Multimedia Tools and Applications (2020) 79:28431–28452
https://doi.org/10.1007/s11042-020-09240-7

A robust tracker integrating particle filter into correlation filter framework

Weirong Liu[1] • Huiling Gao[1] • Jie Liu[2] • Chaorong Liu[3] • Binshan Li[1] • Xuhui Song[1]

Received: 28 September 2019 / Revised: 16 June 2020 / Accepted: 24 June 2020 /
Published online: 3 August 2020

Abstract

The location and scale filters in discriminative correlation filter methods are lack of accurate rotation representation capability and updated with fixed intervals, which leads to tracking failure and time-consuming in complex scenarios. In this manuscript, a robust tracker integrating particle filter into correlation filter is presented to cope with sharp rotation and remarkable deformation. The target position and scale factor are firstly estimated from the correlation filter, and then the rotation factor is determined by similarity between candidates and template based on the particle filter. As a result, target variation can be accurately described with position, scale and rotation factor. Moreover, a long-time and short-time update scheme is proposed to solve target template drifting problem. Extensive experimental results conducted on OTB-2013, OTB-2015 and VOT-2016 show that the proposed tracker improves the accuracy and robustness of discriminative correlation filter methods.

Keywords Object tracking correlation filter particle filter long-time and short-time update scheme

1 Introduction

Visual tracking estimates the target trajectory from subsequent image sequences, with a given initial position. Visual tracking technology plays an important role in intelligent surveillance systems, intelligent transportation and human-computer interaction, etc.. It is still a challenging

✉ Weirong Liu
liu_weirong@163.com

[1] College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou City, China

[2] National Demonstration Center for Experimental Electrical and Control Engineering Education, Lanzhou University of Technology, Lanzhou City, China

[3] Key Laboratory of Gansu Advanced Control for Industrial Processes, Lanzhou University of Technology, Lanzhou City, China

task due to severe occlusions, illumination variation, fast motion, motion blur and scale variation, although lots of excellent visual trackers have been proposed in recent years.

The existing trackers are classified into discriminative trackers and generative trackers. Discriminative trackers regard the tracking problem as a binary classification problem, and the object is obtained by separating target from background. Target tracking method based on correlation filter is an important branch of discriminative trackers, and attracts many researcher's attention because of its fast speed [3], high precision [13] on tracking benchmarks [25, 38]. In 2010, Bolme et al. [3] firstly applied the correlation theory to target tracking and proposed Minimum Output Sum of Squared Error (MOSSE) filter. Based on the MOSSE filter, a large number of excellent research results have been put forward with different strategies to improve robustness and accuracy of trackers. The KCF tracker [20] extends the single channel grayscale feature to the multi channels features, and the paper [29] replaces the HOG feature with layered convolution feature. Based on dimension reduction, Galoogahi et al. [9, 16, 40] fused different features, such as the HOG, color attributes [14]. The CCOT and ECO [11, 13, 39] introduced deep CNN features [8, 28] to improve the representation ability of features. Danelljan et al. [12, 22, 26, 30] employed scale adaptive to estimate target size. These multi channel and feature fusion can obtain more information of target appearance changes and increase the robustness of visual trackers in complex scenes such as illumination change, posture variation and fast motion. In addition, some works [10, 17, 18] optimize the filter learning effect on boundary constrains. These strategies greatly promote adaptive ability of tracker in different complex scenes. However, there are two obvious shortcomings in the DCF-based tracking methods, (1) the tracker is not able to cope well with sharp rotation and remarkable deformation of because of ignoring the rotation effect on tracking accuracy. (2) most of DCF-based trackers [11, 13, 39] can not be updated in time with object changes owing to the fixed intervals updating mechanism, especially when target has large appearance changes and fast motion. On the other hand, generative trackers [31, 34, 35] implement tracking tasks by searching the most similar candidate to the target from the appearance model space [24]. Generative trackers based on the Particle filter (PF) [4, 7, 34] can express and capture target with rotation or deform, but these methods focus on description of target itself and ignore background information, which tends to drift when target is occluded.

Contributions: The accurate rotation representation mechanism of the PF is introduced into the CF in this manuscript, hence the tracker integrating PF into correlation filter framework (CFPF) can solve the inaccurate tracking problem under target sharp rotation and remarkable deformation. The target position and scale factor are determined by a trained correlation filter based on DCF at the first stage. And then a PF tracking method is introduced to get the best rotation factor on the basis of similarity between template and candidates. In the end the target can be accurately captured by position, scale and rotation factor parameters. In order to solve target template drifting problem, for the update of, we propose a long-time and short-time (LS) update scheme of location filter to adaptively capture target changes. We use a short-time update when the maximum score of the output response is a relatively small value, otherwise we employ a long-time update.

The remainder of this manuscript is organized as follows. In Section 2, we review related work. In Section 3, details of the proposed CFPF tracker including: filter training, target detection, and model update are presented. Experiment results and discussions of parameters can be found in Section 4. Finally, we conclude this manuscript and discuss the future works in Section 5.

2 Related work

2.1 Discriminative correlation filter

The tracking methods based on DCF have attracted many researchers because of high accuracy and efficiency. A correlation filter is learned to distinguish target from background appearance, and the entire tracking process is achieved in the Fourier domain. The tracking task includes three steps. Firstly, the correlation filter is trained by modeling input features as a Gaussian distribution, and then output response is obtained using correlation and location filter. Finally, the peak of output response is used to locate the target.

Bolme et al. [3] initially propounded a MOSSE filter that is robust for appearance changes, and a kernelized tracker [21] was introduced to solve linearly inseparable and nonlinearly separable problems, but these methods are restricted to a single feature channel. The performance of more recent work [9, 16] have shown a notable improvement by expanding single-channel grayscale features to multiple-channel features, but scale change does not attract any attention. If the target scale can't be detected, a lot of background or local target information will be learned, which leads to tracker drifting. The achievements [12, 26] significantly improved tracking accuracy thanks to multi-scale detection. However, the samples with densely sampled were highly redundant. Henriques J.F. et al. [19] replaced dense sampling with cyclic-shift sampling, which introduced boundary effects owing to periodic assumption. Besides situation caused by fast motion is inevitable, Galoogahi et al. [17] proposed the Alternating Direction Method of Multipliers (ADMM) to keep the correct filter size to deal with the problem. Danelljan et al. [10] have proposed the Spatially Regularized DCF (SRDCF) that used a larger detection area without enlarging effective filter size, and added space regularization term to punish filter coefficients to near 0 in the boundary area, which increases discriminative ability of classifier.

2.2 Particle filter

The PF is a sequential importance sampling algorithm based on Bayesian [2]. The PF was firstly used for tracking [23] and offered a unified framework to estimate the posterior probability density function of state variables. To guarantee robustness of tracker, particle sampling combining affine transformation must be important to capture the variations in the state space. Numerous importance sampling techniques of particle filter [5, 32, 41] have been presented to obtain better proposals based on the previous position. The PF estimates posterior probability of target position after getting observations in the following formulas (1) and (2).

$$p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t\text{-}1}|y_{1:t-1})dx_{t-1} \tag{1}$$

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})} \tag{2}$$

where $x_{1:t} = \{x_1, x_2, \dots, x_t\}$ donates all available state vectors with time t variation and $y_{1:t} = \{y_1, y_2, \dots, y_t\}$ represents corresponding observations $p(x_t|x_{t-1})$ is named as a dynamic model, and $p(y_t|x_t)$ is an observation model that estimates the likelihood of observing y_t at state x_t.

3 CFPF tracker

As shown in the Fig. 1, the entire tracking process of our CFPF tracker contains two steps. We use correlation filter to define the location and size of the object in the first stage (indicated by CF in the orange rectangle box), and then we use PF method to detect the rotation scale of the object in the second stage (indicated by PF in the green rectangle box). Combining traditional features HOG (Histogram of Oriented Gradient, HOG), CN (Color Name, CN) and deep CNN features, the correlation filter is trained for position and scale. During tracking process for each frame, a long-time and short-time update scheme (shown in Fig. 2) and the two target templates update scheme (shown in Fig. 3) are proposed to update the CF and PF model.

3.1 Training of location and scale filter

In the CF of Fig. 1, we first construct a correlation filter in the target tracking stage, the location and scale are detected simultaneously by ECO [13]. A large number of target tracking algorithms have been proposed to achieve fast and accurate scale estimation. We learn the training of location and scale filter from ECO [13]. We combine traditional features HOG, CN and deep CNN features. The HOG feature represents the structural features of the gradient and describes local shape information. The quantization of position and direction space can control the influence of translation and rotation to some extent, and a normalized histogram in a local area can partially offset the effects of illumination. The paper [13] has drawn that shallow layers contain more low-level features but spatial resolution is high, while features from deep convolutional layers are discriminative and possess more semantic information, namely high-level visual information. Therefore, the shallow features are beneficial to target position. By a summary of the performance of these models on the ILSVRC 2012 validation data, we select the imagenet-vgg-m-2048 net [6] and the outputs of the first and fifth layers are selected as features [13].

If s_n is the total number of scales, every scale level can be expressed as $\upsilon\in\left\{\left[-\frac{s_n-1}{2}\right], \ldots, \left[\frac{s_n-1}{2}\right]\right\}$. A relative scale factor is α and the scale change factor between two adjacent expressed as α^{υ}. The scale factor of the previous frame is s_c and the scale factor of the current frame is $\alpha^{\upsilon}s_c$. We extract image patches according to the scale factor $\alpha^{\upsilon}s_c$. Training samples f are made up of i feature maps $f_1, f_2, \ldots f_i$ extracted from the image patches. There are J feature channels $f_i^1, f_i^2, \cdots f_i^J$ for per sample f_i. N_j is the number of spatial samples in each feature layer $f_i^j \in R^{N_j}$, $f_i^j[n]$ donates a function indexed by the discrete spatial variable $n \in \{0, \cdots, N_j - 1\}$, $\Omega = R^{N_1} \times \cdots R^{N_J}$ is sample space. To fuse different spatial resolution features, we introduce an implicit interpolation model. And these features are translated into a continuous spatial domain, the continuous spatial interval of samples is $[0, T) \subset R$. Here, the scalar T is

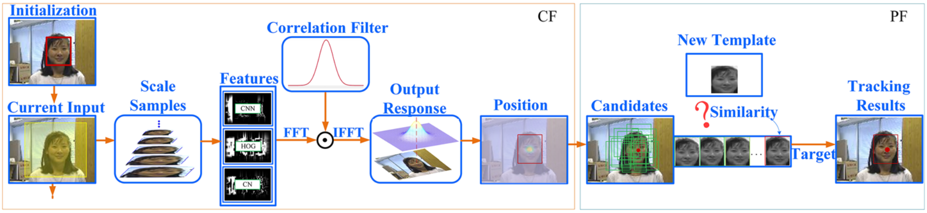

Fig. 1 The tracking flow diagram of the CFPF tracker

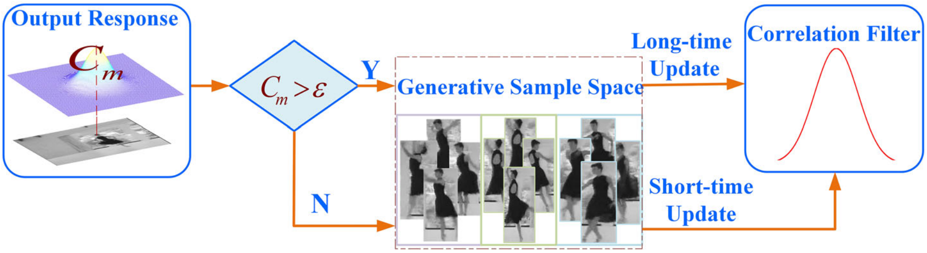

Fig. 2 Long-time and short-time update strategy

arbitrary and it donates scaling of coordinate system. For each feature channel j, the interpolation operator $O_j\{f^j\}(t_p)$is expressed as

$$O_j\{f^j\}(t_p) = \sum_{n=0}^{N_j-1} f^j[n] b_j\left(t_p - \frac{T}{N_j} n\right) \tag{3}$$

where $O_j\{f^j\} \in R^J$represents interpolated feature layer and regarded as a T-periodic function, b_j is an interpolation function with period $T > 0$, t_p stands for the current position. A location filter h is trained by the loss formula (4)

$$E(h) = \sum_{i=1}^{N_s} \alpha_i \left\| C_f\{f_i\} - s_i \right\|_{L^2}^2 + \sum_{j=1}^{J} \left\| \omega h^j \right\|_{L^2}^2 \tag{4}$$

where we provide N_s pairs of samples $\{f_i, s_i\}_1^{N_s}$, α_i are weights of samples, s_i represents a label function that is a Gaussian function defined in the continuous spatial domain. $C_f\{f_i\}$ is actual output response. The last item in formula (4) is a regular term and where ω is regularization weights defined in the entire continuous interval $[0, T)$ [10]. Since samples f_i have J feature channels, we need train a set of filters $h = (h^1, h^2 \ldots h^J)$, where h^j is the continuous filter for feature channel j.

It has shown that many of filters h^j learned incorporate little energy but take more time to calculate [13]. Thus the number of filters can be greatly reduced by Principal Component Analysis (PCA) in the first frame to discard the filters containing less information. The new filters can be expressed as a linear combination $\sum_{p=1}^{P} m_{j,p} h^p$, namely $h = (h^1, h^2 \ldots, h^P)$. The learned coefficients $m_{j,c}$ transform a J-dimensional matrix into a P-dimensional matrix, written as a $J \times P$ matrix $M = (m_{j,p})$. The new filters can be expressed as Mh. The output response is shown in formula (5).

$$C_{Mh}\{f\} = Mh * O\{f\} = \sum_{p,j} m_{p,j} h^p * O_j\{f^j\} = h * M^T O\{f\} \tag{5}$$

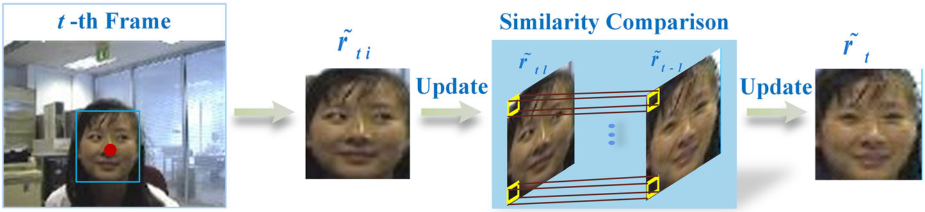

Fig. 3 The updating flow diagram of the object template

We can see that the dimensionality reduction operator about the filter can be transformed to the interpolated feature maps. So, the input of updating filters is the projected feature maps $M^TO\{f\}$ in subsequent frames. The filters h and matrix M can be learned jointly in the first frame, the training process is as follows

$$E(h,M) = \|C_{Mh}\{f\}-s\|_{L^2}^2 + \sum_{p=1}^{P} \|\omega h^p\|_{L^2}^2 + \lambda\|M\|^2 \tag{6}$$

where the last term in formula (6) is defined as L2-norm of M controlled by weight parameterλ. In order to speed up training, the operation of the training filters is carried out in the Fourier domain by Parseval's theorem. $\widehat{O}^j[k] = \widehat{f}^j[k]b_j[k]$ donates the Fourier coefficients of the interpolated feature maps $O_k\{f\}$, and k represents the discrete Fourier transform. The mark $\wedge$ indicates the Fourier coefficients of corresponding variables. The loss (6) in the Fourier domain is derived as

$$E\left(\widehat{h},M\right) = \left\|\widehat{O}^T M\widehat{h}-\widehat{s}\right\|_{l^2}^2 + \sum_{p=1}^{P} \left\|\widehat{\omega}*\widehat{h}^p\right\|_{l^2}^2 + \lambda\|M\|^2 \tag{7}$$

we employ Gauss-Newton and the Conjugate Gradient method in [13] to optimize the quadratic subproblems.

3.2 Target detection

Traditional window-fixed PF tracking algorithms fail to effectively track targets under limited particle number constraints. Especially when the target is in complex scenes, such as occlusion or large deformation, the PF- based tracking methods can not accurately obtain the position and scale of the target. In this manuscript, we use correlation filter trained by scaling samples based on DCF to detect object size and scale, and the method based on PF is used to detect the rotation of the object during the target detection process.

The Fig. 1 shows the entire tracking process of our tracker. It contains two steps. The rectangle box with orange lines indicates the first stage of target detection. As shown by the CF in the box, we use correlation filter method to define the location and size of the object. First of all, different size blocks from the image are cropped based on the center of the previous target and then we extract HOG, CN and CNN features from these blocks. According to Eqs. (5) and (8), the detection score, namely output response, is obtained.

$$\widehat{C}_{Mh}\{f\}[k] = \widehat{O}^T M\widehat{h} = \sum_{j=1}^{J} \widehat{h}^j[k]\widehat{f}^j[k]\widehat{b}_j[k] \tag{8}$$

where $\widehat{C}_{Mh}\{f\}$ is the Fourier coefficients of total detection score for all channels. And the position and scale of the target is obtained by the maximum of detection score in the time domain. As shown by ECO [13], we structure grids to perform rough initial estimation at these positions $c\left(\frac{Tn}{2K+1}\right)$ for $n = 0, \ldots, 2K$, and employ grid search to obtain the peak in the area. Next, the maximum value is treated as initial values of iterative optimization of the Fourier series expansion $c\left(t_p\right) = \sum_{-K}^{K}\widehat{c}[k]e_k\left(t_p\right)$. Similar to other discriminative methods, the score function $c(t_p)$ is defined on the continuous interval $[0, T)$. The target position is determined using the standard Newton's method, and the gradient and Hessian are computed by analytic differentiation of $c(t_p)$. The scale of the object is the sample scale of corresponding to the target position.

The rectangle box with green lines indicates the second stage of target detection. As shown by the PF in the box, we use PF method to detect the rotation factor of the object. First, a target template $\widetilde{r}$ is generated by the affine transformation parameters such as the target position, size and obtained rotation factor. Then according to standard normal distribution, a random number generator is used to obtain a random number matrix that it's dimension is the equal to the number n_p of the particles sampled based on current target position t_p. We combine the random number matrix and initial affine transformation parameters to get the affine parameters of particles with different rotation factors and positions, and different particles based on these particle parameters are generated. The generated particles are the candidates f_c with green boxes in the first image of the PF step in Fig. 1.

Next, for the sake of obtaining rotate factors, we calculate the similarities between candidates f_c and initial object template $\widetilde{r}_i$ instead of probability calculation based on particle filter. The calculation process of the similarity consists of two steps. The first step is calculating the error e_i between candidates and template $\widetilde{r}_i$ using L1-norm. The operator is expressed as

$$e_i = \left\| f_{c,i} - \widetilde{r}_i \right\|_{L^1} \tag{9}$$

where $f_{c,\,i}$ is the i-th candidate. At this time the errors are ranked from small to large. We only select the first few candidates. The second step is obtaining the candidate f_c^* with the highest correlation. For the simple distance calculation can't get the candidate most similar to the template $\widetilde{r}_i$, we calculate the correlation between the template $\widetilde{r}$ and the candidates to get the best candidate f_c^* by formula (10).

$$f_c^* = \arg \max_{f_{c,i}} f_{c,i} \otimes \widetilde{r}_i \tag{10}$$

At this time, the rotation factor of the candidate f_c^* is the most accurate rotation factor. In the end, the target position t_p and size obtained by CF method and the most accurate rotation factor r^* obtained by PF method are regarded as affine parameters to get tracking result of the current frame.

3.3 Model updating

3.3.1 Update location and scale filter by long-time and short-time update mechanism

As shown in Fig. 2, we employ a novel updating mode combined long-time update and short-time update. The leftmost figure is the output response and the c_m represents the maximum response value. The higher the score, the more accurate is the tracking when the score is above a certain threshold. Therefore, we set a threshold ε in accordance with the maximal output response c_m. If the score c_m exceeds the threshold ε, the location filter is updated using long-time update on account of little change of the target. Or else the location and scale filter is updated employing short-time update, because target change is so much that the location filter is difficult to adapt current object status.

We know that collecting samples every frame leads to redundancies. A generative sample space model is used to achieve a compact description of the samples [13]. The approach is based on the joint probability distribution $p(f, s)$ of the samples$f$and corresponding desired output$s$. Location filters h are updated as shown below

$$E(h) = \mathrm{E}\left\{ \left\| O_{h,M}\{f\} - s \right\|_{L^2}^2 \right\} + \sum_{j=1}^{J} \left\| \omega h^j \right\|_{L^2}^2 \tag{11}$$

the expectation E is evaluated over the joint sample distribution $p(f, s)$. Label function s_i in (4) aligns the peak with the object center by shifting the samples f and all $s = s_0$ are identical. We observe that the distribution can be factorized as$p(f, s) = p(f)\delta_{s_0}(s)$, where $\delta_{s_0}(s)$ denotes the Dirac impulse, the joint sample distribution $p(f, s)$ can be obtained by estimating $p(f)$ using a Gaussian Mixture Model (GMM). The location and scale filter updated has better performance due to sample diversity and less redundancy by formula (12).

$$E(h) = \sum_{q=1}^{Q} \pi_q \left\| C_f\{u_q\} - s \right\|_{L^2}^2 + \sum_{j=1}^{J} \left\| \omega h^j \right\|_{L^2}^2 \tag{12}$$

where Q is the number of categories of the sample. The Gaussian means u_q and prior weights π_q directly replace f_i and α_i, respectively, in (4).

3.3.2 Update object template

In order to better integrate PF into the correlation filter framework, the local update strategy is adopted to make the filter adapt to the target change. As shown in the Fig. 3, the two target templates are updated during tracking process for each frame. Firstly, after the object position and size of the current frame is determined, we combine the rotation factor r_{t-1}^* of the t-1-th frame to obtain the initial target template $\widetilde{r}_{ti}$ in t-th frame. The initial target template $\widetilde{r}_{ti}$ is used to track target in t-th frame by formulas (9) and (10) and the rotation factor r_t^* of the t-th frame is obtained. Then the first template $\widetilde{r}_{ti}$ is updated by replacing r_{t-1}^* as r_t^* and we obtain the tracking result $\widetilde{r}_{tl}$. Finally, we use the tracking result $\widetilde{r}_{tl}$ to obtain the final template $\widetilde{r}_t$. As shown in Fig. 3, the first step is the similarity comparison between $\widetilde{r}_{t-1,l'}$ and $\widetilde{r}_{tl,l'}$. The second step is update $\widetilde{r}_{t-1,l'}$ according to formula (13).

$$\widetilde{r}_{t,l'} = \mu \widetilde{r}_{t-1,l'} + (1-\mu)\widetilde{r}_{tl,l'} \quad if \quad \left\| \widetilde{r}_{t-1,l'} - \widetilde{r}_{tl,l'} \right\|^2 < \tau \tag{13}$$

where l' donates l'-th local block. In the t-th frame, the target template $\widetilde{r}_{t,l'}$ contains two parts, one is the template $\widetilde{r}_{t-1,l'}$ of the t-1-th frame, the other is the tracking result $\widetilde{r}_{tl,l'}$ of the t-th frame. The two parts are weighted via a template update factor μ. The threshold τ is an empirically defined parameter indicating the dissimilarity level. The local update method avoids updating background information when the object is occluded. The overall algorithm of our CFPF tracker is summarized into Algorithm 1.

Algorithm 1 Our CFPF Tracker: Iteration at Time Step t

```
Input:
    Image $I_t$.
    Previous object position $p^*_{t-1}$ and scale $s_{c,t-1}$.
    Location and scale filter $h_{t-1}$.
    The regenerated template $\tilde{r}_{t-1}$ in last frame.
Output:
    Estimated object position $p^*_t$ and scale $s_{c,t}$.
    The affine parameters of the best candidate $f^*_c$.
    Updated location and scale filter $h_t$.
    Updated target template $\tilde{r}_t$.
Position and Scale estimation
    Crop search block in an image.
    Extract different features and convert to Fourier Domain by Eq. 3.
    Calculate their corresponding response maps based on Eqs. 4 and 6-7.
    Determine the target location by calculating the maximum score $c^*(t)$ according to Eqs. 5 and 8.
    The scale factor corresponding to the maximum response value is used as the scale factor of the current frame.
Affine parameter detection and Tracking
    Obtain candidates $f_c$ by random particles around the object.
    Generate the first template $\tilde{r}_t$ of current frame.
    Get the tracking result $\tilde{r}_t$ by updating the initial template in the current frame.
    Obtain an optimal candidate $f^*_c$ by similarity of the new template and candidates using Eqs. 9-10.
Update
    Update location and scale filter by the long-time and short-time update strategy according to Eqs. 11-12.
    Update object template according to Eq. 13.
```

4 Experiment

We conduct lots of experiments to evaluate the efficacy of our proposed tracker. First, we compare our tracker with the state-of-the-art trackers and quantitatively analyze the accuracy and success rate of our tracker on benchmark datasets OTB-2013 [37], OTB-2015 [38] and VOT2016 [25]. Second, we evaluate our proposed tracker with adaptive updating strategy against the tracker with fixed interval updating mechanism on OTB-2013 [37].

4.1 Experimental setup

Our tracking algorithm is implemented in MATLAB on a PC with Intel i7–7700 CPU (3.6 GHz) and 32 GB memory. In experiments, some parameters of the algorithms should be properly set in order to obtain acceptable performances. The part of the key experimental parameters in this manuscript are shown in Table 1. our tracker has the highest accuracy and success rate based on these parameters.

In all the experiments, three benchmarks are used. The first evaluation benchmark is the OTB-50 [37] benchmark. It contains results of 29 trackers evaluated on 50 sequences by a no-reset evaluation protocol. In the evaluation benchmark, center location error is the difference between the center of tracked results and the ground truth, where the smaller value means the more accurate result. The Pascal VOC overlap ratio [15] is defined as shown in Eq. (14).

$$r_{VOC} = \left|A_t \cap A_g\right| / \left|A_t \cup A_g\right| \tag{14}$$

where A_t is the area of the tracking result, and A_g is the area of the ground truth. The larger value r_{VOC} means the more accurate result. Based on two evaluation criteria, the benchmark results are reported as success plots and precision plots. The success plot shows portion of

Table 1 Main parameters set in our tracker

Parameters	Value
Score threshold of the long-time and short-time update c_{ls}	0.8
Number of frames of location filter long-time update n_l	6
Number of frames of location filter short-time update n_s	5
Template update block size s_{Tb}	4*4
Dissimilarity level threshold τ	0.1
Number of candidatesn_c	400
Template update factor μ	0.95
Template size s_T	48*48

frames with r_{VOC} greater than a threshold with respect to all threshold values. The precision plot shows similar statistics on the center error. The one-pass evaluation (OPE) is employed to compare our algorithm. We set the threshold of distance precision rate at 20 pixels and the threshold of overlap success rate at 0.5 center location errors. We use the area-under-the-curve (AUC) to rank the different methods. The AUC is displayed in the legend for each tracker.

The second evaluation benchmark is the OTB-100 [38] benchmark, it similar to OTB-50 benchmark, and the only difference is that OTB-100 contains 100 sequences including 50 sequences from OTB-50. The third evaluation benchmark is the VOT2016 [24] benchmark. The dataset contains 60 sequences with improved annotations. The benchmark evaluated a set of 70 trackers which includes the recently published state-of-the-art trackers. In VOT challenge protocol, target is re-initialized whenever tracking fails and the evaluation module reports both accuracy and robustness, which correspond to the bounding box overlap ratio and the number of failures, respectively. In the experiments, we use accuracy-robustness score, accuracy-robustness plot and expected average overlap (EAO) plot to rank the different methods.

To evaluate the impact of introducing PF into CF framework and LS updating mechanism, our tracker CFPF is compared with 9 state-of-the-art trackers, SAMF [26], CCOT [11], DCFNet [36], CREST [33], ECO [13], DeepSTRCF [27], SRDCF [10], SiamFC [1] and KCF [20]. All of these trackers can’t deal with target rotation.

4.2 Quantitative evaluation

To verify the contribution of each component in our algorithm, we implement and evaluate our approach. This section mainly shows our experimental results and we analyze the accuracy and robustness of proposed method in detail by experimental results. Firstly, the part 1) analyzes the impact on the overall performance and under target deformation and rotation of the tracker based on our method introducing PF into CF. Secondly, in the part 2), we analyze the adaptive updating strategy on the basis of experimental results by comparing with fixed interval updating mechanism.

4.2.1 Evaluation of target deformation and rotation

In this section, we evaluate the accuracy and robustness to the target deformation and rotation, while ensure that the overall performance of the tracker is not weakened of our tracker compared with some of state-of-the-art trackers on benchmark datasets OTB-2013 [37], OTB-2015 [38] and VOT2016 [25], respectively.

4.2.2 Evaluation on OTB-2013 dataset

Our CFPF tracker that introduces PF into the CF framework is first in the place for out-plane rotation, in-plane rotation and deformation under good overall performance by precision and success rate in Figs. 4 and 6. Based on the experimental results, we compare and analyze in detail the impact of PF on the tracker.

As shown in Fig. 4, the AUC scores for each tracker are shown in the Fig. 4. It can be seen that the tracking tasks are better achieved compared with other discriminative trackers and it outperforms high competition. The overall precision of our proposed tracker is the highest, and the area-under-the-curve (AUC) score is 0.941, while the CFPF gets a 0.011 improvement upon classic correlation filter method ECO. Compared with state-of-the-art deep learning

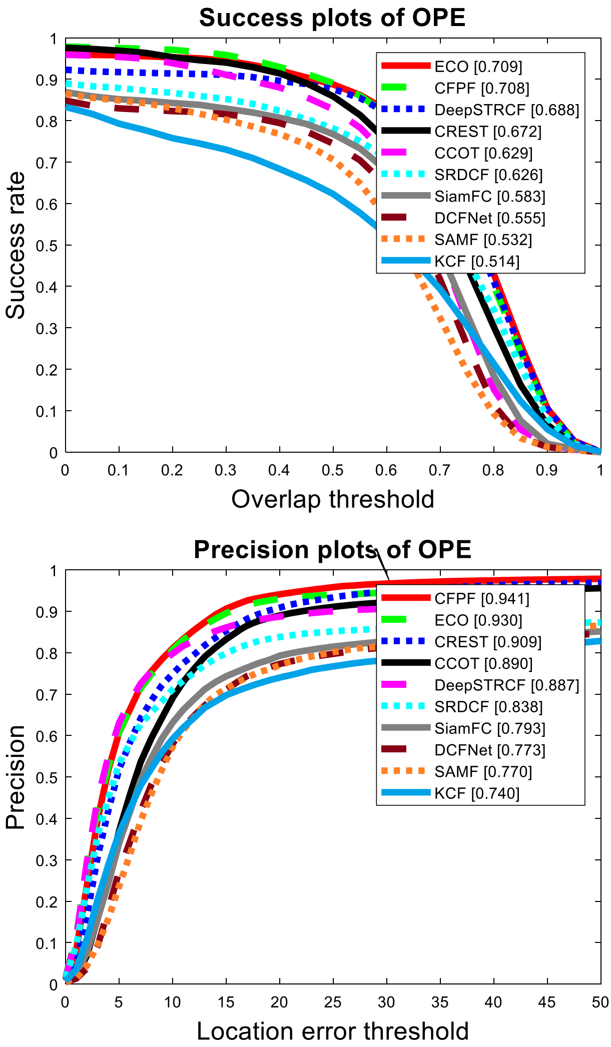

Fig. 4 The tracking success plots on OTB-2013. The area-under-the-curve (AUC) scores for the all trackers are reported in the legend

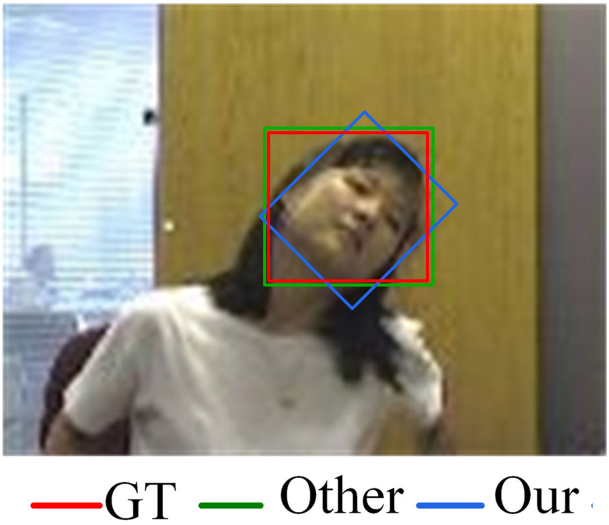

Fig. 5 Tracking results with different tracking methods

trackers CREST, and DCFNet, the CFPF gets a 0.032 improvement at least. For success rate, our tracker ranks the second, and the area-under-the-curve (AUC) score of the success rate is 0.708, it is 0.001 lower than ECO, which mainly attributes to the training labels that are marked by rectangle box. As shown in Fig. 5, the red bounding box represents the tracking result of ground truth. The green bounding box represents the tracking result of other discriminative trackers and the blue bounding box represents the tracking result of our tracker. in the process of calculating the success rate, our tracker considers the deformation and rotation of the target. As formula 14, the overlap rate is calculated and it is slightly unsuitable for our tracker. By contrast, the success rate of our tracker is higher than other discriminative trackers' listed in the figure.

As can be seen from the above analysis, results indicate that the CFPF can effectively capture the target information and the proper PF method is beneficial to improve the accuracy and robustness of our tracker.

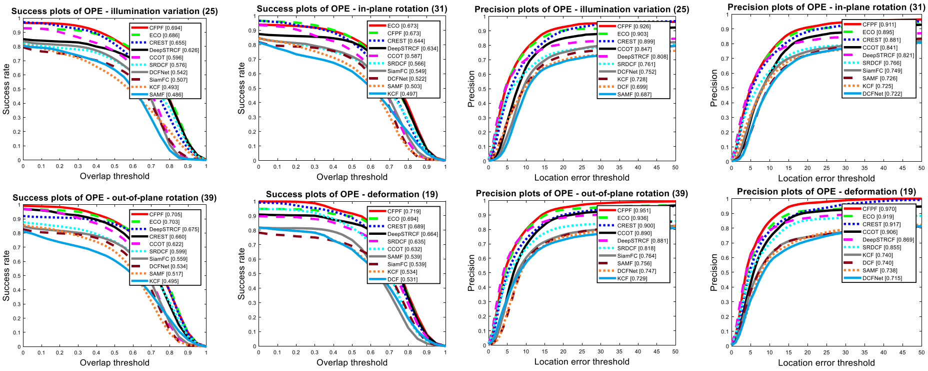

Fig. 6 The success plots over four tracking challenges, including illumination variation, out-of-plane rotation, in-of-plane rotation and deformation on OTB-2013. The area-under-the-curve (AUC) scores for the all trackers are reported in the legend

To evaluate our tracker in complex scenes, such as illumination variation, fast motion, rotation, occlusion, deformation and scale variation, especially the rotation and deformation of the target. We calculate the precision and success rate in these scenarios separately. Figure 6 shows success plots of four different attributes: illumination, out-plane rotation, in-plane rotation and deformation. The area-under-the-curve (AUC) scores for the all trackers are reported in the legend. The CFPF tracker achieves the best performance in terms of both precision and success rate. For target out-of-plane rotation, in-of-plane rotation and deformation, we separately analyze the performance of the tracker introduced into particle filter. The precision and success rate area-under-the-curve (AUC) scores of out-of-plane rotation are respectively 0.951 and 0.705, the CFPF gets 0.015 and 0.002 improvement upon ECO. The precision and success rate area-under-the-curve (AUC) scores of in-of-plane rotation are respectively 0.911 and 0.673, the CFPF gets 0.006 and 0 improvement upon ECO. The precision and success rate area-under-the-curve (AUC) scores of deformation are respectively 0.970 and 0.719, the CFPF gets 0.051 and 0.015 improvement upon ECO. Compared to other trackers based on correlation filter and deep learning, our tracking performance has improved significantly.

In addition, Fig. 7 shows some results of the top performing trackers: ECO [13], DeepSTRCF [27], CREST [33], CCOT [11] and our CFPF on 6 challenge sequences. Our CFPF tracker performs well in sequences with illumination variation, fast motion, rotation, occlusion, deformation and scale variation (ironman, football, lemming, carScale, faceocc2 and tiger 2), especially rotation and deformation, which contributes to PF that makes the tracker better handle the target deformation and rotation problems. And HOG features weaken the effect of light. For other trackers, it is impossible to improve the target's accurate tracking under deformation and rotation (ironman and faceocc2).

These promising results suggest that our CFPF not only has the advantage of high location accuracy, but also the good robustness of particle filter to target deformation and rotation due to PF method. Furthermore, the LS update strategy makes the tracker more adaptable to target deformation and fast motion, and specifically experimental analysis is in the part (2).

4.2.3 Evaluation on OTB-2015 dataset

Our CFPF tracker ranks first for out-plane rotation and deformation under good overall performance by precision and success rate in Figs. 8 and 9.

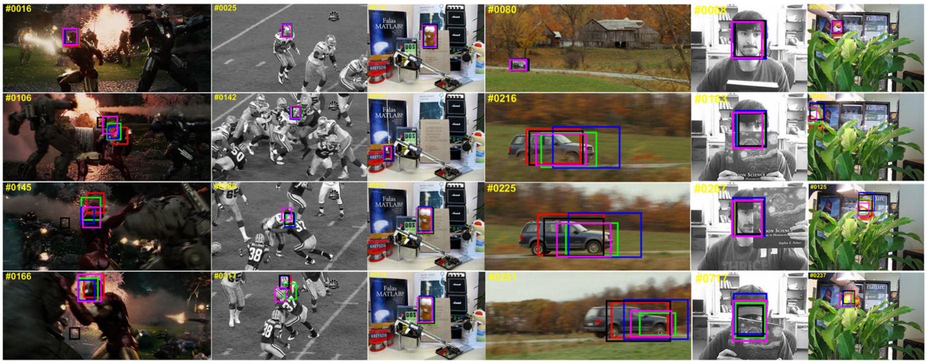

Fig. 7 Sampled tracking results of our tracker for some challenging sequences including illumination variation, fast motion, rotation, occlusion, deformation and scale variation. From left to right: ironman, football, lemming, carScale, faceocc2 and tiger2

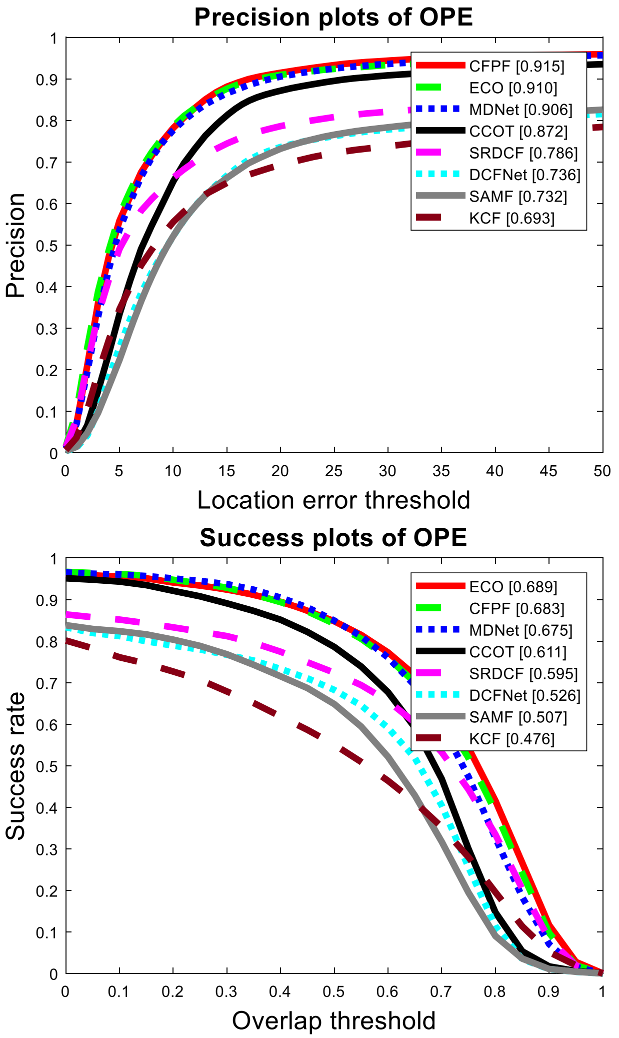

Fig. 8 The tracking success plots conducted on OTB-2015. The area-under-the-curve (AUC) scores for the all trackers are reported in the legend

The AUC scores for each tracker are shown in the Fig. 8. We observe that the overall precision of our proposed tracker is the highest, and the area-under-the-curve (AUC) score is 0.915, while the CFPF gets a 0.005 improvement upon ECO. For success rate, our tracker ranks the second, and the area-under-the-curve (AUC) score of the success rate is 0.683, it is 0.006 similar to this part a), as shown in Fig. 5. Another reason is that lower than ECO. The reason of causing low success rate is the accuracy of our tracker needs to be improved for low resolution sequences. Since the OTB-2015 dataset contains more videos with low resolution, our CFPF tracker does not perform as well as ECO in overlap success. Compared with other state-of-the-art trackers, even if MDNet uses the test sequence to train tracker, our tracker is still more accurate and robust than MDNet.

To verify the first contribution of CFPF method for target rotation and deformation in our algorithm, we also present the evaluation results of the target out-of-rotation and deformation

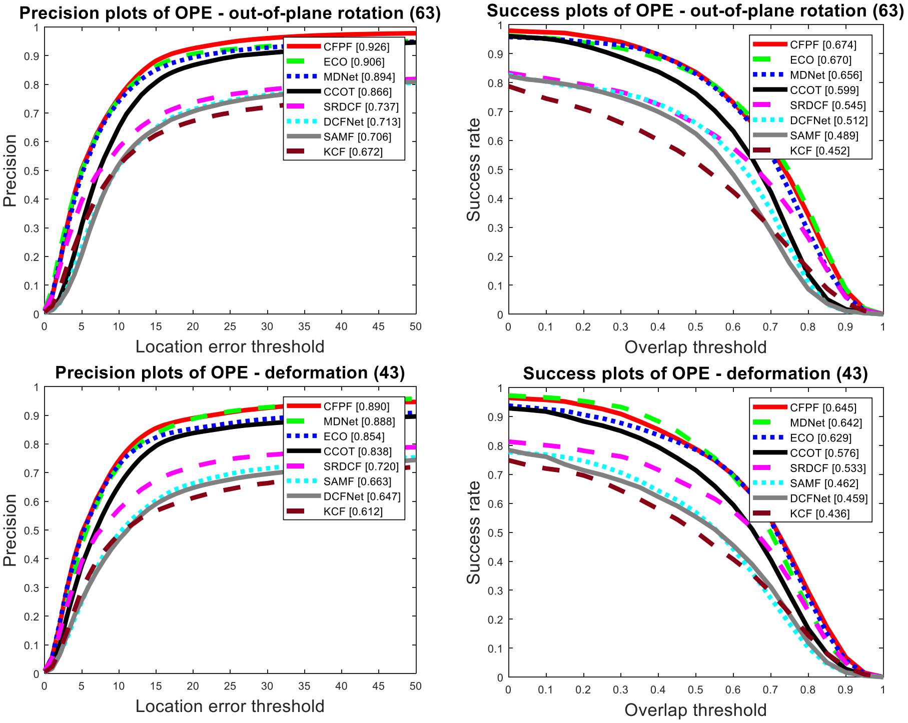

Fig. 9 The success plots over two tracking challenges, including out-of-rotation and deformation on OTB-2015. The area-under-the-curve (AUC) scores for the all trackers are reported in the legend

on OTB-2015. Figure 9 shows success plots of two different attributes and the area-under-the-curve (AUC) scores for the all trackers are reported in the legend. The CFPF tracker ranks first in terms of both precision and success rate. As can be seen from Fig. 9, the precision and success rate area-under-the-curve (AUC) scores of out-of-plane rotation are respectively 0.926 and 0.674, the CFPF gets 0.02 and 0.004 improvement upon ECO. The precision and success rate area-under-the-curve (AUC) scores of in-of-plane rotation are respectively 0.911 and 0.673, the CFPF gets 0.006 and 0 improvement upon ECO. The precision and success rate area-under-the-curve (AUC) scores of deformation are respectively 0.890 and 0.645, the CFPF gets 0.036 and 0.016 improvement upon ECO.

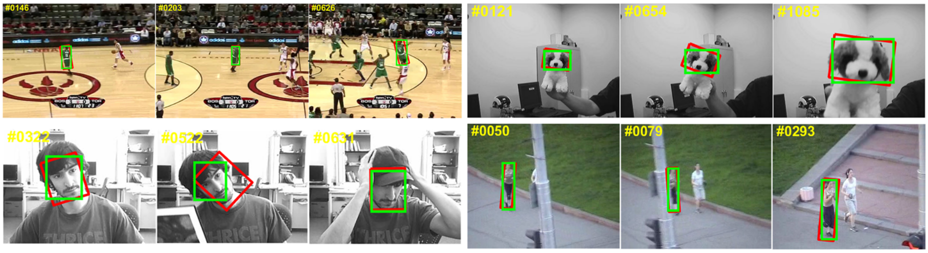

Fig. 10 Sampled tracking results of our tracker for some rotation and deformation sequences on OTB-2015. From up to down: basketball, dog1, faceocc2 and jogging-1

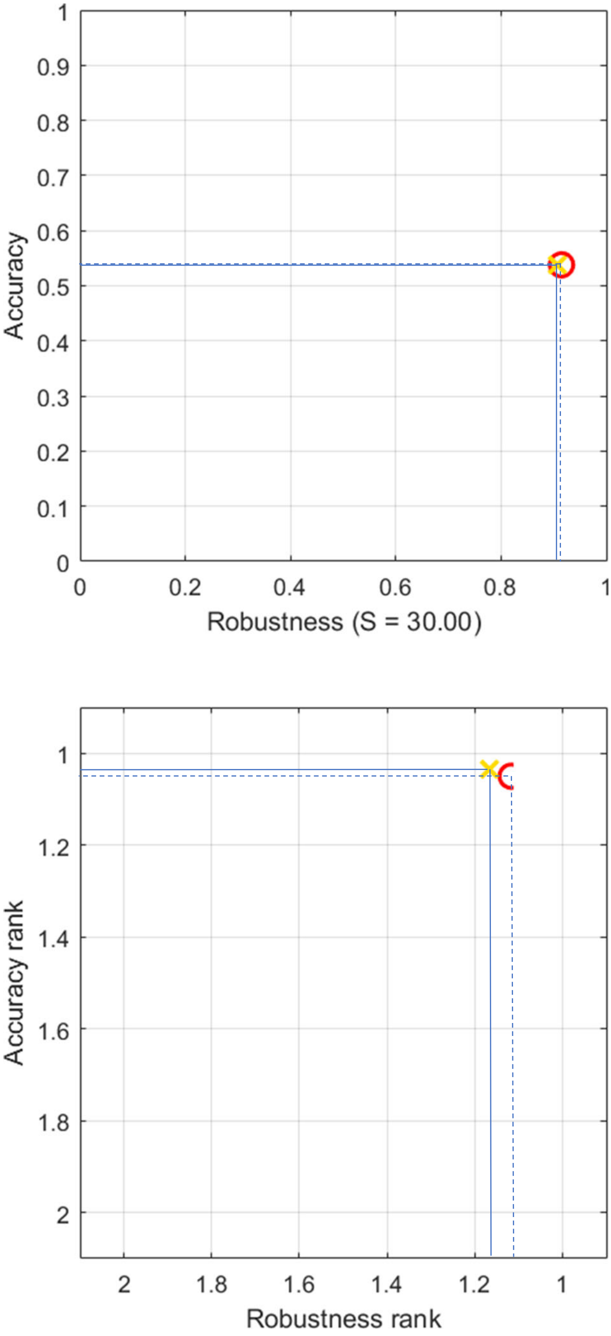

Fig. 11 The robustness-accuracy plots of tested algorithms in VOT2016 dataset. The AR plot (left) shows the accuracy and robustness scores. In the ranking plot (right) the accuracy and robustness rank for each tracker is displayed. The better trackers are located at the upper-right corner

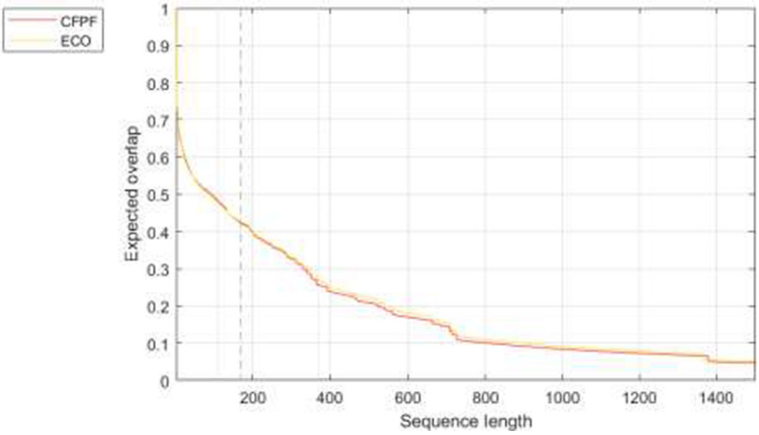

Fig. 12 Expected Average Overlap (EAO) curve on VOT2016. Only the top 2 trackers are shown for clarity

Figure 10 shows some results of the ECO [13] and our CFPF on 4 challenge sequences. The purpose is to visually show that our tracker introduced the particle filter significantly improves the situation that the discriminative correlation filter methods, such as ECO, MDNet, DeepSTRCF, cannot accurately and robustly cope with the target deformation and rotation. As can be seen from the sequence results in the figure, the trackers rotate with targets (dog1, faceocc2). When the targets have deformation (jogging-1, basketball), the tracker can adjust its shape according to the deformation of the target, and accurately track the target to reduce background information. The ECO tracker always tracks the target in a rectangle box, no matter how the target changes, which leads to containing more background information for target. And the background information is used to update the filter, which will reduce the accuracy of the filter. Therefore, the tracker's adaptability to target deformation and rotation is weakened, which makes tracker low robustness.

These experimental results conducted on OTB-2015 and the above analysis suggest that our tracker can be able to adapt to target changes and achieve better accuracy and robustness by the combination of correlation filter and particle filter. Especially in the target deformation and rotation, the CFPF tracker uses the affine transformation parameters and solve the problem rotating with the target for discriminative correlation filter methods.

4.2.4 Evaluation on VOT2016 dataset

To further validate the robustness and accuracy of our tracker, we evaluated it based on the VO2016 dataset. Although the VOT-2016 benchmark takes into account the deformation and rotation of the target during labeling target, the target boxes of the sequences with rotation during testing are transformed into rectangle boxes. As shown in Fig. 5 and formula 2, this dataset will show lower evaluation score compared to the actual performance of our target. In

Table 2 State-of-the-art in terms of EAO

	CFPF	ECO
EAO	0.370	0.374

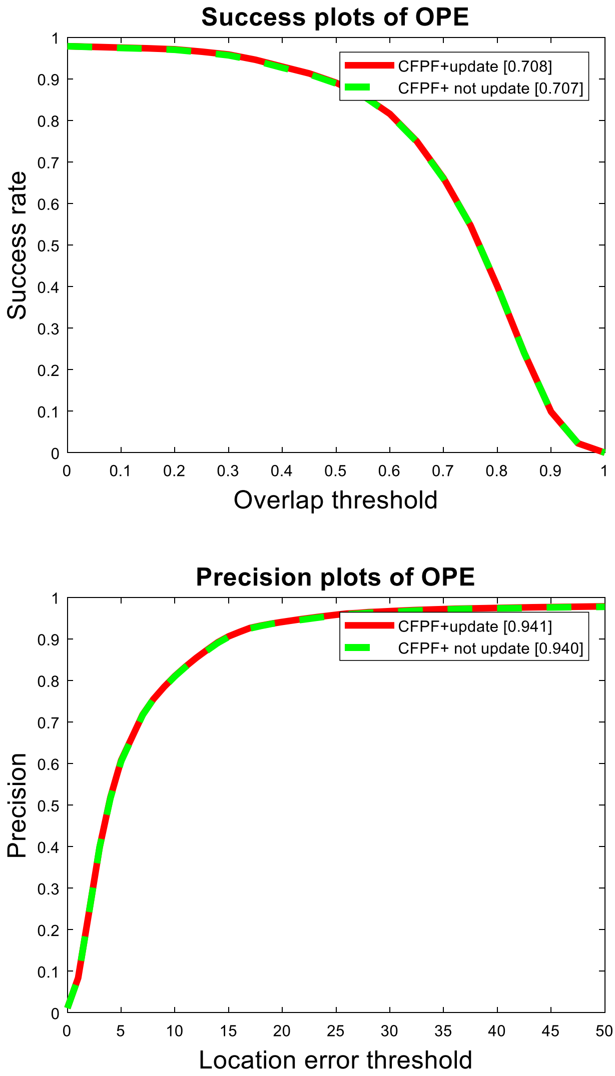

Fig. 13 The tracking success plots conducted on OTB-2013. The area-under-the-curve (AUC) scores for the all trackers are reported in the legend

addition, the dataset can't evaluate individually the performance of target deformation and rotation, so we only test the overall performance of the CFPF tracker based on this dataset.

Figure 11 shows the robustness-accuracy scores and plots of CFPF and ECO trackers. The robustness and accuracy with blue solid line represent ECO's and the robustness and accuracy with blue dotted line represent CFPF's. We can see that the accuracy of ECO is higher than CFPF, but CFPF tracker ranks first in the robustness. Meanwhile, Fig. 12 and Table 2 show expected average overlap (EAO) curvets and scores of ECO and CFPF. It can be seen that CFPF tracker is as good as ECO when sequence length is less than 400 in Fig. 5. If sequence length is longer than 400, the expected average overlap plot of CFPF is falling faster than ECO's. The situation is reflected by the EAO scores in Table 2, and score of CFPF is 0.004

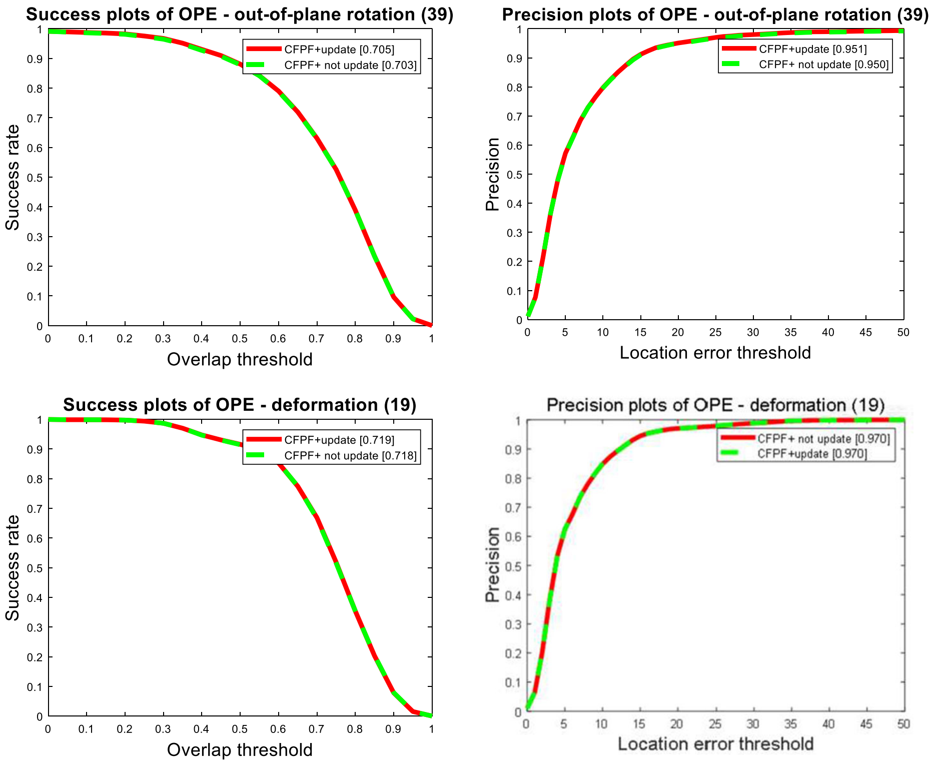

Fig. 14 The success plots over target rotation and deformation on OTB-2013. The area-under-the-curve (AUC) scores for the all trackers are reported in the legend

lower than ECO's. The above experimental results can roughly reflect that our tracker introduced into PF maintain a stable overall performance.

4.2.5 Evaluation of the long-time and short-time update strategy

We conduct experiments on OTB-2013 for validity of the LS strategy. The red line refers to the performance effect that particle filter is introduced to correlation filter framework and combining LS update method, while the green line means no LS update, and the 'CFPF+ update' is CFPF with LS update strategy, but the 'CFPF+not update' is CFPF without LS update in Figs. 13 and 14.

It can be seen briefly that the CFPF with LS gets 0.001 and 0.001 improvement upon CFPF without LS in success rate and precision from Fig. 13. Additionally, Fig. 14 shows success plots of two different attributes: out-plane rotation and deformation. The area-under-the-curve (AUC) scores of tracker with LS update in taget out-plane rotation are separately 0.951 and 0.705. The improvement of success rate and precision are separately 0.001 and 0.002 on CFPF without LS. The area-under-the-curve (AUC) scores of tracker with LS update in taget deformation are separately 0.970 and 0.719. The improvement of success rate is 0.001 on CFPF without LS. It is known from the above experimental results that the LS update strategy improves accuracy and robustness of tracker.

For the tracking speed, the GPU version of CFPF tracker operates at 6–7 FPS (Frames Per Second) and it's about one FPS slower than ECO. The reason is that template matching took some time in PF. But compared with other good performance trackers based on CNN features, the tracking speed of CFPF tracker is relative superior.

5 Conclusion and future work

In this manuscript, we have proposed a new CFPF tracker introduced PF into DCF framework. At first, a correlation filter is used to estimate the position and scale of a given object. Besides, the affine parameters are detected using PF. According to the currently known target position and size, we obtain randomly candidates around the target and get the initial template of the current frame. Next correlation coefficients are calculated between template and candidates. The optimal candidate with the highest degree of correlation and its corresponding rotation factor are obtained. Lastly, according to the tracking principle of PF, the target location, size and rotation factor are used as affine parameters to get target. Experimental results show that our tracker has promising performance in terms of accuracy and robustness, and improves performance in rotation and deformation comparing existing discriminative trackers.

The proposed CFPF tracker can precisely describe target variation in complex scenarios, and further improves accuracy and robustness of discriminative correlation filter methods. Limited by representation capability of target variation in the PF and CF, the proposed CFPF tracker can't accurately capture object when target with wide rotation angle or low resolution. There is still room for improvement of target variation representation mechanism in discriminative correlation filter methods, hence object tracking is still a challenging task because of background cluttering and some special application scenarios.

Acknowledgments The authors would like to thank the editors and the anonymous reviewers, whose comments helped to improve the paper greatly. This work was supported by the National Natural Science Foundation of China (No. 61461028) and the National Natural Science Foundation of China (No. 61861027).

References

1. Bertinetto L, Valmadre J, Henriques JF, Vedaldi A, PHS T (2016) Fully-convolutional Siamese networks for object tracking. European Conference on Computer Vision, pp 850–865
2. Biresaw TA, Cavallaro A, Regazzoni CS (2015) Tracker-level fusion for robust Bayesian visual tracking. IEEE Transactions on Circuits and Systems for Video Technology 25(5):776–789
3. Bolme DS, Beveridge JR, Draper BA, Lui YM (2010) Visual object tracking using adaptive correlation filters. IEEE Conference on Computer Vision and Pattern Recognition, pp 2544–2550
4. Chang C, Ansari R (2005) Kernel particle filter for visual tracking. IEEE Signal Processing Letters 12(3): 242–245
5. Changjiang Y, Duraiswami R, Davis L (2005). Fast multiple object tracking via a hierarchical particle filter. In: EEE International Conference on Computer Vision, 17–21 Oct. 2005. pp 212–219 Vol. 211. https://doi.org/10.1109/ICCV.2005.95
6. Chatfield K, Simonyan K, Vedaldi A, Zisserman A (2014). Return of the devil in the details: delving deep into convolutional nets. Computer Science
7. Choe G, Wang T, Liu F, Choe C, Jong M (2015) An advanced association of particle filtering and kernel based object tracking. Multimed Tools Appl 74(18):7595–7619

8. Cimpoi M, Maji S, Vedaldi A (2015). Deep filter banks for texture recognition and segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, 7–12 June 2015. pp 3828–3836. https://doi.org/10.1109/CVPR.2015.7299007
9. Danelljan M, Khan FS, Felsberg M, JVD W (2014) Adaptive color attributes for real-time visual tracking. IEEE Conference on Computer Vision and Pattern Recognition, pp 1090–1097
10. Danelljan M, Hager G, Khan FS, Felsberg M (2015) Learning spatially regularized correlation filters for visual tracking. IEEE International Conference on Computer Vision, pp 4310–4318
11. Danelljan M, Robinson A, Khan FS, Felsberg M (2016) Beyond correlation filters: learning continuous convolution operators for visual tracking. European Conference on Computer Vision, pp 472–488
12. Danelljan M, Hager G, Khan FS, Felsberg M (2016) Discriminative scale space tracking. IEEE Trans Pattern Anal Mach Intell 39(8):1561–1575
13. Danelljan M, Bhat G, Khan FS, ECO FM (2017) Efficient convolution operators for tracking. IEEE Conference on Computer Vision and Pattern Recognition, pp 6931–6939
14. Danelljan M, Häger G, Khan FS, Felsberg M Coloring Channel Representations for Visual Tracking. In: Paulsen RR, Pedersen KS (eds) . Image Analysis, Cham, 2015// 2015. Springer International Publishing, pp 117–129
15. Everingham M, Winn J (2010). The PASCAL visual object classes challenge IJCV 88 (2):303–338
16. Galoogahi HK, Sim T, Lucey S (2014) Multi-channel correlation filters. IEEE International Conference on Computer Vision, pp 3072–3079
17. Galoogahi HK, Sim T, Lucey S (2015) Correlation filters with limited boundaries. IEEE Conference on Computer Vision and Pattern Recognition, pp 4630–4638
18. Galoogahi HK, Fagg A, Lucey S (2017) Learning background-aware correlation filters for visual tracking. IEEE International Conference on Computer Vision, pp 1144–1152
19. Henriques JF, Caseiro R, Martins P, Batista J (2012) Exploiting the Circulant structure of tracking-by-detection with kernels. European Conference on Computer Vision, pp 702–715
20. Henriques JF, Caseiro R, Martins P, Batista J (2014) High-speed tracking with Kernelized correlation filters. IEEE Trans Pattern Anal Mach Intell 37(3):583–596
21. Henriques JF, Rui C, Martins P, Batista J (2015) High-speed tracking with Kernelized correlation filters. IEEE Trans Pattern Anal Mach Intell 37(3):583–596
22. Huang W, Lin L, Huang T, Lin J, Zhang X (2019) Scale-adaptive tracking based on perceptual hash and correlation filter. Multimed Tools Appl 78(12):16011–16032
23. Isard M, Blake A (1998) CONDENSATION—conditional density propagation for visual tracking. Int J Comput Vis 29(1):5–28
24. Jia X, Lu H, Yang M-H (2012) Visual tracking via adaptive structural local sparse appearance model. IEEE Conference on Computer Vision and Pattern Recognition, pp 1822–1829
25. Kristan M, Leonardis A, Matas J, Felsberg M, Pflugfelder R, Čehovin L, Vojír T, Häger G, Lukežič A, Fernández G (2016) The visual object tracking VOT2016 challenge results. IEEE International Conference on Computer Vision Workshop, pp 191–217
26. Li Y, Zhu JA (2014). Scale adaptive kernel correlation filter tracker with feature integration. In: European conference on computer vision workshop, 2014. Computer Vision - ECCV 2014 Workshops. Springer International Publishing, pp 254–265
27. Li F, Tian C, Zuo W, Zhang L, Yang MH (2018) Learning spatial-temporal regularized correlation filters for visual tracking, IEEE Conference on Computer Vision and Pattern Recognition
28. Liu L, Shen C, van den Hengel A, Ieee (2015) The treasure beneath convolutional layers: cross-convolutional-layer pooling for image classification. IEEE Conference on Computer Vision and Pattern Recognition. IEEE Conference on Computer Vision and Pattern Recognition, pp 4749–4757
29. Ma C, Huang JB, Yang X, Yang MH (2015) Hierarchical convolutional features for visual tracking. IEEE International Conference on Computer Vision, pp 3074–3082
30. Pi J, Hu K, Gu Y, Qu L, Li F, Zhang X, Zhan Y (2016) Robust scale adaptive and real-time visual tracking with correlation filters. IEICE Trans Inf Syst 99(7):1895–1902
31. Qian C, Xu Z (2016) Robust visual tracking via sparse representation under subclass discriminant constraint. IEEE Transactions on Circuits and Systems for Video Technology 26(7):1293–1307. https://doi.org/10.1109/TCSVT.2015.2424091
32. Shaohua Kevin Z, Chellappa R, Moghaddam B (2004) Visual tracking and recognition using appearance-adaptive models in particle filters. IEEE Trans Image Process 13(11):1491–1506. https://doi.org/10.1109/TIP.2004.836152
33. Song Y, Ma C, Gong L, Zhang J, Lau RWH, Yang MHCREST (2017) Convolutional residual learning for visual tracking. IEEE International Conference on Computer Vision, pp 2574–2583
34. Truong MTN, Pak M, Kim S (2018) Single object tracking using particle filter framework and saliency-based weighted color histogram. Multimed Tools Appl 77(22):30067–30088

35. Wang D, Lu H, Bo C (2015) Visual tracking via weighted local cosine similarity. IEEE Transactions on Cybernetics 45(9):1838–1850
36. Wang Q, Gao J, Xing J, Zhang M, Hu W (2017). DCFNet: discriminant correlation filters network for visual tracking. arXiv:170404057
37. Wu Y, Lim J, Yang MH (2013) Online object tracking: a benchmark. IEEE Conference on Computer Vision and Pattern Recognition, pp 2411–2418
38. Wu Y, Lim J, Yang MH (2015) Object tracking benchmark. IEEE Trans Pattern Anal Mach Intell 37(9): 1834–1848
39. Yuan D, Zhang X, Liu J, Li D (2019) A multiple feature fused model for visual object tracking via correlation filters. Multimed Tools Appl 78(19):27271–27290
40. Zhang X, Ding Q, Luo H, Hui B, Chang Z (2019) An adaptive multi-features aware correlation filter for visual tracking. IEEE Access 7:134772–134781. https://doi.org/10.1109/ACCESS.2019.2942047
41. Zia K, Balch T, Dellaert FA (2004). Rao-Blackwellized particle filter for EigenTracking. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 27 June-2 July 2004. pp II-II. https://doi.org/10.1109/CVPR.2004.1315271

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.