Multi-experts Joint Decision with Adaptive Model Updater for Robust Visual Tracking

Da Li¹ · Qixiang Zou¹ · Ke Zhang¹

Received: 15 September 2020 / Accepted: 17 September 2021
© The Author(s) 2021

Abstract

Over these years, correlation filters based trackers have shown edges both in accuracy and speed. However, variations of target appearance caused by heavy occlusion, rotation, background clutters and target deformations are the major challenges for tracking. To solve these problems, many works put on exploiting the power of target representation, such as high-level convolutional features. Nonetheless, these methods make a great compromise between the speed and performance. At the same time, there are few researches on improving the performance of model updater and the ensemble methods. In this paper, a multi-experts joint decision strategy base on kernelized correlation filters is proposed to obtain robust and accurate visual tracking, two trackers with handcrafted features and deep convolutional neural network features are integrated in this framework. We also investigate the mechanism of tracking failure caused by occlusion and background clutters, and propose a novel criterion to evaluate the reliability of samples. Our work includes extending the kernelized correlation filter-based tracker with the capability of handling scale changes as well. The proposed tracker is extensively evaluated on the OTB-2013, OTB-2015 and VOT2015 benchmark datasets. Compared with the state-of-the-art trackers, the distinguished experimental results demonstrate the effectiveness of the proposed framework.

Keywords Correlation filters · Ensemble · Multi-experts · Model updater · Occlusion · Scale estimation · Visual tracking

1 Introduction

In recent years, visual tracking is widely used in intelligent surveillance, autopilot, robotics and many other applications, and becomes one of the most popular fields in computer vision. The research focuses in this on single target, object given and model-free tracking. The tracker is initialized with the location and size of the target in the first frame, and any explicit appearance or prior model cannot be used. In the case where the training sample has only the first frame, it is challenging to estimate the target trajectory throughout the sequence. At the same time, the target also suffers from occlusions, scale variations, motion blur and illumination changes during the tracking process.

To tackle the problem of lacking of training samples, most existing trackers either adopt generative [1–3] or discriminative [4–6] methods to learn appearance model. The

generative algorithms search the candidate regions by the minimum reconstruction error to find the best matching position. The discriminative approaches locate the target by designing and training the classifier to distinguish the background and the foreground. Among the discriminative tracking algorithms, correlation filter-based methods [4, 6, 7] gain much attention for their high accuracy as well as high efficiency. Deep convolutional neural network (CNN) features have achieved great success in many computer vision tasks [8–10]. Correlation filters are difficult to adapt to severe deformation and fast motion, so there is a rising trend for introducing CNN features in tracking framework to improve performance with the help of its rich feature representation. MDNet [11] adopt CNN models trained with tracking datasets including [12, 13] and achieved better performance. HCF [14] integrates CNN features into the correlation filter framework for robust tracking.

However, the original correlation filtering algorithms [4, 6, 7] use fixed-size templates, which leads to a problem. When the size of the target changes drastically, the template either contains extra background or only a part of the target. This may causes tracking failure when scale variation in the

✉ Da Li
frankli26@whut.edu.cn

¹ School of Information Engineering, Wuhan University of Technology, Wuhan 430070, China

presence of other complicated factors such as background clutters, occlusion and motion blur. Meanwhile, many practical application scenarios require accurate target size information in the image. Extensive researches [15–19] have been conducted on how to establish a robust scale estimation strategy. Among them, the scale adaptive kernel correlation filter tracker with feature integration (SAMF) [18] and the discriminative scale space tracker (DSST) [19] are the most widely used approaches. SAMF is a straightforward method to estimate the scale by applying the standard learned two-dimensional filter to samples of multi-resolutions around the target, this exhaustive scale search strategy is computationally demanding. DSST [17] tackles the scale estimation problem by learning two separate correlation filters for explicit translation and scale estimation. First the conventional discriminative correlation filter is employed to find the maximum response indicating the target position. Next, a separate scale detection model is trained to search the optimal scale in the multi-scale spatial pyramid. In this way, the employ of the two independent filters avoids mutual interference. Although the DSST addresses the scale estimation problem to some degree, the conventional correlation filter for translation estimation still suffers from its relatively low discriminative power.

Scalable correlation filter trackers are still not strong enough. Ensemble approaches [18, 20–24] have been developed as another way to improve the performance with combination of multiple trackers for visual tracking. For example, the ensemble method [22] under the acceleration framework [25] incrementally trains the weak trackers of each component to classify the training samples that were previously misclassified by the tracker. As one of the representative works, multiple experts using entropy minimization (MEEM) [20] demonstrates the potential of ensemble, which proposes to address the problem by using the multi-expert restoration scheme to predict target, where an entropy based loss function is defined to determine the confidence of current tracker. HDT [23] estimates the position of the target by fusing the response maps obtained from a correlation filter trained by hierarchical convolutional features of various resolutions as a weak classifier in a coarse-to-fine scheme. The final prediction result is weighted by the adaptive hedging method of weak classifier. In MCCT [24] Wang et al. introduce the concept of feature pool, which includes seven features, and use the different features of the target to learn the correlation filter tracking experts, finally select the most reliable one as the tracking result in each frame.

Although the impact of model updater is significant on performance [26], there are very few researches focus on this component. The model updater determines the frequency and strategy of updating model. Since only the samples of the first frame are reliable, the tracker must

maintain a tradeoff between the collection of new samples during the tracking process and the prevention of the tracker from drifting to the background. Most trackers update the model every frame. In [6], the criterion used to obtain the target position is the naive maximum response value, and update every frame with a moderate learning rate. Entropy minimization is adopted in [20] to identify reliable model update and discard the incorrect ones. Bolme et al. propose a simple measurement of peak strength called the peak-to-sidelobe ratio (PSR) [7]. Wang et al. argue that the robustness of the maximum response value will be heavily degraded due to the presence of some other challenging factors such as motion blur, partial and full occlusion. Thus, in [27] instead of using the naive maximum response value to obtain both the translation estimation, Wang et al. propose a criterion named average peak-to-correlation energy (APCE), when the maximum of the response map and APCE are both great enough, the tracking model will be updated. The above methods either update every frame or discard unreliable samples directly. A reasonable update strategy should adjust the learning rate adaptively according to the confidence level of the sample, so that it does not contaminate the model or lose the information that may be useful for tracking.

To address the problems mentioned above, a multi-experts joint decision framework based on kernelized correlation filter is proposed to carry out robust visual tracking. The main contributions are summarized as follows:

1. First, our trackers are extended with the capability of estimating scale. The size and position of the target can be obtained simultaneously.
2. Then, a multi-experts joint decision strategy based on kernelized correlation filters is presented. Handcrafted features (HOG [28], CN [29]) and CNN features are exploited to build a correlation filter bucket, which contains seven experts. By evaluating the total robustness score of each expert, the most reliable one is selected as tracking result for each frame.
3. Next, the novel criteria of peaks correlation of response map (PCRM) is proposed to evaluate the reliable level of sample. The PCRM of first three response maps are computed, and weigh it to obtain a confidence index for the current sample.
4. Finally, an adaptive model updater strategy is proposed to alleviate the contamination of samples by considering the PCRM of sample and the divergence of experts.

Extensive and comprehensive experiments are conducted on widely used benchmarks OTB-2013 [30], VOT2015 [13] and OTB-2015 [12]. The results validate the improvement in success and precision rate of proposed tracker.

2 Related Works

During the past few decades, substantial progress has been contributed in the field of visual object tracking. In this section, the works closely related to our method are summarized from three perspectives: tracking by correlation filters, tracker ensemble and model update strategy.

2.1 Tracking by Correlation Filters

Due to high efficiency and accuracy, correlation filters-based trackers remain mainstream in practical applications. Bolme et al. [7] utilize the minimum output sum of squared error (MOSSE) to learn the correlation filters, by using the circular correlation, the resulting filter can be computed efficiently using point-wise operation in Fourier domain. Whereafter, Henriques et al. [4], dense sampling is performed by efficiently utilizing the structure of the circulant matrix, and while maintaining high speed, the discriminative ability of the CSK is enhanced as the negative sample is augmented. The above methods are based on grayscale feature. The work is further extended to HOG multichannel feature in kernel space [6]. Staple [31] makes full use of the complementarity of color and gradient information, while running in excess of real-time. Danelljan et al. [32] introduce a spatially regularized component in the learning to penalize CF coefficients depending on their spatial locations and alleviate the boundary effects. With the rising trend of for introducing CNN features into object tracking field, several trackers [14, 33] use deep models pretrained for the object classification task for feature representations, and the performance have been further improved. More recently, C-COT [34] achieves outstanding performance in several benchmarks. C-COT adopts continuous convolution operator to fuse the deep feature maps. After that, in ECO [35], several extra strategies are used to combine the deep and hand-crafted features to speed up in C-COT framework. More extended work, such as scale variation [17, 19] and long-term tracking [36] are added to the correlation filters framework.

2.2 Tracker Ensemble

According to the literature [26], the ensemble approach can improve the performance substantially. In MEEM [20] the entropy minimization is used to exploited the relationship between multi experts and its historical tracker. Then in [37], Li et al. extend it by using the unified discrete graph algorithm to model the multiple experts. Qi et al. [23] propose to develop an improved hedge algorithm that combines weak CNN based trackers from various convolutional layers into

a single stronger tracker. Wang et al. [24] propose the multi-cue correlation filters framework, which constructs parallel experts from different features, and selects the expert with the highest robustness score as tracking result for each frame.

2.3 Model Update Strategy

Although the implementation of the model updater is often treated as tricks, their impact on performance is usually very significant. Unfortunately, few works focus on this component [26]. Santner et al. propose parallel robust online simple tracking (PROST) [38], using a simple template model as a non-adaptive, a novel optical-flow-based mean-shift tracker as highly adaptive element and an online random forest as moderately adaptive appearance-based learner. In MOSSE [7] a criteria PSR is used to quantify the reliability of the tracked sample, and Bolme et al. argue that, when the value of PSR ranges between 20.0 and 60.0 indicates very strong peaks. MEEM [20] tracker is proposed to identify reliable model update and discard the incorrect ones. In KCF [6], the model is updated every frame with a moderate learning rate. Wang et al. [27] propose to employ the maximum response value and the APCE as the criterion to provide a high-confidence update strategy for robustness.

3 Methods

A multi-experts joint decision strategy with adaptive model updater for robust tracking base on kernelized correlation filter is proposed in this work. Firstly, the baseline of our trackers [6] adopts fixed target size, that given in the first frame. Therefore, a robust scale estimation approach [17] is employed to handle target scale changes. Secondly, handcrafted or deep features are extracted, seven experts are obtained by splitting and combining these two features, after jointing the decision of experts, the most reliable one is selected as the tracking result. Thirdly, the novel criteria called peaks correlation of response map (PCRM) is proposed. By evaluating the correlation between the maximum value and other peaks of the response map, the PCRM obtains the confidence level of the sample. Finally, by considering PCRM and the historical divergence of experts, the presented model updater strategy can update the model with an appropriate learning rate. The flowchart in Fig. 1 depicts the main framework of our proposed algorithm.

In Sect. 3.1, the formulas of our baseline tracker multi-channel kernelized correlation filter is treated. The scale estimation approach in our trackers is introduced in Sect. 3.2 and the multi-features construction and ensemble tracking strategy in Sects. 3.3 and 3.4 are demonstrated respectively.

Fig. 1 Flowchart of our proposed algorithm. Different features are extracted for each expert from searching area. Then every expert gets target position by using KCF framework. Next, by self and pair evaluation, the most credible expert is selected as the result for the current

frame. Finally, by combining the fluctuation of response map of first three experts and historical robustness score, the model updates adaptively

In Sect. 3.5, the PCRM and the model updater strategy are proposed.

3.1 The Kernelized Correlation Filter Tracker

Since the discriminative ability of KCF tracker is enhanced with the augmentation of negative samples while maintaining high speed by exploiting structure of circulant matrix with high efficiency, KCF has become the baseline of many trackers [39, 40].

For notational simplicity, one-dimensional signal is considered, more details can be found in [6]. Given one dimensional data $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$, the training goal is to find $f(\mathbf{z}) = \mathbf{w}^T \mathbf{z}$ which minimizes the squared error over training samples $\mathbf{x}_i$ and their regression targets $\mathbf{y}_i$,

$$\min_{\mathbf{w}} \sum_{i=1}^n (f(\mathbf{x}_i) - \mathbf{y}_i)^2 + \lambda_1 \|\mathbf{w}\|. \quad (1)$$

The scalar λ_1 is a regularization parameter that controls overfitting. To allow for a more powerful classifier with nonlinear regression functions $f(\mathbf{z})$, the solution $\mathbf{w}$ is expressed as a combination of the samples:

$$\mathbf{w} = \sum_i \alpha_i \varphi(\mathbf{x}_i), \quad (2)$$

where α_i are the variables under optimization in dual space, $\varphi(\mathbf{x})$ represents a non-linear feature-space, therefore the optimized variables are α , instead of $\mathbf{w}$. According to literature [41], this alternative representation $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_n]$ is said to be in the dual space, as opposed to the primal space $\mathbf{w}$.

The solution to the kernelized version of ridge regression can be obtained as follow

$$\alpha = (K + \lambda_1 I)^{-1} \mathbf{y}, \quad (3)$$

where K is the kernel matrix containing elements $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$, which are computed using the kernel function κ .

For the most commonly used kernels (e.g., Gaussian, linear and polynomial), the circulant matrix trick can also be used to make Eq. (3) diagonal:

$$\hat{\alpha} = \frac{\hat{\mathbf{y}}}{\hat{\mathbf{k}}^{\text{xx}} + \lambda_1}, \quad (4)$$

where $\mathbf{k}^{\text{xx}}$ is the kernel correlation and $\hat{\cdot}$ denotes the Discrete Fourier Transform (DFT) of a vector, $\hat{\mathbf{y}} = \mathcal{F}(\mathbf{y})$. The multiplications and divisions in Eq. (3) are performed element-wise. In our trackers, the Gaussian kernel is adopted for its high accuracy as

$$\mathbf{k}^{\text{xx}'} = \exp\left(-\frac{1}{\sigma^2}\left(\|\mathbf{x}\|^2 + \|\mathbf{x}'\|^2 - 2\mathcal{F}^{-1}(\hat{\mathbf{x}}^* \hat{\mathbf{x}}')\right)\right), \quad (5)$$

where kernel function $\kappa(\mathbf{x}_i, \mathbf{x}_j)$ can be simply denoted as $\mathbf{k}^{\text{xx}'}$, $\hat{\mathbf{x}}^*$ is the complex-conjugate of $\hat{\mathbf{x}}$.

In detection process, a patch $\mathbf{z}$ with the same size of $\mathbf{x}$ is extracted at the position provided by the previous frame, and the response map is calculated as

$$\hat{\mathbf{f}}_{\text{tran}}(\mathbf{z}) = \hat{\mathbf{k}}^{\text{xz}} \hat{\alpha}, \quad (6)$$

where the $\hat{\mathbf{k}}^{\text{xz}}$ is the kernelized correlation between $\mathbf{z}$ and $\mathbf{x}$ as defined in Eq. (5), meanwhile $\hat{\alpha}$ is obtained in the previous frame by Eq. (4). Then, the position of the object in the current frame is located by finding the translation with the maximum value in the response map $\mathbf{f}(\mathbf{z})$.

To avoid model corruption, KCF uses interpolation to update the model every frame:

$$\hat{\alpha}_t = (1 - \eta)\hat{\alpha}_{t-1} + \eta\hat{\alpha}_t, \quad (7a)$$

$$\hat{\mathbf{x}}_t = (1 - \eta)\hat{\mathbf{x}}_{t-1} + \eta\hat{\mathbf{x}}_t, \quad (7b)$$

where η is the learning rate and t denotes frame index of image sequences, how to determine the value of η will be discussed in Sect. 3.5. This puts more weight on recent frames and lets the effect of previous frames decay exponentially over time.

3.2 Discriminative Scale Space Tracking

Our scale searching scheme follows the DSST [17] tracker. Unlike SAMF [18] uses one filter to determine translation and scale simultaneously, the DSST applies two kind of correlation filters. One is two-dimensional translation filter for target localization, and the other one is one-dimensional scale filter for scale evaluation, which are independent of each other.

In our trackers, KCF is employed to locate the target, and learn a separate 1-dimension filter to estimate the scale information. Tens of patches I_n of size $a^n P \times a^n R$ are extracted centered around the target to construct the training sample $\mathbf{x}_{t,\text{scale}}$, here $P \times R$ denotes the target size in the current frame and S is the size of the scale filter, $n \in \left\{\left\lfloor -\frac{S-1}{2} \right\rfloor, \dots, \left\lfloor \frac{S-1}{2} \right\rfloor\right\}$, a represents the scale factor between feature layers. The aim is to train a scale correlation filter $\mathbf{h}_{\text{scale}}$ consisting of one filter $\mathbf{h}_{\text{scale}}^n$ per scale level, this can be solved by minimizing the L^2 error compared the desired output g , here 1-dimensional Gaussian is adopted,

$$\varepsilon = \left\| g - \sum_{n=1}^S \mathbf{h}_{\text{scale}}^n \star \mathbf{x}_{\text{scale}}^n \right\|^2 + \lambda_2 \sum_{n=1}^S \|\mathbf{h}_{\text{scale}}^n\|^2, \quad (8)$$

here, the $\star$ denotes circular correlation, and the second term is a regularization with a weight parameter λ_2 .

The value $\mathbf{x}_{\text{scale}}^n$ of the training samples $\mathbf{x}_{\text{scale}}$ at scale level n , is set to the d -dimensional feature descriptor of I_n . The solution to the problem above is as follows:

$$\hat{\mathbf{h}}_{\text{scale}}^n = \frac{\hat{g}^* \hat{\mathbf{x}}_{\text{scale}}^n}{\sum_{k=1}^S (\hat{\mathbf{x}}_{\text{scale}}^k)^* \hat{\mathbf{x}}_{\text{scale}}^k + \lambda_2}, \quad n = 1, \dots, S, \quad (9)$$

where the fraction denotes pointwise division. Similar to Eq. (7), new sample $\mathbf{x}_{t,\text{scale}}$ is used to update the numerator A_t^n and denominator B_t^n of the scale filter $\mathbf{h}_{t,\text{scale}}$.

$$A_t^n = (1 - \theta)A_{t-1}^n + \theta \hat{g}^* \hat{\mathbf{x}}_{t,\text{scale}}^n, \quad n = 1, \dots, S, \quad (10a)$$

$$B_t = (1 - \theta)B_{t-1} + \theta \sum_{k=1}^S (\hat{\mathbf{x}}_{t,\text{scale}}^k)^* \hat{\mathbf{x}}_{t,\text{scale}}^k. \quad (10b)$$

Here, θ is a learning rate parameter. Many numerical experiments show that $\theta = 0.01$ makes filter quickly adapt to scale variation while still maintaining robust.

To apply the filter in a new frame t , a test sample $\mathbf{z}_{t,\text{scale}}$ is extracted from the location determined by KCF using the same procedure as for the training sample $\mathbf{x}_{t,\text{scale}}$. In Fourier domain, the correlation scores $\hat{\mathbf{f}}_{t,\text{scale}}$ are computed by

$$\hat{\mathbf{f}}_{t,\text{scale}}(\mathbf{z}_{t,\text{scale}}) = \frac{\sum_{n=1}^S (A_{t-1}^n)^* \hat{\mathbf{z}}_{t,\text{scale}}^n}{B_{t-1} + \lambda_2}, \quad (11)$$

where A_{t-1}^n and B_{t-1} are the numerator and denominator of the scale filter updated in $(t-1)$ -th frame. By maximizing the scale correlation score $\hat{\mathbf{f}}_{t,\text{scale}}(\mathbf{z}_{t,\text{scale}}) = \mathcal{F}^{-1}\left\{\hat{\mathbf{f}}_{t,\text{scale}}(\mathbf{z}_{t,\text{scale}})\right\}$, the relative change in scale is obtained compared to the previous frame.

3.3 Multi-experts Construction

HOG [28], gray and ColorNames [29] (CN) are the most popular handcrafted features in tracking filed for their high extraction efficiency. The HOG features, called histogram of oriented gradient, are constructed by calculating and counting the gradient direction histogram in the local region of the image, which reflect the edge shape information of region block. CN features have rich expressiveness and high identification. They are obtained by transforming RGB space to CN space which can reflect the 11-dimensional thematic color information of the region [29]. The gray features are simple features that contain only brightness information. Different from handcrafted features, CNN features contain rich high-level semantic information and are strong at distinguishing objects of different categories.

In our handcrafted features version tracker, only two low-level features (HOG and CN) are adopted to build the experts. Diversity is crucial in ensemble methods [26], to create more experts, 32-dimension HOG is divided into two 16-dimension features, called HOG_1 and HOG_2 . Through permutation and combination of these features, $C_3^1 + C_3^2 + C_3^3 = 7$ experts are obtained in the feature bucket. In our deep features version tracker, HOG, *conv4-4* and *conv5-4* are extracted as low, middle and high features, respectively. Details about the experts is shown in Table 1. After these seven experts are generated, results (bounding boxes) are provided from their own perspective, how to choose a reliable one will be discussed in Sect. 3.4.

3.4 Ensemble Tracking

In each frame, seven experts track the target and generate cues (bounding boxes) simultaneously, inspired by MCCT [24], pair-evaluation and self-evaluation [24] are adopted to evaluate their robustness degree. And then, the expert with the highest score is selected as the tracking result of the current frame. The procedure of ensemble tracking framework is shown in Fig. 2.

3.4.1 Pair-Evaluation

E_1 to E_7 denote Expert 1 to Expert 7 respectively. Every expert is treated as a black box, the bounding box of Expert i in the t -th frame is written as B'_{E_i} , which is a four-

Fig. 2 Schematic diagram of ensemble tracking framework. Nodes in the hexagon represent the output of experts (bounding box). In a frame, the pair evaluation between two experts is marked with a light gray line. The pair evaluation of the same experts is marked with colorful line

Table 1 Our trackers both consist of 7 experts, the separation and combination of features increase the diversity of experts, each expert track from their perspective

Expert bucket	Handcrafted features version
Expert 1	HOG_1
Expert 2	HOG_2
Expert 3	ColorNames
Expert 4	HOG_1, HOG_2
Expert 5	HOG_1 , ColorNames
Expert 6	HOG_2 , ColorNames
Expert 7	HOG_1, HOG_2 , ColorNames
Expert bucket	Deep features version
Expert 1	HOG
Expert 2	Middle
Expert 3	High
Expert 4	HOG, middle
Expert 5	HOG, high
Expert 6	Middle, high
Expert 7	HOG, middle, high

dimensional vector containing the position and target size. The overlap ratios $O_{(E_i, E_j)}^t$ between any two experts E_i and E_j in frame t computed as,

$$O_{(E_i, E_j)}^t = \frac{\text{Area}(B_{E_i}^t \cap B_{E_j}^t)}{\text{Area}(B_{E_i}^t \cup B_{E_j}^t)}, \quad (12)$$

where, Area represents the area of the intersection or union of the bounding boxes. To reduce the gap between the high and low value of $O_{(E_i, E_j)}^t$, it is converted to nonlinear by adopting exponential function

$$O_{(E_i, E_j)}^t = \exp\left(-\left(1 - O_{(E_i, E_j)}^t\right)^2\right). \quad (13)$$

The fluctuation extent of overlap ratios in a short period Δt (e.g., 5 frames in [24]) reveals the stability of overlap evaluation between E_i and other experts. However, $O_{(E_i, E_j)}^t$ only represents the overlap ratio between two experts in present t -th frame, thus previous value of overlap ratio should be taken into consideration, the variance of the $K = 7$ experts is given as follows,

$$V_{E_i}^t = \sqrt{\frac{1}{K} \sum_{j=1}^K \left(O_{(E_i, E_j)}^t - \overline{O_{(E_i, E_j)}^{t-\Delta t+1:t}} \right)^2}. \quad (14)$$

Here, $\overline{O_{(E_i, E_j)}^{t-\Delta t+1:t}} = \frac{1}{\Delta t} \sum_{\tau} O_{(E_i, E_j)}^{\tau}$ and $\tau \in [t - \Delta t + 1, t]$.

The $M_{E_i}^t = \frac{1}{K} \sum_{j=1}^K O_{(E_i, E_j)}^t$ is the mean value of overlap ratios, reflects the consistency between Expert i and other experts. In a short period, the closer to the current frame, the greater the relationship between this score and the current frame. Thus, an increasing sequence $\mathbf{W} = \{\rho^0, \rho^1, \dots, \rho^{\Delta t-1}\}$, ($\rho > 1$) put more confidence on recent scores. Then the average weighted mean and standard variance are computed through: $M_{E_i}^t = \frac{1}{N} \sum_{\tau} W_{\tau} M_{E_i}^{\tau}$ and $V_{E_i}^t = \frac{1}{N} \sum_{\tau} W_{\tau} V_{E_i}^{\tau}$, respectively, and W_{τ} denotes the $(\tau - t + \Delta t)$ -th element in sequence in $\mathbf{W}$, and N is defined by $N = \sum_{\tau} W_{\tau}$.

The pair-evaluation score of Expert i in t -th frame is computed as below,

$$R_{\text{pair}}^t(E_i) = \frac{M_{E_i}^t}{V_{E_i}^t + \xi}, \quad (15)$$

the existence of small constant ξ is to avoid the pair-wise robustness score infinity when the denominator is zero. Equation (15) indicates that, a higher value of $R_{\text{pair}}^t(E_i)$ means greater consistency and less volatility between different experts.

3.4.2 Self-Evaluation

The Euclidean distance between the bounding box $B_{E_i}^{t-1}$ in $(t-1)$ -th frame and the one $B_{E_i}^t$ in t -th frame reflects the reliability of the tracking output of each expert, which is defined by $D_{E_i}^t = \left\| c(B_{E_i}^{t-1}) - c(B_{E_i}^t) \right\|$, Here, $c(B_{E_i}^t)$ is the central coordinate of $B_{E_i}^t$. The trajectory smoothness degree of Expert i is given as follows,

$$S_{E_i}^t = \exp\left(-\frac{1}{2\sigma_{E_i}^2} (D_{E_i}^t)^2\right), \quad (16)$$

where $\sigma_{E_i}^2 = \frac{1}{2} [W(B_{E_i}^t) + H(B_{E_i}^t)]$, $W(B_{E_i}^t)$ and $H(B_{E_i}^t)$ denote the width and height of the Expert i .

As mentioned before, to avoid performance fluctuation of the experts, scores in the short term should be considered. Thus, the self-wise expert trajectory smoothness score is given by $R_{\text{self}}^t(E_i) = \frac{1}{N} \sum_{\tau} W_{\tau} S_{E_i}^{\tau}$, again $N = \sum_{\tau} W_{\tau}$. The higher self-evaluation score means the better reliability of the tracking trajectory.

3.4.3 Joint Decision

The final robustness score $R_{E_i}^t$ of the Expert i in frame t requires self evaluation score $R_{\text{self}}^t(E_i)$ and pair-evaluation score $R_{\text{pair}}^t(E_i)$ to be weighted by the coefficient μ :

$$R^t(E_i) = \mu \cdot R_{\text{pair}}^t(E_i) + (1 - \mu) \cdot R_{\text{self}}^t(E_i), \quad (17)$$

finally, the expert with the highest final robustness score is selected as the output result in each frame.

The main advantage of this ensemble method is that, only twice feature extraction (the heaviest computational burden in the tracking process) is needed, one for the training by Eq. (4), the other for detection by Eq. (6), instead of fourteen ($K = 7, 7 \times 2 = 14$) in each frame. This approach considers both diversity and effectiveness, thus our trackers can maintain real-time performance while achieving high accuracy. Furthermore, by sharing the rectified results of target position and model update, the drift and tracking failure of weak experts are effectively alleviated.

3.5 A Novel Model Updater

The model updater determines both the strategy and frequency of model update, most of the existing trackers adopt two model update methods: (1) Schemes like [4, 6, 17, 31] update tracking models every frame with constant learning rate without considering whether the sample is credible or not. This may cause tracking failure due to the model corruption when the

target is detected inaccurately, severely occluded or totally missing in the current frame; (2) Approaches like [7, 27] uses indicators (PSR and APCE respectively) to assess the fluctuation of the response map, and update the model when the indicator meets a certain conditions. This method alleviates tracking failure caused by corruption of the model, however, the learning rate of the model updater is still constant, unable to fully adapt to the needs in some particular scenes. In addition, models that are discarded since they don't meet the conditions can be valuable. In our trackers, we establish an adaptive strategy to update by utilizing the feedback of tracking results.

It is demonstrated through many experiments that the number and value of the peaks of the response map can reflect the confidence of the tracking results. The ideal response map should have only one sharp peak, and the other area are relatively flat. The sharper peak can get better tracking accuracy. On the contrary, there are more than one peaks in the response map and the fluctuations are severe, whose pattern is significantly different from ideal response maps. If the updater still adopts the same learning rate, the model corruption will lead to tracking failure. Therefore, we propose a feedback adaptive updating mechanism with a criterion, called peaks correlation of response map (PCRM), which is defined as follows,

$$\text{PCRM} = \frac{(\mathbf{f}_{\max} - \mathbf{f}_{\min})^2}{\text{mean}\left(\sum_{i=1}^d (\mathbf{f}_{\text{peaks}}^i - \mathbf{f}_{\min})^2\right)}, \quad (18)$$

where $\mathbf{f}_{\max}$ and $\mathbf{f}_{\min}$ denote the maximum and minimum of the response map $\mathbf{f}_{\text{tran}}(\mathbf{z})$ in Eq. (6) respectively, and $\mathbf{f}_{\text{peaks}}^d$ denotes all peaks value in response map, $\mathbf{f}_{\text{peaks}}^i \in \{\mathbf{f}_{\text{peaks}}^1, \dots, \mathbf{f}_{\text{peaks}}^d\}$, d is the number of peaks. PCRM reflects the fluctuation degree of response map and the confidence level of detected target. When the target appears completely and obviously in the detection area, the response map is similar to that of a cone with a sharp peak and a smooth descent to a relatively flat area, and the PCRM will become larger. Otherwise, PCRM will significantly decrease if the object is occluded or missing.

Seven response maps are generated from seven experts, due to they are repetitive, only the first three experts are taken to compute the weighted PCRM of different features to evaluate the t -th tracking result:

$$\text{PCRM}^t = \sum_i v_i \text{PCRM}_{E_i}^t, \quad (19a)$$

$$v_i = \frac{\text{PCRM}_{E_i}}{\text{PCRM}_{E_1} + \text{PCRM}_{E_2} + \text{PCRM}_{E_3}}. \quad (19b)$$

Here, $i = 1, 2, 3$, PCRM_{E_i} denotes the PCRM values of Expert i response map. Compared with the average PCRM,

the weighted PCRM can reflect the overall fluctuation more comprehensive.

When occlusion or severe deformation occurs, as mentioned above, the PCRM drops down rapidly, and at the same time it can be seen in our experiment that the average robustness score of the experts $R_{\text{mean}}^t = \frac{1}{K} \sum_{i=1}^K R^t(E_i)$ decreases significantly as well, which indicates that the experts have divergence when they encounter unreliable samples. By integrating weighted PCRM and average robustness score, a comprehensive criterion $\text{SC}^t = \text{PCRM}^t \cdot R_{\text{mean}}^t$ is presented, called sample confidence score. Considering that KCF learns the target and background of the sample through dense sampling, even unreliable samples have their own value, it is unreasonable to discard them directly. When the current sample confidence score SC_t much far less than its past mean value $\text{SC}_{\text{mean}}^{1:t} = \frac{1}{t} \sum_{i=1}^t \text{SC}^i$, the learning rate η in Eq. (7) is determined as follows:

$$\eta = \begin{cases} \text{lr} & \text{if } \text{SC}^t > \alpha \cdot \text{SC}_{\text{mean}}^{1:t}, \\ \text{lr} \cdot [\text{SC}^t / (\alpha \cdot \text{SC}_{\text{mean}}^{1:t})]^\beta & \text{otherwise,} \end{cases} \quad (20)$$

where lr is the constant learning rate in original KCF, α and β are the confidence threshold and the power exponent of the power function, respectively. This update system can effectively prevent the tracking failure caused by penalizing samples with low sample confidence score.

Figure 3 illustrates the mechanism of the proposed update strategy. As shown in the Fig. 3, in the beginning, the response map shows the one ideal sharp peak without the occlusion of the target, while there are many low-energy regional peaks around it, so that the value of PCRM is relatively large, and the model is updated with medium learning rate η . When the target occluded severely, the response map fluctuates fiercely in the second row, so that PCRM drops to 5.84, the learning rate is computed as $\eta = 3.23 \times 10^{-4}$ adaptively. It should be noted that, under this circumstance the unreliable samples which may contain valuable information for later tracking are not simply discarded. Therefore, by combining PCRM and the historical robustness score, the model will be updated with low learning rate in this frame under the proposed strategy. Then the tracking model is not corrupted and the target can be tracked successfully in the subsequent frames. Figure 4 intuitively shows the PCRM and learning rate distribution on the *basketball* sequence. The athlete is fully and partially occluded in 17-th and 54-th frame respectively, the corresponding PCRM and learning rate value drops to low points. The subsequent low points indicate that, the proposed model update strategy can also react timely to rotation, deformation, illumination variation and background clutters. To validate the effectiveness of our model updater, more experiments will be conducted in the following section.

Fig. 3 The left column is the snapshots of image sequence *Basketball*, where the red bounding box indicates the tracking results of our trackers, the yellow one indicates the searching area shared by all experts. Seven response maps is generated for seven experts in each frame, for simplicity, only response maps of Expert 2 in our hand-crafted version tracker are shown in the right column, and the target and a part of regional peaks in response map is pointed out in the left column

An overview of our trackers is summarized in Algorithm 1.

Algorithm 1 Iteration in the t -th frame in our algorithm.

Input: Image I_t ; Previous target position p_{t-1} and scale s_{t-1} ; Scale model A_{t-1} , B_{t-1} ; Translation model α_{t-1} , $\mathbf{x}_{t-1}$.

Output: Estimated target position p_t and scale s_t ; Scale model A_t , B_t ; Translation model α_t , $\mathbf{x}_t$.

- 1: **for every** t **do**
- 2: Extract translation samples $\mathbf{z}$ for seven experts from I_t with size s_{t-1} at position p_{t-1} .
- 3: Compute the seven response maps using (6).
- 4: Experts joint decision using section 3.4, select an expert with highest robustness score.
- 5: Set p_t to the target position that maximizes the response map of the expert is selected.
- 6: Extract scale samples $\mathbf{z}_{t,scale}$ from I_t at p_t with size s_{t-1} .
- 7: Compute correlation scores $\hat{f}_{t,scale}(\mathbf{z}_{t,scale})$ scale using (11).
- 8: Set s_t to the scale that maximizes $\mathbf{f}_{t,scale}(\mathbf{z}_{t,scale})$.
- 9: Extract samples $\mathbf{x}$ and $\mathbf{x}_{t,scale}$ at p_t with size s_t .
- 10: Compute learning rate η in (7) using section 3.5.
- 11: Train translation and scale filters using (4) and (9) respectively.
- 12: Update translation and scale filters model using (7) and (10) respectively.
- 13: **end for**

4 Experiments

In this section, comprehensive experiments are employed to evaluate our method. Firstly, the implementation details of our trackers are described. Secondly, the effectiveness of the model updater in our trackers is validated by comparing

Fig. 4 Analysis of PCRM and learning rate on the *basketball* sequence. Blue and orange curves represent the value of PCRM and learning rate, respectively. The text in the plot indicates the status and frame of the target. See the snapshots in the second column for specific status

other approaches. Finally, our trackers are compared with state-of-the-art trackers.

We first conduct experiments on two benchmark datasets, OTB-2013 [30] and OTB-2015 [12]. The former has 51 video sequences, and the latter extends to 100. All these sequences are annotated with 11 attributes which cover various challenging factors, including illumination variation (IV), motion blur (MB), deformation (DEF), fast motion (FM), out-of plane rotation (OPR), scale variation (SV), occlusion (OCC), background clutters (BC), out-of-view (OV), in-plane rotation (IPR), low resolution (LR).

Two indicators are used: success plot and precision plot. The success plot represents the percentage of successful frames whose overlap rate between the tracked bounding box and the ground-truth. The precision plot is defined as the percentage of frame in which average distance (in pixel) between the output bounding box and the ground-truth is less than the given threshold. To rank the trackers, two types of ranking metrics are used as: the representative precision score at threshold = 20 for the distance precision plot (DP), and the area under the curve (AUC) metric for the success plot.

For fair evaluation, the third dataset VOT2015 [13] is also used, which contains 60 annotated sequences.

4.1 Implementation Details

The regularization parameters in Eqs. (4) and (9) are set to $\lambda_1 = 0.0001$ and $\lambda_2 = 0.01$, respectively. The learning rate in Eq. (10) is set to $\theta = 0.025$. Number of scales S is 33 and scale factor a is 1.02, respectively. For ensemble tracking, the parameter ρ in the weigh sequence $\mathbf{W}$ is set to 1.1, the weighting factor μ is set to 0.1. In model updater, α and β in Eq. (20) are set to 0.6 and 3, respectively. All experts adopt the same parameters. All mentioned parameters are shown in Table 2.

Our experiments are implemented in MATLAB 2019a on a computer with Intel I5-3450 3.1 GHz CPU and 16 GB RAM. The MatConvNet toolbox [42] is used for extracting the deep features from VGG-19 [8]. Our deep features version trackers run at about 1.5 FPS in OTB *basketball* sequence. The speed of our handcrafted features version tracker is about 25 FPS in the same sequence, which is sufficient for real-time applications.

4.2 Analyses of Our Trackers

To evaluate the effectiveness of each component in our framework, we compare our trackers with different versions of itself on OTB-2013 and OTB-2015. Our trackers are denoted as Ours and Ours_deep, we first compare our trackers with the Expert 7, which denoted as Expert7. Then to demonstrate the effect of the update mechanism, some

popular methods are embedded into our trackers, such as PSR in [7], APCE in [27] and interpolation from original KCF [6], which are denoted as Our_with_PSR, Our_with_APCE and Our_with_Interpolation respectively. In all compared trackers, only Ours_deep adopts CNN features.

As shown in Fig. 5, our trackers Ours and Ours_deep show the best tracking accuracy and robustness in both OTB-2013 and OTB-2015 datasets. Ours outperform the Expert7 obviously, it is worthy to mention that, Expert7 has achieved a quite sufficient, which illustrates that our ensemble method improves its performance about 5% higher. Besides, Ours_with_Interpolation adopts the constant learning rate to update the model every frame by Eq. (7), Ours_with_APCE and Our_with_PSR simply discard the unreliable sample, which may value to the tracker, due to the limitations of these three kinds, all of them get poor performance both in precision and success. However, our novel strategy by considering the fluctuation degree of the response map and divergence among seven experts boosts the performance further.

4.3 Comparison with the State-of-the-Art

Beside tracking results provided by the benchmark [30], we compare our trackers with stat-of-the-art trackers as well, including MDNet [11], ECO [35], C-COT [34], ADNet [43], SRDCFdecon [44], ACFN [45], CNN-SVM [46], DLSSVM [47], SiamFC-tri [48], MEEM [20], STAPLE [31], KCF [6], DSST and fDSST [17], LCT [21], SAMF [18], DCFNet [49], SRDCF [32]. It should be noted that, where MDNet [11],

Fig. 5 Performance comparison of different versions of our trackers on the OTB-2013 and OTB-2015 datasets. The evaluations on OTB-2013 are in the upper row, on OTB-2015 are in the second row. In the legend, the DP at a threshold of 20 pixels and AUC are reported in the left and right figures, respectively

ECO [35], C-COT [34], ADNet [43], ACFN [45], CNN-SVM [46], DLSSVM [47] and SiamFC-tri [48] are based on deep learning.

4.3.1 OTB-2013 Dataset

In general, according to the evaluation metrics by OTB-2013, the one pass evaluation (OPE) score in precision and success plots are shown in Fig. 6. The “deep” in the brackets of the legend represents that the tracker is based on deep learning. As shown in the plots, our approach achieves promising results compared to many advanced trackers. Ours scores reach 64.3% success rate and 84.4% precision rate. With the help of CNN features, Ours_deep achieves 67.5% success rate and 89% precision rate, ranks fourth and fifth respectively among all the compared trackers. As the baseline of our trackers, the KCF obtains 51.4% success rate and 74.0% precision rate as they reported, meanwhile, our method employs the scale estimation from DSST, which gets 56.5% success rate and 75.4% precision rate. These observations indicate that the proposed framework works better than the original two. Particularly, MEEM similar to our approach is also based on historical tracker ensemble. Our ensemble mechanism exceeds the MEEM significantly by 7.7% of the AUC score and 1.4% of the DP score. The proposed trackers also show comparable performance with the state-of-the-art trackers, MDNet [11], ECO [35], C-COT [34], ADNet [43] in both precision and success rate.

Table 2 Parameters in our trackers

Parameters	λ_1	λ_2	θ	S	a
Value	0.0001	0.01	0.025	33	1.02
Parameters	ρ	μ	α	β	
Value	1.1	0.1	0.6	3	

4.3.2 OTB-2015 Dataset

To further validate the effectiveness of our trackers, we conduct our experiment on relatively large dataset OTB-2015 containing 100 annotated targets, thus, OTB-2015 dataset is more comprehensive than its predecessor. As shown in Fig. 7, top fifteen trackers are colored in plots, and DP scores for precision and AUC scores for success are reported in the legends. The proposed tracker Ours_deep achieving DP score of 88.3% and AUC score of 66.3%, ranks fourth in both criteria. As for the scores of Ours are only lower than deep features based trackers, ECO, MDNet, C-COT and ADNet, however, also ranks higher than MEEM in both plots. It is worth mentioning that in this more comprehensive evaluation, our handcrafted features version method provides a gain 19.4 and 20.1% in DP score, 28.9 and 18.7% in AUC score compared to the KCF and DSST respectively. This demonstrates the effectiveness and validity of our framework again. In general, the proposed trackers have demonstrated competitiveness on the OTB benchmark.

Fig. 6 The success plots and precision plots on OTB-2013 dataset, 50 image sequence with 51 targets are quantitatively analyzed by using one pass evaluation (OPE), only trackers with scores in the top fifteen

shown up, and the other trackers are plotted in light gray curves. The legend illustrates the area under the curve (AUC) for the success plot, and the percentage of the threshold 20 (DP) for the precision

Fig. 7 The success plot and precision plot on OTB-2015 dataset with 100 image sequences, the DP and the AUC scores are illustrated in the left and right plots, respectively. The trackers after the fifteenth

is drawn in light gray in the plots. In both metrics, our approach has achieved the best results among the compared trackers

4.3.3 Attribute-Based Comparison

We further use the image sequences annotated by eleven attributes to comprehensively evaluate the performance of tracker in different scenarios. Figure 8 shows the AUC scores for eleven different attributes, since the AUC score measures the tracker performance more accurately than DP that is with one threshold. For clarity, the results for top fifteen trackers are reported in the legend. As illustrated in the plots, the proposed trackers achieve excellent results on most of attributes. In sequences annotated with the scale variation attribute, our handcrafted features approach outperforms the DSST, due to the joint decision strategy in our high discriminative ability kernelized correlation filters. Moreover, our trackers are at the forefront in the three attributes of occlusion, out of view and background clutters, this shows that the proposed model updater mechanism boosts the performance much higher in these three distractive scenarios. In addition, targets in sequences annotated out-of-plane rotation and in-plane rotation have multiple views, therefore, the strength and frequency of updating the model is particularly critical. Due to the proposed model updater, our trackers handle well in both attributes. Our approach provides favorable results in attributes of deformation, illumination variation, fast motion and motion blur as well.

4.3.4 Qualitative Evaluation

Here qualitative comparative experiments of our approach with other trackers are performed on twelve image sequences are shown in Fig. 9. In nine trackers, only KCF is incapable

of estimating scale variation. Among all the test sequences, *Dog1* and *Singer1* have significant scale variation. *Girl2*, *Jogging2*, *Skating2* go through part or whole occlusion. Targets in *MotorRolling* and *Biker* have the attributes of motion blur and fast motion. Target in *DragonBaby* are suffer from frequent appearance variations. *Diving*, *Bird1*, *Skiing*, *Surfer* and *KiteSurf* have sever deformation.

In *Dog1* and *Singer1*, both ACFN and LMCF suffer from a significant scale drift in the presence of fast scale change and illumination variation, while our approach performs well. Although Staple can adapt to scale variation and in-plane-rotation in *Dog1* and *Singer1*, it does not perform well in presence of occlusion, background clutters and fast motion in *Jogging2* and *DragonBaby*. In *Girl2*, when the adult completely blocks the girl, DSST and most other trackers are drifted by the occlusion. While our proposed model updater can avoid the model corruption, after the girl appears again, our trackers correct the drift and continues to track the real target. A similar phenomenon can also be observed in *Jogging2* and *Skating2*. This demonstrates the superior performance of our trackers not only due to ensemble tracking, but also due to the model update scheme. *Diving*, *Bird1*, *Skiing*, *MotorRolling* and *Biker* are the most challenging sequences in OTB, with the boost of CNN features, our deep version tracker can track the targets, and even perform better than ECO in *Bird1* and *MotorRolling*.

4.4 VOT2015 Dataset

For completeness, we also present the evaluation results on VOT2015 dataset [13], which contains 60 sequences

Fig. 8 The success plots for attribute-based evaluation of trackers on OTB-2015, the AUC scores for top fifteen trackers are reported in the legend. The number of videos related to the attribute is in parentheses above each plot

with substantial variations. Unlike the unsupervised OTB datasets, in the VOT2015 methodology, when the region overlap rate is below the threshold, the tracker is re-initialized. The evaluation reports accuracy and robustness, corresponding to the bounding box overlap rate and the

number of failures, respectively. Please refer to [13, 50] for more information.

For clarity, we compare our algorithm with part of trackers provided by the dataset. The overall experimental

Fig. 9 Comparison of the proposed method with the state-of-the-art trackers: ECO [35], ADNet [43], LMCF [27], MEEM [20], Staple [31], KCF [6] and DSST [17] on OTB-2015 over fifteen typical

sequences. From top-left to bottom-right, the video illustrations are: *Dog1*, *DragonBaby*, *Girl2*, *Jogging2*, *KiteSurf*, *Singer1*, *Skating2*, *Diving*, *Bird1*, *Skiing*, *MotorRolling*, *Biker*

results are illustrated in accuracy and robustness ranking plot, as shown in Fig. 10. The accuracy and failure, as well as expected overlap for dozens of competitive trackers are listed in Table 3. From the plot, it is observed that our deep version tracker resides in the top right corner, which means only MDNet (the VOT2015 winner) ranks higher than Our_deep. It is worth noting that our handcrafted features based tracker outperforms most of the compared trackers. Because relying on iterative optimization operators online, the speeds of MDNet and DeepSRDCF are even lower than 1 FPS, which is far from meeting the real-time requirements. However, the speed of Ours_depp is about 1.5 FPS, our handcrafted features version tracker can reach 25 FPS, which is much faster than the trackers mentioned above. In addition, the proposed method ranks higher than KCF, MEEM and DSST, which demonstrates the effectiveness of the proposed framework again.

5 Conclusion

In this paper, a multi-experts joint decision framework for visual tracking embedded with adaptive model updater is proposed, which fully explore the strength of multiple features not only in feature-level, but also in decision-level by using high discriminative power of kernelized correlation filters. Moreover, our trackers are extended with an effective scale estimation approach to address the problem of fixed template size. Furthermore, a novel criterion called peaks correlation of response map (PCRM) is proposed to assess confidence of sample through response map, and establish an adaptive model update strategy by considering both PCRM and historical robustness score of experts to alleviate the model corruption problem. Three widely used datasets are adopted to conducted extensive experiments. We compare our approach with state-of-the-art trackers

Fig. 10 The AR ranking plot for baseline. The accuracy and robustness rankings are plotted along the vertical and horizontal axis respectively. Our trackers are denoted by the red circle and yellow cross. The tracker is better if it is closer to the top right corner of the plot

Table 3 The table shows accuracy, the average number of failure and expected average overlap of state-of-the-art trackers on the VOT2015 [13]

Tracker	A-R rank		EAO
	Overlap	Failures	
Ours_deep	0.5617	8.8204	0.2800
Ours	0.5375	17.1979	0.2118
MDNet	0.5862	7.0173	0.3951
OACF	0.5864	19.4397	0.2073
DeepSRDCF	0.5515	10.9695	0.2947
EBT	0.4486	7.2362	0.3374
srdef	0.5326	15.3944	0.2675
LDP	0.4820	13.4005	0.2535
SODLT	0.5521	18.9969	0.2115
NSAMF	0.5141	15.3700	0.2236
s3tracker	0.5037	14.6671	0.2232
MEEM	0.4620	15.1711	0.2405
DAT	0.4565	15.9841	0.2486
RobStruck	0.4683	15.0813	0.2369
MCT	0.4610	16.7442	0.2132
HMMTxD	0.5192	20.2657	0.2036

red: rank1, blue: rank2, green: rank3.

on OTB-2013 and OTB-2015, the results show the effectiveness and validity of components in our trackers. The proposed trackers are at the front in most different kind of evaluations. Our approach gains outstanding results on VOT2015 as well. The conducted experiments demonstrate the proposed trackers perform competitively against state-of-the-art approaches. It is worthy to emphasize that, the proposed approach not only performs superiorly, but also can run at high speed on average machines to meet real-time application scenarios.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61801340 and Grant 52075404, and in part by the Application Foundation Frontier Special Project of Wuhan Science and Technology Bureau under Grant 2020010601012176.

Declarations

Conflict of interest The authors declare that they have no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral histogram. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), vol. 1, pp. 798–805 (2006). <https://doi.org/10.1109/CVPR.2006.256>
- Jia, X., Lu, H., Yang, M.: Visual tracking via adaptive structural local sparse appearance model. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1822–1829 (2012). <https://doi.org/10.1109/CVPR.2012.6247880>
- Oron, S., Bar-Hillel, A., Levi, D., Avidan, S.: Locally orderless tracking. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1940–1947 (2012). <https://doi.org/10.1109/CVPR.2012.6247895>
- Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) Computer Vision—ECCV 2012, pp. 702–715. Springer, Berlin (2012)
- Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. IEEE Trans. Pattern Anal. Mach. Intell. **34**(7), 1409–1422 (2012). <https://doi.org/10.1109/TPAMI.2011.239>
- Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern

- Anal. Mach. Intell. **37**(3), 583–596 (2015). <https://doi.org/10.1109/TPAMI.2014.2345390>
7. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2544–2550 (2010). <https://doi.org/10.1109/CVPR.2010.5539960>
8. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
9. Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. **25**, 1097–1105 (2012)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer vision and pattern recognition, pp. 770–778 (2016). <https://doi.org/10.1109/CVPR.2016.90>
11. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4293–4302 (2016). <https://doi.org/10.1109/CVPR.2016.465>
12. Wu, Y., Lim, J., Yang, M.: Object tracking benchmark. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1834–1848 (2015). <https://doi.org/10.1109/TPAMI.2014.2388226>
13. M. Kristan et al.: The visual object tracking VOT2015 challenge results. In: IEEE International Conference on Computer Vision Workshop (ICCVW), pp. 564–586 (2015). <https://doi.org/10.1109/ICCVW.2015.79>
14. Ma, C., Huang, J., Yang, X., Yang, M.: Hierarchical convolutional features for visual tracking. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 3074–3082 (2015). <https://doi.org/10.1109/ICCV.2015.352>
15. Zhang, M., Xing, J., Gao, J., Hu, W.: Robust visual tracking using joint scale-spatial correlation filters. In: 2015 IEEE International Conference on Image Processing (ICIP), pp. 1468–1472 (2015). <https://doi.org/10.1109/ICIP.2015.7351044>
16. Huang, D., Luo, L., Wen, M., Chen, Z., Zhang, C.: Enable scale and aspect ratio adaptability in visual tracking with detection proposals. In: Proceedings of the British Machine Vision Conference (BMVC), pp. 185.1–185.12. BMVA Press (2015). <https://doi.org/10.5244/C.29.185>
17. Danelljan, M., Häger, G., Khan, F.S., Felsberg, M.: Discriminative scale space tracking. IEEE Tran. Pattern Anal. Mach. Intell. **39**(8), 1561–1575 (2017). <https://doi.org/10.1109/TPAMI.2016.2609928>
18. Li, Y., Zhu, J.: A scale adaptive kernel correlation filter tracker with feature integration. In: Agapito, L., Bronstein, M.M., Rother, C. (eds.) Computer Vision—ECCV 2014 Workshops, pp. 254–265. Springer International Publishing, Cham (2015)
19. Danelljan, M., Häger, G., Khan, F.S., Felsberg, M.: Accurate scale estimation for robust visual tracking. In: British Machine Vision Conference 2014 (2014). <https://doi.org/10.5244/C.28.65>
20. Zhang, J., Ma, S., Sclaroff, S.: MEEM: robust tracking via multiple experts using entropy minimization. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision—ECCV 2014, pp. 188–203. Springer International Publishing, Cham (2014)
21. Ma, C., Yang, X., Zhang, C., Yang, M.H.: Long-term correlation tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5388–5396 (2015). <https://doi.org/10.1109/CVPR.2015.7299177>
22. Avidan, S.: Ensemble tracking. IEEE Trans. Pattern Anal. Mach. Intell. **29**(2), 261–271 (2007). <https://doi.org/10.1109/TPAMI.2007.35>
23. Qi, Y., Zhang, S., Qin, L., Yao, H., Huang, Q., Lim, J., Yang, M.: Hedged deep tracking. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4303–4311 (2016). <https://doi.org/10.1109/CVPR.2016.466>
24. Wang, N., Zhou, W., Tian, Q., Hong, R., Wang, M., Li, H.: Multi-cue correlation filters for robust visual tracking, pp. 4844–4853 (2018). <https://doi.org/10.1109/CVPR.2018.00509>
25. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. In: Vitányi, P. (ed.) Computational Learning Theory, pp. 23–37. Springer, Berlin (1995)
26. Wang, N., Shi, J., Yeung, D., Jia, J.: Understanding and diagnosing visual tracking systems. In: 2015 IEEE international conference on computer vision (ICCV), pp. 3101–3109 (2015). <https://doi.org/10.1109/ICCV.2015.355>
27. Wang, M., Liu, Y., Huang, Z.: Large margin object tracking with circulant feature maps. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4800–4808 (2017). <https://doi.org/10.1109/CVPR.2017.510>
28. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, pp. 886–893 (2005). <https://doi.org/10.1109/CVPR.2005.177>
29. van de Weijer, J., Schmid, C., Verbeek, J., Larlus, D.: Learning color names for real-world applications. IEEE Trans. Image Process. **18**(7), 1512–1523 (2009). <https://doi.org/10.1109/TIP.2009.2019809>
30. Wu, Y., Lim, J., Yang, M.: Online object tracking: a benchmark. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2411–2418 (2013). <https://doi.org/10.1109/CVPR.2013.312>
31. Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., Torr, P.H.S.: Staple: Complementary learners for real-time tracking. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1401–1409 (2016). <https://doi.org/10.1109/CVPR.2016.156>
32. Danelljan, M., Hager, G., Khan, F.S., Felsberg, M.: Learning spatially regularized correlation filters for visual tracking. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 4310–4318 (2015). <https://doi.org/10.1109/ICCV.2015.490>
33. Wang, L., Ouyang, W., Wang, X., Lu, H.: Visual tracking with fully convolutional networks. In: The IEEE International Conference on Computer Vision (ICCV), pp. 3119–3127 (2015). <https://doi.org/10.1109/ICCV.2015.357>
34. Danelljan, M., Robinson, A., Shahbaz Khan, F., Felsberg, M.: Beyond correlation filters: learning continuous convolution operators for visual tracking. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision—ECCV 2016, pp. 472–488. Springer International Publishing, Cham (2016)
35. Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M.: Eco: efficient convolution operators for tracking. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6931–6939 (2017). <https://doi.org/10.1109/CVPR.2017.733>
36. Hong, Z., Zhe, C., Wang, C., Xue, M., Tao, D.: Multi-store tracker (muster): a cognitive psychology inspired approach to object tracking. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 749–758 (2015). <https://doi.org/10.1109/CVPR.2015.7298675>
37. Li, J., Deng, C., Xu, R.Y.D., Tao, D., Zhao, B.: Robust object tracking with discrete graph based multiple experts. IEEE Trans. Image Process. **26**, 2736–2751 (2017)
38. Santner, J., Leistner, C., Saffari, A., Pock, T., Bischof, H.: PROST: parallel robust online simple tracking. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 723–730. <https://doi.org/10.1109/CVPR.2010.5540145>
39. Tang, M., Feng, J.: Multi-kernel correlation filter for visual tracking. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 3038–3046 (2015). <https://doi.org/10.1109/ICCV.2015.348>

40. Montero, A.S., Lang, J., Laganière, R.: Scalable kernel correlation filter with sparse feature integration. In: 2015 IEEE International Conference on Computer Vision Workshop (ICCVW), pp. 587–594 (2015). <https://doi.org/10.1109/ICCVW.2015.80>
41. Schölkopf, B., Smola, A. J., Bach, F.: Learning with kernels: support vector machines, regularization, optimization, and beyond[M]. MIT press (2002)
42. Vedaldi, A., Lenc, K.: MatConvNet: convolutional neural networks for MATLAB. In: Proceedings of the 23rd ACM international conference on Multimedia, pp. 689–692 (2015). <https://doi.org/10.1145/2733373.2807412>
43. Yun, S., Choi, J., Yoo, Y., Yun, K., Choi, J.Y.: Action-decision networks for visual tracking with deep reinforcement learning. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1349–1358 (2017). <https://doi.org/10.1109/CVPR.2017.148>
44. Danelljan, M., Häger, G., Khan, F.S., Felsberg, M.: Adaptive decontamination of the training set: a unified formulation for discriminative visual tracking. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1430–1438 (2016). <https://doi.org/10.1109/CVPR.2016.159>
45. Choi, J., Chang, H.J., Yun, S., Fischer, T., Demiris, Y., Choi, J.Y.: Attentional correlation filter network for adaptive visual tracking. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4828–4837 (2017). <https://doi.org/10.1109/CVPR.2017.513>
46. Hong S, You T, Kwak S, et al. Online tracking by learning discriminative saliency map with convolutional neural network[C]// International conference on machine learning. PMLR, pp. 597–606 (2015)
47. Ning, J., Yang, J., Jiang, S., Zhang, L., Yang, M.: Object tracking via dual linear structured SVM and explicit feature map. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4266–4274 (2016). <https://doi.org/10.1109/CVPR.2016.462>
48. Dong, X., Shen, J.: Triplet loss in siamese network for object tracking[C]. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 459–474 (2018)
49. Wang, Q., Gao, J., Xing, J., Zhang, M., Hu, W.: DCFNet: discriminant correlation filters network for visual tracking (2017). [arXiv: 1704.04057](https://arxiv.org/abs/1704.04057)
50. Kristan, M., Matas, J., Leonardis, A., Vojir, T., Pflugfelder, R., Fernandez, G., Nebel, G., Porikli, F., Čehovin, L.: A novel performance evaluation methodology for single-target trackers. IEEE Trans. Pattern Anal. Mach. Intell. **38**(11), 2137–2155 (2016). <https://doi.org/10.1109/TPAMI.2016.2516982>

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