EURASIP Journal on Advances in Signal Processing

RESEARCH Open Access

Long-term target tracking combined with re-detection

Juanjuan Wang[1], Haoran Yang[1], Ning Xu[1], Chengqin Wu[1], Zengshun Zhao[1,2,3*], Jixiang Zhang[1*] and Dapeng Oliver Wu[3]

* Correspondence: zhaozs@sdust.edu.cn; zjxhii@163.com
[1]College of Electronic and Information Engineering, Shandong University of Science and Technology, Qingdao 266590, P.R. China
Full list of author information is available at the end of the article

Abstract

Long-term visual tracking undergoes more challenges and is closer to realistic applications than short-term tracking. However, the performances of most existing methods have been limited in the long-term tracking tasks. In this work, we present a reliable yet simple long-term tracking method, which extends the state-of-the-art learning adaptive discriminative correlation filters (LADCF) tracking algorithm with a re-detection component based on the support vector machine (SVM) model. The LADCF tracking algorithm localizes the target in each frame, and the re-detector is able to efficiently re-detect the target in the whole image when the tracking fails. We further introduce a robust confidence degree evaluation criterion that combines the maximum response criterion and the average peak-to-correlation energy (APCE) to judge the confidence level of the predicted target. When the confidence degree is generally high, the SVM is updated accordingly. If the confidence drops sharply, the SVM re-detects the target. We perform extensive experiments on the OTB-2015 and UAV123 datasets. The experimental results demonstrate the effectiveness of our algorithm in long-term tracking.

Keywords: Learning adaptive discriminative correlation filters, Long-term tracking, Re-detection

1 Introduction

While visual object tracking as a hot research topic in computer vision has been widely applied in various fields, many challenges are still not resolved especially in target disappearance, partial occlusion, and background clutter, and studying a general and powerful tracking algorithm is a tough task.

A typical scenario of visual tracking is to track an unknown object in subsequent image frames by giving the initial state of a target in the first frame of the video. In the past few decades, visual object tracking technology has made significant progress [1–10]. These methods are very effective for short-term tracking tasks, which the tracked object is almost always in the field of view. However, in realistic applications, the requirement for tracking is not only to track correctly, but also to track for a longer period of time [11]. During the period of time, the tracking output is wrong in the absence of the target objects. And the training samples will be incorrectly annotated,

which leads to a risk of model drifts. Therefore, it is important to long-term trackers to determine whether the target is absent and have the capability of re-detection.

Long-term tracking task also requires the tracker as well as short-term tracking to maintain high accuracy in the challenges of disappearance and occlusion, especially to stably capture the target object in a long-term video [12]. Therefore, the long-term tracking presents more challenges from two aspects. The first issue is how to determine the confidence degree of the tracking results. In [13], the maximum response value of the target is used to determine the confidence of the tracking result. When the maximum peak value of the response map is lower than the threshold value, the result is determined to be unreliable. However, the response map may fluctuate drastically when the object in occlusion or disappear condition, only using the maximum response value to judge confidence is incredibility. The average peak-to-correlation energy (APCE) criterion in [14] indicates the degree of fluctuation of the response map. If the target is undergoing fast motion, the value of APCE will be low even if the tracking is correct. However, the APCE criterion is commonly used to update trackers in [14]. Secondly, how to relocate the out-of-view targets remains unresolved. The tracking-learning-detection (TLD) [15] algorithm exploits an ensemble of weak classifiers for global re-detection of the out of view. The method fails to classify the target object due to the huge number of scanning windows. The long-term correlation tracking (LCT) [13] algorithm proposes a random fern re-detection model to detect the out-of-view target. In [16], it learns a spatial-temporal filter in a lower-dimensional discriminative manifold to alleviate the influences of boundary effects. But the method still cannot solve the target disappearance problem.

This paper proposes a tracking algorithm combining the learning adaptive discriminative correlation filter tracker and re-detector. The proposed method aims to perform robust re-detection and relocate the target when target tracking fails. Our main contributions can be summarized as follows:

i) We propose a stable long-term tracking strategy to track the targets that may disappear or deform heavily in long-term tracking. With the confidence strategy adopted, the learning adaptive discriminative correlation filters (LADCF) tracks the accurate target online. And the support vector machine (SVM) is updated when the confidence degree is generally high. In contrast, if the response maps fluctuate heavily, the SVM is used as a re-detector to relocate the target.
ii) We not only utilize the maximum response but also adopt the APCE criterion to the re-detection component. The fusion of the two criteria can accurately determine the state of the tracker and improve the accuracy of the tracking system.
iii) We evaluate the proposed tracking algorithm on the OTB-2015 [17] and UAV123 [18] datasets; the experimental results show that the proposed algorithm performs more stable and accurate tracking performance in the case of occlusion, background clutter, etc. during the long-term tracking.

The structure of the rest of the paper is as follows: Section 2 overviews the related work. Section 3 presents the proposed method. Section 4 reports the experimental results and experimental analysis. Section 5 concludes the paper.

2 Related works

2.1 Correlation filter

Correlation filters have shown outstanding results for target tracking [17, 19]. These methods exploit the circular correlation of the filter in the frequency domain to locate the target object. Bolme et al. [4] propose the pioneering MOSSE tracker, using only gray image features to train the filter. The circulant structure of tracking-by-detection with kernels (CSK) tracker [20] employs the illumination intensity features and applies DCFs in a kernel space. The kernelized correlation filters (KCF) [6] further improves CSK by the use of the multi-channel histogram of oriented gradient (HOG) features. Danelljan et al. [5] exploit the color attributes of the target object and learn an adaptive correlation filter. The literature [21] proposes a patch-based visual tracker that divides the object and the candidate area into several small blocks evenly and uses the average score of the overall small blocks to determine the optimal candidate, which greatly improves under the occlusion circumstances. The literature [22] proposes an online representative sample selection method to construct an effective observation module that can handle occasional large appearance changes or severe occlusion.

The estimation of the target scale is another important aspect for testing an outstanding tracker. It not only improves better performance, but also provides computational efficiency. The discriminative scale space tracking (DSST) tracker [23] performs translation estimation and scale estimation separately, using a scale pyramid to respond to the scale change. Li and Zhu [24] present an effective scale adaptive scheme, which defines a scale pool to turn the samples of each scale into the same size as the initial sample by the bilinear interpolation method.

The formulation of DCFs exploits the circular correlation which implements learning efficiently by applying fast Fourier transform (FFT). However, it induces the circular boundary effects, which has a drastic negative impact on tracking performance. Danelljan et al. [25] suggest reducing these boundary effects by introducing a spatial regularization component. Nevertheless, regularization will make the cost of the model optimization higher. Galoogahi et al. [26] propose an idea to the pre-multiply a fixed masking matrix containing the target regions to address such deficiency of DCFs. Then, they apply the alternating direction method of multipliers (ADMM) [27] algorithm to solve the constrained optimization problem in real time. The context-aware correlation filter tracking (CACF) [28] algorithm selects the background reference around the target by considering the global information and adds the background penalty to the closed solution of the filter. The discriminative correlation filter with channel and spatial reliability (CSRDCF) [29] method distinguishes the foreground and background by segmenting the colors in the search area. The learning adaptive discriminative correlation filters (LADCF) [16] approach adds adaptive spatial feature selection and temporal consistency constraints to alleviate the spatial boundary effects and temporal filter degradation problems that exist in the DCF method.

2.2 Long-term tracking

Kalal et al. [15] propose a tracking-learning-detection (TLD) algorithm, which decomposes the tracking task into tracking, learning, and detection. Among them, tracking and detection facilitate each other, the short-term tracker provides training examples

for the detector, while the detectors are implemented as a cascade to reduce computational complexity. Enlightened by the TLD framework, Ma et al. [13] propose a long-term correlation filter tracker using a KCF as a baseline algorithm and a random fern classifier as a detector. The FCLT-A fully correlational long-term tracker (FCLT) [30] trains several correlation filters on different time scales as a detector and exploits the correlation response to link the short-term tracker and long-term detector.

3 Methods

In this section, we describe our tracker. In Section 3.1, we introduced the main tracking framework of our algorithm, which is shown in Fig. 1. In Section 3.2, we introduce the tracker based on LADCF correlation filtering. In Section 3.3, we introduce the composite evaluation criteria of the confidence degree and the SVM based re-detector.

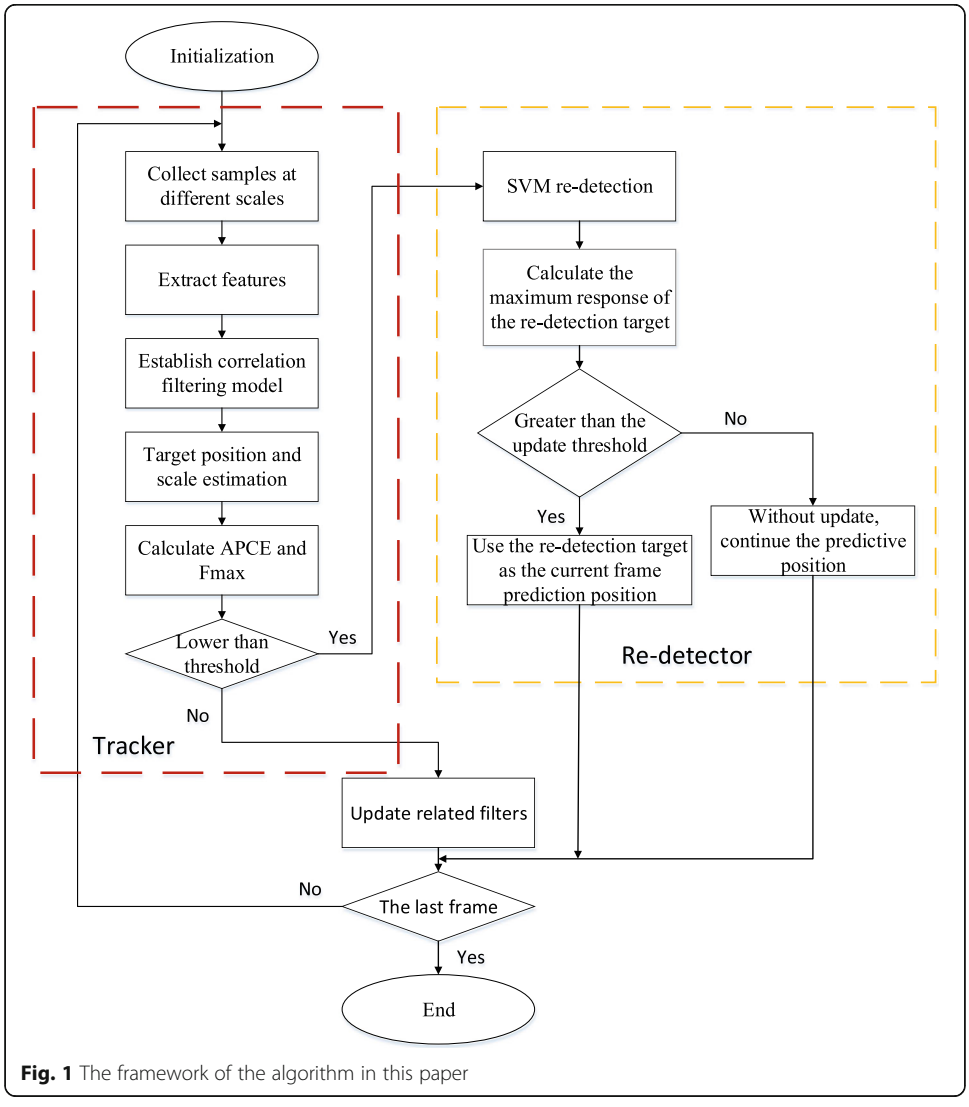

Fig. 1 The framework of the algorithm in this paper

3.1 The main framework of the algorithm

The proposed algorithm aims to combine both the DCF tracker and the re-detector for long-term tracking. First, the baseline correlation filter tracker is adopted to estimate the translation in the tracking stage. Second, the maximum response value and the APCE criterion are utilized to judge the confidence level of the target. Finally, when the value of confidence is higher than the threshold, the baseline tracker achieves the tracking target alone. When the confidence level drops sharply, it indicates tracking failure. We do not update the model and exploit the SVM model to re-detect the target object in the current frame. The structure of the algorithm in this paper is shown in Fig. 1.

The tracking framework is summarized as follows:

(1) Position and scale detection: We utilize DSST to achieve the target position and scale prediction. The $t-th$ frame target is I_t, and the filter model is θ_{model}. When a new frame I_t appears, we extract multiple scale search windows $[I_t^{\text{patch}}\{s\}]$ from it, $s = 1, 2, \ldots, S$, with S denoting the number of scales. For each scale s, the search window patch is centered around the target center position p_{t-1} with a size of $a^N n \times a^N n$ pixels, where a is the scale factor and $N = \lfloor \frac{2s-S-1}{2} \rfloor$. The size of the basic search window size is $n \times n$, which is determined by the target size $\omega \times h$ and padding parameter as $n = (1+\varrho)\sqrt{\omega \times h}$. So, the bilinear interpolation is applied to resize each patch into $n \times n$. Then, we extract multi-channel features for each scale search window as $\chi(s) \epsilon \mathbb{R}^{D^2 \times L}$. Given the filter template, the response score can efficiently be calculated in the frequency domain as [16]:

$$\hat{f}(s) = \hat{x}(s) \odot \hat{\theta}^*_{model} \tag{1}$$

After the implementation of the IDFT on each scale, the maximum value of $f \in \mathbb{R}^{D^2 \times S}$ is the relative position and scale.

(2) Updating: We adopt the same updating strategy as the traditional DCF method:

$$\theta_{model} = (1-\alpha)\theta_{model} + \alpha\theta \tag{2}$$

where α is the updating rate. More specifically, since θ_{model} is not available in the learning stage for the first frame, we use a pre-defined mask that only the target region is activated to optimize θ as in BACF. And then, we initialize $\theta_{\text{model}} = \theta$ after the learning stage of the first frame.

3.2 Correlation filter tracker

In this paper, we set LADCF [16] as the baseline algorithm of our tracking approach.

The LADCF algorithm proposes a new DCF-based tracking method, which utilizes the adaptive spatial feature selection and temporal consistent constraints to reduce the impact of spatial boundary effect and temporal filter degradation. The feature selection process is to select several specific elements in the filter to retain distinguishable and

descriptive information, forming a low-dimensional and compact feature representation. Considering an $n \times n$ image patch $x \in \mathbb{R}^{n^2}$ as a base sample for the DCF design, the circulant matrix for this sample is generated by collecting its full cyclic shifts, $X^T = [x_1, x_2, \ldots, x_{n^2}]^T \in \mathbb{R}^{n^2 \times n^2}$ with the corresponding Gaussian-shaped regression labels $y = [y_1, y_2, \ldots, y_{n^2}]$. The spatial feature selection embedded in the learning stage can be expressed as:

$$\begin{aligned} &\underset{\theta,\phi}{argmin} \|\theta \circledast x - y\|_2^2 + \lambda_1 \|\phi\|_0 \\ &s.t. \theta = \theta_\phi = diag(\phi)\theta, \end{aligned} \tag{3}$$

where θ denotes the target model in the form of DCF, and $\circledast$ denotes the circular convolution operator. The indicator vector ϕ can potentially be expressed by θ and $\|\phi\|_0 = \|\theta\|_0$, and diag($\phi$) is the diagonal matrix generated from the indicator vector of selected features ϕ. The ℓ_0-norm is non-convex, and the ℓ_1-norm is widely used to approximate the sparsity [24], so a temporal consistency is constructed by ℓ_1-norm relaxation spatial feature selection model [16]:

$$\underset{\theta}{\text{argmin}} \|\theta \circledast x - y\|_2^2 + \lambda_1 \|\theta\|_1 + \lambda_2 \|\theta - \theta_{model}\|_1 \tag{4}$$

where λ_1 and λ_2 are tuning parameters, and $\lambda_1 << \lambda_2$. θ_{model} denotes the model parameters estimated from the previous frame.

The ℓ_2-norm relaxation is adopted to further simplify the following expression:

$$\underset{\theta}{argmin} \|\theta \circledast x - y\|_2^2 + \lambda_1 \|\theta\|_1 + \lambda_2 \|\theta - \theta_{model}\|_2^2 \tag{5}$$

where the lasso regularization controlled by λ_1 select the spatial feature. In the above formula, the filter template model is used to increase smoothness between consecutive frames to promote time consistency. In this way, the temporal consistency of spatial feature selection can be preserved to extract and retain the diversity of the static and dynamic appearance.

Since the multi-channel features share the same spatial layout [16], the multi-channel input is represented as $X = \{x_1, x_2, \ldots, x_L\}$, and the corresponding filter is represented as $\theta = \{\theta_1, \theta_2, \ldots, \theta_L\}$. By minimization, the goal can be extended to multi-channel functions with structured sparsity [16]:

$$\underset{\theta}{\text{argmin}} \sum_{i=1}^{L} \|\theta_i \circledast x_i - y\|_2^2 + \lambda_1 \left\| \sqrt{\sum_{i=1}^{L} \theta_i \odot \theta_i} \right\|_1 + \lambda_2 \sum_{i=1}^{L} \|\theta_i - \theta_{model\ i}\|_2^2 \tag{6}$$

where θ^j is the jth element of the ith channel feature vector $\theta_i \in \mathbb{R}^{D^2}$. $\odot$ denotes the element-wise multiplication operator. The structured spatial feature selection term calculates the ℓ_2-norm of each spatial location and then executes the ℓ_1-norm to achieve joint sparsity.

Subsequently, utilizing ADMM [27] to optimize the above formula, we introduce the relaxation variables to construct the goals based on convex optimization [31]. Then, we could obtain the global optimal solution of the model through ADMM and form an enhanced Lagrange operator [16]:

$$\mathcal{L} = \sum_{i=1}^{L} \|\theta_i \circledast x_i - y\|_2^2 + \lambda_1 \sum_{j=1}^{D^2} \left\| \sqrt{\sum_{i=1}^{L} \left(\theta_i^j\right)^2} \right\|_1 + \lambda_2 \sum_{i=1}^{L} \|\theta_i - \theta_{model\ i}\|_2^2 + \frac{\mu}{2} \sum\nolimits_{i=1}^{L} \left\| \theta_i - \theta_i^{'} + \frac{\eta_i}{\mu} \right\|_2^2 \tag{7}$$

where $\mathcal{H} = \{\eta_1, \eta_2, ..., \eta_L\}$ are the Lagrange multipliers, and $\mu > 0$ is the corresponding penalty parameter controlling the convergence rate [16, 32]. As $\mathcal{L}$ is convex, ADMM is exploited iteratively to optimize the following sub-problems with guaranteed convergence:

$$\begin{cases} \theta = \arg\min\limits_{\theta} \mathcal{L}\left(\theta, \theta^{'}, \mathcal{H}, \mu\right) \\ \theta^{'} = \arg\min\limits_{\theta^{'}} \mathcal{L}\left(\theta, \theta^{'}, \mathcal{H}, \mu\right) \\ \mathcal{H} = \arg\min\limits_{\mathcal{H}} \mathcal{L}\left(\theta, \theta^{'}, \mathcal{H}, \mu\right) \end{cases} \tag{8}$$

3.3 Re-detector

3.3.1 Confidence criterion

Most existing trackers do not consider whether the detection is accurate or not. In fact, once the target is detected incorrectly in the current frame, severely occluded, or completely missing, this may cause the tracking failure in subsequent frames.

We introduce a measure to determine the confidence degree of the target objects, which is the first step in the re-detection model. The peak value and the fluctuation of the response map can reveal the confidence about the tracking results. The ideal response map should have only one peak while all the other regions are smooth. Otherwise, the response map will fluctuate intensely. If we continue to use the uncertain samples to track the target in the subsequent frames, the tracking model will be destroyed. Thus, we exploit to fuse two confidence degree evaluation criteria. The first one is the maximum response value $F_{\max}$ of the current frame.

The second one is the APCE measure which is defined as:

$$APCE = \frac{|F_{max} - F_{min}|^2}{mean\left(\sum_{w,h}\left(F_{w,h} - F_{min}\right)^2\right)} \tag{9}$$

where the $F_{\max}$ and $F_{\min}$ are the maximum response and the minimum response of the current frame, respectively. $F_{w,\ h}$ is the element value of the wth row and the hth column of the response matrix.

If the target is moving slowly and is easily distinguishable, the APCE value is generally high. However, if the target is undergoing fast motion with significant deformations, the value of APCE will be low even if the tracking is correct.

3.3.2 Target re-detection

In this section, we describe the re-detection mechanism used in the case of tracking failure. In the re-detection module, when the confidence level is lower than the threshold, the SVM [33] is used for re-detection. Considering a sample set (x_1, y_1), (x_2, y_2), …, (x_i, y_i), …, $x_i \in R^d$, including positive and negative samples, where d is the dimension of

the sample, $y_i \in (+1, -1)$ is sample labels, SVM can make segmentation of positive and negative samples to obtain the best classification hyperplane. The classification plane is defined as [33]:

$$\omega^{\mathrm{T}} x + b = 0 \tag{10}$$

where ω represents the weight vector, and b denotes the bias term. In the case of the linearly classifiable, for a given dataset T and classification hyperplane, the following formula is used for classification judgment:

$$\begin{cases} \omega^{\mathrm{T}} x + b \le 1, y_i = -1 \\ \omega^{\mathrm{T}} x + b \ge 1, y_i = +1 \end{cases} \tag{11}$$

Combining the two equations, we can abbreviate it as:

$$y\left(\omega^T x + b\right) \ge 1 \tag{12}$$

The distance from each support vector to the hyperplane can be written as:

$$d = \frac{\left|\omega^T x + b\right|}{\|\omega\|} \tag{13}$$

The problem of solving the maximum partition hyperplane of the SVM model can be expressed as the following constrained optimization problem:

$$\begin{aligned} &min \frac{1}{2}\|\omega\|^2 \\ &s.t. y_i\left(\omega^T x_i + b\right) \ge 1 \end{aligned} \tag{14}$$

Next, the paper introduces the Lagrangian function to solve the above problem [33].

$$L(\omega, \lambda, c) = \frac{1}{2}\left\|\omega\right\|^2 - \sum_{j=1}^{l} c_i y_i (\omega \cdot x_i + b) + \sum_{i=1}^{l} c_i \tag{15}$$

where $c_i > 0$ is the Lagrange multiplier, the solution of the optimization problem satisfies the partial derivative of $L(\omega, \lambda, c)$ to ω and b be 0. The corresponding decision function is expressed as:

$$f(x) = \mathrm{sign}(\omega^* \cdot x + b^*) = \mathrm{sign}\left\{\left(\sum_{\mathrm{j}=1}^{\mathrm{l}} c_j^* y_j (x_j \cdot x_i)\right) + b^*\right\} \tag{16}$$

Then, the new sample points are imported into the decision function to get the sample classification.

In the case of linear inseparability, we use the kernel function to map it to the high-dimensional space. In this work, we use the Gaussian kernel function as follows:

$$k\left(x_i, x_j\right) = e^{\left(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}\right)} \tag{17}$$

When a frame is re-detected, an exhaustive search is performed on the current frame using a sliding window, and the HOG features are extracted for each image patch as the X vector in the above formula. And the $f(x)$ is calculated by formula (16). Then, we obtain the sample area with the largest $f(x)$. When the response value is greater than the threshold, it will be used as the location of the tracking target again.

The training process of SVM is as follows [33]. By the confidence level, we determine the quality of the sample. Then, samples with high confidence are used as the positive samples, and samples with low confidence are used as the negative samples. The HOG features from positive and negative samples are extracted to obtain the feature vectors. The feature vectors are represented as (x_i, y_i), $i = 1, 2, \ldots, n$, where n denotes the number of training samples, x_i represents the HOG feature vector, and y_i represents the attribute of the extracted sample. If the training sample is positive, then $y_i = 1$, and if the sample is negative, then $y_i = -1$. For the binary classification problem of our samples, the loss function is defined as formula (18).

$$Loss(x, y, \omega) = - \max(0, 1 - y(x \cdot \omega)) \tag{18}$$

When the value of loss is negative, the parameters of SVM are updated as follows.

$$\omega^* = \sum\nolimits_{j=1}^{l} c_j^* y_j x_j \tag{19}$$

$$b^* = y_i - \sum\nolimits_{j=1}^{l} y_j c_j^* (x_j \cdot x_i) \tag{20}$$

where c_j is the Lagrangian coefficient, x is the feature vector extracted from the sample, and y is the label corresponding to the sample.

4 Experimental results and discussion

In this section, we evaluate the proposed algorithm on OTB-2015 and UAV123 benchmarks [17] with comparisons to other detection-based tracking algorithms and classical correlation filtering tracking algorithms. Section 4.1 introduces the experimental platform and parameter settings of the experiments. Section 4.2 introduces the experimental datasets and the evaluation criteria for the experiments. Section 4.3 describes the quantitative evaluation of the results and describes the qualitative evaluation in Section 4.4

4.1 Experimental setups

The experimental software environment is MATLAB R2016a, and the hardware environment is Intel Core i5-4200M processor, 4GB memory, Windows 8 operating system.

The regularization parameters λ_1 and λ_2 are set to 1 and 15, respectively; the initial penalty parameter $\mu = 1$; the maximum penalty parameter $\mu_{\max} = 20$; the maximum number of iterations $K = 2$; the padding parameter as $\varrho = 4$; the scale factor as $a = 1.01$; the threshold for re-detection is set to $tr = 0.13$; and the update threshold is set to $tu = 0.20$.

4.2 Experimental datasets and evaluation criteria

The OTB-2015 dataset has a total of 100 video sequences, including 11 challenges, namely, illumination variation (IV), scale variation (SV), occlusion (OCC), deformation (DEF), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of-plane rotation (OPR), out-of-view (OV), background clutter (BC), and low resolution (LR). The UAV123 consists of 143 challenging sequences, including 123 short-term sequences and 20 long-term sequences. Their evaluation criteria adopt the distance precision and overlap precision in one-pass evaluation (OPE) as the criteria of the evaluation algorithm. The overlap precision is defined as the percentage of overlap ratios exceeding 0.5. The distance precision shows the percentage of location error within 20 pixels.

4.3 Quantitative evaluation

In this paper, we compare our algorithm with 6 state-of-the-art trackers on the OTB-2015 dataset, including 2 tracking-by-detection algorithms, such as LCT [13] and large margin object tracking with circulant feature maps (LMCF) [14], and 4 mainstream correlation filtering tracking algorithms, such as CSK [20], KCF [6], DSST [23], and background-aware correlation filters (BACF) [26]. Figure 2 shows the OPE success rate and precision plots of these algorithms. It can be seen from Fig. 2 that the proposed algorithm has been significantly improved compared with other algorithms. The precision and success rate of our method are 81.4% and 59.9%, respectively. Through experiments, we found that the short-term target trackers learn some wrong information, when the target is occluded or disappears. Thus, the template is polluted by the wrong information and unable to track the target correctly in subsequent frames. Therefore, compared with the BACF algorithm, our method improves the precision and success rate by 14.8% and 7.8%, respectively. The LCT exploits the random fern algorithm to re-detect targets, which is slow to operate. Thus compared with the tracking-by-detection LCT algorithm, the proposed algorithm improves the precision and success rate by 8% and 9.3%, respectively. Compared with the LMCF algorithm with multi-peak detection, our method increased the precision and success rate by 11.2% and 11.1%, respectively.

In order to further verify the superiority of our method, we analyze the tracking performance through attribute-based comparison in Table 1, which shows the area under the curve (AUC) scores of the success plots with 11 different attributes.

As shown in Table 1, the proposed algorithm in this paper achieves the best performance on 11 attributes. In the case of OCC, our algorithm score is 10.1% higher than that of the LMCF algorithm (tracking-by-detection style) and 12% higher than the algorithm BACF algorithm (short-term correlation filtering style). For FM images, our algorithm is 4.6% higher than the second-ranked BACF algorithm and 5.1% higher than the LCT algorithm using random fern re-detection. In the above condition, the target model may be contaminated, which makes target tracking difficult. Meanwhile, our model can solve this problem by accurate re-detection via SVM. In the case of OPR, LCT achieves a score of 48.5%. And our tracker provides a gain of 8.7%. This is because the baseline algorithm applied in this paper solves the influence of boundary effects to

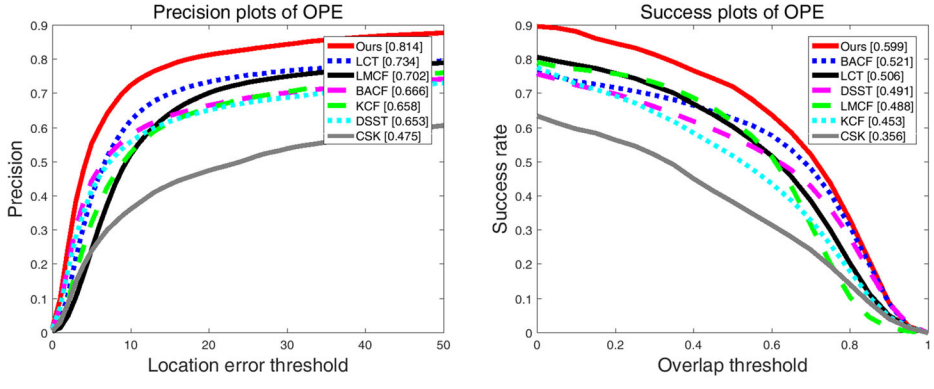

Fig. 2 Precision and success rate plots of the proposed method and state-of-art methods over OTB-2015 benchmark sequences using one-pass evaluation (OPE)

Table 1 The AUC scores of success plots on OTB-2015 sequences with different attributes

	CSK	KCF	DSST	BACF	LCT	LMCF	Ours
IV	0.357	0.470	0.533	0.523	0.509	0.524	**0.610**
SV	0.299	0.385	0.454	0.505	0.420	0.456	**0.563**
OCC	0.313	0.429	0.452	0.456	0.462	0.475	**0.576**
DEF	0.309	0.404	0.414	0.465	0.457	0.446	**0.513**
MB	0.287	0.431	0.439	0.505	0.498	0.471	**0.548**
IPR	0.354	0.441	0.482	0.475	0.511	0.453	**0.565**
FM	0.297	0.434	0.422	0.489	0.484	0.447	**0.535**
OPR	0.332	0.440	0.450	0.483	0.485	0.469	**0.572**
OV	0.230	0.371	0.350	0.468	0.423	0.440	**0.507**
BC	0.382	0.481	0.503	0.539	0.501	0.502	**0.597**
LR	0.248	0.290	0.381	0.502	0.281	0.399	**0.526**

a certain extent and can achieve higher accuracy when the target rotation occurs. In the case of OV, the score of our algorithm is 50.7%, which is 3.9% higher than the BACF algorithm. The reason is that our template stops updating when the target goes out of view; the SVM is used to detect the target again. When the target reappears in the field of view, our model is not contaminated and can continue tracking the target correctly.

Furthermore, we present the OPE success rate and precision plots on UAV123 in Fig. 3.

As shown in Fig. 3, our method beats other algorithms on the UAV123 dataset. Specifically, our method achieves the AUC scores of 65.2% and 46.1%, which is better than LCT by 13.1% and 13.4%. At the same time, the proposed method is 16.1% and 10.5% higher than BACF, because the proposed re-detection approach provides a novel solution to re-detect the low-confidence targets to improve tracking accuracy.

4.4 Qualitative evaluation

We selected 7 representative benchmark sequences from OTB-2015 to demonstrate the effectiveness of our algorithm. The visual evaluation results are shown in Fig. 4. As it can be seen from Fig. 4, in the "Jogger" sequence, the target is blocked at the 70th

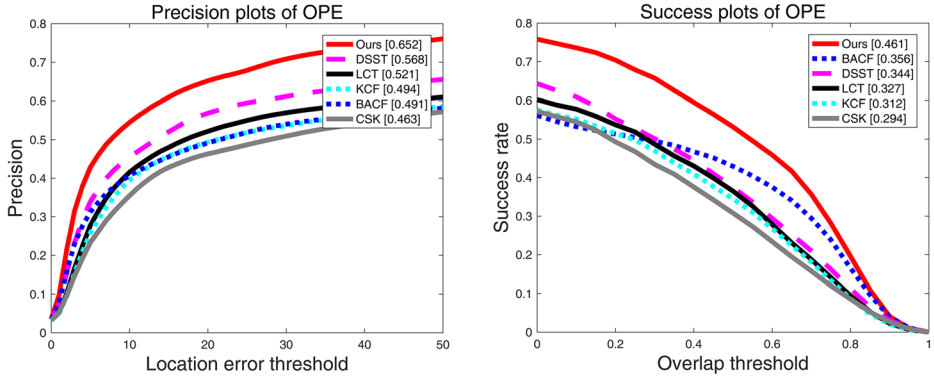

Fig. 3 Precision and success rate plots of the proposed method and state-of-art methods over UAV123 benchmark sequences using one-pass evaluation (OPE)

Fig. 4 The tracking results of each algorithm on 7 video sequences (from top to bottom are Jogging, Soccer, Matrix, Car4, Shaking, Bolt, Dog, respectively)

frame and the target reappears in the field of view at the 84th frame. Due to the re-detection mechanism, our tracker can track the target correctly. But the short-time correlation filter tracking algorithm learns error information during occlusion, which leads to tracking errors in subsequent frames. In the “Soccer” and “Matrix” sequences, due to background clutter, the algorithms such as LCT and BACF lose the target. In contrast, the proposed algorithm can successfully handle such situations. In the “Car4” sequence, due to the scale change problem, the scale-based DSST algorithm and the proposed algorithm both show better performance. In the “Shaking” sequence, the proposed algorithm loses its target in the 17th frame due to issues such as similar lighting changes and background. However, owing to the supplement of a re-detection mechanism, the proposed algorithm relocates the target at the 18th frame and keeps tracking correctly. In the “Bolt” sequence, our algorithm follows the target very closely even in the case of rapid motion of the target. In the “Dog” sequence, when the target is deformed, our algorithm can accurately track the target, while the BACF and LMCF algorithms have a

certain offset. It can be seen from the above description that our algorithm achieves higher accuracy in these 7 sequences.

Furthermore, we compare our method with the baseline tracker using 7 representative benchmark sequences of OTB-2015 in Fig. 5. The first three rows are short-term

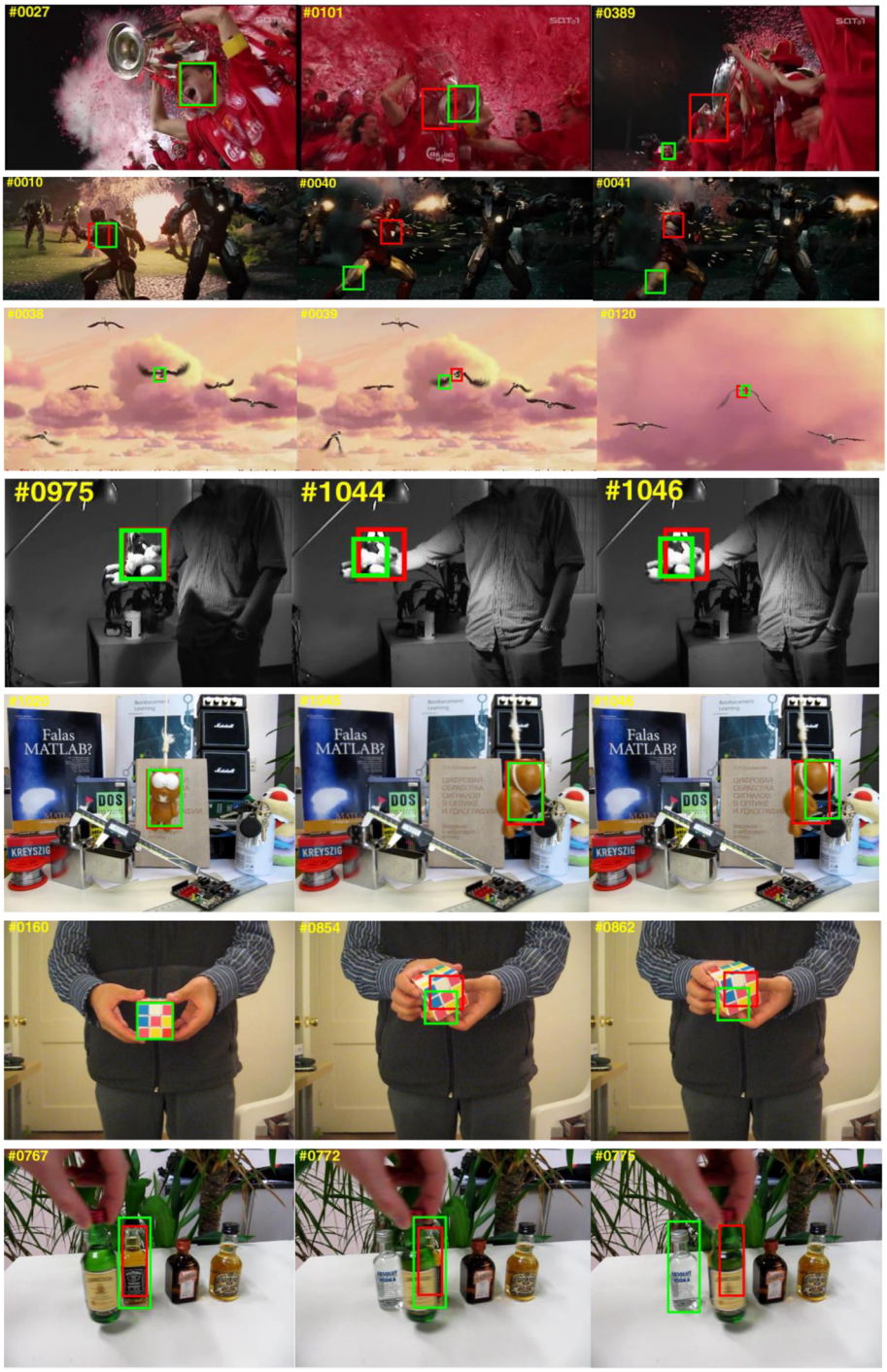

Fig. 5 The performance comparison of two algorithms on 6 video sequences (from top to bottom are Soccer, Ironman, Bird1, Sylvester, Lemming, Rubik, Liquor, respectively)

sequences which none of which exceeds 1000 frames, and the last four rows are long-term sequences, which all exceed 1000 frames.

As shown in Fig. 5, in the experiments for the short-term sequences, the LADCF tracker drifts when the target objects undergo heavy occlusions (Soccer) and does not re-detect targets in the case of tracking failure. Moreover, the LADCF tracker fails to handle background clutter and deformation (Ironman, Bird1), since only the tracking component without the re-detection mechanism makes it less effective to discriminate targets from the cluttered background. In contrast, our method can track the object correctly on these challenging sequences because the trained detector effectively re-detects the target objects.

In the Sylvester and Lemming sequences, the LADCF algorithm tracks incorrectly due to the rotating conditions encountered in these sequences, while our method provides better robustness to these conditions. In the Liquor sequence, the LADCF tracking algorithm is similar to our algorithm before the target is occluded. But when the target is occluded, the LADCF method fails to locate the occluded target. In the Rubik sequence, since the target object has undergone deformation and color variation at the 854th frame, the LADCF tracker fails to track correctly. Our method is able to track successfully due to re-detection. In our method, if the tracking fails, we perform the re-detection procedure and initialize the tracker so that the target can be re-detected. Thus, our method can correctly track the target all the time.

Overall, our method performs well in estimating the positions of the target objects, which can be attributed to three reasons. Firstly, the combined confidence criterion of our method can correctly identify the target even in very low-confidence cases. Secondly, our re-detection component effectively re-detects the target objects in the case of tracking failure. Thirdly, our baseline tracker achieves adaptive discriminative learning ability on a low-dimensional manifold and improves the tracking effect.

5 Conclusions

This paper proposes a long-term target tracking algorithm, where the two main components are a state-of-the-art LADCF short-term tracker which estimates the target translation and a re-detector which re-detect the target objects in the case of tracking failure. Besides, the algorithm introduces a robust confidence criterion to evaluate the confidence value of the predicted target. When the confidence value is lower than the specified threshold, the SVM model is utilized to re-detect the target objects and the template is not updated. The algorithm is suitable for long-term tracking because it can detect the target accurately in real time and update the template with high reliability. Numerous experimental results show that the proposed algorithm achieves better performances than the other tracking algorithms.

Abbreviations

LADCF: Learning adaptive discriminative correlation filters; APCE: Average peak-to-correlation energy; SVM: Support vector machine; OPE: One-pass evaluation

Acknowledgements

Thanks to the anonymous reviewers and editors for their hard work.

Authors' contributions

ZZ and DOW proposed the original idea of the full text. JZ and JW designed the experiment. JW and NX performed the experiment. JW and HY wrote the manuscript under the guidance of ZZ. CW, JZ, and JW revised the manuscript. All authors read and approved this submission.

Funding
This work was supported in part by the China Postdoctoral Science Special Foundation Funded Project (2015T80717), the Natural Science Foundation of Shandong Province (ZR2020MF086).

Availability of data and materials
The datasets used during the current study are the OTB2015 dataset [17] and the UAV123 dataset [18], which are available online or from the corresponding author on reasonable request.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]College of Electronic and Information Engineering, Shandong University of Science and Technology, Qingdao 266590, P.R. China. [2]School of Control Science & Engineering, Shandong University, Jinan 250061, China. [3]Department of Electrical & Computer Engineering, University of Florida, Gainesville, FL 32611, USA.

Received: 29 July 2020 Accepted: 11 December 2020
Published online: 06 January 2021

References

1. D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. **24**(5), 603–619 (2002)
2. M.S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Trans. Signal Process. **50**(2), 174–188 (2002)
3. H. Yang, J. Wang, Y. Miao, Y. Yang, Z. Zhao, Z. Wang, Q. Sun, D.O. Wu, Combining spatio-temporal context and Kalman filtering for visual tracking. Mathematics **7**(11), 1–13 (2019)
4. D.S. Bolme, J.R. Beveridge, B.A. Draper, Y.M. Lui, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Visual object tracking using adaptive correlation filters, (IEEE, San Francisco, 2010), pp. 2544–2550
5. M. Danelljan, F. Shahbaz Khan, M. Felsberg, J. Van de Weijer, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Adaptive color attributes for real-time visual tracking (2014), pp. 1090–1097
6. J.F. Henriques, R. Caseiro, P. Martins, J. Batista, High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. **37**(3), 583–596 (2015)
7. H. Nam, B. Han, in *2016 IEEE Conference on Computer Vision and Pattern Recognition*. Learning multi-domain convolutional neural networks for visual tracking (2016), pp. 4293–4302
8. L. Bertinetto, J. Valmadre, J.F. Henriques, et al., in *European Conference on Computer Vision Workshop*. Fully-convolutional Siamese networks for object tracking, vol 9914 (2016), pp. 850–865
9. E. Gundogdu, A.A. Alatan, Good features to correlate for visual tracking. IEEE Trans. Image Process. **27**(5), 2526–2540 (2018)
10. M. Asadi, C.S. Regazzoni, Tracking using continuous shape model learning in the presence of occlusion. EURASIP J. Adv. Signal Process. **2008**, 250780 (2008)
11. T. Li, S. Zhao, Q. Meng, et al., A stable long-term object tracking method with re-detection strategy. Pattern Recognit. Lett. **127**, 119–127 (2018)
12. B. Yan, H. Zhao, D. Wang, H. Lu, X. Yang, in *IEEE/CVF International Conference on Computer Vision*. 'Skimming-perusal' tracking: a framework for real-time and robust long-term tracking (2019), pp. 2385–2393
13. C. Ma, X. Yang, C. Zhang, M.H. Yang, in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Long-term correlation tracking (2015), pp. 5388–5396
14. M. Wang, Y. Liu, Z. Huang, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Large margin object tracking with circulant feature maps (2017), pp. 4800–4808
15. Z. Kalal, K. Mikolajczyk, J. Matas, Tracking-learning-detection. IEEE Trans. Pattern Anal. Mach. Intell. **34**(7), 1409–1422 (2012)
16. T. Xu et al., Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual object tracking. IEEE Trans. Image Process. **28**(11), 5596–5609 (2019)
17. Y. Wu, J. Lim, M.H. Yang, Object tracking benchmark. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1834–1848 (2015)
18. M. Mueller, N. Smith, B. Ghanem, in *European Conference on Computer Vision*. A benchmark and simulator for UAV tracking, (Springer, Amsterdam, 2016), pp. 445–461
19. Y. Wu, J. Lim, M.H. Yang, in *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*. Online object tracking: a benchmark (2013), pp. 2411–2418
20. J.F. Henriques, R. Caseiro, P. Martins, J. Batista, in *European Conference on Computer Vision*. Exploiting the circulant structure of tracking-by-detection with kernels (2012), pp. 702–715
21. W. Ou, D. Yuan, D. Li, et al., Patch-based visual tracking with online representative sample selection. J. Electron. Imaging **26**(3), 033006 (2017)
22. W. Ou, D. Yuan, Q. Liu, et al., Object tracking based on online representative sample selection via non-negative least square. Multimed. Tools Appl. **77**(9), 10569–10587 (2018)
23. M. Danelljan, G. Häger, F.S. Khan, M. Felsberg, in *Proceedings of the British Machine Vision Conference*. Accurate scale estimation for robust visual tracking, (BMVA Press, Nottingham, 2014), pp. 1–5
24. Y. Li, J. Zhu, in *European Conference on Computer Vision Workshop*. A scale adaptive kernel correlation filter tracker with feature integration, (Springer, Zurich, 2014), pp. 254–265
25. M. Danelljan, G. Hager, F. Shahbaz Khan, M. Felsberg, in *Proceedings of the IEEE International Conference on Computer Vision*. Learning spatially regularized correlation filters for visual tracking (2015), pp. 4310–4318
26. H. Kiani Galoogahi, A. Fagg, S. Lucey, in *Proceedings of the IEEE International Conference on Computer Vision*. Learning background-aware correlation filters for visual tracking (2017), pp. 1135–1143

27. S. Boyd, N. Parikh, E. Chu, et al., Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. Now Publishers Inc **3**(1), 1–122 (2011)
28. M. Mueller, N. Smith, B. Ghanem, in *IEEE Conference on Computer Vision & Pattern Recognition*. Context-aware correlation filter tracking (2017), pp. 1387–1395
29. A. Lukezic, T. Vojir, L.C. Zajc, J. Matas, M. Kristan, in *IEEE Conference on Computer Vision and Pattern Recognition*. Discriminative correlation filter with channel and spatial reliability (2017), pp. 4847–4856
30. A. Lukežič, L. Čehovin Zajc, T. Vojíř, J. Matas, M. Kristan, in *Asian Conference on Computer Vision*. FCLT - a fully-correlational long-term tracker (2017)
31. R. Jenatton, J. Mairal, et al., Structured sparsity through convex optimization. Stat. Sci. **27**(4), 450–468 (2012)
32. D.P. Bertsekas, *Constrained optimization and Lagrange multiplier methods,* (Academic, Pittsburgh, 1982)
33. T. Joachims, in *Advances in kernel methods support vector learning. Chapter 11*, ed. by B. Scholkopf, C. Burges, A. Smola. Making large-scale SVM learning practical (MIT Press, Cambridge, 1999), pp. 169–184

Publisher's Note