Real-time Visual Tracking via Incremental Covariance Tensor Learning

Yi Wu[1,2], Jian Cheng[2], Jinqiao Wang[2], Hanqing Lu[2]

[1]College of Computer and Software, Nanjing University of Information Science & Technology, Nanjing 210044, China
E-mail: ywu.china@gmail.com

[2]Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
E-mail: {jcheng, jqwang, luhq}@nlpr.ia.ac.cn

Abstract

Visual tracking is a challenging problem, as an object may change its appearance due to pose variations, illumination changes, and occlusions. Many algorithms have been proposed to update the target model using the large volume of available information during tracking, but at the cost of high computational complexity. To address this problem, we present a tracking approach that incrementally learns a low-dimensional covariance tensor representation, efficiently adapting online to appearance changes for each mode of the target with only $O(1)$ computational complexity. Moreover, a weighting scheme is adopted to ensure less modeling power is expended fitting older observations. Both of these features contribute measurably to improving overall tracking performance. Tracking is then led by the Bayesian inference framework in which a particle filter is used to propagate sample distributions over time. With the help of integral images, our tracker achieves real-time performance. Extensive experiments demonstrate the effectiveness of the proposed tracking algorithm for the targets undergoing appearance variations.

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. Pose variation and shape deformation can be viewed as the intrinsic appearance variations, whereas the extrinsic variations include illumination changes, camera viewpoint, and occlusions. Consequently, it is imperative for a robust tracking algorithm to model such appearance variations.

In recent years, much work has been done in visual tracking to model the appearance variations of a target. Hager and Belhumeur [1] extended the gradient-based optical flow method using parametric models to handle the appearance variations caused by illumination changes. Black *et al.* [2] encoded the appearance changes into a mixture model to estimate image motion. In [3] a more elaborate mixture model with an online EM algorithm is proposed to explicitly model appearance changes during tracking. Yu *et al.* [4] proposed a spatial-appearance model which captures non-rigid appearance variations and recovers all motion parameters efficiently. In [5] a generalized geometric transform is used to handle the deformation, articulation, and occlusion of appearance. Zhou *et al.* [6] embedded appearance adaptive models into a particle filter to achieve a robust visual tracking.

Generative models, which are used to learn the appearance of an object, have been exploited to handle the variability of a target. The object model is often updated online to adapt to appearance changes. Black *et al.* [7] presented a subspace learning based tracking algorithm with view-based appearance models. In [8], Ross *et al.* proposed a generalized visual tracking framework based on the incremental image-as-vector subspace learning methods with a sample mean update.

Supervised discriminative methods for classification have also been exploited to handle appearance changes, where a classifier is trained and updated online to distinguish the object from the background. SVT [9] integrates an offline trained SVM classifier into an optic-flow-based tracker. In [10], the most discriminative RGB color combination is learned online to build a confidence map in each frame. In [11], an ensemble of online learned weak classifiers is used to label a pixel as belonging to either the object or the background. To accommodate object appearance changes, the ensemble is updated at every frame by using new weak classifiers to replace part of old ones that do not perform well. To encode the object appearance variations, Yu *et al.* [12] proposed to use co-training to combine generative and discriminative models to learn an appearance model on-the-fly.

For visual 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 [13]. Further, Ross *et al.* [8] provided an analysis of its effect on the resulting eigenbasis. Skocaj and Leonardis [14] presented an incremental method, which sequentially updates the principal subspace considering weighted influence of individual images as well as individual pixels

2009 IEEE 12th International Conference on Computer Vision (ICCV)
978-1-4244-4419-9/09/$25.00 ©2009 IEEE

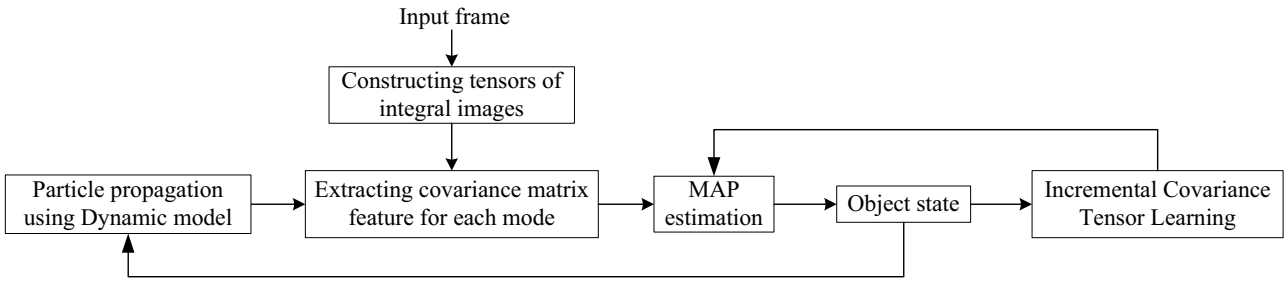

Figure 1. The architecture of the tracking framework.

an image.

However, the appearance models adopted in the above mentioned tracking approaches are usually sensitive to the variations in illumination, view, and pose. This is because they lack a competent object description criterion that captures both statistical and spatial properties of object appearance. Recently, Tuzel *et al.* [15] proposed a covariance region descriptor to characterize the object appearance, which is capable of capturing the correlations among extracted features inside an object region and is robust to the variations in illumination, view, and pose. Since a nonsingular covariance matrix is a symmetric positive definite matrix lying on a connected Riemannian manifold, statistics for covariance matrices of image features may be computed through Riemannian geometry. One existing algorithm for statistics on a Riemannian manifold is based on the affine-invariant Riemannian metric, under which the Riemannian mean has no closed form and is computed by an iterative numerical procedure [16]. In the recently proposed covariance tracking approach [17], 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 the long-term tracking. Based on the Log-Euclidean Riemannian metric [18], Li *et al.* [19] 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. For the covariance computation, their approach could not take advantage of integral images. As a result, their approach is also very time-consuming and cannot be directly used to real applications.

Adopting the covariance descriptor as appearance model, we propose a novel tracking approach via incremental covariance tensor learning. In contrast to the covariance tracking algorithm [17], with the tensor analysis, we simplify the complex model update process on Riemannian manifold by computing weighted sample covariance which can be updated incrementally during the object tracking process. Thus our appearance model can update more efficiently. This is the main contribution of our work. Further, our method uses a particle filter [20] for motion parameter estimation rather than the exhaustive search-based method [17] which is very time-consuming and often distracted by outliers. Moreover, integral image data structure [15] is adopted to accelerate the tracker.

2. The Framework for Visual Tracking

2.1. Overview of the framework

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

2.2. Object representation

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

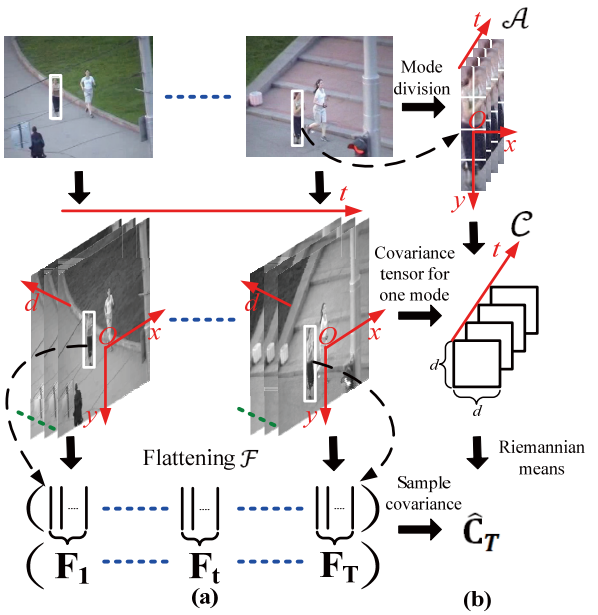

Figure 2. Illustration of object representation, the flattening of $\mathcal{F}$ and two different formulations for $\hat{C}_T$. The input *Jogging* sequence is shown in the upper part of (a) while the 4th-order object feature tensor $\mathcal{F}$ is displayed in the middle one of (a). The result of flattening $\mathcal{F}$ is exhibited in the lower part of (a). The object appearance tensor $\mathcal{A}$ with mode division is shown in the upper part of (b) while the covariance tensor for one mode is displayed in the middle one of (b). The lower part of (b) displays two different formulations for $\hat{C}_T$.

As time progresses, all the object appearances form an object appearance tensor $\mathcal{A} = \{A_t \in R^{m\times n}\}_{t=1,2,\ldots,T}$, 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):

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

where $f_{t,x,y}$ 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

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

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

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

where μ_t is the column mean of F_t. This covariance can be viewed as an informative region descriptor for an object [15]. All the covariance matrices up to time T, $\{C_t \in R^{d\times d}\}_{t=1,2,\ldots,T}$, constitute a covariance tensor $\mathcal{C} \in R^{d\times d\times T}$. We need to track the changes of $\mathcal{C}$ and have to update the compact representation of $\mathcal{C}$ as new data arrive.

2.2.1 *Covariance tensor representation on Riemannian manifold*

A straightforward compact representation of $\mathcal{C}$ is the mean of $\{C_t \in R^{d\times d}\}_{t=1,2,\ldots,T}$. Porikli *et al.* [17] 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 $\left\|log\left(\mathrm{X}^{-\frac{1}{2}}\cdot\mathrm{Y}\cdot\mathrm{X}^{-\frac{1}{2}}\right)\right\|$. An equal form [22] is

$$\rho(\mathrm{X},\mathrm{Y}) = \sqrt{\sum_{k=1}^{d} ln^2\lambda_k(\mathrm{X},\mathrm{Y})} \quad (2)$$

where $\{\lambda_k(\mathrm{X},\mathrm{Y})\}$ are the generalized eigenvalues of X and Y. Under this metric, an iterative numerical procedure [16] is applied to compute the Riemannian mean. The computational cost for this Riemannian mean grows linearly as time progresses. Under the Log-Euclidean Riemannian metric [18], the distance between two points X and Y is calculated by $\|log(\mathrm{Y}) - log(\mathrm{X})\|$. Based on this metric, Li *et al.* [19] presented an online subspace learning algorithm. The covariance computation of their approach could not take advantage of integral images.

In the following, we propose a novel compact representation of $\mathcal{C}$, which can be updated efficiently without computing Riemannian mean. The computational complexity is $\mathcal{O}(1)$, which means that the computation time of the compact tensor representation remains the same even if T becomes very large.

2.3. Incremental Covariance Tensor Learning

From a generative perspective, μ_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 can get the representation by computing the column covariance of F:

$$\hat{C}_T = \frac{1}{N\cdot T-1}\sum_{t=1}^{T}\sum_{n=1}^{N}\left(f_{t,i}-\hat{\mu}_T\right)\left(f_{t,i}-\hat{\mu}_T\right)^T \quad (3)$$

where $\hat{\mu}_T$ is the column mean of F. Although this formulation 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 can treat this 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, this problem can be reformulated as estimating

the weighted sample covariance of F. Further, under this formulation, it is not necessary to normalize the object appearance to the same size as [19]. In the following, we use N_t to denote the size of the object region at time *t*.

One of the critical issues for our formulation is the design of the sample weight. Four issues are taken into account to design the sample weight: 1) The weight of each sample should be varying over time *T*; 2) The samples from current time *T* should have the highest weight; 3) The weight should not affect the fast covariance computation using integral images; 4) The covariance tensor representation could be obtained incrementally. 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 definition and theorem.

Definition 1 *Denote the weighted samples up to current time T as* $\hat{F}_T = \{f_{t,i}, w_{T,t,i}\}_{t=1,\dots,T;i=1,\dots,N_t}$, *where* $w_{T,t,i}$ *is the weight of sample* $f_{t,i}$. *Let the number of samples in* $\hat{F}_T$ *be* $\widehat{N}_T$ *and the sum of weights in* $\hat{F}_T$ *be* $\widehat{w}_T$, *namely* $\widehat{N}_T = \sum_{t=1}^{T} N_t$ *and* $\widehat{w}_T = \sum_{t=1}^{T}\sum_{i=1}^{N_t} w_{T,t,i}$.

Note*. The sample weight* $w_{T,t,i}$ *is varying over time T.*

Definition 2 *Let* C_t, μ_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:*

$$\hat{C}_T = \frac{1}{1-\bar{w}_T^2}\sum_{t=1}^{T}\sum_{i=1}^{N_t}\frac{w_{T,t,i}}{\widehat{w}_T}\left(f_{t,i}-\hat{\mu}_T\right)\left(f_{t,i}-\hat{\mu}_T\right)^T \quad (4)$$

Where

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

$$\hat{\mu}_T = \frac{1}{\widehat{w}_T}\sum_{t=1}^{T}\sum_{i=1}^{N_t} w_{T,t,i} f_{t,i} \quad (5)$$

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

$$C_t = \frac{1}{N_t-1}\sum_{i=1}^{N_t}\left(f_{t,i}-\mu_t\right)\left(f_{t,i}-\mu_t\right)^T \quad (6)$$

$$\mu_t = \frac{1}{N_t}\sum_{i=1}^{N_t} f_{t,i} \quad (7)$$

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

$$\hat{C}_T = \frac{1}{\widehat{w}_T\cdot\left(1-\bar{w}_T^2\right)}\Big\{w\cdot\widehat{w}_{T-1}\cdot(1-\bar{w}_{T-1}^2)\hat{C}_{T-1} + (N_T-1)C_T + \frac{w\cdot\widehat{w}_{T-1}\cdot N_T}{\widehat{w}_T}(\mu_T-\hat{\mu}_{T-1})(\mu_T-\hat{\mu}_{T-1})^T\Big\} \quad (8)$$

where $\widehat{w}_T = w\cdot\widehat{w}_{T-1} + N_T$, $\hat{\mu}_T = \frac{w\cdot\widehat{w}_{T-1}}{\widehat{w}_T}\hat{\mu}_{T-1} + \frac{N_T}{\widehat{w}_T}\mu_T$, $\bar{w}_T^2 = \frac{\left(\widehat{w}_{T-1}{}^2\cdot\bar{w}_{T-1}^2 - N_{T-1}\right)\cdot w^2 + N_T}{(w\cdot\widehat{w}_{T-1}+N_T)^2}$. *The initial condition is* $\hat{C}_1 = C_1$, $\hat{\mu}_1 = \mu_1$, $\widehat{w}_1 = N_1$, $\bar{w}_1^2 = \frac{1}{N_1}$.

The proof of this theorem appears in the Appendix.

If we let *w* be equal to 1, which means all samples are treated equally, we can obtain the sample covariance of *F* from Eq. (8):

$$\hat{C}_T = \frac{1}{\widehat{N}_T-1}\Big\{(\widehat{N}_{T-1}-1)\widehat{C}_{T-1} + (N_T-1)C_T + \frac{N_T\widehat{N}_{T-1}}{\widehat{N}_T}\left(\mu_T-\hat{\mu}_{T-1}\right)\left(\mu_T-\hat{\mu}_{T-1}\right)^T\Big\} \quad (9)$$

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

$$\widehat{C}_T = \sum_{t=1}^{T} w_{t,C} C_t + \sum_{t=2}^{T} w_{t,\mu}\left(\mu_t-\hat{\mu}_{t-1}\right)\left(\mu_t-\hat{\mu}_{t-1}\right)^T \quad (10)$$

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 Table 1.

Table 1. The incremental covariance tensor learning algorithm.

Given C_T, μ_T, N_T, $\hat{C}_{T-1}$, $\hat{\mu}_{T-1}$, $\widehat{w}_{T-1}$, N_{T-1}, $\bar{w}_{T-1}^2$, as well as $w_{T,t,i} = w^{T-t}$, $w \in [0,1]$, compute $\hat{C}_T$:

1. Update the sum of sample weights up to time *T*: $\widehat{w}_T = w\cdot\widehat{w}_{T-1} + N_T$.
2. Update the squared sum of normalized sample weights up to time *T*: $\bar{w}_T^2 = \frac{\left(\widehat{w}_{T-1}{}^2\cdot\bar{w}_{T-1}^2 - N_{T-1}\right)\cdot w^2 + N_T}{(w\cdot\widehat{w}_{T-1}+N_T)^2}$.
3. Update the weighted mean of all the samples up to time *T*: $\hat{\mu}_T = \frac{w\cdot\widehat{w}_{T-1}}{\widehat{w}_T}\hat{\mu}_{T-1} + \frac{N_T}{\widehat{w}_T}\mu_T$.
4. Finally, update the weighted covariance of all the samples up to time *T*:
$$\hat{C}_T = \frac{1}{\widehat{w}_T\cdot\left(1-\bar{w}_T^2\right)}\Big\{w\cdot\widehat{w}_{T-1}\cdot(1-\bar{w}_{T-1}^2)\hat{C}_{T-1} + (N_T-1)C_T + \frac{w\cdot\widehat{w}_{T-1}\cdot N_T}{\widehat{w}_T}(\mu_T-\hat{\mu}_{T-1})(\mu_T-\hat{\mu}_{T-1})^T\Big\}.$$

The initial condition is $\hat{C}_1 = C_1$, $\hat{\mu}_1 = \mu_1$, $\widehat{w}_1 = N_1$, $\bar{w}_1^2 = \frac{1}{N_1}$.

2.4. Bayesian inference for visual tracking

In the Bayesian perspective, object tracking can be viewed as a state estimation problem. The purpose of tracking is to estimate $p(x_t|y_{1:t})$, which stands for the distribution of target state x_t given all observations $y_{1:t}$ up to time *t*. In our case, the state refers to an object's 2D location and scale.

The density propagation of $p(x_t|y_{1:t})$ can be formulated by the well-known two-step recursion:

Prediction: $p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})\,dx_{t-1}$

Update: $p(x_t|y_{1:t}) \propto p(y_t|x_t)\,p(x_t|y_{1:t-1})$ (11)

The posterior $p(x_t|y_{1:t})$ is approximated by a set of weighted samples. The recursive inference is implemented with resampling and importance sampling processes. The state dynamics $p(x_t|x_{t-1})$ is assumed to be Gaussian distribution as $\mathcal{N}(x_t; x_{t-1}, \Sigma)$, where Σ is a diagonal covariance matrix whose diagonal elements are $\sigma_x^2, \sigma_y^2, \sigma_s^2$, respectively. 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 eight modes $\{\hat{C}_{T,i}\}_{i=1,2,\ldots,8}$. Each mode $C_i(x_t)$ of the candidate sample x_t is compared to the corresponding model by Eq. (2). Thus $p(y_t|x_t)$ can be formulated as:

$$p(y_t|x_t) \propto exp\{-\lambda \cdot \sum_{i=1}^{8} \omega_i \cdot \rho^2[\hat{C}_{T,i}, C_i(x_t)]\} \quad (12)$$

where ω_i is the weight for the i^{th} mode ($\sum_{i=1}^{8} \omega_i$=1 and ω_i =0.125 in the experiments). After MAP estimation, we just use the covariance matrices of image features associated with the estimated target state to update the compact covariance tensor model for each mode.

2.5. Fast covariance computation

By our definition for the object state, each particle corresponds to an up-right rectangle. Therefore, it is possible to improve the computational complexity of covariance computation using integral histogram techniques [21]. After constructing tensors of integral images corresponding to each feature dimension and multiplication of any two feature dimensions, the covariance matrix of any arbitrary rectangular region can be computed independent of the region size. Refer to [15] for more details.

3. Experiments

During the visual tracking, the object region is divided into eight modes. For each pixel, a 7-dimesional feature vector is extracted:

$$\left(x, y, R(x,y), G(x,y), B(x,y), I_x(x,y), I_y(x,y)\right)$$

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 7×7 symmetric matrix. Further, the compact covariance tensor representation is learned online every frame. For the particle filtering in the visual tracking, the number of particles is set to be 100. The three diagonal elements $(\sigma_x^2, \sigma_y^2, \sigma_s^2)$ of the covariance matrix Σ for the state dynamics are assigned as $(5^2, 5^2, 0.02^2)$. λ in Eq. (12) and w in Eq. (8) is set to 0.1 and 0.95, respectively. The approach was implemented using C++ and performed on a PC with an Intel Pentium E2140 CPU (1.6-GHz). Without code optimization, our algorithm can achieve around 20 *fps* for image sequences with resolution 320×240.

3.1. Speed comparison for model update

From Eq. (8), 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 [17] 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. As visible, our method has clearly $\mathcal{O}(1)$ time complexity and it is significantly faster than the approach used in [17].

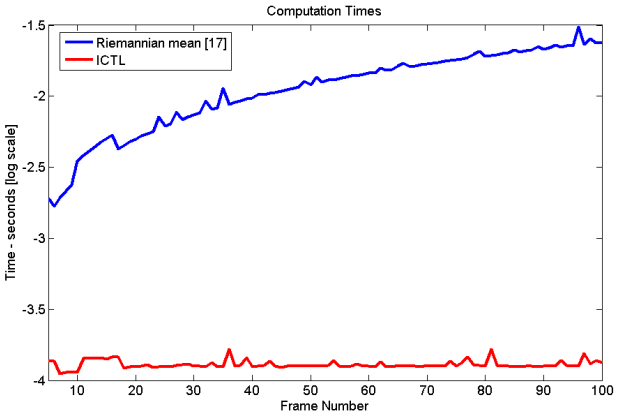

Figure 3. Speed comparison for model update.

3.2. Experimental results

We first test our algorithm using the sequence, *Jogging*, studied in [17]. Figure 4 shows the empirical results using our proposed method. Note that our method is able to track the target undergoing gradual scale changes (#45, #223). Further, our method is able to track the target with severe full occlusion (#71, #78), which lasts around 20 frames. Compared with the results reported in [17], 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 this outperformance.

The second image sequence, shown in Fig. 5, 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 occlusion (#69), large scale variation (#499, #542), and lighting variation (#69, #284). 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. Nevertheless, our tracker eventually fails to recover the true scale after frame 499 as a result of a combination of drastic scale change. Note that [23], which is the first paper to

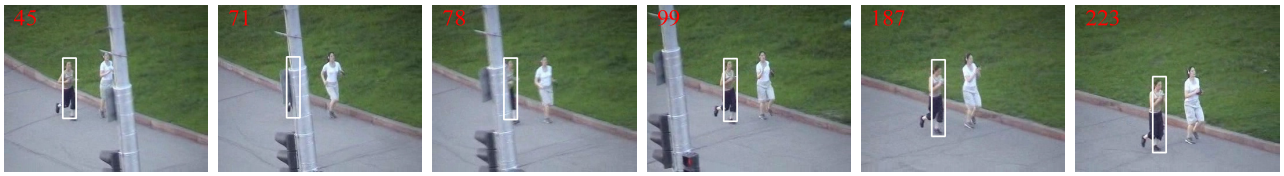

Figure 4: *Jogging*: The tracking results over representative frames under full occlusion.

Figure 5: *Woman*: The tracking results over representative frames under partial occlusions, illumination variations and sudden scale changes.

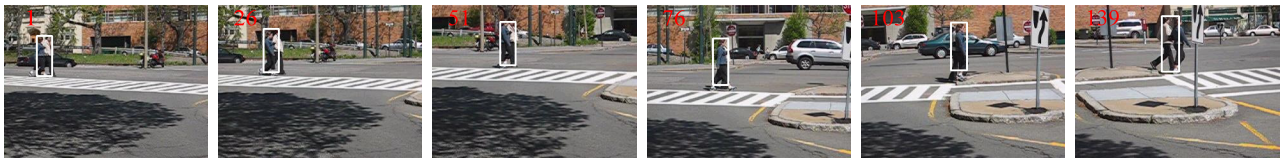

Figure 6: *Couple*: The tracking results over representative frames under hand-held camera.

Figure 7. *Crossing*: The tracking results over representative frames where the target has indistinctive color and texture.

Figure 8. *Subway*: The tracking results over representative frames where the target has indistinctive color and texture

study this sequence, did not report the tracking results after frame 456.

Figure 6 shows the tracking results using a challenging sequence, captured from a hand-held camera, in which a couple is walking and the appearance of the couple is changing over time. Notice that there is also a large scale variation in the target relative to the camera (#1, #139). Even with the significant camera motion and low frame rate, our algorithm is able to track the target throughout the sequence. Furthermore, the compact tensor representation is constructed from scratch and is updated to reflect the appearance variation of the target.

Figure 7 shows the results of tracking a pedestrian, as he is crossing the street and halfway through the sequence, he is standing in front of a car that has the same color as he does. 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. 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 frame 62, the target rectangular window contains some light pixels. The weighted incremental model update adapts the target model to this background changes. The results show that our algorithm faithfully models the appearance of an arbitrary object in the presence of noisy background pixels. Our algorithm is also able to track objects in clutter environment, such as the sequence of a human walking in the subway, shown in Fig. 8. Despite many similar objects in the scenario, and indistinctive texture feature to background, our algorithm is able to track the human well.

3.3. Qualitative comparison

As a qualitative benchmark, we ran two state-of-the-art algorithms, the covariance tracker [17] and Mean Shift [24] tracker, on all the sequences. The results are demonstrated in Fig. 9. As can be seen in the figure (and corresponding videos), our method provides the best performance. The covariance tracker simply exhaustively searches in the whole image for the region that best matches the model descriptor. This maximal likelihood estimation is very time-consuming and easily runs into problems by the background clutter, as demonstrated in Fig. 9. In the sequence *Couple*, our tracker loses the target in some frames. This is due to the significant camera motion and low frame rate (which makes the motions between frames more significant, as when tracking fast-moving objects). With the use of a particle filter, our tracker is able to recover from temporary drifts. In our experiments, we found that the covariance tracker [17] is sensitive to the initialization of target. This may be the reason that the results of sequence *Jogging* and *Subway* are not consistent with those shown in [17].

On the other hand, the Mean Shift tracker performs poorly, experiencing significant drift off the target objects. This is due to local optimization of the Mean Shift tracker. It cannot recover the target after the occlusions.

Figure 9. A comparison of our tracker (indicated with a white box) with the Covariance tracker [17] (shown in red rectangle) and the Mean Shift [24] (depicted by a blue box) on all the test video sequences.

3.4. Qualitative analysis of *ICTL*

We use the sequence *Crossing* to test the effectiveness of the proposed *ICTL*. Three trackers are exploited for this qualitative analysis: Tracker-A uses the proposed approach with default parameter setting; Tracker-B uses the sample covariance for model update, namely, the parameter w in equation (8) is set to 1; Tracker-C is a tracker without model update. The results are illustrated in Fig. 10. As can be seen in the figure, all these three trackers work well before frame 52. When the target window includes more background clutter (white pixels), the Tracker-C drifts first and loses the target after frame 77. The Tracker-B drifts from frame 76 and lost the target in frame 79. While our proposed Tracker-A is able to track the target throughout the sequence. This outperformance can be attributed to the weighting scheme adopted in the proposed *ICTL*.

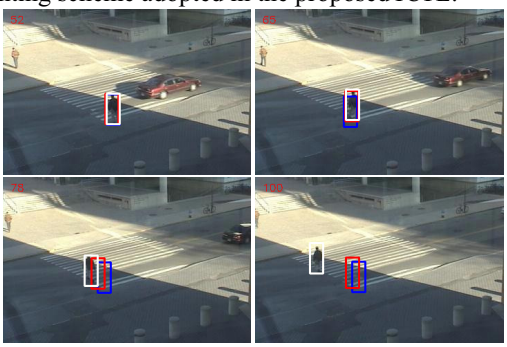

Figure 10. The effectiveness test of *ICTL* using three trackers: Tracker-A(indicated with a white box), Tracker-B (shown in red rectangle) and Tracker-C (depicted by a blue box).

3.5. Discussion

In summary, the experimental results show that our approach is robust and insensitive to occlusions, pose variations, and background clutter. The proposed *ICTL* constructs a compact covariance tensor representation to capture varying object appearance in eight modes, where the spatial structure of object appearance is incorporated into the multi-mode representation. Even if the information of some modes is partially lost or drastically varies, *ICTL* is capable of recovering the information using the cues of the information from other local modes. Consequently, *ICTL* is an effective appearance model update algorithm which performs well in modeling appearance changes of an object in many complex scenarios. Further, the proposed appearance learning algorithm has only $O(1)$ computational complexity and with the use of integral images, our tracker achieves real-time performance.

4. Conclusion

In this paper, we present a real-time visual tracking approach with incremental covariance model update. In the proposed method, the covariance matrix of image features has been used to represent object appearance. Further, an incremental covariance tensor learning algorithm has been proposed to reflect the appearance changes of an object. Moreover, our method uses a particle filter for motion parameter estimation, and with the use of integral images our tracker achieves real-time performance. Compared with the state-of-art covariance tracking method [17], the

proposed algorithm is faster and more robust to occlusions and object pose variations. Experimental results demonstrate that the proposed method is promising.

5. Acknowledgement

This work is partially supported by the National Natural Science Foundation of China (Grant No. 60605004 and 60833006), Natural Science Foundation of Beijing (Grant No. 4072025), and 973 Program (Project No. 2010CB327900，2010CB327905).

6. Appendix

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

Lemma 1. *If* $w_{T,t,i} = w^{T-t}$, $w \in [0,1]$, *we have*

$\widehat{w}_T = w \cdot \widehat{w}_{T-1} + N_T$, *and* $\overline{w}_T^2 = \frac{(\widehat{w}_{T-1}{}^2 \cdot \overline{w}_{T-1}^2 - N_{T-1}) \cdot w^2 + N_T}{(w \cdot \widehat{w}_{T-1} + N_T)^2}$.

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

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

Lemma 3.*If weights of all the samples at time T are equal,*

$\sum_{i=1}^{N_T}(f_{T,i} - \hat{\mu}_T)(f_{T,i} - \hat{\mu}_T)^T$

$= (N_T - 1)C_T + N_T(\mu_T - \hat{\mu}_T)(\mu_T - \hat{\mu}_T)^T$

Lemma 4. *If* $w_{T,t,i} = w^{T-t}$, $w \in [0,1]$, *we have*

$\hat{\mu}_T = \frac{w \cdot \widehat{w}_{T-1}}{\widehat{w}_T}\hat{\mu}_{T-1} + \frac{N_T}{\widehat{w}_T}\mu_T$,

$\hat{\mu}_{T-1} - \hat{\mu}_T = \frac{N_T}{\widehat{w}_T}(\mu_T - \hat{\mu}_{T-1})$,

$\mu_T - \hat{\mu}_T = \frac{w \cdot \widehat{w}_{T-1}}{\widehat{w}_T}(\mu_T - \hat{\mu}_{T-1})$.

Lemma 5. *If* $w_{T,t,i} = w^{T-t}$, $w \in [0,1]$, *we have*

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

$= w \cdot \widehat{w}_{T-1} \cdot (1 - \widehat{w}_{T-1}{}^2)\hat{C}_{T-1}$

$+ w \cdot \widehat{w}_{T-1}(\hat{\mu}_{T-1} - \hat{\mu}_T)(\hat{\mu}_{T-1} - \hat{\mu}_T)^T$

Proof of Theorem 1:

$\hat{C}_T = \frac{1}{1-\overline{w}_T^2}\sum_{t=1}^{T}\sum_{i=1}^{N_t}\frac{w_{T,t,i}}{\widehat{w}_T}(f_{t,i} - \hat{\mu}_T)(f_{t,i} - \hat{\mu}_T)^T$

thus

$\widehat{w}_T \cdot (1 - \overline{w}_T^2)\hat{C}_T$

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

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

$+ \sum_{i=1}^{N_T}(f_{T,i} - \hat{\mu}_T)(f_{T,i} - \hat{\mu}_T)^T$ (Using lemma 3 and 5)

$= w \cdot \widehat{w}_{T-1} \cdot (1 - \overline{w}_{T-1}^2)\hat{C}_{T-1}$

$+ w \cdot \widehat{w}_{T-1}(\hat{\mu}_{T-1} - \hat{\mu}_T)(\hat{\mu}_{T-1} - \hat{\mu}_T)^T$ (Using lemma 4)

$+ (N_T - 1)C_T + N_T(\mu_T - \hat{\mu}_T)(\mu_T - \hat{\mu}_T)^T$

$= w \cdot \widehat{w}_{T-1} \cdot (1 - \overline{w}_{T-1}^2)\hat{C}_{T-1} + (N_T - 1)C_T$

$+ w \cdot \widehat{w}_{T-1}\left(\frac{N_T}{\widehat{w}_T}\right)^2(\mu_T - \hat{\mu}_{T-1})(\mu_T - \hat{\mu}_{T-1})^T$

$+ N_T\left(\frac{w \cdot \widehat{w}_{T-1}}{\widehat{w}_T}\right)^2(\mu_T - \hat{\mu}_{T-1})(\mu_T - \hat{\mu}_{T-1})^T$ (Using lemma1)

$= w \cdot \widehat{w}_{T-1} \cdot (1 - \overline{w}_{T-1}^2)\hat{C}_{T-1} + (N_T - 1)C_T$

$+ \frac{w \cdot \widehat{w}_{T-1} \cdot N_T}{\widehat{w}_T}(\mu_T - \hat{\mu}_{T-1})(\mu_T - \hat{\mu}_{T-1})^T$ □

7. References

[1] G. Hager and P. Belhumeur. Real-time tracking of image regions with changes in geometry and illumination. In: CVPR.1996

[2] M. J. Black, D. J. Fleet, and Y. Yacoob. A framework for modeling appearance change in image sequence. In: ICCV.1998

[3] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi. Robust Online Appearance Models for Visual Tracking. In: CVPR.2001

[4] T. Yu and Y. Wu. Differential Tracking based on Spatial-Appearance Model (SAM). In: CVPR.2006

[5] J. Li, S. K. Zhou and R. Chellappa. Appearance Modeling under Geometric Context. In: ICCV.2005.

[6] S. K. Zhou, R. Chellappa, and B. Moghaddam. Visual Tracking and Recognition Using Appearance-Adaptive Models in Particle Filters. *IEEE Trans. On Image Processing,* Vol. 13, pp.1491-1506 , November 2004.

[7] M. J. Black and A. D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using view-based representation. In: ECCV.1996.

[8] D. Ross, J. Limy, R. Lin andM. Yang. Incremental Learning for Visual Tracking. In: IJCV 2007

[9] Avidan, S. Support vector tracking. In: PAMI.2004

[10] Collins, R.T., Liu, Y. and Leordeanu, M. Online selection of discriminative tracking features. In: PAMI. vol 27. 2005 1631–1643

[11] Avidan, S. Ensemble tracking. In: CVPR. 2005

[12] Q. Yu, T. B. Dinh and G. Medioni. Online Tracking and Reacquisition Using Co-trained Generative and Discriminative Trackers. In: ECCV. 2008

[13] A. Levy and M. Lindenbaum. Sequential Karhunen-Loeve basis extraction and its application to images. *IEEE* Transactions on Image Processing, 9(8):1371–1374, 2000.

[14] D. Skocaj and A. Leonardis. Weighted and Robust Incremental Method for Subspace Learning. In ICCV2003.

[15] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In ECCV 2006.

[16] X. Pennec, P. Fillard, and N. Ayache. A Riemannian Framework for Tensor Computing. In: IJCV 2006.

[17] F. Porikli, O. Tuzel, and P. Meer. Covariance tracking using model update based on Lie algebra. In CVPR 2006.

[18] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Geometric Means in a Novel Vector Space Structure on Symmetric Positive-Definite Matrices. *SIAM* Journal on Matrix Analysis and Applications, 2006.

[19] X. Li, W. Hu, Z. Zhang, X. Zhang and G. Luo. Visual Tracking via Incremental Log-Euclidean Riemannian Subspace Learning. In: CVPR. 2008

[20] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In: ECCV.1996

[21] F. Porikli. Integral histogram: A fast way to extract histograms in Cartesian spaces. In: CVPR 2005

[22] W. Förstner and B. Moonen. A metric for covariance matrices. Technical report, Dept. of Geodesy and Geoinformatics, Stuttgart University, 1999.

[23] A. Adam, E. Rivlin and I. Shimshoni. Robust Fragments-based Tracking using the Integral Histogram. In CVPR 2006.

[24] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. In: PAMI 2003.