IEICE TRANS. INF. & SYST., VOL.E101–D, NO.5 MAY 2018

PAPER

Long-Term Tracking Based on Multi-Feature Adaptive Fusion for Video Target

Hainan ZHANG[†], ***Nonmember***, **Yanjing SUN**[†a)], ***Member***, **Song LI**[†], **Wenjuan SHI**[†], ***and*** **Chenglong FENG**[†], ***Nonmembers***

SUMMARY The correlation filter-based trackers with an appearance model established by single feature have poor robustness to challenging video environment which includes factors such as occlusion, fast motion and out-of-view. In this paper, a long-term tracking algorithm based on multi-feature adaptive fusion for video target is presented. We design a robust appearance model by fusing powerful features including histogram of gradient, local binary pattern and color-naming at response map level to conquer the interference in the video. In addition, a random fern classifier is trained as re-detector to detect target when tracking failure occurs, so that long-term tracking is implemented. We evaluate our algorithm on large-scale benchmark datasets and the results show that the proposed algorithm have more accurate and more robust performance in complex video environment.

key words: *target tracking, correlation filter, feature, fusion*

1. Introduction

Visual tracking is an important research in computer vision for its numerous applications in robotics, video surveillance and human computer interaction [1]. It can be split into different subfields according to different applications such as multi-target tracking and single-target tracking, the focus here is the latter one which assumes that the single object is indicated by a bounding box in the first frame, and the tracker identify the object in the following frame according to the box information rather than other prior knowledge such as category and shape. Currently, lots of trackers which are preoccupied with single-target tracking have been proposed, the performance of the algorithms is still constrained by the complex video environment [2] because the challenging factors such as illumination variation, occlusion and deformation would change object appearance and affect the features' ability of describing the target.

Vast trackers that focus on establishing a robust object appearance model have been explored [3], [4], which can be divided into generative-method and discriminative-method. And we prefer the discriminative-method, for it applies the machine learning to improve the tracking performance. The discriminative-method transforms target tracking into a binary classification problem that separates target from background using classifier. In [5], Avidan et al. train a classifier via extracting the feature vector of each pixel, and they identify whether a pixel belongs to target or background by utilizing the trained classifier in a new frame. Tang et al. [6] propose a method to realize tracking by taking advantage of Support Vector Machine (SVM) and Co-Tracking, which fuses the confidence maps learned from different independent features and generates a new sample using the fused confidence map. Zhang et al. [7] utilize sparse measurement matrix for compressing feature space in real-time compressive tracking algorithm, then classify the samples of foreground and background by Bayes classifier.

In [8], Bolme et al. propose MOOSE that introduce the correlation filter into the tracker for the first time. From then on, the tracking algorithms based on correlation filter become an important theoretical branch among the discriminative-method due to the higher efficient and higher precision. Henriques et al. [9] improve the discriminating ability of the classifier by using the cyclic shift which can construct numerous training samples. Then they introduce kernel trick into tracking algorithm with KCF method [10]. Furthermore, Ma et al. [11] propose LCT which is a correlation filter-based tracker that performs well in long-term visual tracking because of the applying of the re-detection module. Although performing better than classical algorithms, these correlation filter-based trackers are not always effective for coping with the complex video environment because the object appearance models built by single gray features or HOG features are not robust. Li et al. [12] adopt a multiple feature integration scheme in SAMF, which demonstrates the effectiveness of multiple features integration for boosting tracking performance. But the advantage of SAMF is limited owing to the inelasticity between the way of feature integration and the specific conditions of video sequences. In addition, it fails to re-detect the objects in case of tracking failure when long-term occlusion or out-of-view occurs.

In this paper, we propose a novel long-term tracker whose structure is similar to [13]. The approach establishes a robust appearance model by fusing multiple features adaptively including HOG, LBP and CN at response map level to enhance the tracking performance in the challenging scenarios. Meanwhile, we train a random ferns classifier as re-detector to re-detect the target in case of tracking failure to further achieve long-term tracking. We evaluate the proposed tracker on a large-scale benchmark [1]. The experimental results demonstrate that the performance of our

Manuscript received August 1, 2017.
Manuscript revised December 25, 2017.
Manuscript publicized February 2, 2018.

[†]The authors are with the School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, 221116, China.
a) E-mail: yjsun@cumt.edu.cn

DOI: 10.1587/transinf.2017EDP7245

Copyright © 2018 The Institute of Electronics, Information and Communication Engineers

algorithm is favorable against state-of-the-art methods in the aspect of accuracy and robustness.

2. Kernelized Correlation Filter

Kernelized correlation filter (KCF) tracker exploring the structure of the circulant matrix and kernel trick is proposed by Henriques et al. in [10]. In the training phase, we sample an image patch surrounding the target as base sample $\mathbf{x}$ which has $M \times N$ pixels, and generate training negative samples $\mathbf{x}_i$ by the cyclic shift of $\mathbf{x}, \mathbf{x}_i = P_i\mathbf{x}, i \in \{0, 1, \ldots, m-1\} \times \{0, 1, \ldots, n-1\}$, where P is the permutation matrix

$$P = \begin{bmatrix} 0 & 0 & 0 & \ldots & 1 \\ 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 & 0 \end{bmatrix} \tag{1}$$

The purpose of training is to find a $\mathbf{w}$ constructing a function $f(\mathbf{x}) = \mathbf{w}^t\mathbf{x}$ to solve the following minimization problem:

$$\min_{\mathbf{w}} \sum_i (f(\mathbf{x}_i) - y_i)^2 + \lambda \|\mathbf{w}\|^2 \tag{2}$$

Where λ is a regularization parameter, and y_i is Gaussian function label corresponding to $\mathbf{x}_i$.

To handle the problem of no-linear regression, kernel function $\kappa(\mathbf{x}, \mathbf{x}') = \varphi(\mathbf{x}) \cdot \varphi(\mathbf{x}')$ is introduced, where $\varphi(\mathbf{x})$ map $\mathbf{x}$ to Hilbert space. We can re-express $\mathbf{w}$ by using $\mathbf{w} = \sum_i \alpha_i \varphi(\mathbf{x}_i)$, and finally have function $f(\mathbf{x}) = \mathbf{w}^t \cdot \mathbf{x} = \sum_i \alpha_i \kappa(\mathbf{x}, \mathbf{x}_i)$, so Eq. (2) can be described as follows:

$$\min_{\mathbf{w}} \sum_i (\langle \mathbf{w}, \varphi(\mathbf{x}_i) \rangle - y_i)^2 + \lambda \|\mathbf{w}\|^2 \tag{3}$$

As the minimizer has a closed-form according to [14]

$$\boldsymbol{\alpha} = (K + \lambda I)^{-1}\mathbf{y} \tag{4}$$

Where K is kernel function with elements $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ and $\boldsymbol{\alpha}$ is the vector of coefficients α_i. $\boldsymbol{\alpha}$ can be obtained from Eq. (5) according to the character of circulant matrix and convolution:

$$\hat{\boldsymbol{\alpha}} = \frac{\hat{\mathbf{y}}}{\hat{\mathbf{k}}^{\mathbf{xx}} + \lambda} \tag{5}$$

Where a hat $\hat{}$ represents Fourier transform, $\hat{\mathbf{k}}^{\mathbf{xx}}$ is defined as kernel correlation and has elements

$$\mathbf{k}^{\mathbf{xx}'} = \exp\left(-\frac{1}{\sigma^2}(\|\mathbf{x}\|^2 + \|\mathbf{x}'\|^2 - 2F^{-1}(\hat{\mathbf{x}}^* \odot \hat{\mathbf{x}}'))\right) \tag{6}$$

Where $*$ is complex-conjugate and $\odot$ denotes the Hadamard product.

In the testing phase, the tracker computes the response map on a new image patch $\mathbf{z}$ with the same search window size of $M \times N$ as follows:

$$f(\mathbf{z}) = F^{-1}(\hat{\mathbf{k}}^{\mathbf{xz}} \odot \hat{\boldsymbol{\alpha}}) \tag{7}$$

Where $\mathbf{k}^{\mathbf{xz}}$ represents the kernel correlation of $\mathbf{x}$ and $\mathbf{z}$, like the definition in Eq. (6). And F^{-1} denotes the inverse FFT transform. f is the response map which can detect the new position of the target by locating the maximal value.

3. Long-Term Tracking Based on Multi-Feature Adaptive Fusing for Video Target

As we aim to develop a long-term tracking algorithm that is robust to the change of appearance caused by challenging factors in video, we decompose the task into adaptive fusing multiple features and re-detector, where the purpose of fusing features is to establish a robust appearance model to enhance tracking performance and the reason of using re-detector is to re-detect object in case of tracking failure to achieve long-term tracking. The block diagram is shown in Fig. 1. Firstly, we extract HOG, CN and LBP features of the image patch, respectively. Then, we train a correlation filter for each feature and get corresponding reponse map. We compute the final response map by fusing the three reponse maps adaptively and estimate the position of target by searching for the location of maximum value of the final response map. During tracking, we re-detect target when tracking failure occurs with the sign that the maximum value of final response map is less than threshold. Finally, we update model parameter and target template with a learning rate η.

3.1 Features Adaptive Fusing

Considering features fusing, we choose HOG, CN and LBP which are complementary to each other and the integration of them can make appearance model robust in complex video scenarios. HOG is histogram of gradient [15] which focuses on gradient information and shows effective response to illumination variation and background clutters; LBP is local binary pattern [16], an operator that describes

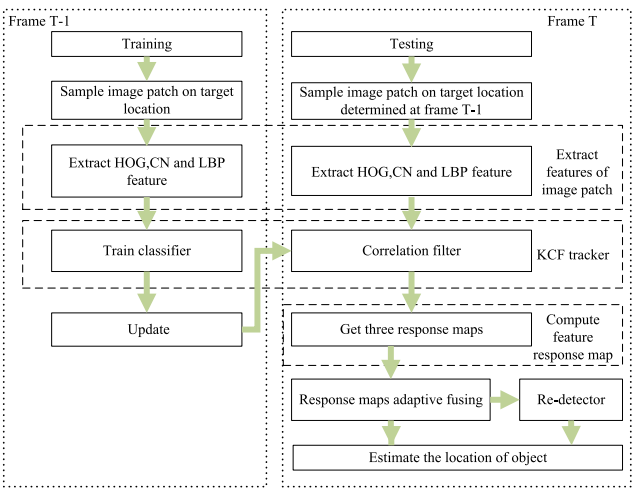


Fig. 1 The block diagram of the proposed algorithm

the local texture features of images and has ability to deal with motion blur and fast motion; CN is color-naming [17] which represents the linguistic color label assigned by human to describe the color. CN copes well with partial occlusion and deformation. We learn a discriminative correlation filter and get a response map for each feature. Then we obtain the final response map by combining three response maps with weights which are computed according to the importance of features in challenging situations. Firstly, we compute the variance between the estimated response and the desired response:

$$s_t^c = (\max(f_t^c) - \max(f_i^c)), c \in C \tag{8}$$

Where f_t^c is the response map under the c feature in frame t, and f_i^c denotes the desired response map computed from the first frame, $C = \{HOG, CN, LBP\}$ represents features set.

Secondly, The importance of feature c is calculated as follows:

$$\tilde{w}_t^c = 1 - \frac{s_t^c}{s_t^{HOG} + s_t^{LBP} + s_t^{CN} + \varepsilon} \tag{9}$$

Where ε is a tiny regularization parameter and we set it to $\varepsilon = 10^{-5}$.

Finally, we define the weight:

$$w_{t+1}^c = \frac{\tilde{w}_t^c}{\tilde{w}_t^{HOG} + \tilde{w}_t^{LBP} + \tilde{w}_t^{CN} + \varepsilon} \tag{10}$$

Where w_t^c is the weight of response map of c feature in frame t. Therefore, we get the final response map

$$f_{t+1} = \sum w_{t+1}^c \times f_{t+1}^c \tag{11}$$

Where f^c is the response map of c feature computed by Eq. (7), and we can detect the location of target by searching for the position of the maximum value of f_{t+1}.

3.2 Re-Detector

In the process of tracking, we train a random ferns classifier as re-detector to re-detect object in case of tracking failure to futher implement long-term tracking. The re-detector is applied to the whole frame with sliding window. For improving the efficiency of the algorithm, we use a threshold to activate the re-detector when $\max(f) < \Gamma_r$ rather than re-detect object on each frame like [19]. Suppose that we have class labels $c_i, i \in \{0, 1\}$ and binary features $f_j, j \in \{1, 2, \ldots, N\}$. Group the features into M which contains features with the number of $S = N/M$ and treat the group as ferns. The joint distribution of features in each fern is shown as follows:

$$P(f_1, f_2, \ldots, f_N \,|\, C = c_i) = \prod_{k=1}^{M} P(F_k \,|\, C = c_i) \tag{12}$$

Where $F_k = \{f_{\sigma_{k,0}}, f_{\sigma_{k,1}}, \ldots, f_{\sigma_{k,S}}\}$ denotes the $k-th$ fern, and $\sigma(k, n)$ represents a permutation function with range 1 to N. The conditional probability of F_k is:

$$P(F_k \,|\, C = c_i = \frac{N_{k,c_i}}{N_k}) \tag{13}$$

Where N_{k,c_i} is the number of training samples of class c_i and N_k represents the total number of training samples. Finally, we can get the optimal class according to the Bayesian perspective:

$$\hat{c_i} = \arg\max_{c_i} \prod_{k=1}^{M} P(F_k \,|\, C = c_i) \tag{14}$$

Meanwhile, we adopt K-nearest neighbor classifier to choose the most confident tracked results as target when k nearest features vectors in the training set all have positive labels, and we set to $k = 5$ in this paper.

3.3 Module Update

We define a threshold for updating when $\max(f) > \Gamma_u$ to solve the problem of tracking drift even failure caused by the introduction of error messages and accumulation in case of updating frame by frame. So we update model parameter and target template as follows when $\max(f) > \Gamma_u$:

$$\hat{\boldsymbol{\alpha}}_t = \eta\hat{\boldsymbol{\alpha}}_t + (1 - \eta)\hat{\boldsymbol{\alpha}_{t-1}} \tag{15}$$

$$\hat{\mathbf{x}}_t = \eta\hat{\mathbf{x}}_t + (1 - \eta)\hat{\mathbf{x}_{t-1}} \tag{16}$$

Where η is learning rate, $\hat{\boldsymbol{\alpha}}_t, \hat{\boldsymbol{\alpha}_{t-1}}$ respectively represent the model parameter of the current frame and the previous frame and $\hat{\mathbf{x}}_t, \hat{\mathbf{x}_{t-1}}$ express the target template of the current frame and the previous frame.

And we keep the model parameter and target template unchanged when $\max(f) < \Gamma_u$

$$\hat{\boldsymbol{\alpha}}_t = \hat{\boldsymbol{\alpha}_{t-1}} \tag{17}$$

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}_{t-1}} \tag{18}$$

3.4 Implementation

The implementation details are discussed in Algorithm1.

4. Experimental Results and Analysis

In this section, we evaluate the performance of the proposed LTMF using distance precision ratio (DP) and overlap success ratio (OS) under one-pass evaluation (OPE), temporal robustness evaluation (TRE) and spatial robustness evaluation (SRE). TRE initializes the evaluation at 20 different frames when SRE initializes the bounding boxes with various perturbation. The two evaluation schemes can better demonstrate the robustness of the trackers as additional evaluations to OPE. DP means the percentage of frames where the Center Location Error (CLE) between estimated location and actual location is smaller than a certain threshold in all frames. OS refers to the percentage of frames where the Overlap between estimated location and actual location is greater than a certain threshold in all frames. Moreover, we

Algorithm 1 : **LTMF**

Input: Initial target position x_0
Output: Estimated object position x_t

1: **for** $t = 1$ to T (T is the total number of video frames)
2: // Extract image patch features, including HOG, CN and LBP
Crop out the searching window in frame t according to x_{t-1} and extract HOG, CN and LBP features respectively
3: // Compute response map
Train correlation filter for each feature and get corresponding response map f^{HOG}, f^{LBP}, f^{CN} by using Eq. (7)
4: // Fuse multi-feature adaptively
For multi-feature adaptive fusing, we give three response maps corresponding weights according to the importance of features.
We obtain the weights using Eq. (10), and compute the final response map f_t using Eq. (11).
5: // Estimate the new position
if $\max(f) < \Gamma_r$
Use re-detector to detect the target x_t;
else
Estimate the target x_t by locating the maximal value of f_t
end if
6: // Update model parameter and target template
if $\max(f) > \Gamma_u$
Update model parameter and target template using Eq. (15) and Eq. (16)
Update re-detector;
end if
end for

also assess the tracking speed by displaying video frames that the algorithm reads per second.

4.1 Experimental Environment and Parameter Settings

Our algorithm is implemented in MATLAB R2014a on Windows7 with Intel(R)C-ore(TM)i5-4590CPU. And the algorithm is validated on a large benchmark dataset [1]. We choose all seventy three objects in seventy one fully marked color videos with eleven different challenging factors: Illumination Variation (IV), Scale Variation (SV), Occlusion (OCC), Deformation (DEF), Motion Blur (MB), Fast Motion (FM), In-Plane Rotation (IPR), Out-of-Plane Rot-ation (OPR), Out-of-View (OV), Background Clutters (BC) and Low Resolution (LR). Meanwhile, we set the regularization parameter to $\lambda = 10^{-4}$, and the learning rate is set to $\eta = 0.01$, the thresholds are set to $\Gamma_r = 0.25$, $\Gamma_u = 0.38$, $M \times N$ is 1.8 times larger than the target size.

4.2 Component Analysis

We prove the effectiveness of LTMF by comparing with four trackers LT_HOG, LT_LBP, LT_CN and TMF. The LT_HOG, LT_LBP and LT_CN are trackers which describe the target appearance using HOG, LBP and CN features alone under the same framework with the LTMF, while the TMF is a tracker without re-detector. As shown in Table 1, our LTMF outperforms the LT_HOG, LT_LBP and LT_CN because the fusion of multiple features establish a robust appearance model and make the method effective in challenging videos. And LTMF also has higher precision against the TMF because the re-detector deals well with tracking failure.

Table 1 The comparisons of 5 trackers under OPE, TRE and SRE. We present the comparisons of distance precision ratio (DP) at a threshold of 20 pixels, overlap success ratio (OS) at an overlap threshold 0.5 and the tracking speed. The first and second best values are highlighted by bold and underline.

		LTMF	TMF	LT_HOG	LT_CN	LT_LBP
DP(%)	OPE	**0.726**	0.711	0.652	0.659	0.623
	TRE	**0.750**	0.747	0.695	0.697	0.669
	SRE	**0.679**	0.667	0.631	0.618	0.607
OS(%)	OPE	**0.577**	0.562	0.521	0.533	0.510
	TRE	**0.650**	0.649	0.609	0.612	0.589
	SRE	**0.555**	0.542	0.508	0.509	0.496
MEAN SPEED (fps)		22.34	**58.97**	26.17	24.65	29.01

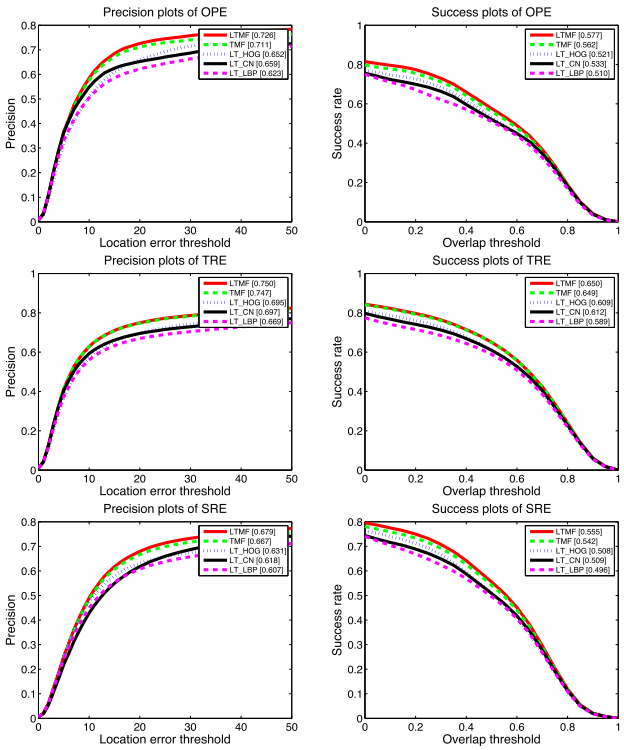


Fig. 2 Distance precision and overlap success plots using one-pass evaluation (OPE), temporal robustness evaluation (TRE) and spatial robustness evaluation (SRE). The legend of distance precision contains threshold scores at 20 pixels while the legend of overlap success contains threshold scores at 0.5. The LTMF performs favorably against the other trackers.

We show the corresponding results in one-pass evaluation (OPE), temporal robustness evaluation (TRE) and spatial robustness evaluation (SRE) using distance precision ratio (DP) plot and overlap success ratio (OS) plot in Fig. 2. The legend of distance precision contains threshold scores at 20 pixels while the legend of overlap success contains threshold scores at 0.5.

4.3 Quantitative Evaluation

In order to evaluate performance quantitatively, we compare our LTMF with 8 state-of-the-art trackers, including

Table 3 Comparisons of the 9 algorithms under OPE, TRE and SRE. We present the comparisons of distance precision ratio (DP) at a threshold of 20 pixels, overlap success ratio (OS) at an overlap threshold 0.5 and the tracking speed. The first and second best values are highlighted by bold and underline.

		LTMF	KCF	LCT	SAMF	CSK	STRUCK	TLD	CT	CXT
DP(%)	OPE	**0.726**	0.638	0.709	0.708	0.453	0.565	0.531	0.323	0.455
	TRE	**0.750**	0.682	0.708	0.742	0.541	0.639	0.549	0.376	0.525
	SRE	**0.679**	0.586	0.650	0.664	0.418	0.556	0.494	0.287	0.435
OS(%)	OPE	0.577	0.506	**0.659**	0.639	0.351	0.465	0.452	0.244	0.369
	TRE	0.650	0.596	0.672	**0.687**	0.459	0.559	0.468	0.300	0.432
	SRE	0.555	0.471	**0.604**	0.597	0.325	0.449	0.404	0.182	0.365
MEAN SPEED (fps)		22.34	162.34	15.66	11.47	**198.05**	10.37	30.96	59.53	15.70

Table 2 The differences of the 9 algorithms

Name	Features	Re-detect	Correlation filter-based
LTMF	HOG, CN, LBP	Y	Y
LCT	HOG	Y	Y
KCF	HOG	N	Y
CSK	GRAY	N	Y
SAMF	HOG, CN	N	Y
STRUCK	HAAR	N	N
TLD	BINARY	Y	N
CT	HAAR	N	N
CXT	AURF	N	N

LCT [11], KCF [10], CSK [9], SAMF [12], STRUCK [18], TLD [19], CT [7] and CXT [20]. Table 2 shows their differences.

We present the 9 algorithms' comparisons of distance precision ratio at a threshold of 20 pixels, overlap success ratio at an overlap threshold 0.5 and the tracking speed in Table 3. Figure 3 shows the corresponding results in one-pass evaluation (OPE), temporal robustness evaluation (TRE) and spatial robustness evaluation (SRE) using distance precision ratio plot and overlap success ratio plot.

Table 3 exhibits that LTMF, KCF, LCT, SAMF which are correlation filter-based trackers outperform STRUCK, TLD, CT and CXT. In terms of distance precision ratio (DP), our LTMF achieves the best performance under all three evaluation schemes with scores of 72.6%, 75.0% and 67.9%. In terms of overlap success ratio (OS), LTMF shows considerable margin compared to KCF, STRUCK, TLD, CSK, CT and CXT while SAMF and LCT outperform it. The reason is that LTMF does not search over scales as opposed to SAMF and LCT which has positive impact to overlap success ratio when most objects change their scales frequently in evaluation environment. Furthermore, the concentration on scale estimation makes the speed of SAMF and LCT be 11.47 fps and 15.66 fps which are slower than our LTMF who runs at around 22.34 fps.

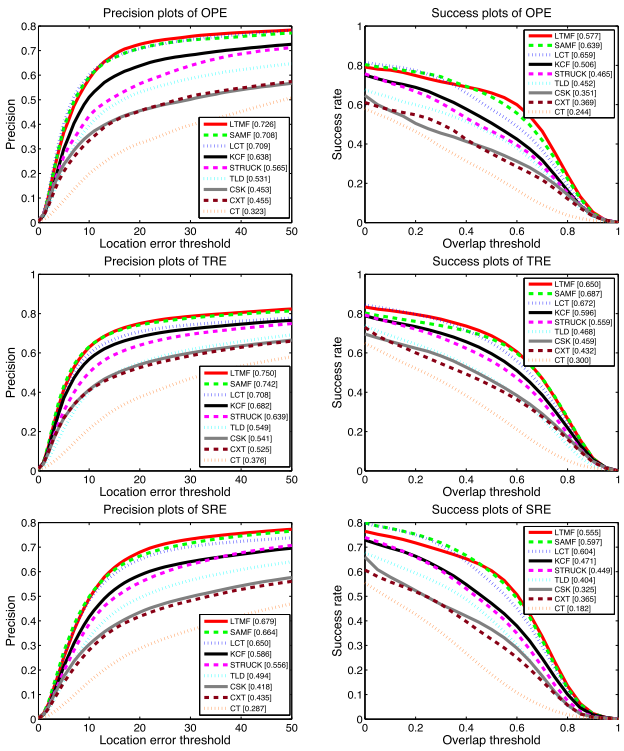


Fig. 3 Distance precision and overlap success plots of the 8 algorithms using one-pass evaluation (OPE), temporal robustness evaluation (TRE) and spatial robustness evaluation (SRE). The legend of distance precision contains threshold scores at 20 pixels while the legend of overlap success contains threshold scores at 0.5.

Table 4 Challenging factors in the sequences

Name	Challenging factors
Shaking	IV SV IPR OPR BC
Tiger2	IV OCC DEF MB FM IPR OPR OV
Jogging2	OCC DEF OPR
Box	IV SV OCC MB IPR OPR OV BC LR
DragonBaby	SV OCC MB FM IPR OPR OV
Diving	SV DEF IPR
Deer	MB FM IPR BC LR
Couple	SV DEF FM OPR BC

4.4 Qualitative Evaluation

We evalute LTMF with comparisons to 4 trackers (KCF, SAMF, LCT and TLD) on 8 challenging sequences. KCF is a correlation filter-based tracker. SAMF is a tracker based on multiple features fusing. LCT and TLD are trackers with re-detector. The partial tracking results are shown in Fig. 4. And Table 4 shows the challenging factors included in these sequences.

(*a*) Shaking: the video exists illumination variation and background clutter etc. At frame #60, LTMF and LCT stay up with target while other algorithms drift away in the case that background suffers from highlight. And at frame #195,

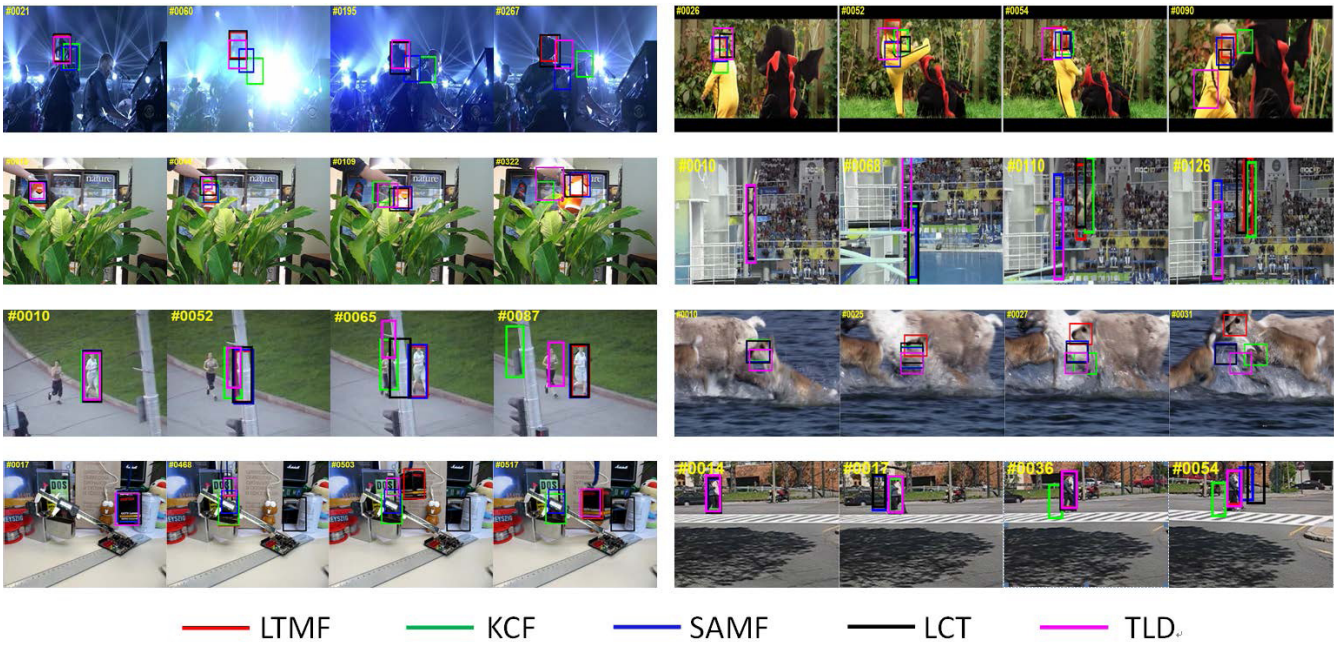


Fig. 4 Tracking results of LTMF, KCF, SAMF, LCT and TLD trackers (from top to down and left to right are Shaking, Tiger2, Jogging2, Box, DragonBaby, Diving, Deer and Couple, respectively)

KCF locates the wrong target and tracks failure in the next sequences.

(*b*) Tiger2: the video sequences are plagued by deformation and occlusion etc. At frame #44, LCT and TLD drift away from target because of the motion blur. KCF and TLD fall in failure when the target endures bright light and rotation as shown at frame #109 and frame #322.

(*c*) Jogging2: the video sequences undergo occlusion and deformation etc. At first, five trackers track target equally well. At frame #52, target stays under the condition of overall occultation. When the object appears again at frame #65, LTMF and SAMF track it immediately while LCT re-detects successfully at later frame #87. Meanwhile KCF tracks failure and TLD re-detects the false target.

(*d*) Box: the video suffers from rotation and occlusion etc. LCT shows terrible tracking efficiency in this video and locates failure because of the background clutter. LTMF re-detects the target successfully when it is obscured for a long time from frame #468 to frame #503, TLD re-detects the target at #517 while SAMF and KCF lose it.

(*e*) DragonBaby: the video sequences are plagued by rotation and occlusion etc. LTMF tracks the target accurately while the other trackers drift away when the target rotates in the image.

(*f*) Diving: the video sequences endure deformation and fast motion etc. SAMF and TLD track failure with the deformation of object while KCF and LCT drift away. LTMF performs well in the whole video and keeps following the target.

(*g*) Deer: the video sequences suffer from motion blur and fast motion etc. KCF, SAMF, LCT and TLD drift away from the target when motion blur and fast motion occur from frame #25 to frame #31, while LTMF provides precise results.

(*h*) Couple: the video sequences undergo motion blur and fast motion etc. LTMF and TLD perform well in sequences with camera motion while KCF, SAMF and LCT drift.

With the arguments above, we can conclude that LTMF performs well in the complex video environment because the fusion of multiple features establishes a robust appearance model. Meanwhile, the use of re-detector helps LTMF to handle the tracking failure caused by prolonged occlusion.

5. Conclusions

In this paper, we propose a long-term tracking algorithm based on multi-feature adaptive fusion for video target. We propose a multiple features fusion scheme to establish a effective appearance model to further boost tracker's robustness in challenging video environment. The multiple features fusion scheme is that we adaptively fuse three features response maps obtained from three discriminative correlation filters. Meanwhile, the proposed LTMF learns a random ferns classifier to re-detect object when tracking failure occurs. The extensive empirical evaluations demonstrate that the proposed algorithm has more accurate and more robust performance in complex video environment.

Acknowledgments

This work is supported by National Natural Science Foundation of China (51504214, 51504255, 51274202), National

Key Research and Development Program (2016YFC0801403), the Fundamental Research and Development Foundation of Jiangsu Province (BE2015040), Natural Science Foundation of Jiangsu province of China (BK20150204, BK20130199, BK20131124), the Perspective Research Foundation of Production Study and Research Alliance of Jiangsu Province under Grant (BY2014028-01), the Transformation Program of Scientific and Technological Achievements of Jiangsu Province (BA2012068).

References

[1] Y. Wu, J. Lim, and M.-H. Yang, "Object Tracking Benchmark," IEEE Trans. Pattern Anal. Mach. Intell., vol.37, no.9, pp.1834–1848, 2015.
[2] J. Wang, H. Zhu, S. Yu, and C. Fan, "Object tracking using color-feature guided network generalization and tailored feature fusion," Neurocomputing, vol.238, pp.387–398, 2017.
[3] Q. Miao, C. Zhang, and L. Meng, "Feature-Based On-Line Object Tracking Combining Both Keypoints and Quasi-Keypoints Matching," IEICE Trans. Inf. & Syst., vol.E99-D, no.4, pp.1264–1267, 2016.
[4] Q. Miao, C. Shi, L. Meng, and G. Cheng, " "On-Line Rigid Object Tracking via Discriminative Feature Classification," IEICE Trans. Inf. & Syst., vol.E99-D, no.11, pp.2824–2827, 2016.
[5] S. Avidan, "Ensemble tracking," IEEE Trans. Pattern Anal. Mach. Intell., vol.29, no.2, pp.261–271, 2007.
[6] F. Tang, S. Brennan, Q. Zhao, and H. Tao, "Co-Tracking Using Semi-Supervised Support Vector Machines," 2007 IEEE 11th International Conference on Computer Vision, pp.1–8, 2007.
[7] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," Computer Vision ECCV 2012, vol.7574, pp.864–877, 2012.
[8] D.S. Bolme, J.R. Beveridge, B.A. Draper, and Y.M. Lui, "Visual object tracking using adaptive correlation filters," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp.2544–2550, 2010.
[9] J.F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the Circulant Structure of Tracking-by-Detection with Kernels," Computer Vision – ECCV 2012, Lecture Notes in Computer Science, vol.7575, pp.702–715, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
[10] J.F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-Speed Tracking with Kernelized Correlation Filters," IEEE Trans. Pattern Anal. Mach. Intell., vol.37, no.3, pp.583–596, 2015.
[11] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Long-term correlation tracking," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.5388–5396, 2015.
[12] Y. Li and J. Zhu, "A Scale Adaptive Kernel Correlation Filter Tracker with Feature Integration," Computer Vision - ECCV 2014 Workshops, Lecture Notes in Computer Science, vol.8926, pp.254–265, Springer International Publishing, Cham, 2015.
[13] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P.H.S. Torr, "Staple: Complementary Learners for Real-Time Tracking," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.1401–1409, 2016.
[14] R. Rifkin, G. Yeo, and T. Poggio, "Regularized least-squares classification," Nato Science Series Sub Series III Computer and Systems Sciences, vol.190, pp.131–154, 2003.
[15] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," IEEE Trans. Pattern Anal. Mach. Intell., vol.32, no.9, pp.1627–1645, 2010.
[16] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," IEEE Trans. Pattern Anal. Mach. Intell., vol.24, no.7, pp.971–987, 2002.
[17] J. van de Weijer, C. Schmid, and J. Verbeek, "Learning Color Names from Real-World Images," 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp.1–8, 2007.
[18] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S.L. Hicks, and P.H.S. Torr, "Struck: Structured Output Tracking with Kernels," IEEE Trans. Pattern Anal. Mach. Intell., vol.38, no.10, pp.2096–2109, 2016.
[19] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N learning: Bootstrapping binary classifiers by structural constraints," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp.49–56, 2010.
[20] T.B. Dinh, N. Vo, and G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," CVPR 2011, pp.1177–1184, 2011.



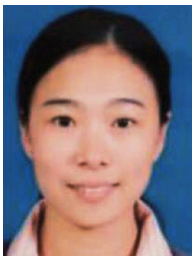
Hainan Zhang received the B.S. degree in communication Engineering from Harbin University Of Science And Technology in 2015. She is currently working the M.S. degree at the school of Information and Control Engineering, China University of Mining and Technology. Her current research interests include computer vision, visual tracking and so on.



Yanjing Sun received the Ph.D. degree in Information and Communication Engineering from China University of Mining and Technology in 2008. He is currently a professor in School of Information and Control Engineering, China University of Mining and Technology. He is also a vice director of Coal Mine Electrical Engineering and Automation Laboratory in JiangSu Province. His current research interests include IBFD communication, Embedded real-time system, Wireless sensor networks, Cyber-physical system and so on.



Song Li received the Ph.D. degree in signal and information processing from Beijing University of Posts and Telecommunications in 2012. He is currently an associate professor in School of Information and Control Engineering, China University of Mining and Technology. His current research interests are full-duplex communication, cyber-physical system and cooperative communication.



Wenjuan Shi received her M.S. degree in mechatronics from Soochow University, Suzhou, China, in 2006. She is currently a PhD student in China University of Mining and Technology. Her current research interests include video quality assessment, data mining and so on.



Chenglong Feng received the B.S. and M.S. degrees in school of Information and Control Engineering, China University of Mining and Technology in 2014 and 2017. His current research interests include computer vision, visual tracking and so on.