*

**Lemma 4.** *If $w_{T,t,i}=w^{T-t}, w\in[0,1]$, we have $\hat{\mu}_T=\frac{w\hat{w}_{T-1}}{\hat{w}_T}\hat{\mu}_{T-1}+\frac{N_T}{\hat{w}_T}\mu_T$, $\hat{\mu}_{T-1}-\hat{\mu}_T=\frac{N_T}{\hat{w}_T}(\mu_T-\hat{\mu}_{T-1})$, and $\mu_T-\hat{\mu}_T=\frac{w\hat{w}_{T-1}}{\hat{w}_T}(\mu_T-\hat{\mu}_{T-1})$.*

**Lemma 5.** *If* $w_{T,t,i} = w^{T-t}, w \in [0,1]$*, we have* $\sum_{t=1}^{T-1}\sum_{i=1}^{N_t} w_{T,t,i}(f_{t,i} - \hat{\mu}_T)(f_{t,i} - \hat{\mu}_T)^T = w\hat{w}_{T-1}(1 - \bar{w}^2_{T-1})\hat{C}_{T-1} + w\hat{w}_{T-1}(\hat{\mu}_{T-1} - \hat{\mu}_T)(\hat{\mu}_{T-1} - \hat{\mu}_T)^T$.

***Proof of Theorem 1:***

By definition, $\hat{C}_T = \frac{1}{1-\bar{w}^2_T}\sum\limits_{t=1}^{T}\sum\limits_{i=1}^{N}\frac{w_{T,t,i}}{\hat{w}_T}(f_{t,i} - \hat{\mu}_T)(f_{t,i} - \hat{\mu}_T)^T$, thus we have

$$
\begin{aligned}
&\hat{w}_T(1 - \bar{w}^2_T)\hat{C}_T \\
&= \sum_{t=1}^{T}\sum_{i=1}^{N_t} w_{T,t,i}(f_{t,i} - \hat{\mu}_T)(f_{t,i} - \hat{\mu}_T)^T \\
&= \sum_{t=1}^{T-1}\sum_{i=1}^{N_t} w_{T,t,i}(f_{t,i} - \hat{\mu}_T)(f_{t,i} - \hat{\mu}_T)^T \\
&\quad + \sum_{i=1}^{N_t} w_{T,t,i}(f_{T,i} - \hat{\mu}_T)(f_{T,i} - \hat{\mu}_T)^T \quad (Lemmas\ 3\ and\ 5) \\
&= w\hat{w}_{T-1}(1 - \bar{w}^2_{T-1})\hat{C}_{T-1} \\
&\quad + w\hat{w}_{T-1}(\hat{\mu}_{T-1} - \hat{\mu}_T)(\hat{\mu}_{T-1} - \hat{\mu}_T)^T \quad (Lemma\ 4) \\
&\quad + (N_T - 1)C_T + N_T(\mu_T - \hat{\mu}_T)(\mu_T - \hat{\mu}_T)^T \\
&= w\hat{w}_{T-1}(1 - \bar{w}^2_{T-1})\hat{C}_{T-1} \\
&\quad + (N_T - 1)C_T + w\hat{w}_{T-1}(\frac{N_T}{\hat{w}_T})^2(\mu_T - \hat{\mu}_{T-1})(\mu_T - \hat{\mu}_{T-1})^T \\
&\quad + N_T(\frac{w\hat{w}_{T-1}}{\hat{w}_T})(\mu_T - \hat{\mu}_{T-1})(\mu_T - \hat{\mu}_{T-1})^T \quad (Lemma\ 1) \\
&= w\hat{w}_{T-1}(1 - \bar{w}^2_{T-1})\hat{C}_{T-1} + (N_T - 1)C_T \\
&\quad + \frac{w\hat{w}_{T-1}N_T}{\hat{w}_T}(\mu_T - \hat{\mu}_{T-1})(\mu_T - \hat{\mu}_{T-1})^T \ .
\end{aligned}
$$

□

If we treat all samples equally, i.e., set $w$ to 1, we can obtain the sample covariance of $F$ from (3):

$$
\begin{aligned}
\hat{C}_T =& \frac{1}{\hat{N}_T - 1}\{(\hat{N}_{T-1} - 1)\hat{C}_{T-1} + (N_T - 1)C_T \\
&+ \frac{N_T\hat{N}_{T-1}}{\hat{N}_T}(\mu_T - \hat{\mu}_{T-1})(\mu_T - \hat{\mu}_{T-1})^T\}
\end{aligned}
$$

When $w$ is set to 0, $\hat{C}_T$ is equal to $C_T$, which means only information at the current time is used to represent the covariance tensor.

Expanding $\hat{C}_{T-1}$ in Theorem 1 iteratively, we can reformulate $\hat{C}_T$ as follows:

$$
\hat{C}_T = \sum_{t=1}^{T} w_{t,C}C_t + \sum_{t=2}^{T} w_{t,\mu}(\mu_t - \hat{\mu}_{t-1})(\mu_t - \hat{\mu}_{t-1})^T .
$$

where $w_{t,C} = \frac{w^T(N_t-1)}{w^t\hat{w}_T(1-\bar{w}^2_T)}$, $w_{t,\mu} = \frac{w^T\hat{w}_{t-1}N_t}{w^{t-1}\hat{w}_t\hat{w}_T(1-\bar{w}^2_T)}$.

It is interesting to see that our formulation is a mixture model which is a weighted sum of all the covariance up to time $T$ with a regularization term, and the weight of each kernel covariance is adapted dynamically.

Consequently, the proposed incremental covariance tensor learning algorithm is shown in Algorithm 1.

**Algorithm 1** The incremental covariance tensor learning algorithm

```
Given $C_T, \mu_T, N_T, \hat{C}_{T-1}, \hat{\mu}_{T-1}, \hat{w}_{T-1}, N_{T-1}, \bar{w}^2_{T-1}$, as well as $w_{T,t,i} = w^{T-t}, w \in [0,1]$, compute $\hat{C}_T$:
Update the sum of sample weights up to time $T$: $\hat{w}_T = w\hat{w}_{T-1} + N_T$;
Update the squared sum of normalized sample weights up to time $T$: $\bar{w}^2_T = \left((\hat{w}^2_{T-1}\bar{w}^2_{T-1} - N_{T-1})w^2 + N_T\right)/\hat{w}^2_T$;
Update the weighted mean of all samples up to time $T$: $\hat{\mu}_T = \frac{w\hat{w}_{T-1}}{\hat{w}_T}\hat{\mu}_{T-1} + \frac{N_T}{\hat{w}_T}\mu_T$;
Update the weighted covariance $\hat{C}_T$ by Theorem 1.
The initial conditions are $\hat{C}_1 = C_1$, $\hat{\mu}_1 = \mu_1$, $\hat{w}_1 = N_1$, and $\bar{w}^2_1 = 1/N_1$.
```

## 5 Experiments

In our experiments, the target is initialized manually. The tracking parameters are tuned on one sequence and applied to all the other sequences. During the visual tracking, a 7-dimensional feature vector is extracted for each pixel:

$$
(x, y, R(x,y), G(x,y), B(x,y), I_x(x,y), I_y(x,y)) ,
$$

where $(x, y)$ is the pixel location, $R, G, B$ are the RGB color values and $I_x, I_y$ are the intensity derivatives. Consequently, the covariance descriptor of a color image region is a $7 \times 7$ symmetric matrix. The state in the particle filter refers to an object's 2D location and scale, namely $(x, y, s)$. The state dynamics $p(x_t|x_{t-1})$ is assumed to be a Gaussian distribution as $N(x_t; x_{t-1}, \Sigma)$, where $\Sigma$ is a diagonal covariance matrix whose diagonal elements are $(\sigma^2_x, \sigma^2_y, \sigma^2_s) = (5^2, 5^2, 0.02^2)$, respectively. The number of particles is set to 100 for our tracker and $w$ in (3) is set to 0.95. The observation model $p(y_t|x_t)$ is the crucial part for finding the ideal posterior distribution. It reflects the similarity between a candidate sample and the learned compact covariance tensor representation. The target appearance model is represented by $M$ modes $\{\hat{C}_{T,i}\}_{i=1}^M$. Each mode $C_i(x_t)$ of the candidate sample $x_t$ is compared with the corresponding model by (1). Thus $p(y_t|x_t)$ can be formulated as:

$$
p(y_t|x_t) \propto \exp\{-\lambda\Sigma_{i=1}^M \omega_i\rho^2[\hat{C}_{T,i}, C_i(x_t)]\} ,
$$

where $\omega_i$ is the weight for the $i$-th mode ($\omega_i = 1/M$ in our experiments). After the MAP estimation, we use the covariance matrices of image features associated with the estimated target state to update the compact covariance tensor model for each mode.

By our definition, each particle corresponds to an up-right rectangle. Therefore, it is possible to improve the computational complexity of covariance computation using the integral histogram techniques [32]. After constructing tensors of integral images for each feature dimension and multiplication of any two feature dimensions, the covariance matrix of any arbitrary rectangular region can be computed independent of the region size. In our case, 28 integral images are constructed for fast covariance computation. The approach was implemented using C++ and performed on a PC with a 1.6-GHz CPU.

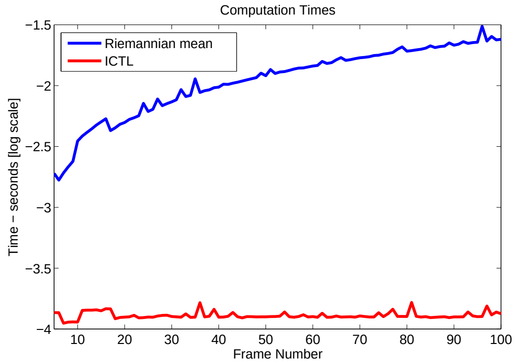


Fig. 3. Speed comparison for model update.

Without code optimization, our tracker can achieve around 20 *fps* for image sequences with resolution $320 \times 240$.

We compared the proposed ICTL tracker with nine state-of-the-art visual trackers, namely, generalized kernel-based Tracker (GKT) [35], multi-instance learning based tracker (MIL) [5], incremental PCA based tracker (IVT) [34], online boosting based tracker (OAB) [11], visual tracking decomposition tracker (VTD) [18], fragments based tracker (Frag) [1], color based particle filtering tracker (CPF) [31], covariance tracker (COV) [33] and Mean Shift tracker (MS) [8]. In our experiments using the public trackers we used the same parameters as the authors. Eleven sequences, most of them have been widely tested before, are used in the comparison experiments. The quantitative results are summarized in Table 1, 2, 3 and Fig. 10. Below is a more detailed discussion of the comparison tracking results.

### 5.1 Speed comparison for model update

From (3), it is clear that the update for $\hat{C}_T$ is independent of $T$ and needs only $\mathcal{O}(d^2)$ arithmetic operations, while the computational complexity of the Riemannian mean used in [33] is $\mathcal{O}(Td^3)$. In our experiment setting, when $T = 50$ and $d = 7$, the computational time for both algorithms are 0.1 ms and 10 ms respectively.

The computation times for model update are given in Fig. 3 in log-linear scale. The figure shows that the proposed ICTL has a constant time complexity and is significantly faster than the original covariance tracker.

### 5.2 Qualitative Evaluation

**Pedestrian tracking.** We first test our ICTL algorithm to track a pedestrian using the sequence, *crossing*, *couple*, *jogging*, *subway* and *woman*.

Fig. 4(a) shows the comparative results on *crossing*. Although the target has the similar color feature as the background, our tracker is able to track the target well, which can be attributed to the descriptive power of the covariance feature and the model update scheme. Notice that the non-convex target is localized within a rectangular window, and thus it inevitably contains some background pixels in its appearance representation. From #48, the target rectangular window contains some light pixels. The weighted incremental model update adapts the target model to the background changes. The results show that our algorithm faithfully models the appearance of an arbitrary object in the presence of noisy background pixels.

Fig. 4(b) shows the tracking results using sequence *couple*, captured from a hand-held camera. The couple represents a situation of group tracking where one or more objects move together in a sequence. Notice that there is a large scale variation in the target relative to the camera (#3, #139). Even with the significant camera motion and low frame rate, our ICTL algorithm is able to track the target better than other trackers (see Table. 1). Although our tracker loses the target in #91 due to the sudden fast camera motion, it re-detects the target in #116 and tracks the target to the end. Furthermore, the compact tensor representation is constructed from scratch and is updated to reflect the appearance variation of the target.

Fig. 4(c) shows the tracking results on the sequence *jogging*. Note that our ICTL method is able to track the target undergoing gradual scale changes (#22, #300). Further, our method is able to track the target with severe full occlusion (#68, #77), which lasts around 20 frames. Compared with the results of COV, our method is able to efficiently learn a compact representation while tracking the target without using Riemannian means. Moreover, our tracker is more stable when the target is under occlusion. The multi-mode representation and Bayesian formulation contribute to the successful performance.

Our algorithm is also able to track objects in cluttered environment, such as the sequence of a human walking in the subway, shown in Fig. 4(d). Despite many similar objects in the scenario, and indistinctive texture feature to background, our algorithm is able to track the human well.

Sequence *woman*, as shown in Fig. 4(e), contains a woman moving in different occlusion, scale, and lighting conditions. Once initialized in the first frame, our algorithm is able to track the target object as it experiences long-term partial occlusions (#68, #146, #324), large scale variation (#540), and sudden global lighting variation (#45, #46). Notice that some parts of the target are occluded, and thus it inevitably contains some background information in its appearance model. The multi-mode representation enables the tracker to work stably and estimate the target location correctly.

**Vehicle tracking.** Sequence *race*, as shown in Fig. 5(a), contains a car moving in different scale and pose, where the background has a similar color as the target. Once initialized in the first frame, our tracker is able to follow the target object as it experiences large scale changes (#4,#64,#254), and pose variations (#4, #185). Notice that the COV tracker cannot handle scale changes and is not stable during the tracking sequence.

Fig. 5(b) shows the tracking results on the sequence *car*. The target is undergoing long-term partial occlusions (#165, #170), which lasts around 40 frames, and large scale variation (#16, #197). In this sequence, GKT loses the target quickly and all the other trackers cannot estimate the scale as well as the ICTL method. When the car changes its pose (#252) together with scale variation, only our tracker can follow the target. The tracking success for partial occlusions and scale variation results from the part-based representation and the proposed model update approach.

Fig. 5(c) shows the tracking results on the sequence *turn-*

| | GKT [35] | MIL [5] | MS [8] | CPF [31] | COV [33] | IVT [34] | OAB [11] | VTD [18] | Frag [1] | **ICTL** |
|---|---|---|---|---|---|---|---|---|---|---|
| car | 12.7242 | 1.5211 | 3.6382 | 4.5717 | 3.3430 | 6.0876 | 3.2358 | 3.0916 | 2.6072 | **0.9118** |
| dog | 0.1933 | 0.1350 | 0.1757 | **0.0946** | 0.2124 | 0.7230 | 0.1843 | 0.1372 | 0.1434 | 0.1671 |
| face | 0.1490 | 0.2798 | 0.1877 | 0.2757 | 0.4031 | 0.1611 | 0.2026 | 0.1897 | **0.1005** | 0.1256 |
| race | 0.0768 | 0.0541 | 0.0784 | 0.0931 | 0.2537 | **0.0317** | 0.0523 | 0.0320 | 0.0533 | 0.0456 |
| turnpike | 0.0213 | 0.0210 | 0.0271 | 0.3961 | 0.2563 | 0.0080 | 0.0091 | **0.0051** | 0.0168 | 0.0127 |
| noise | 0.4706 | 0.0199 | 0.0539 | 0.3000 | 0.1406 | 0.0081 | 0.0065 | **0.0061** | 0.0258 | 0.0209 |
| crossing | 0.4564 | 0.0196 | 0.0351 | 0.2254 | 0.0883 | 0.1902 | **0.0110** | 0.0974 | 0.2359 | 0.0144 |
| couple | 1.3426 | 1.0522 | 3.4404 | 2.6670 | 0.4898 | 2.1280 | 3.0110 | 2.5734 | 1.0009 | **0.3433** |
| jogging | 0.3069 | 0.8211 | 0.7028 | 0.1885 | 0.0865 | 0.7808 | 0.0570 | 0.7916 | 0.6383 | **0.0364** |
| woman | 0.6305 | 0.6972 | 0.5714 | 0.2813 | 0.3178 | 0.5846 | 0.6700 | 0.7337 | 0.0664 | **0.0366** |
| subway | 3.0896 | 0.1061 | 3.0340 | 0.5036 | 0.2772 | 2.9237 | 3.2289 | 3.2080 | 0.1577 | **0.0880** |
| Ave. | 1.7692 | 0.4297 | 1.0859 | 0.8725 | 0.5335 | 1.2388 | 0.9699 | 0.9878 | 0.4588 | **0.1639** |

TABLE 1
The tracking error. The error is measured using the Euclidian distance of two center points, which has been normalized by the size of the target from the annotation.

| | GKT [35] | MIL [5] | MS [8] | CPF [31] | COV [33] | IVT [34] | OAB [11] | VTD [18] | Frag [1] | **ICTL** |
|---|---|---|---|---|---|---|---|---|---|---|
| car | 0.0176 | 0.3939 | 0.2924 | 0.2186 | 0.2791 | 0.4664 | 0.3978 | 0.4224 | 0.3902 | **0.5547** |
| dog | 0.2876 | 0.3423 | 0.3187 | 0.3753 | 0.2665 | 0.1865 | 0.2939 | **0.3962** | 0.3524 | 0.3087 |
| face | 0.7346 | 0.5792 | 0.6714 | 0.5800 | 0.5138 | 0.6901 | 0.6572 | 0.6301 | **0.7822** | 0.7118 |
| race | 0.4984 | 0.5236 | 0.4784 | 0.4275 | 0.3516 | 0.6430 | 0.5334 | **0.6906** | 0.5216 | 0.6372 |
| turnpike | 0.6568 | 0.6506 | 0.6344 | 0.1643 | 0.3582 | 0.7780 | 0.7628 | **0.8118** | 0.7129 | 0.7560 |
| noise | 0.2112 | 0.6580 | 0.4873 | 0.1969 | 0.5343 | **0.7985** | 0.7856 | 0.7828 | 0.6250 | 0.6567 |
| crossing | 0.0133 | 0.6078 | 0.4876 | 0.0836 | 0.3285 | 0.2696 | **0.6258** | 0.4076 | 0.2941 | 0.5947 |
| couple | 0.3422 | 0.4396 | 0.0517 | 0.0363 | 0.4337 | 0.2129 | 0.0675 | 0.0647 | 0.2317 | **0.5125** |
| jogging | 0.3449 | 0.1761 | 0.1449 | 0.4141 | 0.5369 | 0.1339 | 0.5333 | 0.1694 | 0.1643 | **0.6838** |
| woman | 0.0202 | 0.0767 | 0.0521 | 0.0856 | 0.0996 | 0.0641 | 0.0740 | 0.0661 | 0.5455 | **0.5938** |
| subway | 0.1146 | **0.5724** | 0.0540 | 0.1623 | 0.3644 | 0.0707 | 0.0808 | 0.0781 | 0.5404 | 0.5589 |
| Ave. | 0.2947 | 0.4564 | 0.3339 | 0.2495 | 0.3697 | 0.3921 | 0.4375 | 0.4109 | 0.4691 | **0.5972** |

TABLE 2
The tracking quality. The quality is measured using the area coverage between the tracking result and the annotation.

| | #frame | GKT [35] | MIL [5] | MS [8] | CPF [31] | COV [33] | IVT [34] | OAB [11] | VTD [18] | Frag [1] | **ICTL** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| car | 252 | 246 | 118 | 123 | 188 | 163 | 88 | 118 | 102 | 127 | **36** |
| dog | 127 | 89 | 71 | 77 | 69 | 91 | 95 | 96 | **56** | 73 | 80 |
| face | 890 | **0** | 201 | **0** | 140 | 214 | **0** | 10 | 2 | **0** | **0** |
| race | 320 | 33 | 22 | 52 | 46 | 132 | **0** | 22 | **0** | 27 | 43 |
| turnpike | 290 | **0** | **0** | **0** | 235 | 142 | **0** | **0** | **0** | 14 | **0** |
| noise | 290 | 190 | **0** | 48 | 202 | 60 | **0** | **0** | **0** | 14 | **0** |
| crossing | 120 | 118 | **0** | 7 | 114 | 63 | 71 | 2 | 44 | 68 | **0** |
| couple | 140 | 58 | 44 | 128 | 133 | 39 | 94 | 128 | 128 | 95 | **25** |
| jogging | 300 | 119 | 231 | 232 | 82 | 43 | 236 | 35 | 231 | 232 | **1** |
| woman | 542 | 531 | 478 | 510 | 483 | 503 | 474 | 474 | 490 | 84 | **48** |
| subway | 154 | 125 | **3** | 147 | 127 | 70 | 137 | 136 | 137 | 15 | 8 |
| Total | 3425 | 1509 | 1168 | 1324 | 1819 | 1520 | 1195 | 1021 | 1190 | 749 | **241** |

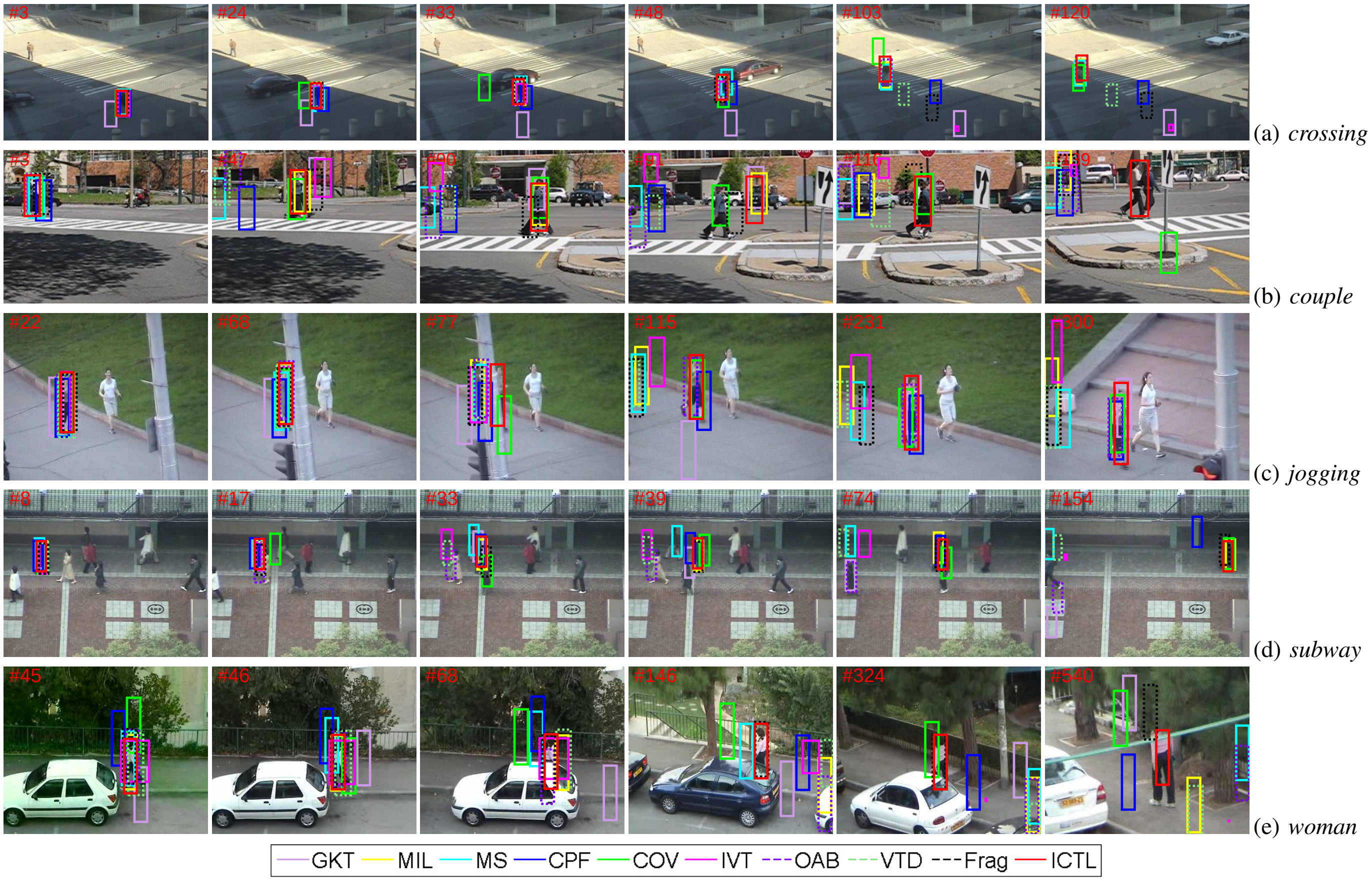
TABLE 3
Failed tracking statistics. The number for each sequence is calculated using a threshold ($1/3$ is used to generate this table) to filter the area coverage between the tracking result and the ground truth.

*pike*. The color based CPF tracker drifts off the target from #60 and then quickly loses the target. Similarly, the COV tracker also loses the target and is attracted to the nearby car with similar color.

**Noise.** To test the robustness to noise, Gaussian noise was added to sequence *turnpike* and the generated sequence is named *noise*. The comparative results are shown in Fig. 5(d). Compared with Fig. 5(c), we can see that the performance of GKT is decreased dramatically. The poor performance of the GKT is because its appearance model is not robust to the noise. Note that the covariance descriptor is robust to the Gaussian noise and the performance of our tracker is almost the same as noise-free sequence.

**Long term sequence tracking.** Long term sequence tracking has recently drawn many researchers' attention [15] due to its challenges and practical applications. We test the proposed method on one long sequence, *doll* [24], which is taken by a hand held camcorder and lasts 3871 frames. Some samples of the tracking results are shown in Fig. 6. It shows the tracking capability of our method under scale, pose changes and occlusions.

**More other cases.** Fig. 7(a) and Fig. 7(b) show more tracking results on the sequence *face* and *dog*, respectively. In *face*, the target is frequently undergoing long-term partial occlusion. Our tracker again outperforms all the other trackers. The successful performance can be attributed to the adopted



(a) *crossing*

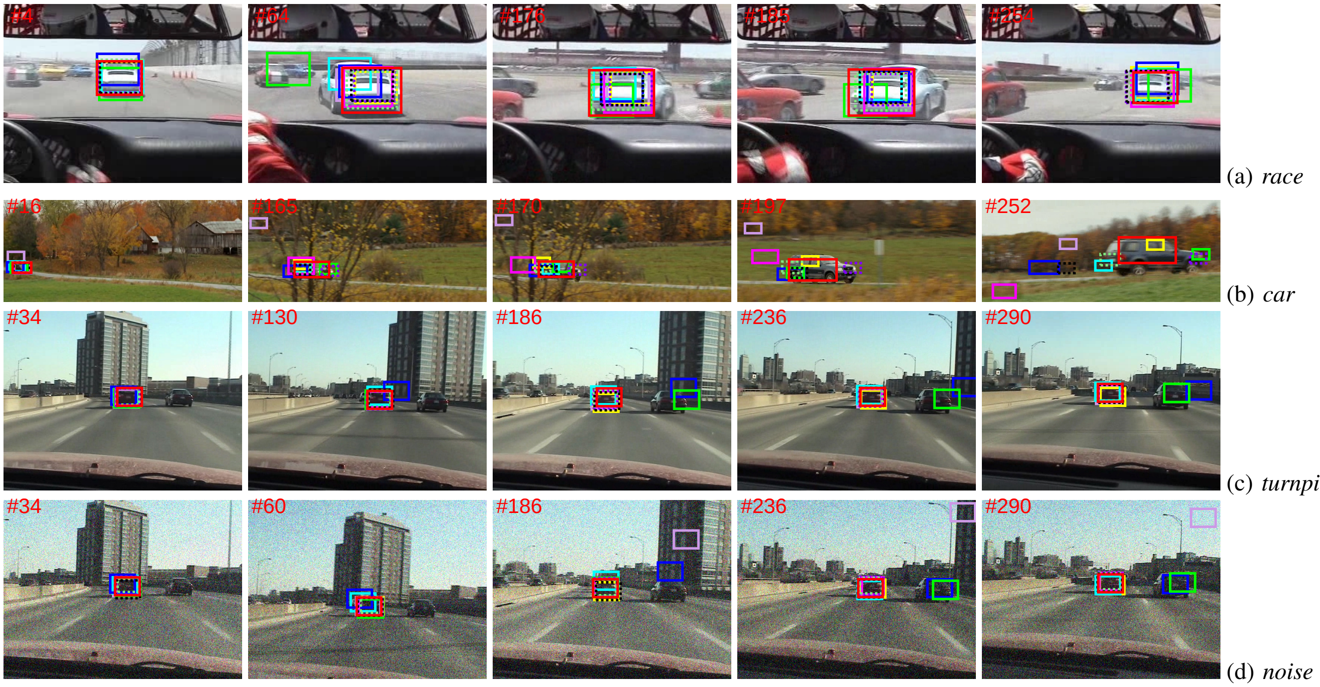
(b) *couple*

(c) *jogging*

(d) *subway*

(e) *woman*

Fig. 4. Pedestrian tracking results of different algorithms.



(a) *race*

(b) *car*

(c) *turnpike*

(d) *noise*

Fig. 5. Vehicle tracking results. Legend is the same as in Fig.4.

part-based representation. COV performs poorly on this sequence. In sequence *dog*, the dog is running and undergoing large pose variation. Although our tracker cannot estimate the accurate scale of the target due to the severe pose change, our ICTL tracker follows the dog throughout the sequence.

### 5.3 Qualitative analysis of ICTL

We use the sequence *crossing* to test the effectiveness of the proposed ICTL. Three trackers are exploited for the qualitative analysis: Tracker-A uses the proposed ICTL approach with default parameter setting; Tracker-B uses the sample covari-

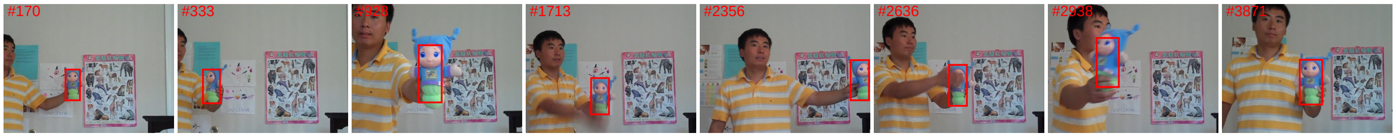
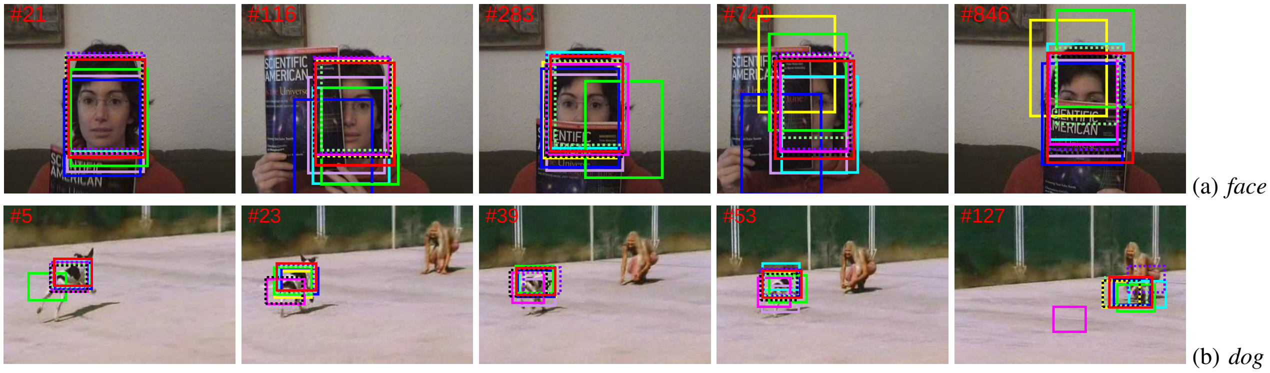


Fig. 6. Tracking results on a long sequence, *doll*. There are pose, scale changes and occlusions in the sequence.



(a) *face*

(b) *dog*

Fig. 7. Tracking results of different algorithms on *face* and *dog*. Legend is the same as in Fig.4.

ance for model update, namely, the parameter $w$ in Eq.3 is set to 1; and Tracker-C is a tracker without model update. To test if adding more features could improve the tracking performance, we construct Tracker-D by adding Tracker-C with two additional features: two directional second-order intensity derivatives, and the size of covariance descriptor for Tracker-D is $9 \times 9$. The results are illustrated in Fig. 8. As can be seen in the figure, when the target window includes more background clutter (white pixels), Tracker-C drifts and loses the target after #77. Tracker-B drifts from #76 and loses the target in #79. Even with more visual features, Tracker-D could not track the target robustly. While our proposed Tracker-A is able to track the target throughout the sequence. The success of the ICTL performance can be attributed to the weighting scheme adopted in the proposed ICTL.

To further realize the tracking performance with respect to the weight selection, we carried out different trackers with different weights on sequence *crossing*, where the weight's range is from 0 to 1 with space 0.05. This is illustrated in Fig.9. We can see that an improper weight may degenerate the performance. Weights in the range $[0.8, 0.95]$ may be a good choice for the tracker.

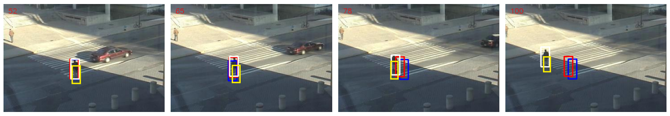


Fig. 8. The effectiveness test of ICTL using three modification of ICTL: Tracker-A (white), Tracker-B (red), Tracker-C (blue) and Tracker-D (yellow).

### 5.4 Quantitative Evaluation

To quantitatively evaluate all the trackers, we manually labeled the bounding box of the target in each frame. In Table 1 we give the average tracking errors of each approach in all sequences. From the statistical results, we can see that although many of the state-of-the-art tracking approaches have difficulty tracking the targets throughout the sequence, our proposed tracker can track the target robustly.

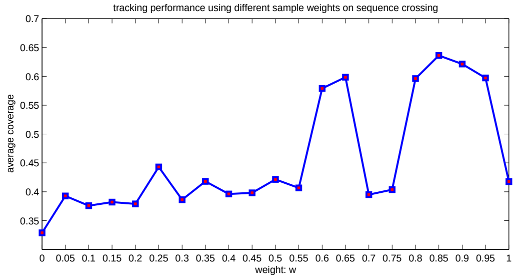


Fig. 9. Tracking performance w.r.t the weight selection.

To measure the tracking quality of each approach, we use the area coverage between the tracking result and the annotation as the criterion. The range of this measure is [0,1]. The average quality is shown in Table 2. If we treat the coverage lower than $1/3$ as poor tracking result, we can get the poor tracking statistics table as shown in Table 3. We can see that all the approaches cannot perform well on *dog* sequence due to the target is undergoing large deformation together with scale change. *car* and *race* are also challenging sequences due to the large scale variation. Especially on *jogging* and *woman*, our tracker perform much better than other trackers.

Fig. 10 illustrates the tracking error plot for each algorithm on each testing sequence. Each subfigure corresponds to one testing sequence, and in each subfigure, different colored lines represent different trackers. Our proposed tracker performs excellently in comparison with other state-of-the-art trackers.

The reason that our ICTL tracker performs well is three-folded: 1) multiple covariance feature matrices are used to characterize the object appearance; 2) the particle filter is adopted for posterior distribution propagation over time; and 3) the ICTL learns the compact covariance model to handle appearance variation.

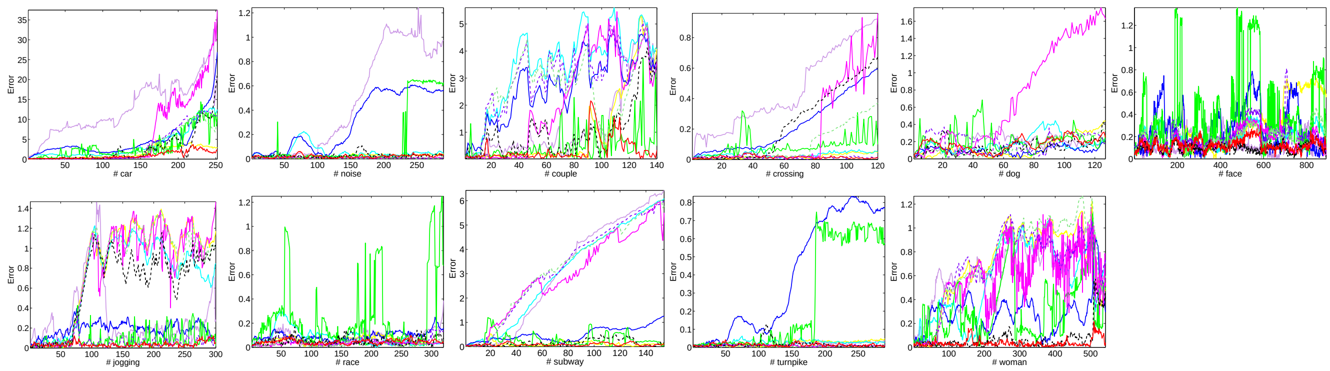


Fig. 10. The tracking error plot. Legend is the same as in Fig.4.

### 5.5 Discussion

Our proposed tracker is based on the multi-mode representation, covariance descriptor, incremental appearance learning and particle filter. The robustness of the tracking performances are joint result of these components. In particular, multi-mode representation addresses partial occlusion and scale estimation; covariance matrix brings rich information for target representation; and particle filter is more powerful than searching based approach. That said, there are challenging cases our tracker meet problems, such as when dealing with severe motion blurs, large and fast scale change, abrupt motion or moving out of the frame, etc. These challenges are likely to happen especially in long sequences. Fig. 11 shows some failure or inaccurate tracking results of the proposed tracker.

Some of the compared trackers are without a model update procedure, such as the CPF tracker. As a result, they cannot handle the appearance variations of the target. Their tracking performance could be improved by adopting some advanced model update scheme, such as the approach adopted in [26] for the CPF tracker. This may also give a good motivation of choosing covariance descriptor. We would investigate this in our future work.

To fairly compare different trackers is not an easy work. Different evaluation criterion may generate different performance. For example, on the sequence *subway* center error criterion does not consistent with area coverage criterion. Center error criterion is widely used in visual tracking domain while area coverage criterion is commonly used in object detection area. From the performance generated by area coverage we can get much more information than center error, e.g. the quality of tracking. Therefore, we think the area coverage is a better criterion for tracking performance measurement.

## 6 Conclusion

In this paper, we presented a real-time probabilistic visual tracking approach with incremental covariance model updating. In the proposed method, the covariance matrix of image features represents the object appearance. Further, an incremental covariance tensor learning (ICTL) algorithm adapts and reflects the appearance changes of an object due to intrinsic and extrinsic variations. Moreover, our probabilistic ICTL method uses a particle filter for motion parameter estimation, the covariance region descriptor for object appearance, and with the use of integral images achieves real-time performance. Use of a part-based representation of the object model in addition to the ICTL and Bayesian PF updates also affords tracking through scale, pose, and illumination changes. Compared with many state-of-the-art trackers, the proposed algorithm is faster and more robust to occlusions and object pose variations. Experimental results demonstrate that the proposed method is promising for robust real-time tracking for many security, surveillance, and monitoring applications.

The proposed probabilistic tracker is more suitable for multi-target tracking. Due to the integral images used for fast calculations of covariance matrix, when tracking multi-objects, the computational cost grows less than the linear of the tracked target number. When covariance-based object detector [40] is used to initialize the targets, the computational cost would lower than the independent detector and tracker. This is because the detector shares the same base features (integral images) with the tracker. Furthermore, the boosted particle filter [27] can be used to improve the multi-object tracking performance.

## Appendix A
## Proof of all lemmas

***Proof of Lemma 1:***

$$\hat{w}_T = \sum_{t=1}^{T}\sum_{i=1}^{N_t} w_i^t = \sum_{t=1}^{T} N_t w^{T-t} = \sum_{t=1}^{T-1} N_t w^{T-t} + N_T.$$

Since $\hat{w}_{T-1} = \sum_{t=1}^{T-1} N_t w_{T-1-t}$, we have $\hat{w}_T = w\hat{w}_{T-1} + N_T$. Thus

$$\begin{aligned}\bar{w}_T^2 &= \sum_{t=1}^{T}\sum_{i=1}^{N_t} \frac{w_{T,t,i}}{\hat{w}_T}^2 = \frac{1}{\hat{w}_T^2}\sum_{t=1}^{T}\sum_{i=1}^{N_t} w^{2(T-t)} \\ &= \frac{1}{\hat{w}_T^2}\sum_{t=1}^{T} N_t w^{2(T-t)} = \frac{1}{\hat{w}_T^2}(\sum_{t=1}^{T-1} N_t w^{2(T-t)} + N_T) \ .\end{aligned}$$

Since $\bar{w}_{T-1}^2 = \frac{1}{\hat{w}_{T-1}^2}(\sum_{t=1}^{T-1} N_t w^{2(T-1-t)} + N_{T-1})$ , and $\sum_{t=1}^{T-1} N_t w^{2(T-t)} = (\hat{w}_{T-1}^2\bar{w}_{T-1}^2 - N_{T-1})w^2$ , we have

$$\bar{w}_T^2 = \frac{(\hat{w}_{T-1}^2\bar{w}_{T-1}^2 - N_{T-1})w^2 + N_T}{(w\hat{w}_{T-1} + N_T)^2}.$$

□

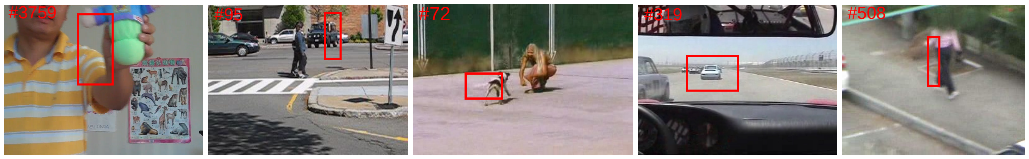


Fig. 11. Some failed or inaccurate tracking results by the proposed tracker.

*Proof of Lemma 2:*

$$\sum_{t=1}^{T}\sum_{i=1}^{N_t} w_{T,t,i}(f_{t,i}-\hat{\mu}_T) = \sum_{t=1}^{T}\sum_{i=1}^{N_t} w_{T,t,i}f_{t,i} - \sum_{t=1}^{T}\sum_{i=1}^{N_t} w_{T,t,i}\hat{\mu}_T.$$

By definition we have $\hat{w}_T = \sum_{t=1}^{T}\sum_{i=1}^{N_t} w_{T,t,i}$ and $\hat{\mu}_T = \frac{1}{\hat{w}_T}\sum_{t=1}^{T}\sum_{i=1}^{N_t} w_{T,t,i}f_{t,i}$, therefore

$$\begin{aligned}
&\sum_{t=1}^{T}\sum_{i=1}^{N_t} w_{T,t,i}f_{t,i} - \sum_{t=1}^{T}\sum_{i=1}^{N_t} w_{T,t,i}\hat{\mu}_T \\
&= \hat{w}_T\hat{\mu}_T - \hat{\mu}_T\sum_{t=1}^{T}\sum_{i=1}^{N_t} w_{T,t,i} = \hat{w}_T\hat{\mu}_T - \hat{\mu}_T\hat{w}_T = 0,
\end{aligned}$$

and

$$\sum_{t=1}^{T}\sum_{i=1}^{N_t} w_{T,t,i}(f_{t,i}-\hat{\mu}_T)^T = \left\{\sum_{t=1}^{T}\sum_{i=1}^{N_t} w_{T,t,i}(f_{t,i}-\hat{\mu}_T)\right\}^T = 0.$$

□

*Proof of Lemma 3:*

$$\begin{aligned}
&\sum_{i=1}^{N_T}\left(f_{T,i}-\hat{\mu}_T\right)\left(f_{T,i}-\hat{\mu}_T\right)^T \\
&= \sum_{i=1}^{N_T}\left(f_{T,i}-\mu_T+\mu_T-\hat{\mu}_T\right)\left(f_{T,i}-\mu_T+\mu_T-\hat{\mu}_T\right)^T \\
&= \sum_{i=1}^{N_T}\left(f_{T,i}-\mu_T\right)\left(f_{T,i}-\mu_T\right)^T + \left(\sum_{i=1}^{N_T}\left(f_{T,i}-\mu_T\right)\right)\left(\mu_T-\hat{\mu}_T\right)^T \\
&\quad + \left(\mu_T-\hat{\mu}_T\right)\sum_{i=1}^{N_T}\left(f_{T,i}-\mu_T\right)^T + N_T\left(\mu_T-\hat{\mu}_T\right)\left(\mu_T-\hat{\mu}_T\right)^T \\
&= \left(N_T-1\right)C_T + N_T\left(\mu_T-\hat{\mu}_T\right)\left(\mu_T-\hat{\mu}_T\right)^T
\end{aligned}$$

Since

$$\begin{aligned}
\left(N_t-1\right)C_t &= \sum_{i=1}^{N_t}\left(f_{t,i}-\mu_t\right)\left(f_{t,i}-\mu_t\right)^T, \\
\sum_{i=1}^{N_T}\left(f_{T,i}-\mu_T\right) &= \sum_{i=1}^{N_T} f_{T,i} - N_T\mu_T = 0,
\end{aligned}$$

we have

$$\begin{aligned}
&\sum_{i=1}^{N_T}\left(f_{T,i}-\hat{\mu}_T\right)\left(f_{T,i}-\hat{\mu}_T\right)^T \\
&= \left(N_T-1\right)C_T + N_T\left(\mu_T-\hat{\mu}_T\right)\left(\mu_T-\hat{\mu}_T\right)^T
\end{aligned}$$

□

*Proof of Lemma 4:*

$$\begin{aligned}
\hat{\mu}_T &= \frac{1}{\hat{w}_T}\sum_{t=1}^{T}\sum_{i=1}^{N_t} w_{T,t,i}f_{t,i} = \frac{1}{\hat{w}_T}\sum_{t=1}^{T} w^{T-t}\sum_{i=1}^{N_t} f_{t,i} \\
&= \frac{1}{\hat{w}_T}\sum_{t=1}^{T-1} w w^{T-1-t}\sum_{i=1}^{N_t} f_{t,i} + \frac{1}{\hat{w}_T}\sum_{i=1}^{N_T} f_{t,i} \\
&= \frac{w\hat{w}_{T-1}}{\hat{w}_T}\hat{\mu}_{T-1} + \frac{N_T}{\hat{w}_T}\mu_T
\end{aligned}$$

Since by definition $N_t\mu_t = \sum\limits_{i=1}^{N_t} f_{t,i}$ and $\hat{\mu}_T = \frac{1}{\hat{w}_T}\sum\limits_{t=1}^{T}\sum\limits_{i=1}^{N_t} w_{T,t,i}f_{t,i} = \frac{1}{\hat{w}_T}\sum\limits_{t=1}^{T} w^{T-t}\sum\limits_{i=1}^{N_t} f_{t,i}$, we have

$$\begin{aligned}
\hat{\mu}_T &= \frac{w\hat{w}_{T-1}}{\hat{w}_T}\hat{\mu}_{T-1} + \frac{N_T}{\hat{w}_T}\mu_T \\
\hat{\mu}_{T-1} - \hat{\mu}_T &= \hat{\mu}_{T-1} - \left(\frac{w\hat{w}_{T-1}}{\hat{w}_T}\hat{\mu}_{T-1} + \frac{N_T}{\hat{w}_T}\mu_T\right) \\
&= \frac{N_T}{\hat{w}_T}\left(\mu_T - \hat{\mu}_{T-1}\right) \\
\mu_T - \hat{\mu}_T &= \mu_T - \left(\frac{w\hat{w}_{T-1}}{\hat{w}_T}\hat{\mu}_{T-1} + \frac{N_T}{\hat{w}_T}\mu_T\right) \\
&= \frac{w\hat{w}_{T-1}}{\hat{w}_T}\left(\mu_T - \hat{\mu}_{T-1}\right)
\end{aligned}$$

□

*Proof of Lemma 5:*

$$\begin{aligned}
&\sum_{t=1}^{T-1}\sum_{i=1}^{N_t} w_{T,t,i}\left(f_{t,i}-\hat{\mu}_T\right)\left(f_{t,i}-\hat{\mu}_T\right)^T \\
&= \sum_{t=1}^{T-1}\sum_{i=1}^{N_t} w_{T,t,i}\left(f_{t,i}-\hat{\mu}_{T-1}+\hat{\mu}_{T-1}-\hat{\mu}_T\right) \\
&\qquad \cdot\left(f_{t,i}-\hat{\mu}_{T-1}+\hat{\mu}_{T-1}-\hat{\mu}_T\right)^T \\
&= \sum_{t=1}^{T-1}\sum_{i=1}^{N_t} w^{T-t}\left(f_{t,i}-\hat{\mu}_{T-1}\right)\left(f_{t,i}-\hat{\mu}_{T-1}\right)^T \\
&+ \left\{\sum_{t=1}^{T-1}\sum_{i=1}^{N_t} w_{T,t,i}\left(f_{t,i}-\hat{\mu}_{T-1}\right)\right\}\left(\hat{\mu}_{T-1}-\hat{\mu}_T\right)^T \\
&+ \left(\hat{\mu}_{T-1}-\hat{\mu}_T\right)\sum_{t=1}^{T-1}\sum_{i=1}^{N_t} w_{T,t,i}(f_{t,i}-\hat{\mu}_{T-1})^T \\
&+ \left(\hat{\mu}_{T-1}-\hat{\mu}_T\right)\left(\hat{\mu}_{T-1}-\hat{\mu}_T\right)^T\sum_{t=1}^{T-1}\sum_{i=1}^{N_t} w^{T-t}
\end{aligned}$$

By definition $\hat{w}_{T-1}\left(1-\bar{w}_{T-1}^2\right)\hat{C}_{T-1} =$

$\sum\limits_{t=1}^{T-1}\sum\limits_{i=1}^{N_t} w^{T-1-t}\left(f_{t,i}-\hat{\mu}_{T-1}\right)\left(f_{t,i}-\hat{\mu}_{T-1}\right)^T$, we have

$$\begin{aligned}&\sum_{t=1}^{T-1}\sum_{i=1}^{N_t} w^{T-t}\left(f_{t,i}-\hat{\mu}_{T-1}\right)\left(f_{t,i}-\hat{\mu}_{T-1}\right)^T\\&= w\hat{w}_{T-1}\left(1-\bar{w}_{T-1}^2\right)\hat{C}_{T-1}.\end{aligned}$$

By using lemma 2, we have

$$\left(\hat{\mu}_{T-1}-\hat{\mu}_T\right)^T\sum_{t=1}^{T-1}\sum_{i=1}^{N_t} w_{T,t,i}\left(f_{t,i}-\hat{\mu}_{T-1}\right)=0,$$

$$\left(\hat{\mu}_{T-1}-\hat{\mu}_T\right)\sum_{t=1}^{T-1}\sum_{i=1}^{N_t} w_{T,t,i}(f_{t,i}-\hat{\mu}_{T-1})^T=0.$$

By definition $\hat{w}_{T-1} = \sum\limits_{t=1}^{T-1} N_t w^{T-1-t}$, we have $\sum\limits_{t=1}^{T-1}\sum\limits_{i=1}^{N_t} w^{T-t} = \sum\limits_{t=1}^{T-1} N_t w^{T-t} = w\hat{w}_{T-1}$.

Therefore we have,

$$\begin{aligned}&\sum_{t=1}^{T-1}\sum_{i=1}^{N_t} w_{T,t,i}\left(f_{t,i}-\hat{\mu}_T\right)\left(f_{t,i}-\hat{\mu}_T\right)^T\\&= w\hat{w}_{T-1}\left\{\left(1-\bar{w}_{T-1}^2\right)\hat{C}_{T-1}+\left(\hat{\mu}_{T-1}-\hat{\mu}_T\right)\left(\hat{\mu}_{T-1}-\hat{\mu}_T\right)^T\right\}\end{aligned}$$

□

## References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[2] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM J. on Matrix Analysis and Applications*, 29(1), 2008.

[3] S. Avidan. Support vector tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(8):1064–1072, 2004.

[4] S. Avidan. Ensemble tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(2):261–271, 2008.

[5] B. Babenko, M. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[6] S. Birchfield and S. Rangarajan. Spatiograms versus histograms for region-based tracking. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[7] R. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(10):1631–1643, 2005.

[8] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(5):564–577, 2003.

[9] W. Förstner and B. Moonen. A metric for covariance matrices. Technical report, Stuttgart University, 1999.

[10] K. Guo, P. Ishwar, and J. Konrad. Action change detection in video by covariance matching of silhouette tunnels. In *IEEE Int'l Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2010.

[11] G. Helmut, G. Michael, and B. Horst. Real-Time Tracking via On-line Boosting In *BMVC*, 2006.

[12] X. Hong, H. Chang, S. Shan, X. Chen, and W. Gao. Sigma set: A small second order statistical region descriptor. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[13] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *Int'l J. of Computer Vision*, 29(1):5–28, 1998.

[14] S. Julier and J. Uhlmann. Using covariance intersection for SLAM. *Robotics and Autonomous Systems*, 55(7):3–20, 2007.

[15] Z. Kalal, J. Matas, and K. Mikolajczyk. P-N Learning: Bootstrapping Binary Classifiers by Structural Constraints. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[16] M. Kalandros and L. Pao. Covariance control for multisensor systems. *IEEE Trans. on Aerospace & Electronic Systems*, 38(4):1138-1157, 2002.

[17] S. Kwak, W. Nam, B. Han, and J. Han. Learning Occlusion with Likelihoods for Visual Tracking. In *IEEE Int'l Conf. on Computer Vision (ICCV)*, 2011.

[18] J. Kwon and K. M. Lee. Visual Tracking Decomposition. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[19] A. Levy and M. Lindenbaum. Sequential karhunen-loeve basis extraction and its application to images. *IEEE Trans. on Image Processing*, 9(8):1371–1374, 2000.

[20] X. Li, W. Hu, Z. Zhang, X. Zhang, M. Zhu, and J. Cheng. Visual tracking via incremental Log-Euclidean Riemannian subspace learning. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[21] H. Li, C. Shen, and Q. Shi. Real-time visual tracking with compressed sensing. In *IEEE Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[22] B. Liu, j. Huang, C. Kulikowski, L. Yang. Robust Tracking Using Local Sparse Appearance Model and K-Selection. In *IEEE Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[23] X. Mei, and H. Ling. Robust Visual Tracking using L1 Minimization. In *IEEE Int'l Conf. on Computer Vision (ICCV)*, 2009.

[24] X. Mei and H. Ling. Robust visual tracking and vehicle classification via sparse representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(11):2259–2272, 2011.

[25] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai. Minimum Error Bounded Efficient L1 Tracker with Occlusion Detection. In *IEEE Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[26] K. Nummiaro, E. Koller-Meier, and L. Van Gool. Object Tracking with an Adaptive Color-Based Particle Filter. *Symp. for Pattern Recognition of the DAGM*,2002.

[27] K. Okuma, A. Taleghani, N. Freitas, J. Little, and D. Lowe. A boosted particle filter: Multitarget detection and tracking. In *European Conf. on Computer Vision (ECCV)*, 2004.

[28] S. Paisitkriangkrai, C. Shen, and J. Zhang. Fast pedestrian detection using a cascade of boosted covariance features. *IEEE Trans. on Circuits and Systems for Video Technology*, 18(8):1140–1151, 2008.

[29] Y. Pang, Y. Yuan, and X. Li. Gabor-based region covariance matrices for face recognition. *IEEE Trans. on Circuits and Systems for Video Technology*, 18(7):989–993, 2008.

[30] X. Pennec, P. Fillard, and N. Ayache. A Riemannian framework for tensor computing. *Int'l J. of Computer Vision*, 66(1):41–66, 2006.

[31] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *European Conf. on Computer Vision (ECCV)*, 2002.

[32] F. Porikli. Integral histogram: A fast way to extract histograms in Cartesian spaces. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[33] F. Porikli, O. Tuzel, and P. Meer. Covariance tracking using model update based on Lie Algebra. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[34] D. Ross, J. Lim, R. Lin, and M. Yang. Incremental learning for robust visual tracking. *Int'l J. of Computer Vision*, 77(1):125–141, 2008.

[35] C. Shen, J. Kim, and H. Wang. Generalized kernel-based visual tracking. *IEEE Trans. on Circuits and Systems for Video Technology*, 20(1):119–130, 2010.

[36] R. Sivalingam, D. Boley, V. Morellas, and N. Papanikolopoulos. Tensor sparse coding for region covariances. In *European Conf. on Computer Vision (ECCV)*, 2010.

[37] D. Skočaj and A. Leonardis. Weighted and robust incremental method for subspace learning. In *IEEE Int'l Conf. on Computer Vision (ICCV)*, 2003.

[38] D. Tosato, M. Farenzena, M. Spera, V. Murino, and M. Cristani. Multi-class classification on Riemannian manifolds for video surveillance. In *European Conf. on Computer Vision (ECCV)*, 2010.

[39] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In *European Conf. on Computer Vision (ECCV)*, 2006.

[40] O. Tuzel, F. Porikli, and P. Meer. Human detection via classification on Riemannian manifolds. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[41] M. Wei, G. Chen, E. Blasch, H. Chen, and, J. B. Cruz. Game theoretic multiple mobile sensor management under adversarial environments. In *Int'l Conf. on Information Fusion*, 2008.

[42] Y. Wu, E. Blasch, G. Chen, L. Bai, and H. Ling Multiple Source Data Fusion via Sparse Representation for Robust Visual Tracking. In *Int'l Conf. on Information Fusion (FUSION)*, 2011.

[43] Y. Wu, J. Cheng, J. Wang, and H. Lu. Real-time visual tracking via incremental covariance tensor learning. In *IEEE Int'l Conf. on Computer Vision (ICCV)*, 2009.

[44] Y. Wu, H. Ling, J. Yu, F. Li, X. Mei, and E. Cheng. Blurred Target Tracking by Blur-driven Tracker. In *IEEE Int'l Conf. on Computer Vision (ICCV)*, 2011.

[45] Y. Wu, J. Wang, and H. Lu. Robust Bayesian Tracking on Riemannian Manifolds Via Fragments-based Representation. In *IEEE Int'l Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2009.

[46] Y. Wu, B. Wu, J. Liu, and H. Lu. Probabilistic Tracking on Riemannian Manifolds. In *IEEE Int'l Conf. on Pattern Recognition (ICPR)*, 2008.

[47] Q. Yu, T. Dinh, and G. Medioni. Online tracking and reacquisition using co-trained generative and discriminative trackers. In *European Conf. on Computer Vision (ECCV)*, 2008.

[48] S. Zhou, R. Chellappa, and B. Moghaddam. Visual tracking and recognition using appearance-adaptive models in particle filters. *IEEE Trans. on Image Processing*, 13(11):1491–1506, 2004.