Robust Visual Tracking Based on an Effective Appearance Model

Xi Li[1], Weiming Hu[1], Zhongfei Zhang[2], and Xiaoqin Zhang[1]

[1] National Laboratory of Pattern Recognition, CASIA, Beijing, China
{lixi,wmhu,xqzhang}@nlpr.ia.ac.cn
[2] State University of New York, Binghamton, NY 13902, USA
zhongfei@cs.binghamton.edu

Abstract. Most existing appearance models for visual tracking usually construct a pixel-based representation of object appearance so that they are incapable of fully capturing both global and local spatial layout information of object appearance. In order to address this problem, we propose a novel spatial Log-Euclidean appearance model (referred as *SLAM*) under the recently introduced Log-Euclidean Riemannian metric [23]. *SLAM* is capable of capturing both the global and local spatial layout information of object appearance by constructing a block-based Log-Euclidean eigenspace representation. Specifically, the process of learning the proposed *SLAM* consists of five steps—appearance block division, online Log-Euclidean eigenspace learning, local spatial weighting, global spatial weighting, and likelihood evaluation. Furthermore, a novel online Log-Euclidean Riemannian subspace learning algorithm (*IRSL*) [14] is applied to incrementally update the proposed *SLAM*. Tracking is then led by the Bayesian state inference framework in which a particle filter is used for propagating sample distributions over the time. Theoretic analysis and experimental evaluations demonstrate the promise and effectiveness of the proposed *SLAM*.

1 Introduction

For visual tracking, handling appearance variations of an object is a fundamental and challenging task. In general, there are two types of appearance variations: intrinsic and extrinsic. Pose variation and/or shape deformation of an object are considered as the intrinsic appearance variations while the extrinsic variations are due to the changes resulting from different illumination, camera motion, camera viewpoint, and occlusion. Consequently, effectively modeling such appearance variations plays a critical role in visual tracking.

Hager and Belhumeur [1] propose a tracking algorithm which uses an extended gradient-based optical flow method to handle object tracking under varying illumination conditions. In [3], curves or splines are exploited to represent the appearance of an object to develop the Condensation algorithm for contour tracking. Due to the simplistic representation scheme, the algorithm is unable to handle the pose or illumination change, resulting in tracking failures under a varying lighting condition. Zhao *et al.*[18] present a fast differential EMD tracking method which is robust to illumination changes. Silveira and Malis [16] present a new algorithm for handling generic lighting changes.

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part IV, LNCS 5305, pp. 396–408, 2008.
© Springer-Verlag Berlin Heidelberg 2008

Black *et al.*[4] employ a mixture model to represent and recover the appearance changes in consecutive frames. Jepson *et al.*[5] develop a more elaborate mixture model with an online EM algorithm to explicitly model appearance changes during tracking. Zhou *et al.*[6] embed appearance-adaptive models into a particle filter to achieve a robust visual tracking. Wang *et al.*[20] present an adaptive appearance model based on the Gaussian mixture model (GMM) in a joint spatial-color space (referred to as SMOG). SMOG captures rich spatial layout and color information. Yilmaz [15] proposes an object tracking algorithm based on the asymmetric kernel mean shift with adaptively varying the scale and orientation of the kernel. Nguyen *et al.*[17] propose a kernel-based tracking approach based on maximum likelihood estimation.

Lee and Kriegman [7] present an online learning algorithm to incrementally learn a generic appearance model for video-based recognition and tracking. Lim *et al.*[8] present a human tracking framework using robust system dynamics identification and nonlinear dimension reduction techniques. Black *et al.*[2] present a subspace learning based tracking algorithm with the subspace constancy assumption. A pre-trained, view-based eigenbasis representation is used for modeling appearance variations. However, the algorithm does not work well in the scene clutter with a large lighting change due to the subspace constancy assumption. Ho *et al.*[9] present a visual tracking algorithm based on linear subspace learning. Li *et al.*[10] propose an incremental PCA algorithm for subspace learning. In [11], a weighted incremental PCA algorithm for subspace learning is presented. Limy *et al.*[12] propose a generalized tracking framework based on the incremental image-as-vector subspace learning methods with a sample mean update. Chen and Yang [19] present a robust spatial bias appearance model learned dynamically in video. The model fully exploits local region confidences for robustly tracking objects against partial occlusions and complex backgrounds. In [13], li *et al.* present a visual tracking framework based on online tensor decomposition.

However, the aforementioned appearance-based tracking methods share a problem that their appearance models lack a competent object description criterion that captures both statistical and spatial properties of object appearance. As a result, they are usually sensitive to the variations in illumination, view, and pose. In order to tackle this problem, Tuzel *et al.* [24] and Porikli *et al.*[21] propose a covariance matrix descriptor for characterizing the appearance of an object. The covariance matrix descriptor, based on several covariance matrices of image features, is capable of fully capturing the information of the variances and the spatial correlations of the extracted features inside an object region. In particular, the covariance matrix descriptor is robust to the variations in illumination, view, and pose. Since a nonsingular covariance matrix is a symmetric positive definite (SPD) matrix lying on a connected Riemannian manifold, statistics for covariance matrices of image features may be computed through Riemannian geometry. Nevertheless, most existing algorithms for statistics on a Riemannian manifold rely heavily on the affine-invariant Riemannian metric, under which the Riemannian mean has no closed form. Recently, Arsigny *et al.*[23] propose a novel Log-Euclidean Riemannian metric for statistics on SPD matrices. Under this metric, distances and Riemannian means take a much simpler form than the widely used affine-invariant Riemannian metric.

Based on the Log-Euclidean Riemannian metric [23], we develop a tracking framework in this paper. The main contributions of the developed framework are as follows. First, the framework does not need to know any prior knowledge of the object, and only assumes that the initialization of the object region is provided. Second, a novel block-based spatial Log-Euclidean appearance model (*SLAM*) is proposed to fully capture both the global and local spatial properties of object appearance. In *SLAM*, the object region is first divided into several $p \times q$ object blocks, each of which is represented by the covariance matrix of image features. A low dimensional Log-Euclidean Riemannian eigenspace representation for each block is then learned online, and updated incrementally over the time. Third, we present a spatial weighting scheme to capture both the global and local spatial layout information among blocks. Fourth, while the Condensation algorithm [3] is used for propagating the sample distributions over the time, we develop an effective likelihood function based on the learned Log-Euclidean eigenspace model. Last, the Log-Euclidean Riemannian subspace learning algorithm (i.e., *IRSL*) [14] is applied to update the proposed *SLAM* as new data arrive.

2 The Framework for Visual Tracking

2.1 Overview of the Framework

The tracking framework includes two stages:(a) online *SLAM* learning; and (b)Bayesian state inference for visual tracking.

In the first stage, five steps are needed. They are appearance block division, online Log-Euclidean eigenspace learning, local spatial weighting, global spatial weighting, and likelihood evaluation, respectively. A brief introduction to these five steps is given as follows. First, the object appearance is uniformly divided into several blocks. Second, the covariance matrix feature from Eq. (2) in [14] is extracted for representing each block. After the Log-Euclidean mapping from Eq. (5) in [14], a low dimensional Log-Euclidean Riemannian eigenspace model is learned online. The model uses the incremental Log-Euclidean Riemannian subspace learning algorithm (*IRSL*) [14] to find the dominant projection subspaces of the Log-Euclidean unfolding matrices. Third, the block-specific likelihood between a candidate block and the learned Log-Euclidean eigenspace model is computed to obtain a block related likelihood map for object appearance. Fourth, the likelihood map is filtered by local spatial weighting into a new one. Fifth, the filtered likelihood map is further globally weighted by a spatial Gaussian kernel into a new one. Finally, the overall likelihood between a candidate object region and the learned *SLAM* is computed by multiplying all the block-specific likelihoods after local and global spatial weighting.

In the second stage, the object locations in consecutive frames are estimated by maximum a posterior (MAP) estimation within the Bayesian state inference framework in which a particle filter is applied to propagate sample distributions over the time. After MAP estimation, we just use the block related Log-Euclidean covariance matrices of image features inside the affinely warped image region associated with the highest weighted hypothesis to update the *SLAM*.

These two stages are executed repeatedly as time progresses. Moreover, the framework has a strong adaptability in the sense that when new image data arrive, the Log-

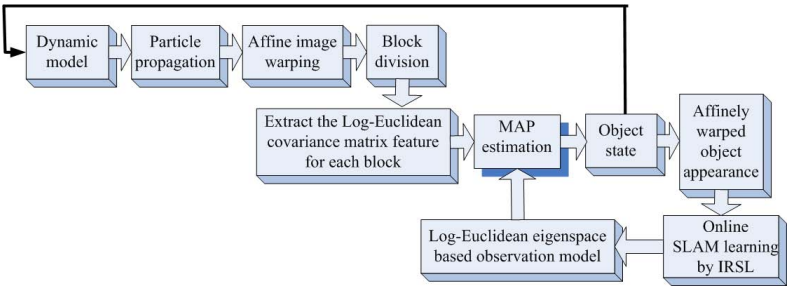

Fig. 1. The architecture of the tracking framework

Euclidean Riemannian eigenspace observation model follows the updating online. The architecture of the framework is shown in Fig. 1.

2.2 Spatial Log-Euclidean Appearance Model (*SLAM*)

The process of learning the SLAM consists of five steps—appearance block division, online Log-Euclidean eigenspace learning, local spatial weighting, global spatial weighting, and likelihood evaluation. The details of these five steps are given as follows.

(1) Appearance block division. Given an object appearance tensor $\mathcal{F} = \{F^t \in R^{m\times n}\}_{t=1,2,\ldots,N}$, we divide the object appearance F^t at any time t into several $p \times q$ blocks ($m = n = 36$ and $p = q = 6$ in the paper), as illustrated in Figs. 2(a) and (b). For each block $F^t_{ij} \in R^{p\times q}$, the covariance matrix feature from Eq. (2) in [14] is extracted for representing F^t_{ij}, i.e., $\mathbf{C}^t_{ij} \in R^{d\times d}$. We call the covariance matrix $\mathbf{C}^t_{ij}$ as the block-(i, j) covariance matrix. In this case, the block-(i, j) covariance matrices $\{\mathbf{C}^t_{ij}\}_{t=1,2,\ldots,N}$ constitute a block-(i, j) covariance tensor $\mathcal{A}_{ij} \in R^{d\times d\times N}$. If $\mathbf{C}^t_{ij}$ is a singular matrix, we replace $\mathbf{C}^t_{ij}$ with $\mathbf{C}^t_{ij} + \epsilon\mathbf{I}_d$, where ϵ is a very small positive constant ($\epsilon = 1\mathrm{e} - 18$ in the experiments), and $\mathbf{I}_d$ is a $d \times d$ identity matrix. By the Log-Euclidean mapping from Eq. (5) in [14], as illustrated in Fig. 2(c), the block-(i, j) covariance subtensor $\mathcal{A}_{ij}$ is transformed into a new one:

$$\mathcal{LA}_{ij} = \{\log(\mathbf{C}^1_{ij}), \ldots, \log(\mathbf{C}^t_{ij}), \ldots, , \log(\mathbf{C}^N_{ij})\} \tag{1}$$

where ϵ is a very small positive constant, and $\mathbf{I}_d$ is a $d \times d$ identity matrix. We call $\mathcal{LA}_{ij}$ as the block-(i, j) Log-Euclidean covariance subtensor, as illustrated in Fig. 2(d). Denote $[\cdot]$ as the rounding operator, m^* as $[\frac{m}{p}]$, and n^* as $[\frac{n}{q}]$. Consequently, all the Log-Euclidean covariance subtensors $\{(\mathcal{LA}_{ij})_{m^*\times n^*}\}_{t=1,2,\ldots,N}$ forms a Log-Euclidean covariance tensor $\mathcal{LA}$ associated with the object appearance tensor $\mathcal{F} \in R^{m\times n\times N}$. With the emergence of new object appearance subtensors, $\mathcal{F}$ is extended along the time axis t (i.e., N increases gradually), leading to the extension of each Log-Euclidean covariance subtensor $\mathcal{LA}_{ij}$ along the time axis t. Consequently, we need to track the changes of $\mathcal{LA}_{ij}$, and need to identify the dominant projection subspace for a compact representation of $\mathcal{LA}_{ij}$ as new data arrive.

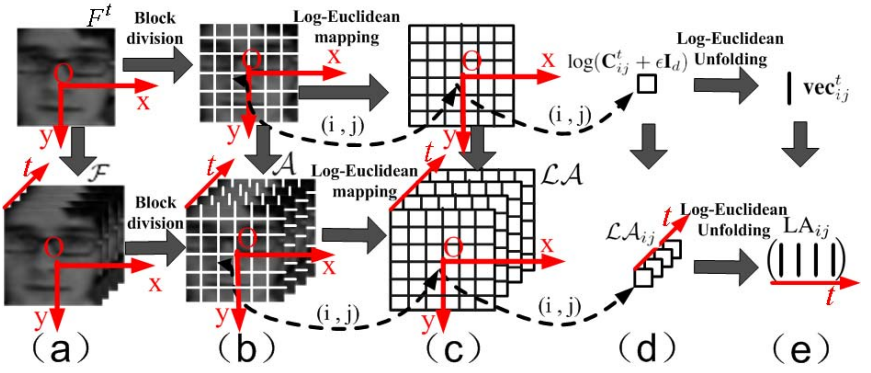

Fig. 2. Illustration of appearance block division, Log-Euclidean mapping, and Log-Euclidean unfolding. A face image F^t at time t is shown in the upper part of (a) while a 3-order face tensor $\mathcal{F} = \{F^t\}_{t=1,2,\ldots,N}$ (i.e., face image ensemble) is displayed in the lower one of (a). The results of appearance block division are exhibited in (b). The Log-Euclidean mapping results are shown in (c). An example of the block-(i, j) Log-Euclidean mapping is given in (d). (e) displays the results of Log-Euclidean unfolding.

Due to the vector space structure of $\log(\mathbf{C}_{ij}^t)$ under the Log-Euclidean Riemannian metric, $\log(\mathbf{C}_{ij}^t)$ is unfolded into a d^2-dimensional vector $\mathbf{vec}_{(i)}^t$ which is formulated as:

$$\mathbf{vec}_{(i)}^t = \mathrm{UT}(\log(\mathbf{C}_{ij}^t)) = (c_1^t, c_2^t, \ldots, c_{d^2}^t)^T \tag{2}$$

where $\mathrm{UT}(\cdot)$ is an operator unfolding a matrix into a column vector. The unfolding process can be illustrated by Figs. 2(e) and 3(a). In Fig. 3(a), the left part displays the covariance tensor $\mathcal{A}_{ij} \in \mathcal{R}^{d\times d\times N}$, the middle part corresponds to the Log-Euclidean covariance tensor $\mathcal{LA}_{ij}$, and the right part is associated with the Log-Euclidean unfolding matrix LA_{ij} with the t-th column being $\mathbf{vec}_{ij}^t$ for $1 \le t \le N$. As a result, LA_{ij} is formulated as:

$$\mathrm{LA}_{ij} = \left(\mathbf{vec}_{ij}^1\ \mathbf{vec}_{ij}^2\ \cdots\ \mathbf{vec}_{ij}^t\ \cdots\ \mathbf{vec}_{ij}^N\right). \tag{3}$$

The next step of the *SLAM* is to learn an online Log-Euclidean eigenspace model for $\mathcal{LA}_{ij}$. Specifically, we will introduce an incremental Log-Euclidean Riemannian subspace learning algorithm (*IRSL*) [14] for the Log-Euclidean unfolding matrix LA_{ij}. *IRSL* applies the online learning technique (R-SVD [12,27]) to find the dominant projection subspaces of LA_{ij}. Furthermore, a new operator $\mathrm{CVD}(\cdot)$ used in *IRSL* is defined as follows. Given a matrix $H = \{\mathrm{K}_1, \mathrm{K}_2, \ldots, \mathrm{K}_g\}$ and its column mean K, we let $\mathrm{CVD}(H)$ denote the SVD (i.e., singular value decomposition) of the matrix $\{\mathrm{K}_1 - \mathrm{K}, \mathrm{K}_2 - \mathrm{K}, \ldots, \mathrm{K}_g - \mathrm{K}\}$.

(2) Online Log-Euclidean eigenspace learning. For each Log-Euclidean covariance subtensor $\mathcal{LA}_{ij}$, *IRSL* [14] is used to incrementally learn a Log-Euclidean eigenspace model (i.e., LA_{ij}'s column mean $\bar{L}_{ij}$ and $\mathrm{CVD}(\mathrm{LA}_{ij}) = \mathrm{U}_{ij}\mathrm{D}_{ij}\mathrm{V}_{ij}^T$) for $\mathcal{LA}_{ij}$. For convenience, we call $\bar{L}_{ij}$ and $\mathrm{CVD}(\mathrm{LA}_{ij})$ as the block-(i, j) Log-Euclidean eigenspace

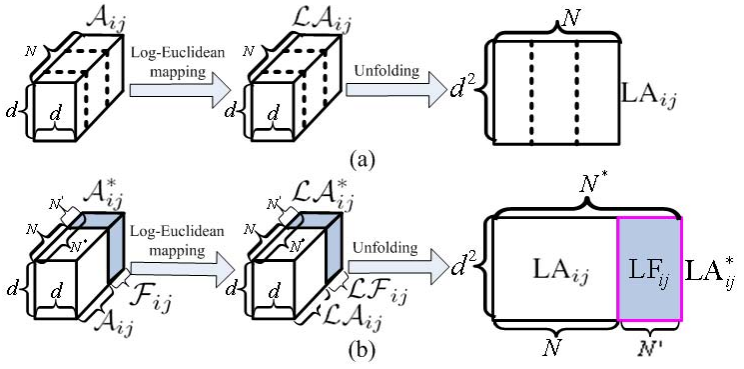

Fig. 3. Illustration of Log-Euclidean unfolding and *IRSL*. (a) shows the generative process of the Log-Euclidean unfolding matrix; (b) displays the incremental learning process of *IRSL*.

model. For a better understanding of *IRSL*, Fig. 3(b) is used to illustrate the incremental learning process of *IRSL*. Please see the details of *IRSL* in [14].

The distance between a candidate sample $B_{i,j}$ and the learned block-(i,j) Log-Euclidean eigenspace model (i.e. LA_{ij}'s column mean $\bar{L}_{ij}$ and $\mathrm{CVD}(\mathrm{LA}_{ij}) = \mathrm{U}_{ij}\mathrm{D}_{ij}\mathrm{V}_{ij}^T$) is determined by the reconstruction error norm:

$$RE_{ij} = \|(\mathbf{vec}_{ij} - \bar{L}_{ij}) - U_{(j)} \cdot U_{(j)}^T \cdot (\mathbf{vec}_{ij} - \bar{L}_{ij})\|^2 \tag{4}$$

where $\|\cdot\|$ is the Frobenius norm, and $\mathbf{vec}_{ij} = \mathrm{UT}(\log(B_{i,j}))$ is obtained from Eq. (2). Thus, the block-(i,j) likelihood p_{ij} is computed as: $p_{ij} \propto \exp(-RE_{ij})$. The smaller the RE_{ij}, the larger the likelihood p_{ij}. As a result, we can obtain a likelihood map $\mathcal{M} = (p_{ij})_{m^* \times n^*} \in R^{m^* \times n^*}$ for all the blocks.

(3) Local spatial weighting. In this step, the likelihood map $\mathcal{M}$ is filtered into a new one $\mathcal{M}_l \in R^{m^* \times n^*}$. The details of the filtering process are given as follows. Denote the

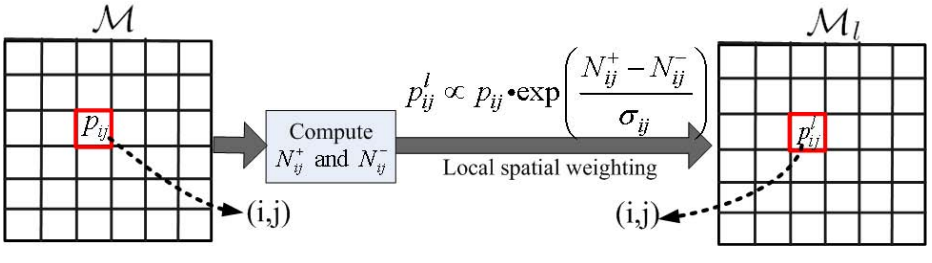

Fig. 4. Illustration of local spatial weighting for the i-th and j-th block. (a) shows the original likelihood map while (b) displays the filtered map by local spatial weighting for the i-th and j-th block.

original map $\mathcal{M} = (p_{ij})_{m^* \times n^*}$, and the filtered map $\mathcal{M}_l = (p^l_{ij})_{m^* \times n^*}$. After filtering by local spatial weighting, the entry p^l_{ij} of $\mathcal{M}_l$ is formulated as:

$$p^l_{ij} \propto p_{ij} \cdot \exp\left(\frac{N^+_{ij} - N^-_{ij}}{\sigma_{ij}}\right), \tag{5}$$

where $N^+_{ij} = \frac{1}{k_{ij}} \sum_{u,v \in \mathbb{N}_{ij}} \text{sgn}\left[\frac{|p_{uv} - p_{ij}| + (p_{uv} - p_{ij})}{2}\right]$, $N^-_{ij} = \frac{1}{k_{ij}} \sum_{u,v \in \mathbb{N}_{ij}} \text{sgn}\left[\frac{|p_{uv} - p_{ij}| - (p_{uv} - p_{ij})}{2}\right]$, $|\cdot|$ is a function returning the absolute value of its argument, sgn$[\cdot]$ is a sign function, σ_{ij} is a positive scaling factor ($\sigma_{ij} = 8$ in the paper), $\mathbb{N}_{ij}$ denotes the neighbor elements of p_{ij}, and $k_{i,j}$ stands for the number of the neighbor elements. In this paper, if all the 8-neighbor elements of p_{ij} exist, $k_{i,j} = 8$; otherwise, $k_{i,j}$ is the number of the valid 8-neighbor elements of p_{ij}. A brief discussion on the theoretical properties of Eq. (5) is given as follows. The second term of Eq. (5)(i.e., exp$(\cdot)$) is a local spatial weighting factor. If N^+_{ij} is smaller than N^-_{ij}, the factor will penalize p_{ij}; otherwise it will encourage p_{ij}. The process of local spatial weighting is illustrated in Fig. 4.

(4) Global spatial weighting. In this step, the filtered likelihood map $\mathcal{M}_l = (p^l_{ij})_{m^* \times n^*}$ is further globally weighted by a spatial Gaussian kernel into a new one $\mathcal{M}_g = (p^g_{ij}) \in R^{m^* \times n^*}$. The global spatial weighting process is formulated as follows.

$$\begin{aligned} p^g_{ij} &\propto p^l_{ij} \cdot \exp\left(-\|pos_{ij} - pos_o\|^2 / 2\sigma^2_{p_{ij}}\right) \\ &\propto p_{ij} \cdot \exp\left(-\|pos_{ij} - pos_o\|^2 / 2\sigma^2_{p_{ij}}\right) \cdot \exp\left(\frac{N^+_{ij} - N^-_{ij}}{\sigma_{ij}}\right) \end{aligned} \tag{6}$$

where pos_{ij} is the block-(i, j) positional coordinate vector, pos_o is the positional coordinate vector associated with the center O of the likelihood map $\mathcal{M}_l$, and $\sigma_{p_{ij}}$ is a scaling factor ($\sigma_{p_{ij}} = 3.9$ in the paper). The process of global spatial weighting can be illustrated by Fig. 5, where the likelihood map $\mathcal{M}_l$ (shown in Fig. 5(a)) is spatially weighted by the Gaussian kernel (shown in Fig. 5(b)).

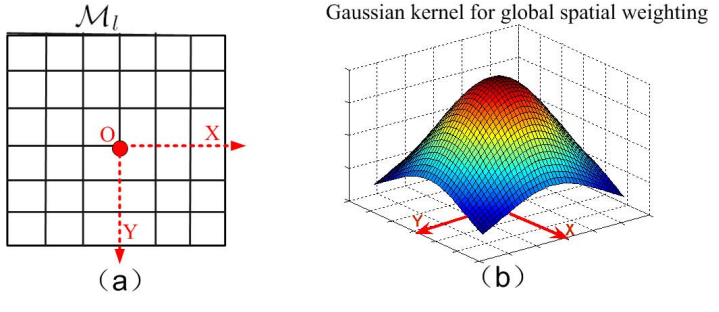

Fig. 5. Illustration of global spatial weighting. (a) shows the original likelihood map $\mathcal{M}_l$ while (b) exhibits the spatial weighting kernel for $\mathcal{M}_l$.

(5) Likelihood evaluation for *SLAM*. In this step, the overall likelihood between a candidate object region and the learned *SLAM* is computed by multiplying all the block-specific likelihoods after local and global spatial weighting. Mathematically, the likelihood is formulated as:

$$\begin{aligned}\mathbb{LIKI} &\propto \prod_{1\le i\le m^*}\prod_{1\le j\le n^*} p_{ij}^{g} \\ &\propto \prod_i\prod_j p_{ij}\cdot \exp\left(-\|pos_{ij}-pos_o\|^2/2\sigma_{p_{ij}}^2\right)\cdot \exp\left(\frac{N_{ij}^{+}-N_{ij}^{-}}{\sigma_{ij}}\right)\end{aligned} \tag{7}$$

2.3 Bayesian State Inference for Visual Tracking

For visual tracking, a Markov model with a hidden state variable is used for motion estimation. In this model, the object motion between two consecutive frames is usually assumed to be an affine motion. Let X_t denote the state variable describing the affine motion parameters (the location) of an object at time t. Given a set of observed images $\mathcal{O}_t = \{O_1, \ldots, O_t\}$, the posterior probability is formulated by Bayes' theorem as:

$$p(X_t|\mathcal{O}_t) \propto p(O_t|X_t)\int p(X_t|X_{t-1})p(X_{t-1}|\mathcal{O}_{t-1})dX_{t-1} \tag{8}$$

where $p(O_t \,|\, X_t)$ denotes the observation model, and $p(X_t \,|\, X_{t-1})$ represents the dynamic model. $p(O_t|X_t)$ and $p(X_t|X_{t-1})$ decide the entire tracking process. A particle filter [3] is used for approximating the distribution over the location of the object using a set of weighted samples.

In the tracking framework, we apply an affine image warping to model the object motion of two consecutive frames. The six parameters of the affine transform are used to model $p(X_t \mid X_{t-1})$ of a tracked object. Let $X_t = (x_t, y_t, \eta_t, s_t, \beta_t, \phi_t)$ where $x_t, y_t, \eta_t, s_t, \beta_t, \phi_t$ denote the x, y translations, the rotation angle, the scale, the aspect ratio, and the skew direction at time t, respectively. We employ a Gaussian distribution to model the state transition distribution $p(X_t \,|\, X_{t-1})$. Also the six parameters of the affine transform are assumed to be independent. Consequently, $p(X_t \,|\, X_{t-1})$ is formulated as:

$$p(X_t|X_{t-1}) = \mathcal{N}(X_t; X_{t-1}, \Sigma) \tag{9}$$

where Σ denotes a diagonal covariance matrix whose diagonal elements are σ_x^2, σ_y^2, σ_η^2, σ_s^2, σ_β^2, σ_ϕ^2, respectively. The observation model $p(O_t \,|\, X_t)$ reflects the similarity between a candidate sample and the learned *SLAM*. In this paper, $p(O_t|X_t)$ is formulated as: $p(O_t \,|\, X_t) \propto \mathbb{LIKI}$, where $\mathbb{LIKI}$ is defined in Eq. (7). After maximum a posterior (MAP) estimation, we just use the block related Log-Euclidean covariance matrices of features inside the affinely warped image region associated with the highest weighted hypothesis to update the block related Log-Euclidean eigenspace model.

3 Experiments

In order to evaluate the performance of the proposed tracking framework, four videos are used in the experiments. The first three videos are recorded with moving cameras

while the last video is taken from a stationary camera. The first two videos consist of 8-bit gray scale images while the last two are composed of 24-bit color images. Video 1 consists of dark gray scale images, where a man moves in an outdoor scene with drastically varying lighting conditions. In Video 2, a man walks from left to right in a bright road scene; his body pose varies over the time, with a drastic motion and pose change (bowing down to reach the ground and standing up back again) in the middle of the video stream. In Video 3, a girl changes her facial pose over the time in a color scene with varying lighting conditions. Besides, the girl's face is severely occluded by a man in the middle of the video stream. In the last video, a pedestrian moves along a corridor in a color scene. In the middle of the video stream, his body is severely occluded by the bodies of two other pedestrians.

During the visual tracking, the size of each object region is normalized to 36×36 pixels. Then, the normalized object region is uniformly divided into thirty-six 6×6 blocks. Further, a block-specific *SLAM* is online learned and online updated by *IRSL* every three frames. The maintained dimension r_{ij} of the block-(i, j) Log-Euclidean eigenspace model (i.e., U_{ij} referred in Sec. 2.2) learned by *IRSL* is obtained from the experiments. For the particle filtering in the visual tracking, the number of particles is set to be 200. The six diagonal elements $(\sigma_x^2, \sigma_y^2, \sigma_\eta^2, \sigma_s^2, \sigma_\beta^2, \sigma_\phi^2)$ of the covariance matrix Σ in Eq. (9) are assigned as $(5^2, 5^2, 0.03^2, 0.03^2, 0.005^2, 0.001^2)$, respectively.

Three experiments are conducted to demonstrate the claimed contributions of the proposed *SLAM*. In these four experiments, we compare tracking results of *SLAM* with those of a state-of-the-art Riemannian metric based tracking algorithm [21], referred here as *CTMU*, in different scenarios including drastic illumination changes, object pose variation, and occlusion. *CTMU* is a representative Riemannian metric based tracking algorithm which uses the covariance matrix of features for object representation. By using a model updating mechanism, *CTMU* adapts to the undergoing object deformations and appearance changes, resulting in a robust tracking result. In contrast to *CTMU*, *SLAM* constructs a block-based Log-Euclidean eigenspace representation to reflect the appearance changes of an object. Consequently, it is interesting and desirable to make a comparison between *SLAM* and *CTMU*. Furthermore, *CTMU* does not need additional parameter settings since *CTMU* computes the covariance matrix of image features as the object model. More details of *CTMU* are given in [21].

The first experiment is to compare the performances of the two methods *SLAM* and *CTMU* in handling drastic illumination changes using Video 1. In this experiment, the maintained eigenspace dimension r_{ij} in *SLAM* is set as 8. Some samples of the final tracking results are demonstrated in Fig. 6, where rows 1 and 2 are for *SLAM* and *CTMU*, respectively, in which five representative frames (140, 150, 158, 174, and 192) of the video stream are shown. From Fig. 6, we see that *SLAM* is capable of tracking the object all the time even in a poor lighting condition. In comparison, *CTMU* is lost in tracking from time to time.

The second experiment is for a comparison between *SLAM* and *CTMU* in the scenarios of drastic pose variation using Video 2. In this experiment, r_{ij} in *SLAM* is set as 6. Some samples of the final tracking results are demonstrated in Fig. 7, where rows 1 and 2 correspond to *SLAM* and *CTMU*, respectively, in which five representative frames (142, 170, 178, 183, and 188) of the video stream are shown. From Fig. 7, it is clear

Fig. 6. The tracking results of *SLAM* (row 1) and *CTMU* (row 2) over representative frames with drastic illumination changes

Fig. 7. The tracking results of *SLAM* (row 1) and *CTMU* (row 2) over representative frames with drastic pose variation

that *SLAM* is capable of tracking the target successfully even with a drastic pose and motion change while *CTMU* gets lost in tracking the target after this drastic pose and motion change.

The last experiment is to compare the tracking performance of *SLAM* with that of *CTMU* in the color scenarios with severe occlusions using Videos 3 and 4. The RGB color space is used in this experiment. r_{ij} for Videos 3 and 4 are set as 6 and 8, respectively. We show some samples of the final tracking results for *SLAM* and *CTMU* in Fig. 8, where the first and the second rows correspond to the performances of *SLAM* and *CTMU* over Video 3, respectively, in which five representative frames (158, 160, 162, 168, and 189) of the video stream are shown, while the third and the last rows correspond to the performances of *SLAM* and *CTMU* over Video 4, respectively, in which five representative frames (22, 26, 28, 32, and 35) of the video stream are shown. Clearly, *SLAM* succeeds in tracking for both Video 3 and Video 4 while *CTMU* fails.

In summary, we observe that *SLAM* outperforms *CTMU* in the scenarios of illumination changes, pose variations, and occlusions. *SLAM* constructs a block-based Log-Euclidean eigenspace representation to capture both the global and local spatial properties of object appearance. The spatial correlation information of object appearance is incorporated into *SLAM*. Even if the information of some local blocks is partially lost or drastically varies, *SLAM* is capable of recovering the information using the cues of the information from other local blocks. In comparison, *CTMU* only captures the statistical properties of object appearance in one mode, resulting in the loss of the local

Fig. 8. The tracking results of *SLAM* and *CTMU* over representative frames in the color scenarios of severe occlusions. Rows 1 and 2 show the tracking results of *SLAM* and *CTMU* for Video 4, respectively. Rows 3 and 4 display the tracking results of *SLAM* and *CTMU* for Video 5, respectively.

spatial correlation information inside the object region. In particular, *SLAM* constructs a robust Log-Euclidean Riemannian eigenspace representation of each object appearance block. The representation fully explores the distribution information of covariance matrices of image features under the Log-Euclidean Riemannian metric, whereas *CTMU* relies heavily on an intrinsic mean in the Lie group structure without considering the distribution information of the covariance matrices of image features. Consequently, *SLAM* is an effective appearance model which performs well in modeling appearance changes of an object in many complex scenarios.

4 Conclusion

In this paper, we have developed a visual tracking framework based on the proposed spatial Log-Euclidean appearance model (*SLAM*). In this framework, a block-based Log-Euclidean eigenspace representation is constructed by *SLAM* to reflect the appearance changes of an object. Then, the local and global spatial weighting operations on the block-based likelihood map are performed by *SLAM* to capture the local and global spatial layout information of object appearance. Moreover, a novel criterion for the likelihood evaluation, based on the Log-Euclidean Riemannian subspace reconstruction error norms, has been proposed to measure the similarity between the test image and the learned subspace model during the tracking. *SLAM* is incrementally updated by the proposed online Log-Euclidean Riemannian subspace learning algorithm (*IRSL*). Experimental results have demonstrated the robustness and promise of the proposed framework.

Acknowledgment

This work is partly supported by NSFC (Grant No. 60520120099, 60672040 and 60705003) and the National 863 High-Tech R&D Program of China (Grant No. 2006AA01Z453). Z.Z. is supported in part by NSF (IIS-0535162). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

1. Hager, G., Belhumeur, P.: Real-time tracking of image regions with changes in geometry and illumination. In: Proc. CVPR, pp. 410–430 (1996)
2. Black, M.J., Jepson, A.D.: Eigentracking: Robust matching and tracking of articulated objects using view-based representation. In: Buxton, B.F., Cipolla, R. (eds.) ECCV 1996. LNCS, vol. 1064, pp. 329–342. Springer, Heidelberg (1996)
3. Isard, M., Blake, A.: Contour tracking by stochastic propagation of conditional density. In: Buxton, B.F., Cipolla, R. (eds.) ECCV 1996. LNCS, vol. 1065, pp. 343–356. Springer, Heidelberg (1996)
4. Black, M.J., Fleet, D.J., Yacoob, Y.: A framework for modeling appearance change in image sequence. In: Proc. ICCV, pp. 660–667 (1998)
5. Jepson, A.D., Fleet, D.J., El-Maraghi, T.F.: Robust Online Appearance Models for Visual Tracking. In: Proc. CVPR, vol. 1, pp. 415–422 (2001)
6. Zhou, S.K., Chellappa, R., Moghaddam, B.: Visual Tracking and Recognition Using Appearance-Adaptive Models in Particle Filters. IEEE Trans. on Image Processing 13, 1491–1506 (2004)
7. Lee, K., Kriegman, D.: Online Learning of Probabilistic Appearance Manifolds for Video-based Recognition and Tracking. In: Proc. CVPR, vol. 1, pp. 852–859 (2005)
8. Lim, H., Morariu3, V.I., Camps, O.I., Sznaier1, M.: Dynamic Appearance Modeling for Human Tracking. In: Proc. CVPR, vol. 1, pp. 751–757 (2006)
9. Ho, J., Lee, K., Yang, M., Kriegman, D.: Visual Tracking Using Learned Linear Subspaces. In: Proc. CVPR, vol. 1, pp. 782–789 (2004)
10. Li, Y., Xu, L., Morphett, J., Jacobs, R.: On Incremental and Robust Subspace Learning. Pattern Recognition 37(7), 1509–1518 (2004)
11. Skocaj, D., Leonardis, A.: Weighted and Robust Incremental Method for Subspace Learning. In: Proc. ICCV, pp. 1494–1501 (2003)
12. Limy, J., Ross, D., Lin, R., Yang, M.: Incremental Learning for Visual Tracking. In: NIPS, pp. 793–800. MIT Press, Cambridge (2005)
13. Li, X., Hu, W., Zhang, Z., Zhang, X., Luo, G.: Robust Visual Tracking Based on Incremental Tensor Subspace Learning. In: Proc. ICCV (2007)
14. Li, X., Hu, W., Zhang, Z., Zhang, X., Luo, G.: Visual Tracking Via Incremental Log-Euclidean Riemannian Subspace Learning. In: Proc. CVPR (2008)
15. Yilmaz, A.: Object Tracking by Asymmetric Kernel Mean Shift with Automatic Scale and Orientation Selection. In: Proc. CVPR (2007)
16. Silveira, G., Malis, E.: Real-time Visual Tracking under Arbitrary Illumination Changes. In: Proc. CVPR (2007)
17. Nguyen, Q.A., Robles-Kelly, A., Shen, C.: Kernel-based Tracking from a Probabilistic Viewpoint. In: Proc. CVPR (2007)
18. Zhao, Q., Brennan, S., Tao, H.: Differential EMD Tracking. In: Proc. ICCV (2007)

19. Chen, D., Yang, J.: Robust Object Tracking Via Online Dynamic Spatial Bias Appearance Models. IEEE Trans. on PAMI 29(12), 2157–2169 (2007)
20. Wang, H., Suter, D., Schindler, K., Shen, C.: Adaptive Object Tracking Based on an Effective Appearance Filter. IEEE Trans. on PAMI 29(9), 1661–1667 (2007)
21. Porikli, F., Tuzel, O., Meer, P.: Covariance Tracking using Model Update Based on Lie Algebra. In: Proc. CVPR, vol. 1, pp. 728–735 (2006)
22. Tuzel, O., Porikli, F., Meer, P.: Human Detection via Classification on Riemannian Manifolds. In: Proc. CVPR (2007)
23. Arsigny, V., Fillard, P., Pennec, X., Ayache, N.: Geometric Means in a Novel Vector Space Structure on Symmetric Positive-Definite Matrices. SIAM Journal on Matrix Analysis and Applications (2006)
24. Tuzel, O., Porikli, F., Meer, P.: Region Covariance: A Fast Descriptor for Detection and Classification. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 589–600. Springer, Heidelberg (2006)
25. Pennec, X., Fillard, P., Ayache, N.: A Riemannian Framework for Tensor Computing. In: IJCV, pp. 41–66 (2006)
26. Rossmann, W.: Lie Groups: An Introduction Through Linear Group. Oxford Press (2002)
27. Levy, A., Lindenbaum, M.: Sequential Karhunen-Loeve Basis Extraction and Its Application to Images. IEEE Trans. on Image Processing 9, 1371–1374 (2000)