International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:5, No:7, 2011

A Robust Visual Tracking Algorithm with Low-Rank Region Covariance

Songtao Wu, Yuesheng Zhu, and Ziqiang Sun

***Abstract*—**Region covariance (RC) descriptor is an effective and efficient feature for visual tracking. Current RC-based tracking algorithms use the whole RC matrix to track the target in video directly. However, there exist some issues for these whole RC-based algorithms. If some features are contaminated, the whole RC will become unreliable, which results in lost object-tracking. In addition, if some features are very discriminative to the background, other features are still processed and thus reduce the efficiency. In this paper a new robust tracking method is proposed, in which the whole RC matrix is decomposed into several low rank matrices. Those matrices are dynamically chosen and processed so as to achieve a good tradeoff between discriminability and complexity. Experimental results have shown that our method is more robust to complex environment changes, especially either when occlusion happens or when the background is similar to the target compared to other RC-based methods.

***Keywords*—**Visual tracking, region covariance descriptor, low-rank region covariance

I. Introduction

VISUAL Tracking has found wide use in human behavior recognition, human-computer interactions, and security & monitoring. However, tracking a moving target precisely in a complex environment is still a challenging issue in practical tracking systems. Consequently, extracting the object features and choosing some of them as an appropriate reference (reference features) for dynamic tracking becomes critical.Oncel Tuzel etc propose a region covariance (RC) descriptor for object detection and classification [1]. A covariance matrix is generated to represent the target by combining several low level image features, such as colors, intensities, gradients and coordinates. The RC descriptor features low dimensionality and good disciminability. Using the elements of Riemannian geometry, Fatih Porikli develops an effective model update algorithm [2] based on the RC descriptor for visual tracking. However, the algorithm is search-based and turns out to be time consuming. In an attempt to avoid that issue, Yi Wu proposes a method [3] based on particle filtering, which only computes the region that the particles occupy to reduce the computational burden and improve the tracking efficiency. Using R-SVD subspace learning algorithm [4], Xi Li etc construct a low order Log-Euclidean eigenspace [5] to represent the target with Log-Euclidean Riemannian metric [6] to enhance the robustness. Yi Wu also proposes an efficient low-dimensional covariance tensor learning method in [7]. However, these RC-based tracking algorithms build on the whole RC matrix to track the given target in video directly. In doing so, two defects arise in practical systems. First, if some features are contaminated, the whole covariance matrix becomes unreliable and leads to tracking failure. Moreover, when some features are very discriminative to the background, no sensitive features can be processed, which results in unnecessary computations. To overcome these problems, a robust tracking method is proposed in the framework of Sequential Monte Carlo (SMC). The basic idea behind is to decompose the whole covariance matrix into several low-rank region covariance (LRC) matrices, only a subset of which are dynamically chosen and processed so that a good tradeoff between discriminability and complexity is achieved.The rest of the paper is organized as follows: The traditional region covariance descriptor and the proposed low-rank region covariance are introduced in section II. In section III, the proposed algorithm based on Sequential Monte Carlo and LRC features are described in detail. Section IV presents the experimental results, where the proposed method and Fatih Porikli's algorithm are compared by handling two important cases which frequently arise in visual tracking. Finally, conclusions are made in section V.

II. Low-Rank Region Covariance Features

Let I be the observed intensity or color image in a typical RC-based tracking algorithm. Its feature image F is a $W \times H \times d$ dimensional matrix extracted from I:

$$F(x,y) = \phi(I,x,y) \tag{1}$$

where ϕ is a mapping operator which can be related to some low level features, such as color, intensity, gradients and coordinates, etc.

For d dimensional feature points inside $R, \{z_k\}_{k=1..N}$, where R is a rectangular region contained in F, the covariance feature C_R of region R is given by

Wu Songtao, Zhu Yuesheng and Sun Ziqiang are with Communication & Information Security Lab, Shenzhen Graduate School, Peking University, Shenzhen, CHINA.
Corresponding author e-mail add: zhuys@szpku.edu.cn

$$C_R = \frac{1}{N-1}\sum_{k=1}^{N}(z_k-\mu)(z_k-\mu)^T \qquad (2)$$

where μ is the mean feature of R , C_R is a $d \times d$ matrix and can reveal the correlation of any two low level features with low dimensionality.

In d^2 dimension Euclidean space all $d \times d$ symmetric positive definite matrices form a smooth manifold. Rather than the Euclidean distance, in this paper the distance [8]
between two covariance matrices is used:

$$dist(C_1, C_2) = tr(\log^2(C_1^{-1}C_2)) \qquad (3)$$

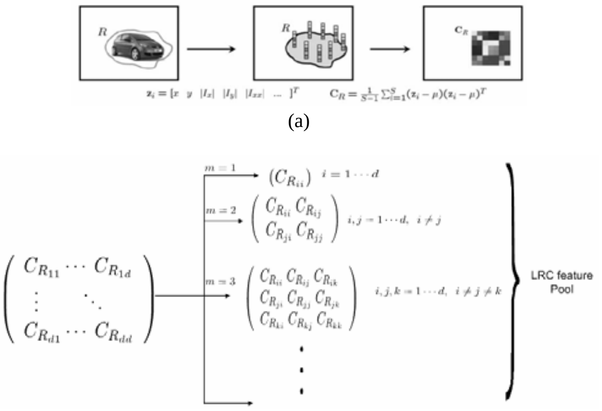
where $tr(.)$ represents the trace of a matrix, $\log(.)$is matrix logarithm. For $C_1, C_2, ..., C_n(n>2)$, the mean of them is computed by the following formula:

$$\bar{C} = argmin_{(C)}\{\sum_{i=1}^{n}\omega_i dist(C, C_i)\} \qquad (4)$$

where ω_i is the weight of C_i, and satisfies $\sum_{i=1}^{n}\omega_i = 1$ ($\omega_i \geqslant 0$). Since the analytic solution of $\bar{C}$ doesn't exist, its approximated solution can be obtained by the iterative method proposed in [9]:

$$\bar{C}_{t+1} = \exp_{\bar{C}_t}\left(\sum_{i=1}^{n}\omega_i \log_{\bar{C}_t}(C_i)\right)$$
$$= \bar{C}_t^{1/2}\exp\left(\sum_{i=1}^{n}\omega_i \log(\bar{C}_t^{-1/2}C_i\bar{C}_t^{-1/2})\right)\bar{C}_t^{1/2} \qquad (5)$$

given an initial estimate $\bar{C}_0$, the iteration is carried out until convergence is achieved.


R
R
C_R
$z_i = [x \;\; y \;\; |I_x| \;\; |I_y| \;\; |I_{xx}| \;\; ... \;]^T$
$C_R = \frac{1}{S-1}\sum_{i=1}^{S}(z_i-\mu)(z_i-\mu)^T$
$m=1$
$(C_{R_{ii}})$ $i = 1\cdots d$
$m=2$
$i,j = 1\cdots d, \; i \neq j$
$m=3$
$i,j,k = 1\cdots d, \; i \neq j \neq k$
LRC feature Pool

(a)

(b)

Fig.1 Comparison of RC feature and proposed LRC features: (a) thetraditional RC feature for a given image; (b) the proposed LRC features is decomposed from original RC matrix C_R

For a $W \times H \times d$ dimensional feature image F , traditional RC-based tracking algorithms calculate the whole $d \times d$ RC matrix and use it to track the target directly. Instead, in our method, m features are selected to generate a $m \times m$ matrix with m ranging from 1 to d. As a result, there are totally $2^d - 1$ cases. In other words, the whole RC matrix is decomposed into $2^d - 1$ RC matrices. Since m is not greater than d, the obtained $2^d - 1$ matrices have lower ranks than the original $d \times d$ RC matrix, which constitute a low-rank RC (LRC) pool for a single image block. This process is depicted by Fig.1. There are at least three advantages of RC matrix decomposition. First, when some features are discriminative to the background, some LRCs can be chosen as reference features which are more accurate than using the entire RC matrix so that computing resources can be saved. It should be noted that although the decomposition might result in an increase in complexity, the increase is moderate, if not negligible, due to the fact that all LRCs share the same elements with original RC matrix. This is justified by its tracking accuracy in later experiments. Moreover, if the most discriminative LRCs are selected as reference features, the object can be tracked more precisely when the background has features similar to the target. Last but not least, if some features are contaminated by noise or disturbance, such as illumination changes and occlusion, the traditional RC-based tracking methods are influenced inevitably, while in the proposed method, LRCs are chosen dynamically to be least affected by environment changes so that the tracking algorithm can be more robust and stable.

III. Visual Tracking With Low-Rank Region Covariance

In this section, a new tracking method based on the SMC framework is proposed, as shown in Fig.2. The method mainly involves object RC extraction, template update, background RC model estimation, LRC features selection, state description and dynamical model, weights computation. These procedures will be elaborated on in what follows.

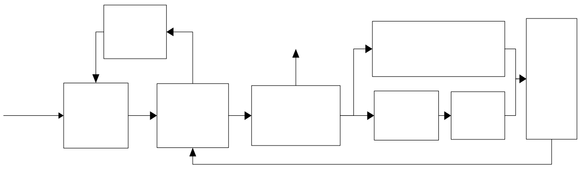


Fig.2 Block Diagram of the Proposed Method

A. Sequential Monte Carlo Based Tracking

Object tracking can be formulated as a state estimation problem. Assume X_t is the state of the target at time t, and $\{Y_t\}_{t=1...T}$ are the observations of $\{X_t\}_{t=1...T}$, then the posterior probability distribution of X_t is:

$$p(X_t|Y_{1...t}) \propto p(Y_t|X_t)\int p(X_t|X_{t-1})p(X_{t-1}|Y_{1...t-1})dX_{t-1} \quad (6)$$

where $p(X_t|X_{t-1})$ is the dynamical model, $p(Y_t|X_t)$is the observation model. Due to the intractability to solve (6) analytically, especially when either the noise is non-Gaussian or the dynamical equation and observation model are nonlinear, we adopt the Sequential Monte Carlo (SMC) method to approximate the posterior distribution of X_t in this paper. The SMC method takes advantage of many weighted particles $\{X_t^i, \omega_t^i\}_{i=1...N}$ to represent the $p(X_t|Y_{1...t})$; that is:

$$p(X_t|Y_{1\ldots t}) \approx \sum_{i=1}^{N} \omega_t^i \delta(X_t - X_t^i) \quad (7)$$

when a new observation Y_{t+1} comes, the approximated posterior distribution of X_{t+1} is computed as:

$$p(X_{t+1}|Y_{1\ldots t+1}) \approx \sum_{i=1}^{N} \omega_{t+1}^i \delta(X_{t+1} - X_{t+1}^i) \quad (8)$$

where ω_{t+1}^i is estimated from ω_t^i:

$$\omega_{t+1}^i \propto \frac{p(Y_{t+1}|X_{t+1}^i)p(X_{t+1}^i|X_t^i)}{q(X_{t+1}^i|X_t^i, Y_{t+1})}\omega_t^i \quad (9)$$

$q(X_{t+1}^i|X_t^i, Y_{t+1})$ is the proposal sampling density for generating new particles. If $q(X_{t+1}^i|X_t^i, Y_{t+1})$ is chosen as $p(X_{t+1}|X_t)$, then the update rule of ω_{t+1}^i is simplified:

$$\omega_{t+1}^i = p(Y_{t+1}^i|X_{t+1}^i)\omega_t^i \quad (10)$$

The dynamical model and observation model in this paper will be given in section *E*.

A disadvantage of SMC is the particle degeneration in many applications. In order to overcome this drawback, a bootstrap resampling process is invoked to make the algorithm work properly [10].

B. Template Update

An appropriate template for target is a key for object tracking. A modified Fatih Porikli's method [2] for template update is developed here by choosing a sliding window model and using the approximated mean of the previous K (K is the size of window) RC matrices as a template as given by (11):

$$\bar{C}^{i+1} = \exp_{\bar{C}^i}\left(\frac{1}{dist^*}\sum_{t=T-K+1}^{T} dist^{-1}(C_t, \bar{C}_t^*)\log_{\bar{C}^i}(C_t)\right) \quad (11)$$

where $\bar{C}_t^*$is the mean RC matrix at t, $dist$ is defined in (3), and $dist^* = \sum_{t=T-K+1}^{T} dist^{-1}(C_t, \bar{C}_t^*)$. The iteration of (11) will continue until $\bar{C}^i$ converges, and the converged RC matrix $\bar{C}_{t+1}^*$ is viewed as the new template of the target at $t+1$. After the template been updated, the LRC features, $\{LRC_{temp,t+1}^i\}_{i=1..2^d-1}$, for the template are obtained by the RC matrix decomposition in section II.

C. Region Covariance Model Estimation of Background

For an estimated state of the target, X_{t+1}^*, n image blocks around it are selected and their RC matrices are derived in our algorithm. The RC model of background is determined by their mean with the same weight ($\omega_i = \frac{1}{n}$) in (5). This process is illustrated in Fig. 3, where each BRCM represents the RC matrix of background in corresponding image block. The LRCs of background, $\{LRC_{bg,t+1}^i\}_{i=1..2^d-1}$, can be obtained by decomposing the RC model of background.

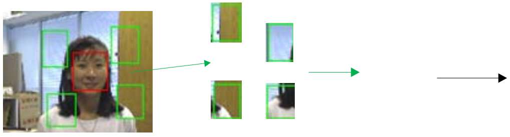


Fig.3 RC model of background extraction process

D. Low-Rank Region Covariance Features Selection

In this section, we will choose a subset LRCs in $\{LRC_{temp,t+1}^i\}_{i=1..2^d-1}$as reference features for next frame tracking. Assume an estimated target state $X_{t+1}^*$is obtained at $t+1$, its RC matrix is computed by (2), and decomposed into LRC features $\{LRC_{t+1}^i\}_{i=1..2^d-1}$. Some of the template LRCs which can discriminate the target from the background obviously will be chosen as reference features. In order to measure the discriminability, a discriminative gap (DG) is defined here:

$$DG_{t+1}^i = \frac{dist(LRC_{bg,t+1}^i, LRC_{t+1}^i)}{dist(LRC_{temp,t+1}^i, LRC_{t+1}^i)} \quad (12)$$

among the LRC features, those with large DG values indicate that they are either close to $LRC_{temp,t+1}$ or far from $LRC_{bg,t+1}$. In order to balance discriminability against complexity, a tradeoff factor (TF) is defined to measure the "importance" of each LRC for reference feature selection:

$$TF_{t+1}^i = \frac{DG_{t+1}^i}{CLRC_{t+1}^i} \quad (13)$$

where $CLRC_{t+1}^i$ represents the complexity of $LRC_{t+1}^i$and it is determined by multiplications involved in calculating the corresponding LRC. It is easy to show that $CLRC_{t+1}^i$ is determined by the dimension of LRC_{t+1}^i. Based on the measurement (13), all LRCs of the template are sorted in a descending order; that is, the original LRC features $\{LRC_{t+1}^i\}_{i=1..2^d-1}$ are rearranged into $\{LRC_{sorted,t+1}^i\}_{i=1...2^d-1}$ according to their TF values. Later, the first M_{t+1} of the LRCs are selected, with M_{t+1} determined by:

$$M_{t+1} = argmax_{(m)}\{\sum_{i=1}^{m} CLRC_{sorted,t+1}^i < CT\} \quad (14)$$

where CT is a complexity threshold and it is used to confine the total number of the chosen LRC features. $CLRC_{sorted,t+1}^i$ is the complexity of $LRC_{sorted,t+1}^i$. Therefore a subset of LRC features,$\{LRC_{temps,t+1}^i\}_{i=1..M_{t+1}}$, is selected.

E. Dynamical and Observation Model

The state at time t, X_t, consists of four parameters, $(x_t, y_t, s_{w_t}, s_{H_t})$, which denotes horizontal location, vertical location, width scale and height scale, respectively. The posterior distribution of X_t at time t is estimated based on SMC. The dynamical model is chosen as a simple Brownian motion:

$$p(X_{t+1}|X_t) = N(X_{t+1}; X_t, \phi) \quad (15)$$

where ϕ is a diagonal matrix, $\phi = diag(\sigma_{x_t}, \sigma_{y_t}, \sigma_{s_{W_t}}, \sigma_{s_{H_t}})$.

The observation $p(Y_{t+1}^i|X_{t+1}^i)$ is calculated by (16) with selected reference features$\{LRC_{temps,t+1}^j\}_{j=1..M_{t+1}}$,

$$p(Y_{t+1}^i|X_{t+1}^i) = \frac{1}{\sqrt[M_{t+1}]{\prod_{j=1}^{M_{t+1}}\left(dist(LRC_{temps,t+1}^j, LRC_{t+1}^{i,j}) + \epsilon\right)}} \quad (16)$$

where $LRC_{t+1}^{i,j}$ is the LRC feature of i-th particle that corresponds to$LRC_{temps,t+1}^j$, ϵ is a given small value to ensure a non-zero denominator in (16). Based on maximum a posterior (MAP) estimation, the particle with largest weight is taken as the tracking result at current frame.

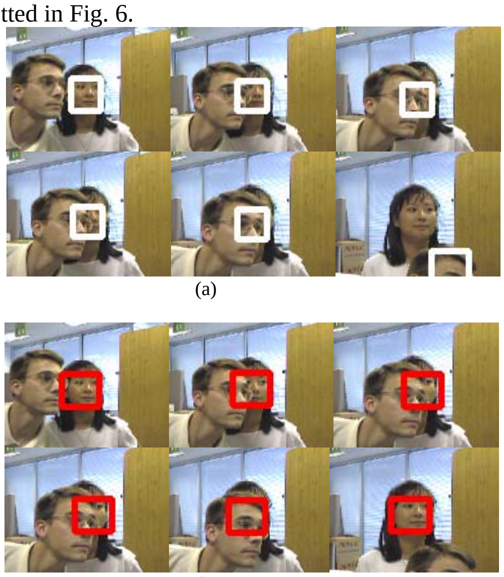
Finally, the proposed algorithm to process individual video frames is summarized below:

Algorithm:

```
For the $(t+1)$-th video frame:

Input: weighted particles $\{\omega_t^i, X_t^i\}_{i=1...N}$ ; target template $\bar{C}_t^*$; selected LRC $\{LRC^i_{temps,t}\}_{i=1..M_t}$.

propagate $\{\omega_t^i, X_t^i\}_{i=1...N}$ according to (15), then the
  new particles$\{\omega_t^i, X_{t+1}^i\}_{i=1...N}$ are obtained;
For i=1,…,N
    extract the RC descriptor of $X_{t+1}^i$ and decompose it
    into LRC features $\{LRC_t^{i,j}\}_{j=1...2^d-1}$;
    calculate the particle's weight by (16);
end
compute the new weight $\omega_{t+1}^i$ of each $X_{t+1}^i$ by (10) and
  normalize them: $\hat{\omega}_{t+1}^i = \frac{\omega_{t+1}^i}{\sum_{i=1}^N \omega_{t+1}^i}$;
estimate the target's state $X_{t+1}^*$by MAP, then compute
  its corresponding RC feature by (2);
update $\bar{C}_t^*$ to $\bar{C}_{t+1}^*$ according to (11), then decompose
  it into LRC features $\{LRC^i_{temp,t+1}\}_{i=1..2^d-1}$;
randomly choose $n$ image patches around $X_{t+1}^*$, and
  calculate $\{LRC^i_{bg,t+1}\}_{i=1..2^d-1}$;
compute $TF_{t+1}^i$ value of each $LRC^i_{temp,t+1}$ via (13)
  and sort them by $TF_{t+1}^i$, then $\{LRC^i_{temp,t+1}\}_{i=1..2^d-1}$
  are changed into $\{LRC^i_{temps,t+1}\}_{i=1..2^d-1}$, choose
  first $M_{t+1}$ of $LRC^i_{temps,t+1}$ by (14);
if $\frac{1}{\sum_{i=1}^N (\hat{\omega}_{t+1}^i)^2} < N_T$, where $N_T$ is a predefined
  threshold for resampling, then resample the particles;

Output: weighted particles $\{\omega_{t+1}^i, X_{t+1}^i\}_{i=1...N}$ ; updated template$\bar{C}_{t+1}^*$; selected LRC $\{LRC^i_{temps,t+1}\}_{i=1..M_{t+1}}$
```

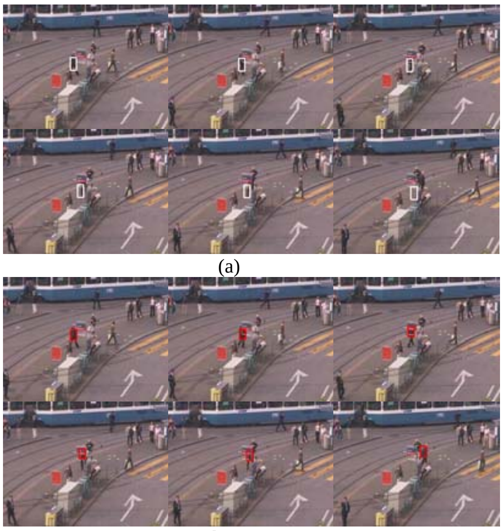
IV. Experimental Results

In this section, experiments are set up to evaluate the performance of the proposed algorithm compared to Fatih Porikli's algorithm. Four image sequences with more than 1500 frames in total are tested. For a color image, $[R, G, B, |I_x|, |I_y|]$ are chosen as low level features, where $|I_x|$ and $|I_y|$ are the intensity derivatives corresponding to x and y, respectively. Since$d = 5$, the total number of LRC features is $2^5 - 1 = 31$. The number of particles involved in tracking process is set to 300, and six image blocks ($n = 6$) around the current target are randomly chosen and used to calculate the RC model of background. The resampling threshold N_T is set as 0.6. The diagonal elements of ϕ depends on the video source, for fast moving objects they are $(12^2, 12^2, 0.01^2, 0.01^2)$ and $(3^2, 3^2, 0.01^2, 0.01^2)$ for slow moving target. Fig.4 and Fig.5 compare Fatih Porikli's method (white rectangle) with the proposed method (red rectangle) in terms of tracking robustness when occlusion happens. It can be observed that while Fatih Porikli's method fails, our method shows no loss of tracking, having more robust performance. The corresponding tracking errors are plotted in Fig. 6.


(a)

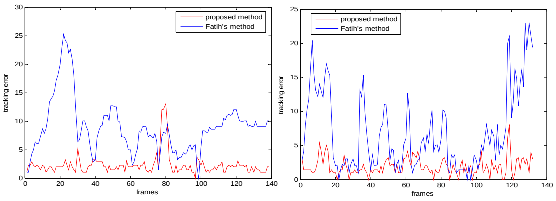
(b)

Fig 4. Girl sequence with occluded face


(a)

(b)

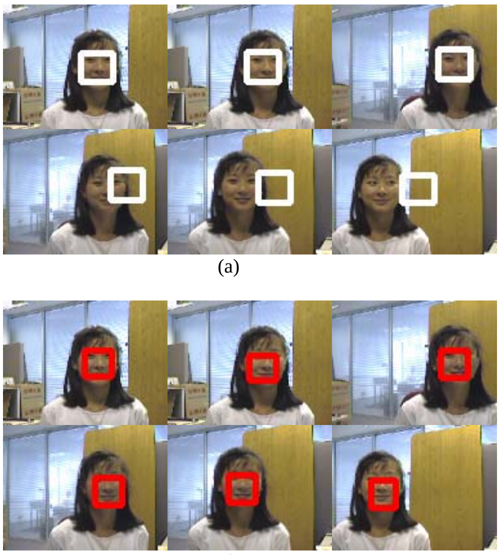
Fig.5 Pedestrian occluded by guidepost



(a) (b)

Fig.6. Error plots for girl sequence (a) and pedestrian sequence (b)

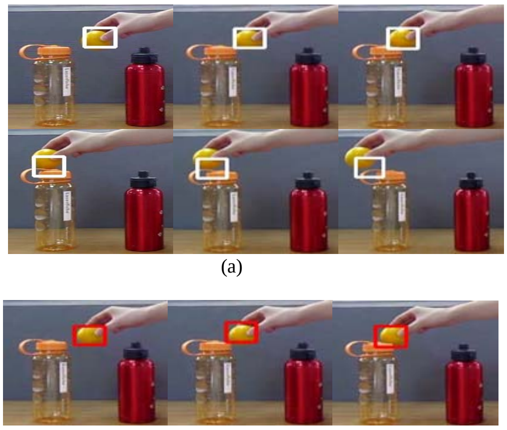
Fig.7 and Fig.8 show the comparison results between Fatih Porikli's method (white rectangle) and our method (red rectangle) when the background has some features similar to the target. In Fatih Porikli's method, an obvious drift happens when the background is similar to the target, while in the proposed method this phenomenon can be overcome successfully. The corresponding tracking errors are plotted in Fig. 9.



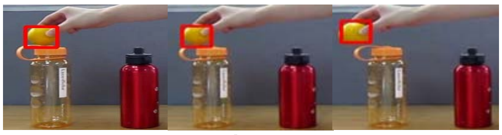
(a)

(b)

Fig.7 girl sequence with similar color background

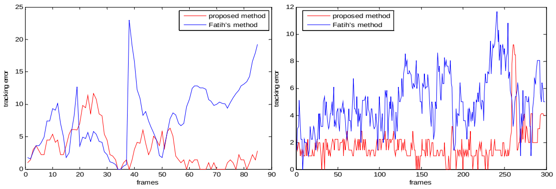


(a)



(b)

Fig.8. Lemon sequence with similar color cup around



(a) (b)

Fig.9. Error plots for girl sequence (a) and lemon sequence (b)

By using the lemon sequence for test, Fig. 10 illustrates the tracking errors with different complexity thresholds (CT). For a small CT, the tracking error is large. Increasing CT, however, does not significantly improve the performance which, on the contrary, results in unnecessary computational burden. As a rule of thumb, CT around 10 is regarded as a cost-efficient value while still achieving satisfying tracking precision.

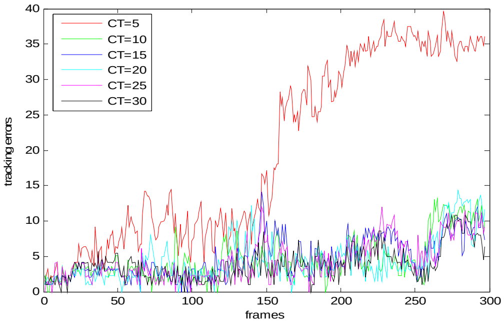


Fig.10 Tracking errors with different complexity thresholds

V. Conclusions

In this paper, a new tracking algorithm with low rank covariance features is proposed and described. The whole RC matrix used in typical RC-based tracking algorithms is decomposed into several low rank parts. A subset of low-rank RC features are dynamically chosen and processed to represent the target according to the current state and background. With an appropriate choice of complexity threshold, a good tradeoff between discriminability and complexity can be achieved. Experimental results have shown that our method is more robust to complex environment changes compared with other algorithms, especially when occlusion happens and when the background is similar to the target.

REFERENCES

[1] Oncel Tuzel, et al, "Region covariance: A fast descriptor for detection and classification", Computer Vision - ECCV 2006, Pt 2, Proceedings, vol. 3952, pp. 589-600, 2006.

[2] Fatih Porikli, et al., "Covariance Tracking using Model Update Based on Means on Riemannian Manifold", 2006 IEEE Conf. on Computer Vision and Pattern Recognition, 2006.

[3] Y. Wu, et al., "Probabilistic Tracking on Riemannian Manifolds," 19th International Conf. on Pattern Recognition, Vols 1-6, pp. 229-232, 2008.

[4] D. A. Ross, et al., "Incremental learning for robust visual tracking," International Journal of Computer Vision, vol. 77, pp. 125-141, May 2008.

[5] X. Li, et al., "Visual tracking via incremental Log-Euclidean Riemannian subspace learning," 2008 IEEE Conf. on Computer Vision and Pattern Recognition, Vols 1-12, pp. 1349-1356, 2008.

[6] V. Arsigny, et al., "Log-Euclidean metrics for fast and simple calculus on diffusion tensors", Magnetic Resonance in Medicine, vol. 56, pp. 411-421, Aug 2006.

[7] Y. Wu, J. Cheng, J. Wang, H. Lu, "Real-time visual tracking via incremental covariance tensor learning", International Conf. on Computer Vision, 2009.

[8] F¨orstner, W. Moonen, B. “A metric for covariance matrices” Technical report Dept. of Geodesy and Geoinformatics, Stuttgart University, 1999.

[9] X. Pennec, et al., "A Riemannian Framework for Tensor Computing", International Journal of Computer Vision, vol. 66, pp. 41-66, 2006.

[10] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian non-linear state space models", J. Comput, Graph Statist, vol. 5, no. 1, pp.1–25, 1996.

[11] A. Daucet, S. Godsill, and C. Andrieu, "on sequential Monte Carlo sampling method for Bayesian filtering", Statistics and Computing, vol. 10, pp. 197-208, 2000.