Multimedia Tools and Applications (2020) 79:25801–25815
https://doi.org/10.1007/s11042-020-09032-z

Improving model drift for robust object tracking

Qiujie Dong[1,2] · Xuedong He[3] · Haiyan Ge[4] · Qin Liu[1] · Aifu Han[1,2] · Shengzong Zhou[1]

Received: 4 June 2019 / Revised: 31 March 2020 / Accepted: 5 May 2020 /
Published online: 7 July 2020

Abstract
Discriminative correlation filters show excellent performance in object tracking. However, in complex scenes, the apparent characteristics of the tracked target are variable, which makes it easy to pollute the model and cause the model drift. In this paper, considering that the secondary peak has a greater impact on the model update, we propose a method for detecting the primary and secondary peaks of the response map. Secondly, a novel confidence function which uses the adaptive update discriminant mechanism is proposed, which yield good robustness. Thirdly, we propose a robust tracker with correlation filters, which uses hand-crafted features and can improve model drift in complex scenes. Finally, in order to cope with the current trackers' multi-feature response merge, we propose a simple exponential adaptive merge approach. Extensive experiments are performed on OTB2013, OTB100 and TC128 datasets. Our approach performs superiorly against several state-of-the-art trackers while runs in real-time.

Keywords Object tracking · Correlation filters · Primary and secondary peaks detection · Confidence function · Adaptive discriminant · Adaptive merge

1 Introduction

With the advent of the automation era, object tracking has gradually become a hot topic in recent years [3, 5, 8, 10, 13, 14, 19, 20, 22, 23, 31]. Tracking is multidisciplinary research that predicts a target position in all subsequent frames given the initial frame information. In the

✉ Shengzong Zhou
zhousz@fjirsm.ac.cn

[1] Fujian Institute of Research on the Structure of Matter Chinese Academy of Sciences, Fuzhou 350002, China

[2] North University of China, Taiyuan 030051, China

[3] School of Intelligent Systems Engineering, Sun Yat-sen University, Shenzhen, China

[4] Shandong University of Technology, Zibo 255049, China

early twenty-first century, the tracking models are generative models such as particle filters, but the trackers only use the features of the tracked target and ignore background information, so their performances are generally poor. Inspired by object detection, the discriminative models that separate the target from the background have become the mainstream.

At present, mainstream trackers based on discriminative models include the correlation filters (CF) trackers [1, 4, 6, 7, 9, 11, 12, 15–17, 25] which use hand-crafted features or CNNs and the Siamese Network trackers [2, 24, 26, 28, 32]. The CF trackers can update the tracking models in real time, so their robustness is better. The Siamese Network trackers adopt the offline pre-training networks, so their accuracy is higher, but the deep networks are larger, tracking models are hard to real-time updated, trackers' robustness is worse. Although trackers based on deep features also have good performance, these trackers need to perform pre-training processing of the network, and the information correlation between each stream is large, so it is difficult to highlight the effectiveness of our method. Therefore, our tracker uses the features of the Histogram of Oriented Gradient (HOG) [17] and Color Name (CN) [7]. The HOG feature focuses on the appearance information of the target, and the CN feature focuses on the color characteristics of the target. This paper proposes a multi-feature adaptive fusion mechanism based on multi-feature trackers, instead of using linear weighting for multi-feature fusion during target tracking. The Visual Tracker Benchmark[1] includes 11 attributes.

Though the CF trackers are nearing maturity, the constant variation of the target's scenes cause easily the background information to be learned into the model, resulting in the models to be polluted [27]. The reason for the contaminated model is that the response map is no longer a single peak, yet a multi-peak in the complex scenes. In the case where the difference between the value of primary peak and the value of secondary peak is not obvious, the model drift is most likely to occur.

Currently, the approach to solving model drift problems is to selectively update the model. Bolme et al. [4] proposed Peak to Sidelobe Ratio (PSR) confidence function, but the actual application effect is not prominent, so it has not been widely used. Wang et al. [25] proposed a novel criterion called average peak-to-correlation energy (APCE), which is better than PSR. Danelljan et al. [11]proposed a simple lazy update mechanism to update the model every five frames. However, model drift will still occur for [11] proposed scheme when the update frame is the image whose target apparent features change greatly, so it does not fundamentally solve the problem of template drift.

Now the CF trackers utilize multiple features instead of a single feature. However, a fixed merge factor is widely used in feature fusion, which reduces the performance of the trackers in the complex tracking scenarios and worst of all causes tracking failure.

In this paper, we consider the problems mentioned above and propose a novel object tracking method called improving model drift for robust object tracking with correlation filters (MDRCF). The main contributions of our works can be summarized as follows:

- A novel feature response map peak detection method is proposed to realize the detection of primary and secondary peaks.
- We propose a new confidence function using the adaptive update discriminative mechanism that can select different thresholds according to different tracking videos. This confidence function has good robustness and can be well transplanted in the others CF trackers.

[1] http://cvlab.hanyang.ac.kr/tracker_benchmark/benchmark_v10.html

- Based on the above method, we explore a novel CF tracker (MDRCF) with the hand-crafted features including Histogram of Oriented Gradient (HOG) and Color Name (CN), which can effectively improve the model drift and enhance tracker's performances in complex scenes. The proposed algorithm is compared with state-of-the-art (SOTA) CF trackers on the large-scale datasets: OTB2013 [29], OTB100 [30] and TC128 [21]. The proposed tracker's performances exceed ECO-HC on all three datasets.
- Finally, aiming at the multi-feature merge problem, we establish a new and simple multi-feature response adaptive merge method which is based on Staple [1], which is called exponential adaptive merge Staple (EAMStaple). The experiment proves that our proposed tracker works well in solving the multi-feature merge problem.

2 Related work

The CF trackers extract the template $\boldsymbol{\beta}$ from the initial frame and continuously train and update $\boldsymbol{\beta}$in subsequent frames. In the t^{th} frame, the template $\boldsymbol{\beta} \in \wp$ is updated where the loss function $L[g(\mathbf{x}_t, \boldsymbol{\beta}), \mathbf{y}_t]$ is minimized for the input sample $\mathbf{x}_t$ and the desired output $\mathbf{y}_t$, that is

$$\boldsymbol{\beta}_t = \mathrm{argmin}_{\boldsymbol{\beta}\in\wp}\sum_{i=1}^{t}\left\{L[g(\mathbf{x}_i, \boldsymbol{\beta}), \mathbf{y}_i] + \lambda\|\boldsymbol{\beta}\|^2\right\} \tag{1}$$

where λ is a regularization parameter to prevent over-fitting, and $\|\cdot\|$ is L_2 norm. $g(\mathbf{x}_t, \boldsymbol{\beta}) = \boldsymbol{\beta}^T\mathbf{x}_t$ is the regression function.

For the loss function, the ridge regression method is used in this paper, and its performance is consistent with the Support Vector Machine (SVM), but the structure is relatively simple and has a closed-form solution [17],

$$L[g(\mathbf{x}_t, \boldsymbol{\beta}), \mathbf{y}_t] = \left[\boldsymbol{\beta}^T\mathbf{x}_t - \mathbf{y}_t\right]^2 \tag{2}$$

Substituting Eq. (2) into Eq. (1),

$$\boldsymbol{\beta}_t = \mathrm{argmin}_{\boldsymbol{\beta}\in\wp}\sum_{i=1}^{t}\left[\left(\boldsymbol{\beta}^T\mathbf{x}_i - \mathbf{y}_i\right)^2 + \lambda\|\boldsymbol{\beta}\|^2\right] \tag{3}$$

According to the [17], the closed-form solution of Eq. (3) is

$$\boldsymbol{\beta} = \left(\mathbf{x}^T\mathbf{x} + \lambda\mathbf{I}\right)^{-1}\mathbf{x}^T\mathbf{y} \tag{4}$$

Where $\mathbf{I}$ is an identity matrix of the same dimension as $\mathbf{x}^T\mathbf{x}$.

Mapping $\boldsymbol{\beta}$ to nonlinear space, the variable to be solved is transformed from $\boldsymbol{\beta}$ to $\boldsymbol{\omega}$, and according to the kernel technique mentioned in [17], Eq. (4) is

$$\boldsymbol{\omega} = (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{y} \tag{5}$$

Where $\mathbf{K}$ is kernel matrix.

The kernel matrix$\mathbf{K}$ is a symmetric matrix, which can be diagonalized. The Eq. (5) by the Discrete Fourier Transform (DFT) can implement time domain convolution to frequency domain Hadamard product, reducing the computational complexity by a margin. Then Eq. (5) expresses

$$\widehat{\omega} = \frac{\widehat{\mathbf{y}}}{\widehat{\mathbf{K}} + \lambda} \tag{6}$$

where hat^ represents the corresponding DFT vector of the matrix and$\widehat{\mathbf{K}}$ is the Gaussian kernel matrix.

In order to preserve the information of the previous frames, the model is updated using a linear interpolation method,

$$\widehat{\omega}_t = \begin{cases} \widehat{\omega} & , t = 1 \\ (1-\eta)\widehat{\omega}_{t-1} + \eta\widehat{\omega}, t > 1 \end{cases} \tag{7}$$

Where$\widehat{\omega}_t$ is the current frame template and$\widehat{\omega}_{t-1}$ is the previous frame template. $\eta \in [0, 1]$ is the learning rate.

The CF tracker's time domain response map of the current frame is

$$\mathbf{r}_t = \mathscr{F}^{-1}\left(\widehat{\omega}_t \odot \widehat{\mathbf{K}}\right) \tag{8}$$

Where$\odot$ is the Hadamard product.

The detailed derivation formulas can be found in [16, 17].

3 Our approach

In this section, we first elaborate the proposed feature response map peak detection method and demonstrate its effect. Next, we deduce a novel confidence function which used to achieve the selective update of the model. Thirdly, the adaptive update discriminating mechanism is proposed to break the tradition of discriminating using a fixed discriminant threshold. In the end, we present a new and simple target multi-feature adaptive merge method.

3.1 Primary and secondary peaks detection

In the ideal case, the CF tracker obtained by Eq. (8) has only one peak in the time domain response, as visualized in Fig. 1(b). However, in the complex scenes of OCC, SV, IV and etc., the time domain response maps are introduced unwanted multi-peak (see Fig. 1(d)).

In the$(t + 1)^{th}$ frame, the input sample$\mathbf{x}_{t+1}$ is traversed using a cyclic matrix, instead of a sliding window, to generate an image segment data set$\mathbf{S}_{\mathbf{t}+1}$, and the image segments in$\mathbf{S}_{\mathbf{t}+1}$ is matched with the model$\boldsymbol{\beta}_t$ so that the matching value function$f(\mathbf{s}, \boldsymbol{\beta}_t)$ is maximized, that is

$$\mathbf{s}_{t+1} = \operatorname{argmax}_{\mathbf{s}\in\mathbf{S}_{t+1}} f(\mathbf{s}, \boldsymbol{\beta}_t) \tag{9}$$

However, when there exists a secondary peak and its value is larger, it is updated into the model$\boldsymbol{\beta}_t$, and when the prediction of the$(t + 1)^{th}$ frame is performed, the image segments$\mathbf{s}^*_{t+1}$ may be matched,

$$\mathbf{s}^*_{t+1} = \operatorname{argmax}_{\mathbf{s}\in\mathbf{S}_{t+1}} f(\mathbf{s}, \boldsymbol{\beta}_t) \tag{10}$$

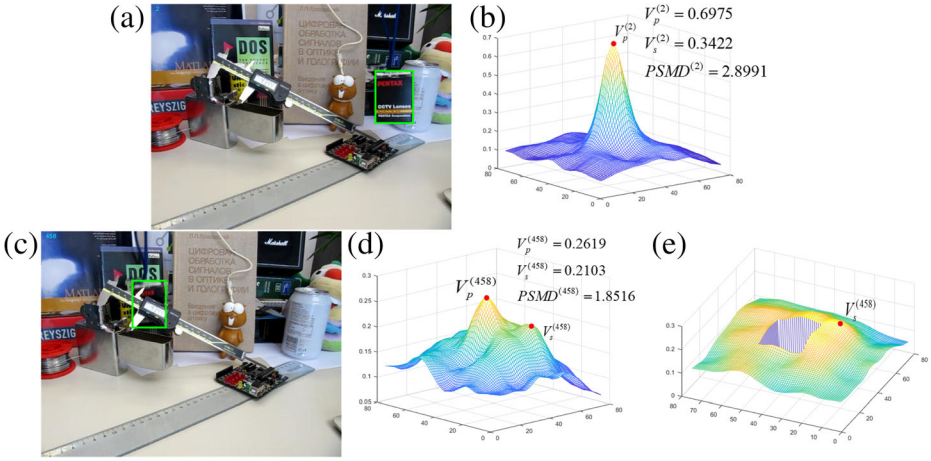

Fig. 1 Response map of different scenes. **a** tracking result **b** response map for frame 2 of box sequence, **c** tracking result **d** response map **e** Secondary Peak response map for frame 458 of box sequence

Compared with$\mathbf{s}_{t+1}$, there is

$$f\left(\mathbf{s}^*_{t+1}, \boldsymbol{\beta}_t\right) > f\left(\mathbf{s}_{t+1}, \boldsymbol{\beta}_t\right) \tag{11}$$

This will result in a tracking offset, and the worst is that the target will be lost. Therefore, it is important to perform primary and secondary peaks detection on the time domain response of the CF tracker.

For the time domain response$\mathbf{r}_t$, the detection of the primary peak's position is relatively easy. Extract the maximum response layer in$\mathbf{r}_t$,

$$l = \text{argmax}_{w \in W, h \in H, l \in L} \mathbf{r}_t(w, h, l) \tag{12}$$

Where W, H and L are the width, height and the number of layers of $\mathbf{r}_t$, respectively.

Then$\mathbf{r}_l^{(t)} \in \mathbf{r}_t$ is the maximum response layer of $\mathbf{r}_t$. The time domain response map's primary peak is located in the domain response maximum response layer,

$$\left[l_{lw}^{(t)}, l_{lh}^{(t)}\right] = \text{argmax}_{lw \in W, lh \in H}\left[\mathbf{r}_l^{(t)}(lw, lh)\right] \tag{13}$$

where $\left[l_{lw}^{(t)}, l_{lh}^{(t)}\right]$ is the primary peak's position of $\mathbf{r}_t$.

Different from the [25], the mask matrix$\boldsymbol{\psi}$ is a binary matrix with the same dimension as $\mathbf{r}_l^{(t)}$ of the time domain response$\mathbf{r}_t$. The elements at the location of the rectangular region centered on $\left[l_{lw}^{(t)}, l_{lh}^{(t)}\right]$ and having the length of the $2N \times 2N$ side are set to 0, while others are set to 1. N is a hyperparameter with a value range of [2, 15].

The secondary peak's response map matrix is the Hadamard product of the maximum response layer $\mathbf{r}_l^{(t)}$ and the mask matrix $\boldsymbol{\Psi}$,

$$\mathbf{r}_s^{(t)} = \mathbf{r}_l^{(t)} \odot \Psi \tag{14}$$

Here, $\mathbf{r}_s^{(t)}$ is the secondary peak's response map matrix (see Fig. 1(e)).

The time domain response map's secondary peak is located in $\mathbf{r}_s^{(t)}$,

$$\left[l_{sw}^{(t)}, l_{sh}^{(t)}\right] = \text{argmax}_{sw \in W, sh \in H}\left[\mathbf{r}_s^{(t)}(sw, sh)\right] \tag{15}$$

Here, $\left[l_{sw}^{(t)}, l_{sh}^{(t)}\right]$ is the secondary peak's position of $\mathbf{r}_t$.

3.2 Confidence function

The traditional models update each frame. Although this method can continuously learn according to the target's appearance characteristics, it is also easy to cause pollution of the model. Reference [4] proposes to use the PSR for peak intensity detection, but the effect is not good. Reference [25] proposed the APCE confidence function, but this function is mainly concerned with the average value, while ignoring the most influence on the primary peak is the secondary peak of $\mathbf{r}_t$.

In the t^{th} frame, in order to have a unified evaluation criterion for the confidence function, we average the maximum response layer $\mathbf{r}_l^{(t)}$ of $\mathbf{r}_t$,

$$V_m^{(t)} = \frac{1}{W \cdot H} \sum_{lw=1, lh=1}^{W,H} \left[\mathbf{r}_l^{(t)}(lw, lh)\right] \tag{16}$$

Where $V_m^{(t)}$ is the mean of the maximum response layer $\mathbf{r}_l^{(t)}$of $\mathbf{r}_t$.

The maximum response value $V_p^{(t)}$ of $\mathbf{r}_t$ is

$$V_p^{(t)} = \max_{w \in W, h \in H, l \in L} \mathbf{r}_t(w, h, l) \tag{17}$$

In the maximum response layer $\mathbf{r}_l^{(t)}$, the maximum response value $V_s^{(t)}$ other than $V_p^{(t)}$ is

$$V_s^{(t)} = \max_{w \in W, w \neq lw, h \in H, h \neq lh} \mathbf{r}_l^{(t)}(w, h) \tag{18}$$

The new confidence function proposed in this paper is called Primary and Secondary Peak Mean Difference Ratio (PSMD), and its expression is

$$PSMD^{(t)} = \frac{V_p^{(t)} - V_m^{(t)}}{\left|V_s^{(t)} - V_m^{(t)}\right|} \tag{19}$$

Here, $|\cdot|$ is the symbol for absolute value.

Figure 1(b) and Fig. 1(d) show the PSMD of the frames in different scenarios, respectively.

3.3 Adaptive update discriminating mechanism

Using the confidence function to perform model update discrimination, both [4, 25] use a fixed discriminant threshold, which is not good for processing different tracking sequences. In this paper, we get adaptive update discrimination threshold according to different sequences, which is better robustness.

Since the initial frame in the tracking is manually specified, image processing is generally not performed on the initial frame. The tracking effect of the second frame is optimal. In this paper, we use the data of the second frame to find the adaptive update threshold.

The adaptive update discrimination thresholds t_{PSMD} of the confidence function PSMD is

$$t_{PSMD} = \left(\frac{V_p^{(2)} - V_m^{(2)}}{\left| V_s^{(2)} - V_m^{(2)} \right|} \right) / \mu \tag{20}$$

Where μ is a hyperparameter with a range of [1, 11].

It is not enough to just update the model with a confidence function. Since the value of PSMD is still large in the early stages of complex scenes such as occlusion, if the model is still updated, it will gradually lead to the model to be polluted. In the case, the maximum response value will change significantly. Therefore, based on the PSMD, we increase the maximum response value adaptive update threshold $t_{\max}$,

$$t_{\max} = V_p^{(2)} / \nu \tag{21}$$

Where ν is a hyperparameter with a range of (0, 5].

The model judgment criteria $V_p^{(t)}$ and $PSMD^{(t)}$ are respectively compared with the adaptive update threshold to obtain the update flag $update_{flag}$, that is

$$\begin{cases} update_{flag} = 1, & if \quad PSMD^{(t)} \geq t_{PSMD} \,\&\, \quad V_p^{(t)} \geq t_{\max} \\ update_{flag} = 0, & others \end{cases} \tag{22}$$

If $update_{flag} = 1$, we use linear interpolation to update the model, the current frame is updated to the model while retaining the previous frame. If $update_{flag} = 0$,the model is not updated.

3.4 MDRCF

We propose a novel CF tracker called MDRCF. This algorithm uses only the hand-crafted features which are HOG and CN and doesn't use depth features.

It should be noted that when the tracking sequences are grayscale, using the CN will increase the unnecessary computational cost. In this paper, when the sequences are grayscale, the intensityChannelNorm6 (IC) [11] is used, which is a simplified CN.

In order to ensure the real-time performance of the algorithm, we employ the fast Discriminative Scale Space Tracking (fDSST) method proposed in [12] to solve the scale change problem in our tracker.

Algorithm1: Our MDRCF approach, iteration at time step t.

Input:

Frame x_t.

Previous target position p_{t-1}.

Model ω_{t-1}.

Output:

Estimated target position p_t.

Updated model.

Frame=1:

Compute model ω_1 **with HOG and CN.**

Frame=2:

Compute adaptive threshold t_{PSMD} **and** $t_{\max}$ **using (20) and (21), respectively.**

Frame ≥ 2**:**

1.Extract sample s **from** x_t **at** p_{t-1}.

2.Compute correlation scores $r_{HOG}^{(t)}$ **and** $r_{CN}^{(t)}$.

3.Compute merge response.

4.Compute the target scale using fDSST.

5.Set P_t **to the target position that maximizes** r_t.

6.Compute $V_m^{(t)}, V_p^{(t)}$ **and** $V_s^{(t)}$ **using (16), (17) and (18).**

7.Compute confidence score $PSMD^{(t)}$ **using (19).**

8.Model update

8.1 If $PSMD^{(t)} \geq t_{PSMD} \,\&\, V_p^{(t)} \geq t_{\max}$**, set** ω_t **to the target model.**

8.2 If $PSMD^{(t)} < t_{PSMD} \,|\, V_p^{(t)} < t_{\max}$**, set** ω_{t-1} **to the target model.**

Hyperparameter N in Section 3.1 and hyperparameters μ and ν in Section 3.2 can be calculated offline. We make use of VOT2018 [18] as the training set and obtain three hyperparameters through training.

Figure 2 is the framework of our algorithm, and algorithm 1 is a brief overview of our algorithm.

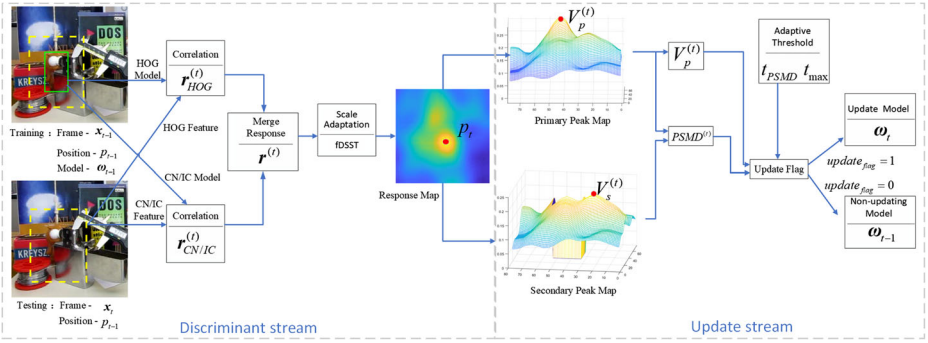

Fig. 2 The overall architecture of the proposed model. The whole model contains two streams: Discriminant stream and update stream. The tracker extracts the HOG feature and CN feature of the $t-1$ frame image and the t frame image, and performs correlation matching. Using the multi-feature fusion mechanism to perform feature fusion on the two features, and then we use the fast discriminative scale space tracking which introduced by fDSST to solve the problem of scale changes in the tracking process. Using Eq. (17) and Eq. (18) to obtain the primary and secondary peak values V_p and $V_s$of the feature map, respectively. We use Eq. (19) to obtain the PSMD value. The template adaptive update discrimination thresholds are obtained from Eq. (20) and Eq. (21), which are $t_{\max}$ and t_{PSMD}. Comparing with the PSMD value and the primary peak value V_p of the response map, we use Eq. (22) to perform the template adaptive update judgment

3.5 Multi-feature adaptive merge method

Taking the Staple tracker as an example, it uses the target HOG feature and color histogram feature to perform feature response merge,

$$\boldsymbol{r} = (1-\lambda)\boldsymbol{r}_{cf} + \lambda \boldsymbol{r}_{ch} \tag{23}$$

Where $\mathbf{r}_{cf}$ is HOG response and $\mathbf{r}_{ch}$ is color histogram response. λ is the merge factor which is 0.3 in Staple.

In practice, the representation ability of HOG is better than the color histogram, so more attention needs to be paid to the HOG feature, it must be guaranteed that

$$(1-\lambda) > \frac{1}{2} \tag{24}$$

When the target probability mean value α in the target color histogram feature is larger, it indicates that the target color feature and the background color feature have obvious differences at this time, so the attention to $\mathbf{r}_{ch}$ should be appropriately increased. However, when background color interference occurs, α will also increase, but at this time, it is necessary to reduce the attention to $\mathbf{r}_{ch}$ as much as possible to avoid updating the background information into the model.

Therefore, the merge factor λ is solved using a piecewise function. To ensure the simplicity and smoothness of the model, an exponential adaptive merge factor is used,

$$\lambda^{(t)} = \begin{cases} \exp\left(\frac{1}{n_\tau}\sum_{\tau=1}^{n_\tau}\alpha_\tau^{(t)}\right) - \phi & ,\alpha^{(t)} < \hbar \\ \exp\left(-\frac{1}{n_\tau}\sum_{\tau=1}^{n_\tau}\alpha_\tau^{(t)}\right)/\varepsilon & ,\alpha^{(t)} \geq \hbar \end{cases} \tag{25}$$

Where $\alpha_\tau^{(t)}$ is the probability value that pixel τ is the target in the t^{th} frame, and n_τ is the number of pixels in the response matrix. $\hbar$ is the discriminant threshold and is a hyperparameter. ϕ and ε are hyperparameters.

4 Experiments

In this section, the experimental parameters and evaluation criteria are given first. Furthermore, the confidence function proposed in this paper is compared with other confidence functions in the current tracking domain to evaluate the performance of the confidence function. Thirdly, the proposed algorithm (MDRCF) and SOTA CF algorithms are quantitatively and qualitatively compared and analyzed. Finally, we evaluate the performances of the EAMStaple proposed in this paper.

4.1 Implementation details

The software platform of this experiment is MATLAB R2017b. The hardware platform configuration is a desktop computer with Intel (R) Core (TM) i7-4790CPU@3.60GHz and 12GB RAM. The experimental hyperparameters of the algorithm are shown in Table 1.

The evaluation datasets are OTB2013, OTB100 and TC128. The OTB2013 contains 51 sequences which contain 11 attributes. The OTB100 includes 100 sequences, which are more

Table 1 Experimental hyperparameters

Parameters	Values
N	10
μ	1.06
ν	0.94
$\hbar$	0.38
ϕ	1.09
ε	2

complex and difficult than the OTB2013. The TC128 has 128 color sequences which are rich in colors. The above benchmark datasets are classic datasets in the field of object tracking and contain most of the situations that can be encountered in the process of tracking. At the same time, considering that most related trackers for performance comparison with MDRCF perform performance tests on the above datasets, we have selected the above three benchmark datasets to test our tracker performance.

The experiments in this paper are based on the one-pass evaluation (OPE) protocol in [29] for quantitative analysis, including precision and area under the curve (AUC). In order to ensure fairness, the precision threshold and AUC threshold of the quantitative analysis are 20 pixels and 0.5 respectively, and all algorithms don't use the target's depth features.

4.2 Comparison of confidence function

The current confidence function in object tracking are PSR proposed by Bolme et al. [4] and APCE proposed by Wang et al. [25]. Based on the Staple algorithm, the confidence function

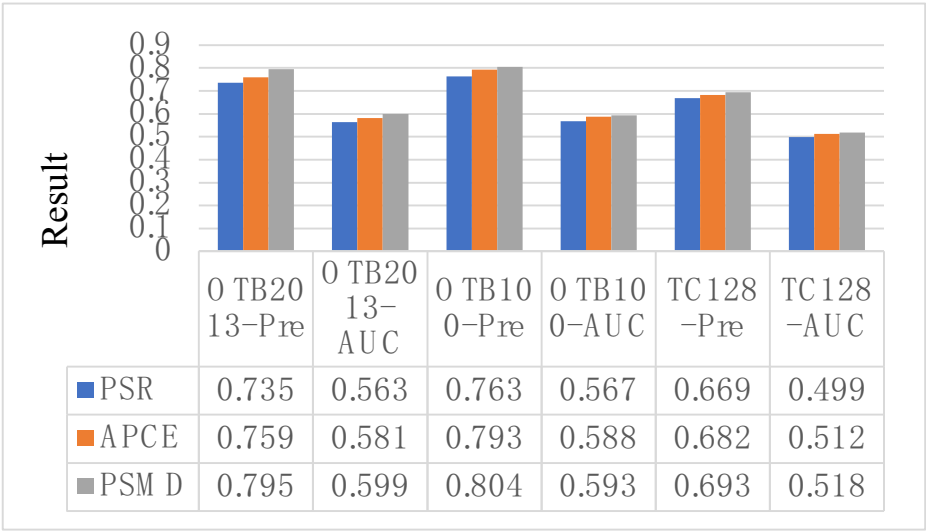

Fig. 3 Tracking result of three confidence functions on different dataset. Notes: Since the four SOTA algorithms we selected in the current CF trackers are excellent, the earlier CF trackers [4, 6, 16, 17] are no longer considered. LMCF doesn't disclose the source code, so we use the experiment data in its paper for comparison. Our tracker uses hand-crafted features of the tracked target instead of depth features, and the tracker does not need to be pre-trained for large datasets, so the manuscript only compares the performance with SOTA trackers using hand-crafted features

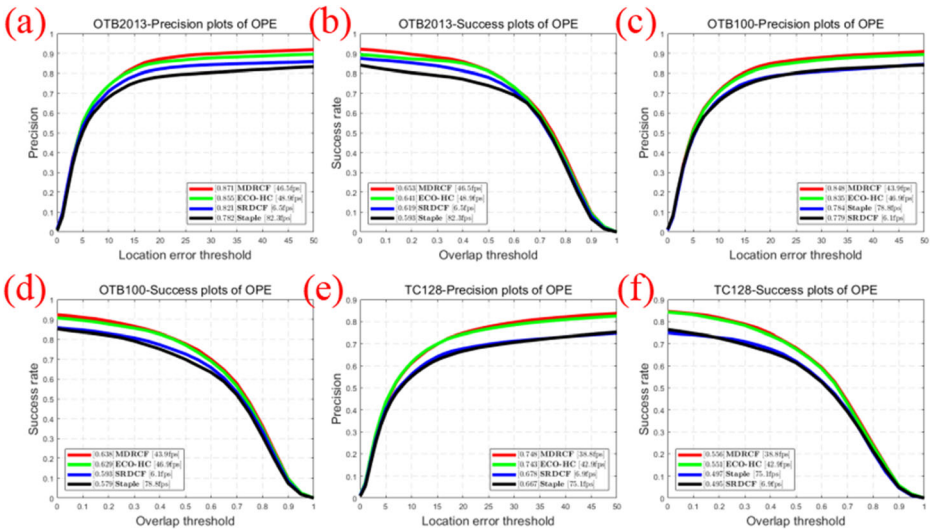

Fig. 4 Tracking result of four trackers. **a** precision **b** AUC for OTB2013, **c** precision **d** AUC for OTB100, **e** precision **f** AUC for TC128

PSMD proposed in this paper is compared with PSR and APCE. Figure 3 shows the precision and AUC values for the three confidence functions based on Staple on different datasets.

It can be seen from Fig. 3 that the PSMD confidence function is the best both the precision and the AUC evaluation criteria on three datasets compared with the PSR and APCE. It can be proved that the performance of the confidence function proposed in this paper is superior.

Compared with PSR and APCE, the confidence function of PSMD focuses on the secondary peak that has the greatest impact on the primary peak, rather than performing simple sidelobe detection as PSR and multi-peak detection as APCE. When the difference between the primary peak value and the secondary peak value of the current frame is small, it can be proved that the current image does not meet the conditions for template update, so the template is not updated. Because the difference between image frames is small during the tracking process, deleting the damaged image frames will not only damage the tracker, but also prevent the model from being contaminated.

Table 2 A comparison with SOTA trackers on the OTB2013, OTB100 and TC128 datasets. The entries in bold red denote the best results and the ones in italic blue indicate the second best. 'No' means no result

Trackers Datasets		Staple	SRDCF	LMCF	ECO-HC	MDRCF
OTB 2013	Precision	0.782	0.821	0.839	*0.855*	**0.871**
	AUC	0.593	0.619	0.624	*0.641*	**0.653**
OTB 100	Precision	0.784	0.779	NO	*0.835*	**0.848**
	AUC	0.579	0.593	0.568	*0.629*	**0.638**
TC 128	Precision	0.667	0.678	NO	*0.743*	**0.748**
	AUC	0.497	0.495	NO	*0.551*	**0.556**

Table 3 Tracking results (in %) of four trackers Staple, SRDCF, ECO-HC and MDRCF on 11 attributes for OTB2013, OTB100 and TC128 datasets. The entries in bold red denote the best results and the ones in italic blue indicate the second best

Attributes	Trackers	IV	OPR	SV	OCC	DEF	MB	FM	IPR	OV	BC	LR
Staple	OTB 2013	56.1	56.9	54.5	58.5	60.7	52.6	50.1	*57.6*	51.8	55.7	*39.6*
SRDCF		56.3	59	57.3	61.5	*63.7*	*56.8*	54.6	55.3	55.5	57.1	**47.1**
ECO-HC		*58.9*	*61.7*	*60.2*	*64.5*	62.7	56.6	*60*	56.6	**69**	*59.8*	37.1
MDRCF		**63.9**	**63.4**	**62.3**	**64.8**	**67.7**	**61.3**	**60.9**	**59.2**	*67.6*	**61.2**	36.6
Staple	OTB 100	59.5	53.4	52	54.3	55	54	54.1	54.9	47.6	56.1	39.9
SRDCF		60	54.2	55.6	55	54.5	58.1	58.8	53.4	46.1	57.1	**51.4**
ECO-HC		*61.4*	*58.8*	*59.6*	*60.1*	*58.9*	*60.8*	**61.9**	*55.3*	*56.5*	*62.7*	*49.9*
MDRCF		**66.3**	**60.7**	**60.6**	**60.9**	**60.5**	**63.8**	*61.8*	**58.1**	**59**	**64.3**	49.5
Staple	TC 128	52	49.4	48.8	46.3	55	41.7	48.1	47	38	49	35.6
SRDCF		51.7	45.8	48.6	48.2	54	*41.9*	45.4	45.8	39.2	49.6	37.6
ECO-HC		*54.1*	*51.5*	*52.5*	*54.2*	*55*	**45.1**	*50.2*	*50.6*	**47**	*56.5*	**49.2**
MDRCF		**57.8**	**52.9**	**52.6**	**54.7**	**57.9**	**45.1**	**50.5**	**52.5**	*46.7*	**57.6**	*46.6*

4.3 Comparison of MDRCF and SOTA CF trackers

In order to verify the performance of the proposed MDRCF, we select four SOTA CF trackers which are Staple, SRDCF, LMCF and ECO-HC to compare with our algorithm and perform quantitative and qualitative analysis.

First, quantitative analysis was performed on five trackers. According to the precision plots and AUC plots of the four trackers in OTB2013, OTB100 and TC128, the proposed algorithm has better performance than the SOTA trackers (see Fig. 4).

Table 2 also shows the precision values and AUC values of the five trackers in the three datasets and MDRCF performs better in the three datasets. Compared with ECO-HC, the precision values of MDRCF on the three datasets are 1.87%, 1.56% and 0.67% higher respectively, and the AUC values are 1.87%, 1.43% and 0.91% higher.

To further quantitatively compare the four trackers, Table 3 exhibits the AUC values of the four trackers on different video attributes of the three datasets. It can be seen that MDRCF has better performance on the 11 attributes of three datasets, especially in IV, OPR, SV, OCC, DEF, MB, IPR and BC. However, MDRCF performs poorly on the video attribute of LR. Because of the low-resolution image, the MDRCF tracker will consider it inappropriate to update the template, resulting in poor tracking results.

From the quantitative analysis data of Fig. 4, Table 2 and Table 3, it can be seen that ECO-HC and MDRCF perform better, so we apply the OTB100 dataset to conduct qualitative analysis on the trackers.

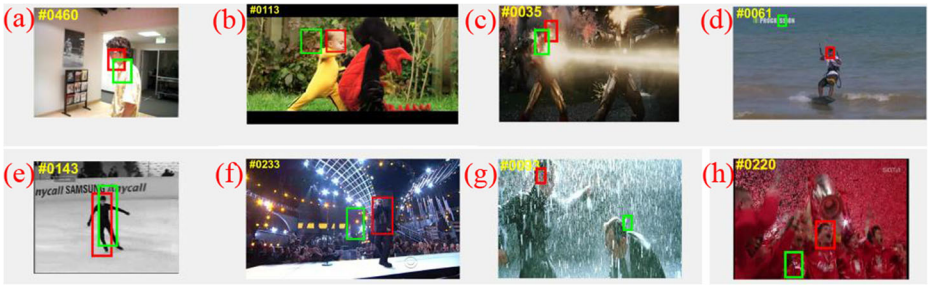

Fig. 5 Qualitative comparison of ECO-HC and MDRCF on 8 sequences. Red is MDRCF, green is ECO-HC. **a** david **b** DragonBaby **c** ironman **d** KiteSurf **e** Skater **f** matrix **g** singer2 **h** soccer

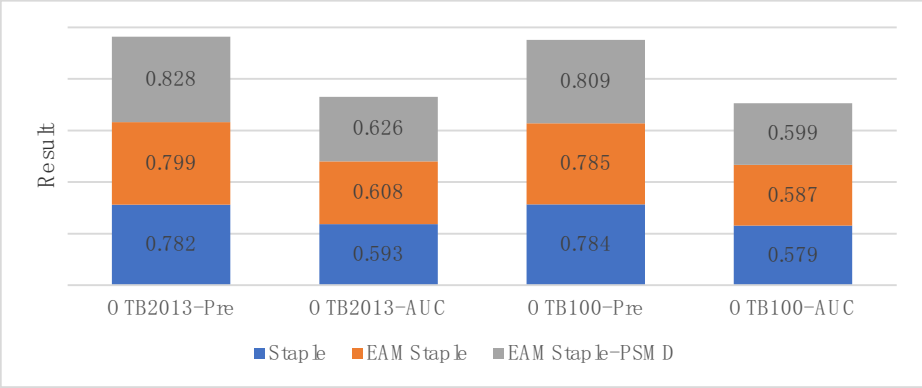

Fig. 6 Tracking result of three Trackers on different datasets

The performance of the MDRCF is superior to that of the ECO-HC, which illustrates the effectiveness of the proposed tracker. Our tracker performs better on different sequences (see Fig. 5), which indicates that the proposed algorithm is robust.

4.4 Exponential adaptive merge

Based on the Staple algorithm, we test the exponential adaptive merge method proposed in this paper. Figure 6 shows the precision and AUC values of the Staple, EAMStaple and EAMStaple-PSMD on OTB2013 and OTB100, where EAMStaple-PSMD is an exponential adaptive merge method with confidence function added. Table 4 shows the Frame Per Second (FPS) of Staple, EAMStaple and EAMStaple-PSMD on OTB2013 and OTB100 datasets.

The EAMStaple presented in this paper performs better (see Fig. 6), and the precision and AUC are improved on different datasets.

It can be seen from Table IV that the EAMStaple proposed in this paper doesn't cut down the FPS performance of the tracker, and even improves the FPS of the tracker to some extent.

The data in Fig. 6 and Table IV also reflect that the new confidence function proposed in this paper has good robustness and better effect.

5 Conclusion

In this paper, we propose a novel method for detecting the primary and secondary peaks of the characteristic response map. At the same time, a new confidence function proposed uses the adaptive update discriminant method on the discriminant mechanism to replace the traditional method of using a fixed threshold. The experimental results on multiple datasets demonstrate that both methods perform well. Furthermore, based on the proposed methods above, we

Table 4 FPS results for Staple, EAMStaple and EAMStaple-PSMD on the OTB2013 and OTB100 datasets

Trackers Datasets	Staple	EAMStaple	EAMStaple-PSMD
OTB2013	82.3	*83.1*	**84.1**
OTB100	78.7	*79.1*	**80.3**

propose a new robust CF tracker – MDRCF that uses hand-crafted features to improve model drift in complex scenes. Experiments indicate that MDRCF has better performances than SOTA CF trackers. Finally, we explore a simple multi-feature adaptive merge method which yield a good effect.

Although the tracker proposed in this paper performs well, there are still many needs to be improved: we solve the scale variation using the scale solution in [12], its performance is not optimal, and the follow-up work will find a solution with good performance. When searching for adaptive update discriminant thresholds, we don't use more complex functions to ensure the simplicity of the algorithm, but the cost is that the performances of the tracker are sub-optimal, so we will look for a more appropriate method in the future. We consider testing the tracker performance by replacing the hand-crafted features with deep features in future work.

Acknowledgements The authors acknowledge the Chinese Academy of Sciences STS Projects. (Grant: 2019 T31020008, 2019 T31020010), the Fujian Provincial Department of Science and Technology Project (Grant: cyzx201805063).

References

1. Bertinetto L, Valmadre J, Golodetz S, Miksik O, Torr PHS (2016) Staple: Complementary Learners for Real-Time Tracking. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 27–30 June 2016. pp 1401–1409. doi:https://doi.org/10.1109/CVPR.2016.156
2. Bertinetto L, Valmadre J, Henriques JF, Vedaldi A, Torr PHS (2016) Fully-convolutional Siamese networks for object tracking. arXiv e-prints
3. Bhat G, Johnander J, Danelljan M, Khan FS, Felsberg M (2018) Unveiling the Power of Deep Tracking. In, Cham, Computer Vision – ECCV 2018. Springer International Publishing, pp 493–509
4. Bolme DS, Beveridge JR, Draper BA, Lui YM (2010) Visual object tracking using adaptive correlation filters. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 13–18 June 2010. pp 2544–2550. doi:https://doi.org/10.1109/CVPR.2010.5539960
5. Chen B, Li P, Sun C, Wang D, Yang G (2019) Lu H (2019) multi attention module for visual tracking. Pattern Recogn 87:80–93. https://doi.org/10.1016/j.patcog.2018.10.005
6. Danelljan M, Häger G, Khan F, Felsberg M (2014) Accurate Scale Estimation for Robust Visual Tracking. In: Proceedings of the British Machine Vision Conference 2014, 1–5 September 2014. pp 65.61–65.11. doi:https://doi.org/10.5244/C.28.65
7. Danelljan M, Khan FS, Felsberg M, Weijer JVD (2014) Adaptive Color Attributes for Real-Time Visual Tracking. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 23–28 June 2014. pp 1090–1097. doi:https://doi.org/10.1109/CVPR.2014.143
8. Danelljan M, Häger G, Khan FS, Felsberg M (2015) Convolutional Features for Correlation Filter Based Visual Tracking. In: 2015 IEEE International Conference on Computer Vision Workshop (ICCVW), 7–13 Dec. 2015. pp 621–629. doi:https://doi.org/10.1109/ICCVW.2015.84
9. Danelljan M, Häger G, Khan FS, Felsberg M (2015) Learning Spatially Regularized Correlation Filters for Visual Tracking. In: 2015 IEEE International Conference on Computer Vision (ICCV), 7–13 Dec. 2015. pp 4310–4318. doi:https://doi.org/10.1109/ICCV.2015.490
10. Danelljan M, Robinson A, Shahbaz Khan F, Felsberg M (2016) Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking. In, Cham, Computer Vision – ECCV 2016. Springer International Publishing, pp 472–488
11. Danelljan M, Bhat G, Khan FS, Felsberg M ECO (2017) Efficient Convolution Operators for Tracking. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 21–26 July 2017. pp 6931–6939. doi:https://doi.org/10.1109/CVPR.2017.733
12. Danelljan M, Häger G, Khan FS, Felsberg M (2017) Discriminative scale space tracking. IEEE Trans Pattern Anal Mach Intell 39(8):1561–1575. https://doi.org/10.1109/TPAMI.2016.2609928
13. Fan H, Ling H (2018) Siamese cascaded region proposal networks for real-time visual tracking. arXiv e-prints
14. Ge B, Zuo X, Hu Y (2018) Review of visual object tracking technology. J Image Graph 23(08):1091–1107

15. Hare S, Saffari A, Torr PHS (2011) Struck: Structured output tracking with kernels. In: 2011 International Conference on Computer Vision, 6–13 Nov. 2011. pp 263–270. doi:https://doi.org/10.1109/ICCV.2011.6126251
16. Henriques JF, Caseiro R, Martins P, Batista J (2012) Exploiting the Circulant Structure of Tracking-by-Detection with Kernels. In, Berlin, Heidelberg. Computer vision – ECCV 2012. Springer Berlin Heidelberg, pp 702–715
17. Henriques JF, Caseiro R, Martins P, Batista J (2015) High-speed tracking with Kernelized correlation filters. IEEE Trans Pattern Anal Mach Intell 37(3):583–596. https://doi.org/10.1109/TPAMI.2014.2345390
18. Kristan M, Leonardis A, Matas J, Felsberg M, Pfugfelder R, Zajc LC, Vojir T, Bhat G, Lukezic A, Eldesokey A, Fernandez G, et al. (2018) The sixth Visual Object Tracking VOT2018 challenge results.
19. Li Y, Zhu J, Hoi SCH, Song W, Wang Z, Liu H (2017) Robust estimation of similarity transformation for visual object tracking. arXiv e-prints
20. Li B, Wu W, Wang Q, Zhang F, Xing J, Yan J (2018) SiamRPN++: evolution of Siamese visual tracking with very deep networks. arXiv e-prints
21. Liang P, Blasch E, Ling H (2015) Encoding Color Information for Visual Tracking: Algorithms and Benchmark. IEEE Trans Image Process 24:–5644. https://doi.org/10.1109/TIP.2015.2482905
22. Ma C, Huang J-B, Yang X, Yang M-HJIJoCV (2018) Adaptive Correlation Filters with Long-Term and Short-Term Memory for Object Tracking 126 (8):771–796. doi:https://doi.org/10.1007/s11263-018-1076-4
23. Nam H, Han B (2015) Learning multi-domain convolutional neural networks for visual tracking. arXiv e-prints
24. Valmadre J, Bertinetto L, Henriques JF, Vedaldi A, Torr PHS (2017) End-to-end representation learning for correlation filter based tracking. arXiv e-prints
25. Wang M, Liu Y, Huang Z (2017) Large margin object tracking with Circulant feature maps. arXiv e-prints
26. Wang Q, Gao J, Xing J, Zhang M, Hu W (2017) DCFNet: discriminant correlation filters network for visual tracking. arXiv e-prints
27. Wang M, Su D, Shi L, Liu Y, Valls Miro J (2017) Real-time 3D human tracking for Mobile robots with multisensors. arXiv e-prints
28. Wang Q, Zhang L, Bertinetto L, Hu W, Torr PHS (2018) Fast online object tracking and segmentation: a unifying approach. In: arXiv e-prints, December 01, 2018
29. Wu Y, Lim J, Yang M-H (2013) Online object tracking: a benchmark. In pp 2411–2418. doi:https://doi.org/10.1109/CVPR.2013.312
30. Wu Y, Lim J, Yang M (2015) Object tracking benchmark. IEEE Trans Pattern Anal Mach Intell 37(9):1834–1848. https://doi.org/10.1109/TPAMI.2014.2388226
31. Zhang Z, Peng H, Wang Q (2019) Deeper and wider Siamese networks for real-time visual tracking. arXiv e-prints
32. Zhu Z, Wang Q, Li B, Wu W, Yan J, Hu W (2018) Distractor-Aware Siamese Networks for Visual Object Tracking. In, Cham, Computer Vision – ECCV 2018. Springer International Publishing, pp 103–119