PLOS ONE

RESEARCH ARTICLE

An adaptive spatiotemporal correlation filtering visual tracking method

Yuhan Liu, He Yan*, Wei Zhang, Mengxue Li, Lingkun Liu

School of Artificial Intelligence, Chongqing University of Technology, Chongqing, China

* yanhe@cqut.edu.cn

OPEN ACCESS

Citation: Liu Y, Yan H, Zhang W, Li M, Liu L (2023) An adaptive spatiotemporal correlation filtering visual tracking method. PLoS ONE 18(1): e0279240. https://doi.org/10.1371/journal.pone.0279240

Editor: Nattapol Aunsri, Mae Fah Luang University, THAILAND

Received: July 7, 2022

Accepted: December 3, 2022

Published: January 6, 2023

Data Availability Statement: OTB dataset are available from the OTB database (website: http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html). VOT dataset are available from the VOT database (website: https://www.votchallenge.net/). GOT10K dataset are available from the GOT database (website: http://got-10k.aitestunion.com/).

Funding: This work is supported by the National Key R&D Plan "Intelligent Robots" Key Project of P.R. China (Grant No.2018YFB1308602), the National Natural Science Foundation of P.R. China

Abstract

Discriminative correlation filter (DCF) tracking algorithms are commonly used for visual tracking. However, we observed that different spatio-temporal targets exhibit varied visual appearances, and most DCF-based trackers neglect to exploit this spatio-temporal information during the tracking process. To address the above-mentioned issues, we propose a three-way adaptive spatio-temporal correlation filtering tracker, named ASCF, that makes fuller use of the spatio-temporal information during tracking. To be specific, we extract rich local and global visual features based on the Conformer network, establish three correlation filters at different spatio-temporal locations during the tracking process, and the three correlation filters independently track the target. Then, to adaptively select the correlation filter to achieve target tracking, we employ the average peak-to-correlation energy (APCE) and the peak-to-sidelobe ratio (PSR) to measure the reliability of the tracking results. In addition, we propose an adaptive model update strategy that adjusts the update frequency of the three correlation filters in different ways to avoid model drift due to the introduction of similar objects or background noise. Extensive experimental results on five benchmarks demonstrate that our algorithm achieves excellent performance compared to state-of-the-art trackers.

1 Introduction

Visual target tracking technology aims to locate a moving target of interest in a video image and then to capture the object's real-time position, motion state and trajectory information. At present, the direction of the considerable demand for tracking tasks involves online tracking of general objects without specific restrictions and requirements regarding the category, shape, tracking scene, and tracking target. In an actual tracking scene (intelligent traffic monitoring [1, 2], unmanned aerial vehicle (UAV) [3], search-and-rescue missions [4], etc.), the tracking target may experience appearance disturbances from the target background or the target itself, such as sudden changes in illumination, target deformation, similar colors between the target and the background, and target occlusion [5–7]. In addition, most trackers do not make reasonable use of spatio-temporal information during the tracking process. In general, strategies for utilizing limited spatio-temporal information to construct a reasonable target tracking

(Grant No.61173184), the Chongqing Natural Science Foundation of P.R. China (Grant No. cstc2018jcyjAX0694).

Competing interests: The authors have declared that no competing interests exist.

model while avoiding appearance interference during the tracking process is still an unsolved problem.

The discriminative correlation filter-(DCF) based method has attracted extensive attention because of its high accuracy and robustness during tracking. Due to the rapid development of deep learning, deep learning networks are being used to mine high-level semantic features in images to provide better target features for trackers. Most of the current correlation filter trackers use pre-trained convolutional neural networks (CNNs) to extract the targets' deep features, and after using deep features instead of traditional hand-crafted features, the trackers' performance significantly improved [8–13]. Another algorithm, the HCF [8] algorithm, uses the deep features of different layers to separately train correlation filters and perform coarse-to-fine fusion. However, HCT does not consider the temporal context during tracking. The MCCT [14] tracker extracts various types of features, trains different correlation filter models, and adaptively selects the model. However, it only uses the adjacent historical frame information to train the correlation filtering model, ignoring the long-distance context information. C-COT [9] and ECO [10] have also achieved very good performance during the same period; these two algorithms use continuous interpolation and filter for joint optimization, but the tracking accuracy cannot be improved by using better deep features [15]. ATOM [16] uses the hard negative mining strategy [17] to update the template while adjusting the learning rate so that the tracking model can quickly adapt to the influence of interference, but the algorithm ignores the impact of occlusion on the training sets. It is therefore easy to mistakenly identify the occluded objects as the tracking target. DiMP [18] removes the hard negative mining strategy, uses a fast update strategy, and performs two recursive optimizations every 20 frames to refine the target model. However, a fixed model update frequency may introduce a large number of meaningless negative samples, which reduces the generalization ability of the model and severely affects the discriminative power of the classifier. Furthermore, contaminated positive samples may cause model degradation and lead to tracking drift. DiMP simply uses the historical frame information to build the tracking model and does not make use of long-distance spatio-temporal information during the tracking process, which may cause the model to lack the ability to effectively handle global context information in complex scenes. Finally, it may lose tracked targets due to tracking challenges such as large deformation of the target body, occlusion by similar objects, and disappearance from the field of view [19].

Aiming at the above problems, we propose a new adaptive target tracking algorithm based on the spatio-temporal correlation filter. Its main idea is to use the Conformer network [20], which performs better in the transformer network [21–23], to extract the tracking target features. Then, to model tracking targets from a different time and space, the initial tracking target is used as the initial template, the current frame tracking search area is used as the search template, and the dynamic target during the tracking process is used as the dynamic template. The three templates train the corresponding correlation filter tracking model to realize three-way tracking; to reasonably select the corresponding model for tracking in different environments, we combine the average peak-to-correlation energy (APCE) [24], peak-to-sidelobe ratio (PSR) [25], and trajectory smoothness degree as the tracking confidence evaluation index and select the tracking result with the best evaluation. At the same time, to quickly adapt to the interference, the tracking state is judged by the tracking confidence proposed in this paper, which adaptively judges whether the model needs to be updated and changes the model's learning rate. In addition, the Conformer network's image classification ability is used to control the dynamic template update.

The main contributions of this paper are summarized as follows:

- We propose a dynamic template update method that can flexibly obtain the target's spatial information and compensate for the defect that some correlation filter tracker only uses historical frame information to make predictions.
- To improve the model's distractor discrimination, we propose an adaptive model update strategy that further exploits the valuable samples selected by the PSR, APCE, and trajectory smoothness degree.
- In contrast to most of the existing DCF-based tracking methods, which use only single-way correlation filters to achieve tracking (i.e., a single tracking result), we propose a three-way tracking algorithm that enables the tracker to flexibly address appearance changes and geometric deformations of the tracked target over time.

To demonstrate the effectiveness of the tracking framework proposed in this paper, we conduct extensive experiments on the following object tracking evaluation datasets: VOT2020 [26], GOT-10K [27], OTB2015 [28], OTB2013 [29] and LBT50 [30]. The experimental results show that our proposed (ASCF) tracker exhibits excellent performance on four benchmarks.

2 Related works

In this section, we review related work on template updating and spatio-temporal information in trackers, and briefly review recent DCFs-based trackers.

2.1 Correlation filter tracking

All correlation filter trackers are online training tracking models. As the first CF-based tracker, MOSSE [25] has high accuracy and achieves the fastest tracking speed. HDT [31] introduced deep learning into correlation filtering, and an algorithm that adaptively changes the weight of the filter under each scale feature was designed. [9] and ECO [10], which are representative correlation filter trackers, achieved very good performance over the same period, but they did not achieve further performance gains when using deeper networks. In [15], scholars investigated how to use deeper networks to improve the accuracy and robustness of correlation filter trackers, and to solve the problem of target scale transformation during the tracking process, scholars proposed CFML [32]. In [33], a background-aware correlation filter model with saliency regularization is established to address boundary effects in correlation filter tracking, and another model, CFNet [11], was proposed to embed correlation filtering into a two-way network for end-to-end training and learning. The research team who proposed ECO [10] drew on the advantages of end-to-end trackers such as CFNet and proposed the use of a better gradient descent method to learn a convolution kernel. This strategy is similar to the correlation filter but has the ability to distinguish foreground and background [16, 18], and its performance made it the current state-of-the-art model.

In this work, ECO is selected as our baseline method. Different from the DCF-based trackers mentioned above, we propose a three-way tracker that builds correlation filters in different spatio-temporal areas to obtain multi-tracking results.

2.2 Tracking model update

To adapt to the target's appearance changes, visual tracking algorithms generally adopt the model update strategy. However, there are also trackers [8, 34] that do not use a model update strategy and build a tracking model by using only the initial frame. This type of tracker is prone to tracking drift when the target in the search area undergoes large deformation. ECO [10] proposes a sparse model update strategy and sets a fixed update interval, while LMCF [24]

proposes the APCE metric to determine the tracking accuracy and thus uses it to decide whether to update the model. However, APCE may introduce negative samples into the model due to inaccurate judgments about the updates, and many trackers [9, 11, 14, 35–38] that use deep learning networks to extract features ignore the image classification capabilities of the pretrained networks they use.

In contrast, we propose an adaptive model update strategy that can better solve the drift phenomenon during tracking by judging whether the model needs to be updated by combining the pre-training training confidence and feature extraction networks.

2.3 Spatio-temporal information in visual tracking

The global context information in the target tracking task includes both temporal information and spatial information. Temporal information involves tracking object state changes across frames; spatial information involves object appearance information and nearby background information. The recently popular offline Siamese trackers [34, 36, 39–41] use only spatial information for tracking and achieve target tracking by using the initial template and the current search area for module matching. In [42], scholars proposed a novel spatial-channel selection and temporal regularized correlation filter (SCSTCF) model that adds spatial-channel constraints to select features along the spatial and channel dimensions, and some trackers [33, 43] enhance the spatio-temporal contextual connections by introducing spatio-temporal saliency. BSTCF [44] introduces background constraints and spatio-temporal regularization to solve the problem that the object background of the traditional CF model is not modeled over time. The CACF [45] tracker, which combines temporal information and spatial information, adds the background near the target to the filter's learning and introduces spatial templates to the correlation filtering, which better solves the boundary effect [46]. Trackers that combine spatio-temporal information additionally utilize temporal or spatial information to improve the gain, and later works [47–50] have achieved higher robustness.

Although trackers have made some progress in terms of making full use of spatio-temporal information in recent years, most trackers use convolutional features, which have a limited receptive field and lack the ability to model long-distance spatial relationships. We use the Conformer network instead of convolutional neural networks to extract tracking object features, use the self-attention mechanism to capture long-distance spatial relationships [23], and capture the tracking target's appearance changes and background changes through dynamic templates to increase the utilization of spatial information.

3 Proposed visual tracking methods

In this section, we describe the proposed ASCF algorithm in detail. ASCF adopts the Conformer backbone network for feature extraction and constructs three-way parallel correlation filters for tracking, as shown in Fig 1. Unlike most deep correlation filter trackers, ASCF can extract global and local features due to our choice of backbone network; at the same time, ASCF uses the Conformer [20] network's classification ability to judge the state of dynamic templates and correlation filter models for updating. Three different correlation filters are trained by different spatio-temporal target features, and then, the best correlation filter model is selected for tracking through joint evaluation by the APCE and PSR. Different tracking models are reasonably selected for different tracking object states to improve the tracking accuracy.

Through the above design, our tracker can reasonably select the correlation filtering model for tracking according to the tracking scene and update the model at the appropriate time to reduce the error caused by invalid model updates.

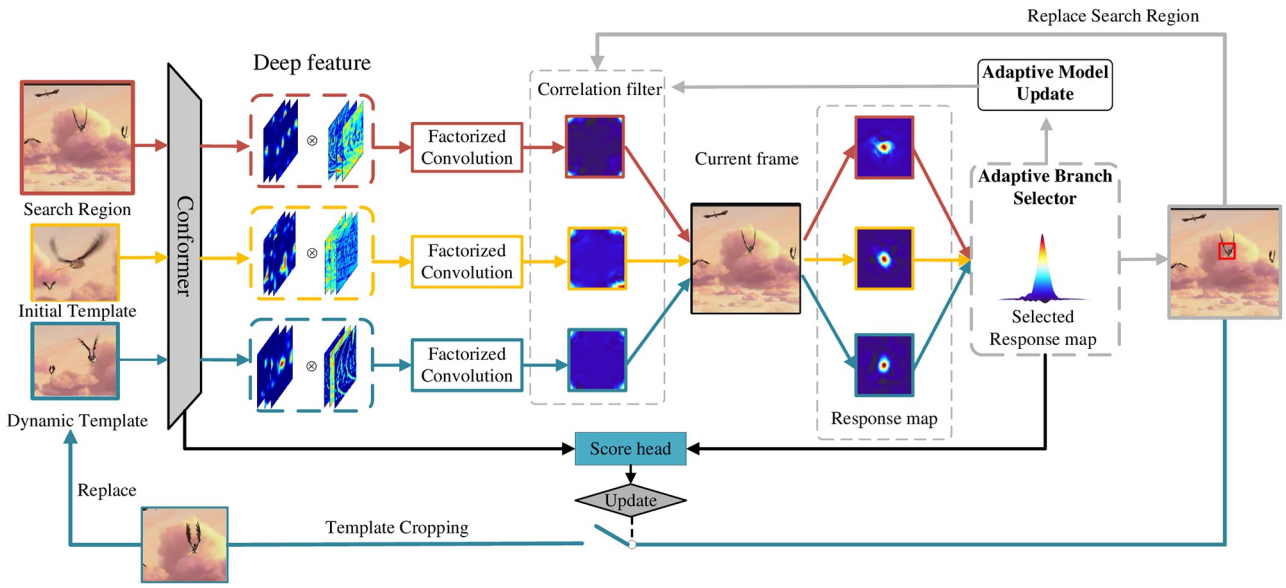

Fig 1. Pipeline of the proposed tracking framework. The initial template refers to initial frame target. The dynamic template is sampled from intermediate frames.

https://doi.org/10.1371/journal.pone.0279240.g001

3.1 Conformer backbone

The quality of the captured features is very important because the target can change greatly over time during tracking. The self-attention mechanism in the Conformer network can capture the global features that are ignored by the convolutional neural network and improve the feature quality. Consistent with correlation filter trackers based on deep learning [8, 31], our proposed ASCF accepts the surrounding images of the tracked object as the input to the backbone network. Overall, there are three inputs to the backbone network: the template image of the initial target object $z \in \mathbb{R}^{3\times H_z\times W_z}$, the search image for the current frame $x \in \mathbb{R}^{3\times H_x\times W_x}$, and the dynamically updated dynamic template image $d \in \mathbb{R}^{3\times H_d\times W_d}$. During the tracking phase, only the search images and dynamic template images are input.

Conformer adopts a concurrent structure and builds CNN and Transformer branches. First, a 7×7 convolution with a stride of two followed by a 3×3 max pooling layer, which also has a stride of two, is used to extract shallow feature maps w_{stem}. hese shallow features are rich in texture and contour information.

First, the image needs to be convoluted and normalized as shown in Eq (1), which is expressed as follows:

$$y = \frac{\mathrm{conv}(x) - \mathrm{mean}(\mathrm{conv}(x))}{\sqrt{\mathrm{Var}(\mathrm{conv}(x)) + \zeta}} \tag{1}$$

where x represents the input image (search template), conv represents the convolution operation, and mean and Var denote the mean and variance of the calculation. ζ increases the value to e^{-5} to avoid the denominator being zero.

After normalization, a 64-channel feature map is obtained using the ReLU activation function and max pooling as follows:

$$w_{\mathrm{stem}} = \mathrm{Max\ Pool}(\mathrm{ReLU}(y)) \tag{2}$$

To obtain local and global features at the same time, a feature coupling unit (FCU) is used to continuously couple local features and global representations in an interactive manner. When the feature map extracted by the CNN enters the Transformer branch, a 1×1 convolution is used to make the feature map consistent with the number of patch embedding channels (the number of channels is 384). Then, a 4×4 downsample with a stride of four is used, and we also add a (1, 384) dimension class_token to complete the spatial alignment. The shape of the patch is nE, and the fusion process is shown in Eq (3), which defines patch i in P_c (denoted as P_c^i) and patch j in P_t (denoted as P_t^i) as follows:

$$P_t^j = P_t^j + \text{Softmax}\left(\frac{(P_t^j W_q)(P_c^i W_k)^T}{\sqrt{E}}\right)(P_c^i W_v) \tag{3}$$

where K and E represent the number of patch embeddings (called P_t) and the channel dimension of the transformer branch, respectively. The feature map is divided into K patches of 14×14, denoted by P_c, W_q, W_k, $W_v \in \mathbb{R}^{3\times H\times W}$,, which are learned linear transformations that map the input and P_t^j to query Q, key K and value V.

When transitioning from the Transformer channel back to the CNN channel, upsampling is performed, and the same attention weights that were employed by the Transformer channel are used, as denoted in Eq (4), which is expressed as follows:

$$\tilde{P}_c^i = \tilde{P}_c^i + \text{Softmax}\left(\frac{(P_t^j W_q)(P_c^i W_k)^T}{\sqrt{E}}\right)^T \tilde{P}_t^j \tag{4}$$

where $\tilde{P}_c^i$ belongs to P_c^i and is processed by the convolutional layer, and $\tilde{P}_c^j$ belongs to P_t^j and is processed by the Transformer block. The feature map $w_{conv\_trans\_10}$ which is rich in global and local features, can be obtained through the convolution operation.

Most DCF-based trackers [8, 10, 16] do not use the backbone network's classification function after extracting the target features because these trackers ignore the pre-trained backbone network's classification ability. Conversely, after the image is entered into the network, we use the pre-trained network's classification function to provide judgments for both subsequent dynamic template updates and model updates. The proposed model performs global pooling on the CNN branch, obtains the class token [51] of the Transformer branch, and then calculates the classification prediction score using a top-k list as follows:

$$S_c = \text{Top}(LN(z_t)) + \text{Top}(LN(z_c)) \tag{5}$$

where z_t represents the Transformer branch's classification, z_c represents the CNN branch's classification, and LN refers to layer normalization. Finally, the predicted score for image classification is obtained.

3.2 Three-way parallel correlation filter tracking

In this subsection, we describe how to implement tracking by using three-way parallel correlation filtering. Conventional correlation filter trackers use a single correlation filter to achieve tracking [25], but if they fail to track at a certain frame, it will cause subsequent continuous tracking failures until the tracking target is lost. To improve the tracker's performance under disturbed conditions such as deformation or occlusion of the tracking target, ASCF uses initial templates (initial frame templates), dynamically updated templates, and search templates to train different correlation filtering models. Subsequent tracking is conducted through the branch selector module, which adaptively selects the model, as shown in Fig 2.

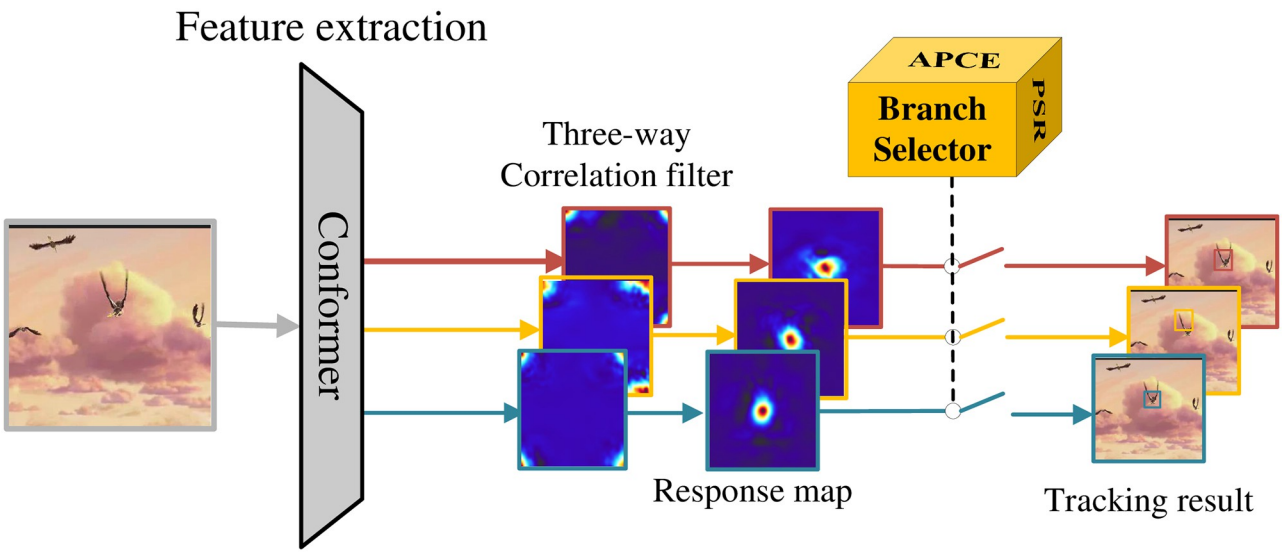

Fig 2. Three-way parallel correlation filter tracking.

https://doi.org/10.1371/journal.pone.0279240.g002

3.2.1 Training correlation filter tracker. All three-way filters are trained in the same way. In this section, using the search template as an example, we explain how to obtain the response map. Suppose that while tracking the t-th frame, the current frame search template is used to train the correlation filter. The search region of the current frame image is considered the detection region $x_t \in \mathbb{R}^{3\times H_x\times W_x}$ of the tracking target. The image is then input into the Conformer network, and the features x_j of the *Conv_stem* and *Conv_trans*_10 layers are extracted as the training samples for correlation filtering. Assuming that there are m training samples in total, the continuous convolution operator [9] is used to transform the features into the continuous space domain. Sample x_j contains a total of D feature channels, and the resolution of sample x_j^d of the d-th feature channel is N_d. $x_i^d[n]$ represents a variable in discrete space, and the sample space is expressed as $\chi = \mathbb{R}^{N_1} \times \ldots \times \mathbb{R}^{N_D}$.

The interpolation operator $J_d : \mathbb{R}^{N_D} \rightarrow L^2(T)$ is used to transform the feature discrete space into the continuous interval $[0, T] \subset \mathbb{R}$, and the $J_d\,(x^d)\,(t)$ interpolation operator is shown in Eq (6), which is expressed as follows:

$$J_d(x^d)\,(t) = \sum_{n=0}^{N_d-1} x^d[n] b_d\left(t - \frac{T}{N_d}n\right) \tag{6}$$

The interpolation function b_d is constructed from the standard cubic spline interpolation kernel, as denoted in Eq (7), which is defined as follows:

$$b(t) = \begin{cases} (a+2)|t|^3 - (a+3)t^2 + 1 & |t| \le 1 \\ a|t|^3 - 5at^2 + 8a|t| - 4a & 1 < |t| \le 2 \\ 0 & |t| > 2 \end{cases} \tag{7}$$

The confidence function S_f uses the convolution filter $f = (f^1, \ldots, f^D) \in L^2(T)^D$ as the parameter, where $f^d \in L^2(T)$ is the feature filter for channel d.

$$S_f\{x\} = \sum_{d=1}^{D} f^d * J_d\{x^d\}, x \in \chi \tag{8}$$

The filter f minimizes the following function through m pairs of training samples $\{(x_i, y_i)\}_1^m \subset \chi \times L^2(T)$. Eq (9) is minimized by m training samples $\{(x_i, y_i)\}_1^m \subset \chi \times L^2(T)$ to obtain filter f as follows:

$$E(f) = \sum_{j=1}^{m} \alpha_j \| S_f\{x_j\} - y_j \|^2 + \sum_{d=1}^{D} \| \omega f^d \|^2 \tag{9}$$

Label y_j is the expected output after applying $S_f\{x_j\}$ to the training sample x_j, and the calculation of the ω penalty coefficient is consistent with [46]. After obtaining filter f, the confidence response S_f can be calculated, and we employ the Gauss-Newton method to optimize the function as shown in Eq (9).

3.2.2 Multi-model adaptive selection. When the training of the three-way correlation filters is completed, tracking can be achieved by adaptively selecting the correlation filters. The current frame image is input, the correlation filter response S_{fd} is calculated and tracked by the dynamic template correlation filter model, the correlation filter response S_{fi} is tracked by the initial template correlation filter model, and the correlation filter response S_{fs} is tracked by the search template correlation filter model, which selects an appropriate correlation filter for tracking.

Among the correlation filters, the PSR can represent the peak sharpness of the correlation filter response (CFR), which is used to evaluate the status of the tracked target and the severity of the interference.

$$PSR\left(S_f\right) = \frac{\max\{S_f\} - \mu(S_f)}{\sigma(S_f)} \tag{10}$$

where $\max\{S_f\}$ is the maximum value of S_f in the correlation filter response, and $\mu(S_f)$ and $\sigma(S_f)$ are the mean and standard deviation of S_f, respectively.

The larger the PSR value, the higher the target tracking confidence and vice versa. However, simply using the PSR to evaluate the target tracking confidence is not sufficient. As shown in Fig 3, at frame 39, PSR==1.2715319. However, at frame 176, the tracking target exhibits dynamic blur and occlusion, among other challenges, and the PSR does not change significantly. But at frame 186 and frame 43, the APCE value is too sensitive.

The APCE [24] evaluation index represents the smoothness of the response graph and is defined in Eq (11) as follows:

$$APCE\left(S_f\right) = \frac{|\max\{S_f\} - \min\{S_f\}|}{\mathrm{mean}\left(\sum_{w,h}(S_{f_{w,h}} - \min\{S_f\})\right)} \tag{11}$$

where $\max\{S_f\}$, $\min\{S_f\}$, and $S_{f_{w,h}}$ are the maximum response value, minimum response value and corresponding position response value respectively. When the peak is sharper and there are fewer interfering peaks, the APCE will be relatively improved, which will be evident in outcomes such as a smooth response graph with only a single peak. Otherwise, when objects are occluded or missing, multimodal responses appear, and the APCE will decrease significantly.

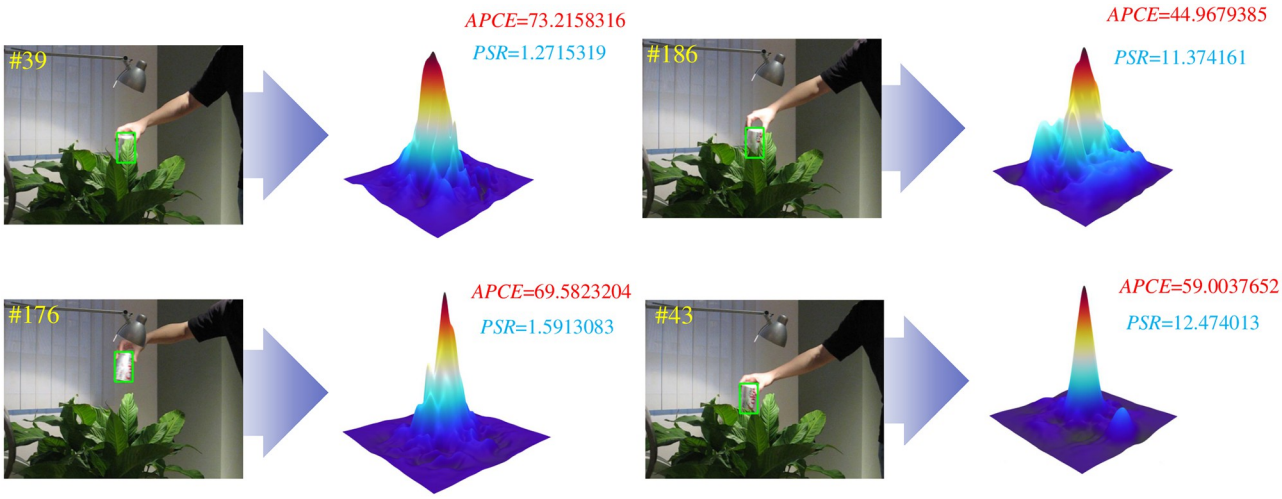

Fig 3. PSR value and APCE value in different tracking environments.

https://doi.org/10.1371/journal.pone.0279240.g003

According to the above analysis and to evaluate the target tracking confidence more reliably, we combine the APCE and PSR as the confidence evaluation indexes of the target tracker to construct a binary function $f(PSR, APCE)$, which is defined as follows:

$$f(PSR, APCE) = (1 - \gamma) \cdot APCE + \gamma \cdot PSR \tag{12}$$

where $\gamma \in [0, 1]$ represents the evaluation weights.

The trajectory smoothness degree indicates the reliability of the tracking results to some extent. To make full use of the historical frame prediction information, we measure the target motion trajectory smoothness degree; by measuring the trajectory smoothness degree between the current frame and the previous five frames, the overall trajectory smoothness degree W is established as follows:

$$W^j = \exp\left(-\left(\sum_{i=j-5}^{j}\left\|\frac{Q^i - Q^{i-1}}{\sqrt{2}\theta_i \times \eta^i}\right\|\right)^2\right) \tag{13}$$

where j is the current frame number, and Q^i is the center position information of the predicted rectangle frame of the ith frame. θ_i is the mean value of the height and width of the frame's predicted rectangular frame. Due to a possible low correlation between the current frame and the previous frame, we set the correlation coefficient θ. $\theta^i = 2$, $\theta^{i-1} = 4$, $\theta^{i-2} = 8$, $\theta^{i-3} = 16$, $\theta^{i-4} = 32$, $\theta^{i-5} = 64$. The score of the current branch at the current frame j is calculated as follows:

$$F_j = (1 - \lambda) \cdot f_j(PSR, APCE) + \lambda \cdot W^j \tag{14}$$

where $\lambda \in [0, 1]$ represents the evaluation weights.

The model then calculates the PSR, APCE, and trajectory smoothness degree of each branch and substitutes these values into Eq (14) to select the tracker with the highest F_j.

3.3 Adaptive update strategy

During the tracking process, if the target deforms too fast, it will cause motion blur, and the tracking frame will drift because the model update after each matching will be inaccurate. If the model's learning rate is low, it can only learn a small part of the information in the current frame, and the model will contain more information from previous frames. In the case of a

large number of occlusions, if the model is updated, many negative samples will be introduced, and the model will not be able to effectively handle the global information in complex scenes or perform robust positioning of the target object. Therefore, it is necessary to be cautious while updating the model and only update it in the appropriate tracking situation.

3.3.1 Update of dynamic template. We judge whether the dynamic template needs to be updated using the classification prediction score and classification prediction category calculated by Eq (5), and we set a reliability threshold δ. If the classification prediction score is greater than the threshold and the classification prediction category is consistent with the category of the last dynamic update template, then it will perform an update; otherwise, it will not be updated. In addition to the spatial information provided, the dynamically updated template can also capture the temporal changes in the target's appearance over time, thus providing additional temporal information.

$$dy_t = \begin{cases} dy_{\mathrm{n}} & S_c > \delta \text{ and } cgy_{\mathrm{n}} = cgy_{\mathrm{o}} \\ dy_{\mathrm{o}} & \text{otherwise} \end{cases} \tag{15}$$

where S_c refers to the classification prediction score calculated in Eq (5), δ refers to the threshold, cgy_{n} refers to the current frame tracking target category, and cgy_{o} refers to the last updated dynamic template classification category.

3.3.2 Model update strategy. Selectively updating the target's tracking model can improve the tracking efficiency and eliminate the influence of negative samples on the model. In this paper, an adaptive update strategy is proposed to refer to the historical average PSR and APCE values of each tracker, and only when the $f(PSR, APCE)$ of the current frame exceeds the historical frame by a certain percentage is the current frame considered a model update sample; the model is then updated. In this way, the time information can be fully utilized, and the model can be updated and judged in combination with the historical frames. The specific learning rate adjustment steps are as follows: First, a fixed base learning rate Lb is set. Then, the ratio of the current frame's APCE value to the average APCE value of the historical frame is calculated, and this value is multiplied by Lb to obtain the latest learning rate. We set a fixed threshold, and if the APCE ratio is lower than the threshold, the learning rate is adjusted to 0. This process is expressed in Eq (16) as follows:

$$l = \begin{cases} L_b, A_t \geq 1 \\ L_b * A_t, 1 > A_t \geq A_T \\ 0, A_t < A_T \end{cases} \tag{16}$$

where l is the updated learning rate, and A_t is the ratio of the APCE value at the current moment to the average APCE value of the historical frame.

Furthermore, the three-way correlation filtering model update strategy is as follows: For the initial template branch, the model is not updated during the entire tracking process $\hat{x}_i = \hat{x}_i$. The ASCF only updates the model for the remaining two-way correlation filtering.

a. For the dynamic update template branch, if the dynamic update template is updated and $f_t(PSR, APCE) \geq f_T(PSR, APCE)$, we update the dynamic update model $\hat{x}_d^p = (1 - l)\hat{x}_d^{p-1} + l\hat{x}_d$.

b. For the search template branch, an update judgment is performed once in five frames, and if $f_t(PSR, APCE) \geq f_T(PSR, APCE)$ is satisfied, $\hat{x}_s^p = (1 - l)\hat{x}_s^{p-1} + l\hat{x}_s$.

The model is updated according to the learning rate obtained by Eq (16), and the newly-learned model is used to track the next image frame. $\hat{x}_i$ represents the initial template tracking model, $\hat{x}_d$ represents the dynamically updated template tracking model, $\hat{x}_s$ represents the search template tracking model, $f_t(PSR, APCE)$ is the frame calculation of t-th, $f_T(PSR, APCE)$ is the historical calculation mean, and $\hat{x}^{p-1}$ is the last updated target model.

Algorithm 1: ASCF

Input: Sequence frames (t-th frame, total of T frames). Initial bounding box of target.
Output: Target bounding box.
for t = 1 to T **do**
  Extract search region feature map w_{stem}, $w_{conv\_trans\_10}$ by Conformer
  if t = 1 **then**
    Using feature maps to training initial template correlation filters by Eq (6) and Eq (9)
  end
  if t > 4 **then**
    Using feature maps to training dynamic tracking template correlation filters by Eq (6) and Eq (9);
    Determine whether the dynamic template needs to be updated by Eq (15);
  end
  Using feature maps to training search template correlation filters by Eq (6) and Eq (9);
  Get the dynamic tracking template, initial template and search template confidence response map S_{fd}, S_{fi}, S_{fs} by Eq (8);
  Select the appropriate tracking template for tracking by Eq (14);
  Update training learning rate by Eq (16);
  Update the dynamic, initial, search tracking model.
end
return Target bounding box.

3.4 Tracking pipeline

We provide a brief overview of this paper's algorithm in Algorithm 1. This algorithm consists of two main modules: adaptive selection of three correlation filters for tracking and an adaptive update strategy. During the tracking process, the target features are extracted through the Conformer network. Then, the correlation filter is trained, and the highest correlation filter from among the initial, dynamic and search branches is selected for tracking evaluation, and the generated tracking frame is used as prior knowledge to crop the current frame as a reference to generate a search tracking template. During the model's update process, the dynamic template is checked and updated at the same time. Selectively updating the target's tracking model not only improves the tracking efficiency but also effectively excludes the impact of negative samples on the mode.

4 Experiments

This section introduces the implementation details of our proposed algorithm, ASCF, and then, we conduct comparative experiments with the current state-of-the-art trackers on target tracking evaluation datasets to prove the superiority of this algorithm. The datasets are as follows: OTB2013, OTB2015, VOT2020, GOT-10K and LBT50. Finally, through ablation experiments, the effectiveness of each tracker module is analyzed.

Our tracker was implemented based on Python3.7 and Pytorch1.7.1 and was tested on a desktop computer using a single NVIDIA GeForce GTX 3070 GPU with a 3.7GHz AMD Ryzen 5 5600X CPU. The Conv_stem and Conv_trans_10 layers of the Conformer network

are extracted as the features of the tracking target, and we set the search region to 4.5 times the tracking box size during detection. The dynamic template is acquired for the first time after four tracking frames, and the dynamic update template is updated after the 4-th frame. The classification confidence threshold δ of the dynamic template is 8.5, as shown in Eq (15). Based on [10], we set the learning rate to 0.01, which is L_{base} in Eq (16). In Eq (16), the learning rate adjustment threshold A_T is 0.65, and in Eq (12), the weight adjustment coefficient γ is set to 0.9 because the APCE value is usually much larger than that of the PSR. In addition, the λ in Eq (14) is set to 0.6. To provide a fair comparison, all of the following experiments were performed under identical training settings.

4.1 Comparison with the SOTA trackers

4.1.1 OTB-2013. OTB-2013 [29] consists of 51 video sequences and is currently one of the most widely tested datasets in the field of visual tracking. This dataset uses the one-pass evaluation (OPE) [29] protocol as the tracking evaluation indicator.

We compare the ASCF trackers with many state-of-the-art trackers, including MCCT [14], Ocean [40], ATOM [16], UDT [52], DaSiamRPN [17], SiamBAN [53], SiamRPN++ [54], ECO [10] and C-COT [9] on OTB2013. We report two metrics, the area-under-the-curve(AUC) score and the distance precision(DP) score.

The center location error (CLF) is the Euclidean distance from the ground-truth center position $(x_g,y_g)$to the predicted center position (x_p, y_p), as shown in Eq (17). The DP is the percentage of the number of frames whose CLF is greater than a certain distance error threshold, known as the location error threshold (LET), to the total number of frames in the video sequence.

$$C_{LF} = \sqrt{(x_p - x_g)^2 + (y_p - y_g)^2} \tag{17}$$

The overlap rate accuracy (OP) refers to the percentage of the number of frames where the overlap rate ϕ of the tracking target frame R^P and the ground-truth bounding box R^G is greater than the overlap rate threshold (OT) to the total number of frames.

$$\phi = \frac{|R^P \cap R^G|}{|R^P \cup R^G|} \tag{18}$$

Fig 4 shows the DP and the AUC on the OTB-2013 dataset compared to the current state-of-the-art tracking algorithms, with the performance scores of each algorithm labeled in the legend. With an AUC of 73% and a DP of 94.4%, our algorithm is the best among the compared trackers in terms of both the AUC and DP evaluation metrics. Our tracker's AUC is 0.4% higher than that of ECO when using the same correlation filter tracker, and its AUC is 2.5% higher overall. Compared with the Siamese tracker, our tracking algorithm improves more dramatically. Specifically, when compared to the Siamese tracker SiamBAN, which is the best-performing Siamese tracker on OTB 2013, the DP improved by 2.6%, and the AUC improved by 3.7%.

4.1.2 OTB-2015. Compared with OTB2013, the OTB2015 [28] dataset is more difficult to track, and the tracking scenarios are more complex, which provides a relatively uniform testing and evaluation environment for tracking algorithms. We compare ASCF with the recent state-of-the-art trackers, including MCCT [14], C-COT [9], UDT [52], DaSiam [17], Ocean [40], Siam_RPN++ [54], ECO [10], SiamBAN [53], ATOM [16] and DiMP [18] on OTB2015. The comparison results are shown in Table 1. The ASCF tracker ranks first in terms of the AUC, DP and OP. Compared with the baseline tracker, ECO, ASCF improves the AUC, DP

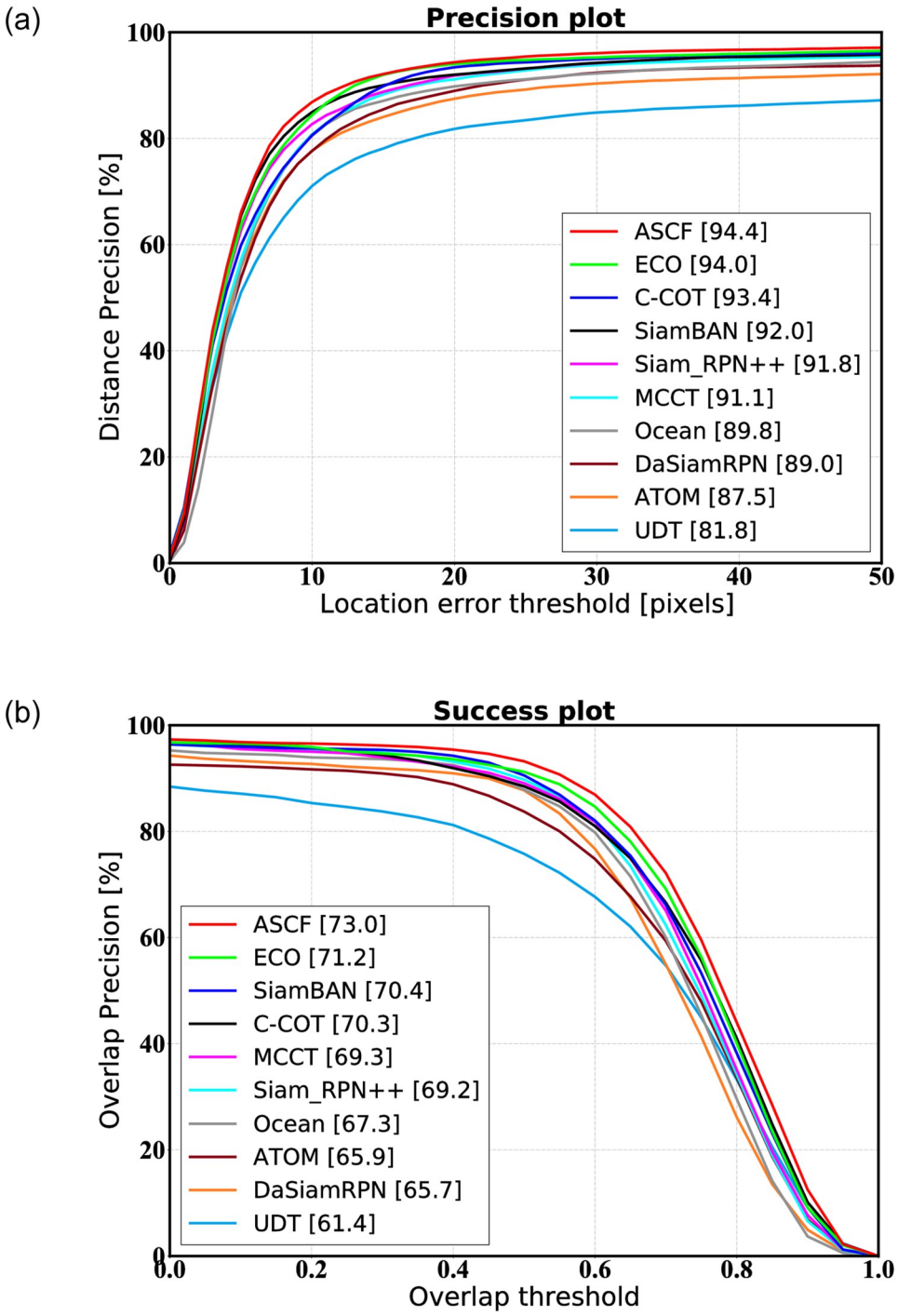

Fig 4. One-pass evaluation, the distance accuracy DP of the tracker and the area under the overlap rate curve accuracy AUC are displayed on the OTB-2013 data set, and the center position error CLF score threshold of the distance accuracy DP is set to 20. (a) OTB2013 (DP). (b) OTB2013 (AUC).

https://doi.org/10.1371/journal.pone.0279240.g004

Table 1. Comparisons on OTB2015 dataset.

	UDT [52]	MCCT [14]	C-COT [9]	DaSiam [17]	Ocean [40]	ATOM [16]	Dimp [18]	ECO [10]	Siam_RPN++ [54]	SiamBAN [53]	ASCF
AUC	61.9	67.8	68.1	65.7	67.1	66.7	68.5	69.2	69.6	69.6	71.3
Precision	82.4	90.7	91.5	88.0	89.9	87.9	89.9	91.4	91.4	91.0	91.7
OP50	75.7	85.5	83.5	86.5	86.6	83.6	86.4	86.7	89.2	89.3	89.7

https://doi.org/10.1371/journal.pone.0279240.t001

and OP by 3%, 0.3%, and 3.4%, respectively. Compared with the best performing Siamese tracker SiamRPN++, ASCF outperforms it in terms of the AUC, DP and OP by 2.4%, 0.3%, and 0.6%, respectively. The OTB2015 dataset divides the video sequence attributes in the test set into 11 categories according to common challenging factors in object tracking, including illumination variation (IV), deformation (DEF), scale variation (SV), occlusion (OCC), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of-plane rotation (OPR), out-of-view (OV), background clutters (BC) and low resolution (LR), each video sequence in the test set contains at least one of the above properties. To further evaluate the effectiveness of our method in different tracking scenarios, we tested it in the above 11 tracking scenes. For the convenience of observation, six representative trackers were selected for comparison with the trackers in this paper. The evaluation results are shown in Fig 5. It can be clearly seen that the tracker proposed in this paper achieves a better AUC in most tracking scenes. Especially in terms of the MB, IV, OV, OPR, FM, DEF, OCC, and SV tracking scenes, it performs significantly better than the other algorithms. These experimental results show that our tracker can adapt to different common challenges in object tracking, and the adaptive update strategy is an important reason why the algorithm we propose can effectively address various challenges.

4.1.3 GOT10K. GOT10K [27] is a large-scale dataset containing more than 10,000 videos that most deep learning trackers use for training. We evaluate the proposed algorithm in this paper on its test set, which contains a total of 180 videos with a total of 150 different categories, and we followed the evaluation guidelines and submitted the tracking results to GOT10k's official online evaluation server. For the first time, this dataset combines categories with evaluation metrics, and it uses the mean average overlap (mAO) and mean success rate (mSR) as metrics. Compare with Ocean [40], DiMP [18], ATOM [16], TRASF [55], DPMT [56], SiamFC++ [39], SiamRPN++ [54], MemTracker [57], C-COT [9], ECO [10], SiamFC [34] and MDNet [58], which are the state-of-the-art trackers, the proposed algorithm performed well on the GOT10k dataset. Its average overlap (AO) and success rate (SR) are evaluated as shown in Table 2, and it can be clearly observed that the ASCF algorithm ranks first in terms of the mAO and mSR75 with values of 61.4% and 52.6%, respectively, which is better than that of DiMP, ATOM, and other deep correlation filter trackers.

The calculation formula of mAO is shown in the following Equation.

$$mAO = \frac{1}{C}\sum_{c=1}^{C}\left(\frac{1}{|S_c|}\sum_{i\in S_c} AO_i\right) \tag{19}$$

where C represents the number of types, S_c represents the number of video sequences under a certain type, and AO represents the average overlap. Similarly, the calculation formula of mSR is shown in Eq (20).

$$mSR = \frac{1}{C}\sum_{c=1}^{c}\left(\frac{1}{|S_c|}\sum_{i\in S_c} SR_i\right) \tag{20}$$

where SR represents the success rate.

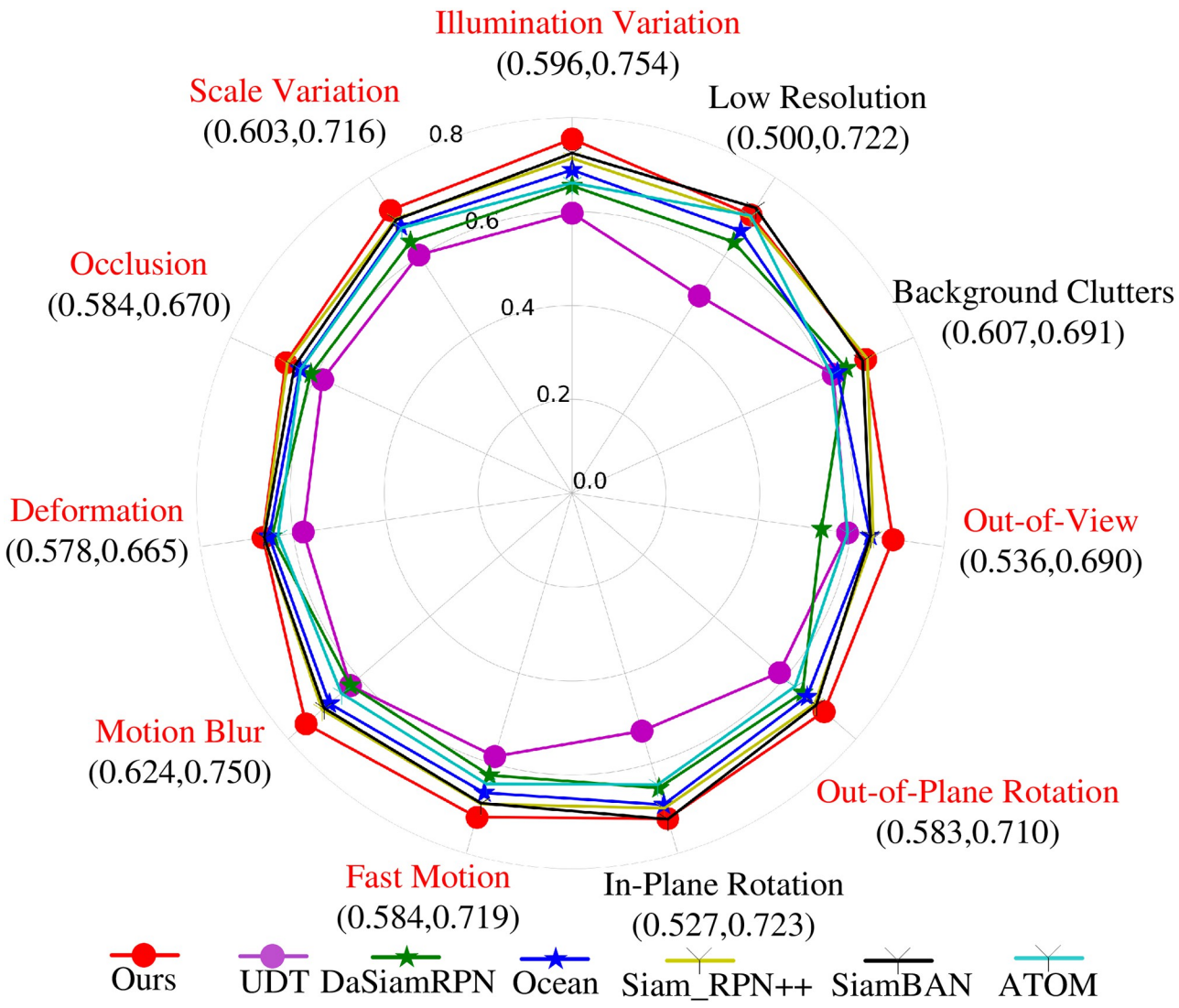

Fig 5. The AUC evaluation index values of each tracker under 11 different challenge factors in OTB-2015.

https://doi.org/10.1371/journal.pone.0279240.g005

4.1.4 VOT2020. The visual object tracking challenge (VOT) is a recently released challenging target tracking evaluation dataset and is the most authoritative and influential evaluation platform dataset in the field of international object tracking. The difference from the previous VOT dataset is that the label format of VOT-ST2020 and VOT-RT2020 has changed from the original rotated rectangular box to a mask, where ST refers to short-term tracking challenges and RT refers to short-term real-time challenges. Inspired by AlphaRef, we introduced the AlphaRef mask branch to achieve mask segmentation of the tracking targets. We selected VOT2020-ST and VOT2020-RT to evaluate the tracker, and the evaluation indicators are the expected average overlap (EAO), accuracy, and robustness. Our tracker is compared with the following state-of-the-art trackers: AFAT [61], DPMT [56], DiMP [18], ATOM [16], CSR-DCF [59], SiamFC [34], TRASF [55], SiamMask [60].

As shown in Fig 6, we visualized the accuracy and robustness of each tracker on the VOT-ST2020 short-term tracking challenge. Our tracker ranks first on the accuracy evaluation

Table 2. Experimental results on GOT10K dataset.

Tracker	Performance			Properties			Venue
	mAO	mSR50	mSR75	CF	Siamese	DL	
ASCF	0.614	0.696	0.526	✓		✓	
Ocean [40]	0.611	0.721	0.473		✓	✓	ECCV'2020
DiMP [18]	0.611	0.717	0.492	✓		✓	ICCV'2019
TRASF [55]	0.604	0.708	0.469			✓	ArXiv'2020
DPMT [56]	0.600	0.716	0.460	✓		✓	PRCV'2020
SiamFC++ [39]	0.595	0.695	0.479		✓	✓	AAAI'2020
ATOM [16]	0.556	0.634	0.402	✓		✓	CVPR'2019
SiamRPN++ [54]	0.517	0.616	0.325		✓	✓	CVPR'2019
MemTracker [57]	0.460	0.524	0.193			✓	ECCV'2018
C-COT [9]	0.406	0.415	0.161	✓		✓	ECCV'2016
ECO [10]	0.395	0.407	0.170	✓		✓	CVPR'2017
SiamFC [34]	0.392	0.426	0.135		✓	✓	ECCV'2016
ECOhc [10]	0.363	0.359	0.154	✓		✓	CVPR'2017
MDNet [58]	0.352	0.367	0.137			✓	CVPR'2016

https://doi.org/10.1371/journal.pone.0279240.t002

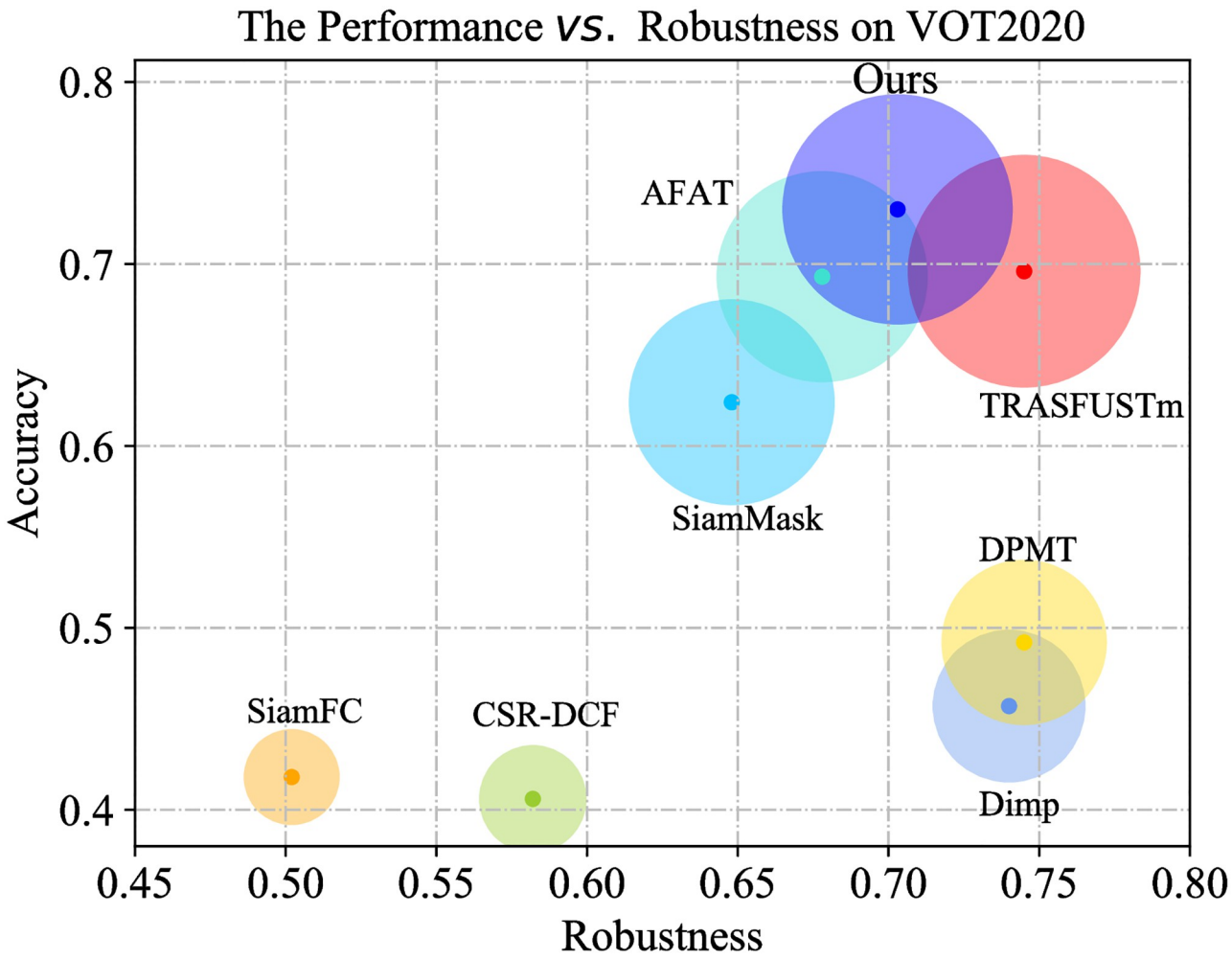

Fig 6. Comparison with state-of-the-arts on VOT-ST2020. We equip ASCF with a refinement module proposed by AlphaRef [62] to generate segmentation masks.

https://doi.org/10.1371/journal.pone.0279240.g006

Table 3. Experimental results on VOT2020-ST dataset.

	SiamFC [34]	CSR-DCF [59]	ATOM [16]	Dimp [18]	UPDT [15]	DPMT [56]	SiamMask [60]	AFAT [61]	TRASF [55]	ASCF
EAO	0.179	0.193	0.271	0.274	0.278	0.303	0.321	0.378	0.424	0.396
Accuracy	0.418	0.406	0.462	0.457	0.465	0.492	0.624	0.693	0.696	0.730
Robustness	0.502	0.582	0.734	0.740	0.755	0.745	0.648	0.678	0.745	0.703

https://doi.org/10.1371/journal.pone.0279240.t003

Table 4. Experimental results on VOT2020-RT dataset.

	SiamFC [34]	CSR-DCF [59]	ATOM [16]	UPDT [15]	Dimp [18]	TRASF [55]	DPMT [56]	SiamMask [60]	AFAT [61]	ASCF
EAO	0.172	0.193	0.237	0.237	0.241	0.282	0.293	0.320	0.372	0.332
Accuracy	0.422	0.405	0.440	0.443	0.434	0.576	0.487	0.624	0.687	0.639
Robustness	0.479	0.580	0.687	0.688	0.700	0.616	0.730	0.645	0.676	0.660

https://doi.org/10.1371/journal.pone.0279240.t004

metric with a value of 0.73; the tracker proposed in this paper also achieved a robustness score of 0.703, which is higher than that of most of the compared trackers. This result verifies that our tracker can maintain good robustness while tracking with high accuracy.

As shown in Table 3, the EAO, accuracy, and robustness values of each tracker on the VOT-ST2020 dataset are shown. The results show that our tracker ranks second only to TRASF on EAO with a value of 0.396, which is 42.4% higher than the correlation filter tracker UPDT. In addition, the accuracy is 4.89% higher than that of the second-highest TRASF tracker among the compared trackers. At the same time, we made a comparison in VOT-RT2020, as shown in Table 4. There are trackers that perform well on short-term tracking challenges but not well in real-time challenges, such as the TRASF tracker. Our tracker achieves high scores on both short-term tracking challenges and real-time tracking challenges.

4.1.5 LBT50. LBT50 [30] proposed a long-term visual object tracking performance evaluation methodology and a benchmark and provides eight different long-term tracking challenge sequences. Fig 7 shows the F-scores over all 50 videos in the dataset. Our method achieves good results on all eight attributes but does not perform well on the out-of-view criterion because ASCF is not configured with a re-detection module. Our ASCF achieves an F-score of 61%, which is competitive with the other trackers. Our approach especially excels in the case of aspect ratio changes and scale variation, demonstrating the impact of our components.

4.2 Component-wise analysis

In this section, we choose to verify the effectiveness of the enhancements to the tracker performance of the proposed components in this paper on GOT-10K dataset.

4.2.1 Deep feature. At present, most correlation filter trackers use the CNN to extract deep features and obtain tracking models through the learning and training of correlation filters. More specifically, we compare the baseline trackers with different deep features to verify the effectiveness of our used deep features. As shown in Table 5, Resnet50 [63], MobileNetv3 [64] and the Conformer network are used to extract the AUC and DP of the tracker (VGG16 was selected as the baseline). The experimental results show that the Conformer network can extract better-quality features in the image, so the AUC and DP of the tracker are improved.

4.2.2 Model adaptive selection. As described in the section titled "Three-way parallel correlation filter tracking", we trained a three-way correlation filter tracker and introduced spatial information. As shown in Table 6, the AUC of only using the Conformer network to extract features is 47.3%, and after adding the adaptive model selection strategy, the AUC improves

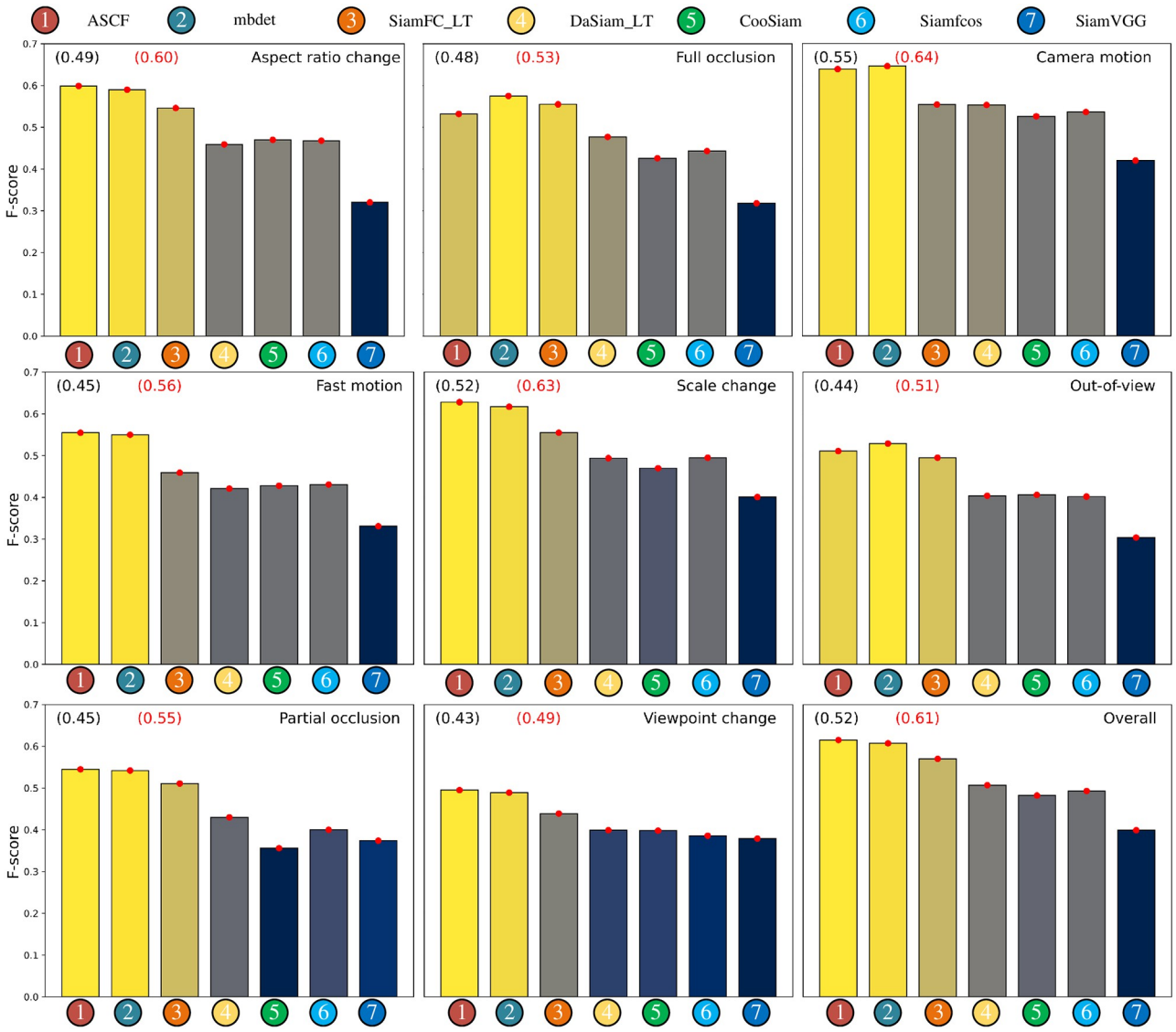

Fig 7. Attribute analysis on the LBT50 dataset.

https://doi.org/10.1371/journal.pone.0279240.g007

significantly to 56.7%. This result verifies that the model adaptive selection strategy proposed in this paper can effectively improve the tracker's accuracy.

4.2.3 Adaptive update strategy. As described in the section titled "Adaptive update strategy", to selectively update the model in different situations, we propose a model adaptive

Table 5. Analysis of deep feature on GOT-10K.

Comparison of features	mAO(%)	mSR50(%)	mSR75(%)
Baseline	39.5	40.7	17.0
Baseline+ResNet50	44.7	48.1	25.9
Baseline+MobileNetv3	38.2	37.3	16.4
Baseline+Conformer	47.3	54.6	28.5

https://doi.org/10.1371/journal.pone.0279240.t005

Table 6. Component-wise analysis. Performance is evaluated on GOT-10K.

#Num	Metrics	mAO(%)	mSR50(%)	mSR75(%)
1.	Baseline	39.5	40.7	17.0
2.	+ Conformer	47.3	54.6	28.5
3.	+ Three-way track	56.7	63.2	46.8
4.	+ Adaptive update strategy	61.4	69.6	52.6

https://doi.org/10.1371/journal.pone.0279240.t006

Table 7. The amount of frames processed per second (fps) with different trackers.

Tracker	Baseline(ECO)	ASCF	DiMP	ATOM	Ocean
Tracking speed(Avg.FPS)	34	26	40*	46*	25*

https://doi.org/10.1371/journal.pone.0279240.t007

update strategy. From the experimental results shown in Table 6, it can be seen that by using the combined Baseline + Conformer + Three-way track + Adaptive update strategy, the tracking accuracy and success rate of the tracker proposed in this paper are significantly better than the baseline or other combinations that use the baseline. Therefore, using an adaptive model update strategy can further improve the tracker's performance.

4.3 Limitations of the proposed method

During the experiments, we found that the ASCF tracker still has some flaws. Although ASCF establishes tracking models in spatio-temporal locations and makes full use of space and time information, tracking drift still occurs when the model experiences tracking challenges such as long-term targets disappearing from the field of view for a long time. For example, ASCF does not perform well on the long-term tracking dataset LBT50 because ASCF lacks the re-detection mechanism implemented in the long-term trackers.

We tested the tracking speed of ASCF against the baseline tracker on the OTB2015 dataset, as shown in Table 7. The baseline and our proposed ASCF were both tested on the RTX 3070 GPU, and the tracking speeds with * are taken from the results published in the original paper. It can be seen that our tracker still has some defects in its tracking speed, so our next goal is to optimize the algorithm to improve the tracking speed.

4.4 Visualization

We selected three representative trackers during the past two years as well as the tracking algorithm proposed in this paper for qualitative evaluation of some selected tracking sequences with different challenges. The results are shown in Fig 8. According to the visualization, our tracker is more robust compared to other trackers and produces more accurate tracking results when encountering occlusions, fast motion, and scale variation.

As shown in Fig 9, we made some visualizations for model selection and observed the tracking results generated by different correlation filtering models in the three channels. In the soccer tracking video sequence, due to the large deformation and occlusion of the tracking target and because the similarity to the initial image is low, it is necessary to update the correlation filter model with a faster frequency, and the current tracking frame uses a dynamic template model. In the basketball tracking video sequence, the tracking object in the current image frame has a high similarity with the initial tracking target, and the initial template correlation filtering model with higher confidence is selected for tracking.

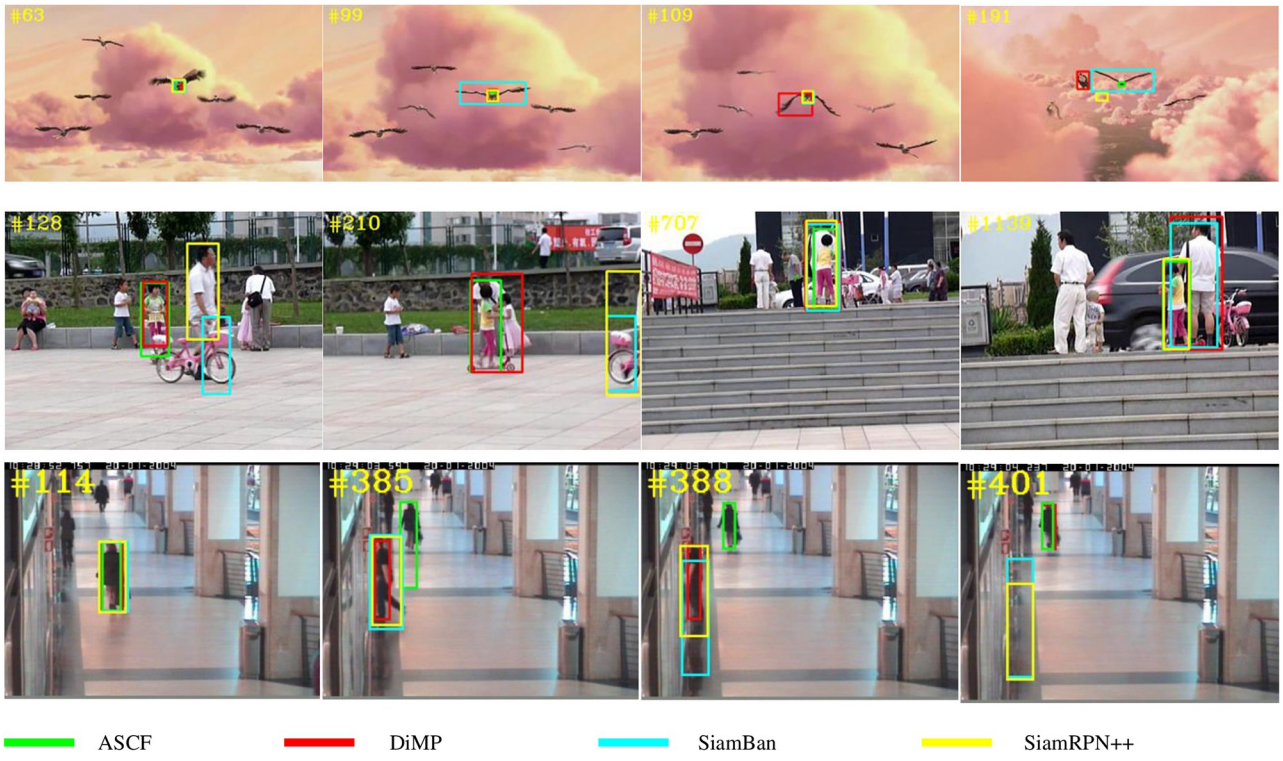

Fig 8. Tracking results on Bird1, Girl2 and Walking2 videos in OTB-2015.

https://doi.org/10.1371/journal.pone.0279240.g008

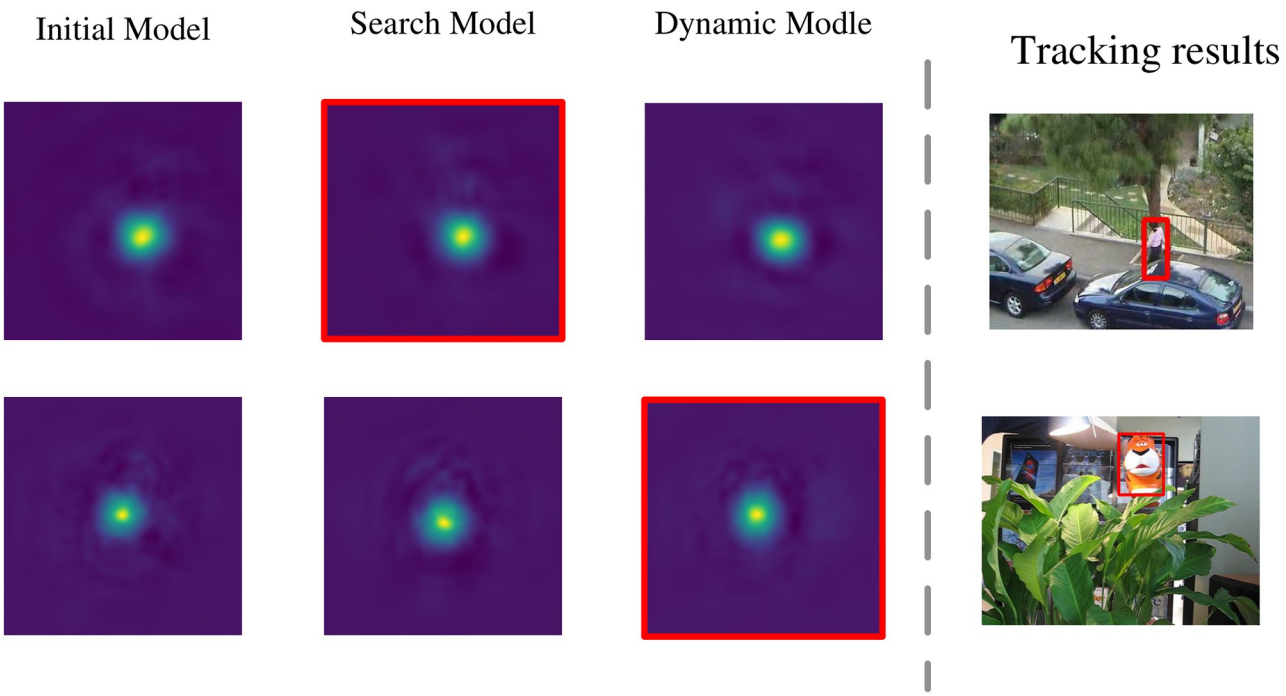

Fig 9. Visualization of response maps of different correlation filtering models. Selected models surrounded by red boxes. The results from top to down are Woman and Tgier1 from OTB-2015.

https://doi.org/10.1371/journal.pone.0279240.g009

5 Conclusions

In this paper, we propose a new ASCF tracking algorithm to spatio-temporally model the tracked target at different points in time and space during the tracking process. It consists of three correlation filters constructed with different spatio-temporal features, and the features are extracted by the Conformer network. The best tracking result is then selected by the adaptive model selection module proposed in this paper. Furthermore, we designed an adaptive model update strategy to avoid introducing disturbing information into the model. Finally, experiments were conducted on the public databases OTB2013, OTB2015, GOT-10K, VOT2020 and LBT50, and they demonstrate the superiority of ASCF and all of its components. By making full use of the spatio-temporal information through three different spatio-temporal tracking models, ASCF can track targets robustly and accurately in complex tracking environments. In future work, we plan to deploy our tracking framework on an end-to-end deep learning framework while improving the tracking efficiency to further improve the algorithm's target tracking performance.

Author Contributions

Conceptualization: Yuhan Liu, He Yan.

Data curation: Yuhan Liu, Mengxue Li, Lingkun Liu.

Formal analysis: Yuhan Liu, He Yan.

Funding acquisition: He Yan.

Investigation: Yuhan Liu, He Yan.

Methodology: Yuhan Liu, He Yan, Wei Zhang.

Project administration: Yuhan Liu, Mengxue Li, Lingkun Liu.

Resources: Yuhan Liu, He Yan.

Software: Yuhan Liu.

Supervision: He Yan.

Validation: Yuhan Liu, Wei Zhang.

Visualization: Yuhan Liu, Wei Zhang.

Writing – original draft: Yuhan Liu, He Yan.

Writing – review & editing: Yuhan Liu, He Yan.

References

1. Liu S, Wang S, Liu X, Dai J, Muhammad K, Gandomi AH, et al. Human inertial thinking strategy: A novel fuzzy reasoning mechanism for IoT-assisted visual monitoring. IEEE Internet of Things Journal. 2022;.
2. Liu S, Wang S, Liu X, Gandomi AH, Daneshmand M, Muhammad K, et al. Human memory update strategy: a multi-layer template update mechanismfor remote visual monitoring. IEEE Transactions on Multimedia. 2021; 23:2188–2198. https://doi.org/10.1109/TMM.2021.3065580
3. Yuan D, Chang X, Li Z, He Z. Learning adaptive spatial-temporal context-aware correlation filters for UAV tracking. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM). 2022; 18(3):1–18. https://doi.org/10.1145/3486678
4. Wang W, Zhang K, Lv M, Wang J. Hierarchical spatiotemporal context-aware correlation filters forvisual tracking. IEEE Transactions on Cybernetics. 2020;.
5. Li P, Wang D, Wang L, Lu H. Deep visual tracking: Review and experimental comparison. Pattern Recognition. 2018; 76:323–338. https://doi.org/10.1016/j.patcog.2017.11.007

6. Abbass MY, Kwon KC, Kim N, Abdelwahab SA, El-Samie FEA, Khalaf AA. A survey on online learning for visual tracking. The Visual Computer. 2021; 37(5):993–1014. https://doi.org/10.1007/s00371-020-01848-y
7. Javed S, Danelljan M, Khan FS, Khan MH, Felsberg M, Matas J. Visual object tracking with discriminative filters and Siamesenetworks: A survey and outlook. arXiv preprint arXiv:211202838. 2021;.
8. Ma C, Huang JB, Yang X, Yang MH. Robust Visual Tracking via Hierarchical Convolutional Features. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2019; 41(11):2709–2723. https://doi.org/10.1109/TPAMI.2018.2865311 PMID: 30106709
9. Danelljan M, Robinson A, Shahbaz Khan F, Felsberg M. Beyond correlation filters: Learning continuous convolution operatorsfor visual tracking. In: European conference on computer vision. Springer; 2016. p.472–488.
10. Danelljan M, Bhat G, Shahbaz Khan F, Felsberg M. Eco: Efficient convolution operators for tracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 6638–6646.
11. Valmadre J, Bertinetto L, Henriques J, Vedaldi A, Torr PH. End-to-end representation learning for correlation filter basedtracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 2805–2813.
12. Yuan D, Chang X, Huang PY, Liu Q, He Z. Self-supervised deep correlation tracking. IEEE Transactions on Image Processing. 2020; 30:976–985. PMID: 33259298
13. Zhang J, Sun J, Wang J, Yue XG. Visual object tracking based on residual network and cascadedcorrelation filters. Journal of ambient intelligence and humanized computing. 2021; 12(8):8427–8440. https://doi.org/10.1007/s12652-020-02572-0
14. Wang N, Zhou W, Tian Q, Hong R, Wang M, Li H. Multi-cue correlation filters for robust visual tracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. p. 4844–4853.
15. Bhat G, Johnander J, Danelljan M, Khan FS, Felsberg M. Unveiling the power of deep tracking. In: Proceedings of the European Conference on Computer Vision (ECCV);2018. p. 483–498.
16. Danelljan M, Bhat G, Khan FS, Felsberg M. Atom: Accurate tracking by overlap maximization. In: Proceedings of the IEEE/CVF Conference on Computer Vision andPattern Recognition; 2019. p. 4660–4669.
17. Zhu Z, Wang Q, Li B, Wu W, Yan J, Hu W. Distractor-aware siamese networks for visual object tracking. In: Proceedings of the European conference on computer vision (ECCV);2018. p. 101–117.
18. Bhat G, Danelljan M, Gool LV, Timofte R. Learning discriminative model prediction for tracking. In: Proceedings of the IEEE/CVF international conference on computer vision; 2019. p. 6182–6191.
19. Yan B, Peng H, Fu J, Wang D, Lu H. Learning spatio-temporal transformer for visual tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021. p. 10448–10457.
20. Peng Z, Huang W, Gu S, Xie L, Wang Y, Jiao J, et al. Conformer: Local features coupling global representations for visualrecognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021. p. 367–376.
21. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S. End-to-end object detection with transformers. In: European conference on computer vision. Springer; 2020. p.213–229.
22. Chen X, Yan B, Zhu J, Wang D, Yang X, Lu H. Transformer tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision andPattern Recognition; 2021. p. 8126–8135.
23. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, et al. Swin transformer: Hierarchical vision transformer using shiftedwindows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021. p. 10012–10022.
24. Wang M, Liu Y, Huang Z. Large margin object tracking with circulant feature maps. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 4021–4029.
25. Bolme DS, Beveridge JR, Draper BA, Lui YM. Visual object tracking using adaptive correlation filters. In: 2010 IEEE computer society conference on computer vision andpattern recognition. IEEE; 2010. p. 2544–2550.
26. Kristan M, Leonardis A, Matas J, Felsberg M, Pflugfelder R,Kämäräinen JK, et al. The eighth visual object tracking VOT2020 challenge results. In: European Conference on Computer Vision. Springer; 2020. p.547–601.
27. Huang L, Zhao X, Huang K. Got-10k: A large high-diversity benchmark for generic object trackingin the wild. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2019; 43(5):1562–1577. https://doi.org/10.1109/TPAMI.2019.2957464
28. Wu Y, Lim J, Yang MH. Object Tracking Benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2015; 37(9):1834–1848. https://doi.org/10.1109/TPAMI.2014.2388226 PMID: 26353130

29. Wu Y, Lim J, Yang MH. Online object tracking: A benchmark. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2013. p. 2411–2418.

30. Lukeźič A, Zajc L, Vojíř T, Matas J, Kristan M. Performance Evaluation Methodology for Long-Term Single-ObjectTracking. IEEE Transactions on Cybernetics. 2021; 51(12):6305–6318. https://doi.org/10.1109/TCYB.2020.2980618 PMID: 32248144

31. Qi Y, Zhang S, Qin L, Huang Q, Yao H, Lim J, et al. Hedging deep features for visual tracking. IEEE transactions on pattern analysis and machine intelligence. 2018; 41(5):1116–1130. https://doi.org/10.1109/TPAMI.2018.2828817 PMID: 29993908

32. Yuan D, Kang W, He Z. Robust visual tracking with correlation filters and metric learning. Knowledge-Based Systems. 2020; 195:105697. https://doi.org/10.1016/j.knosys.2020.105697

33. Zhang J, Yuan T, He Y, Wang J. A background-aware correlation filter with adaptive saliency-awareregularization for visual tracking. Neural Computing and Applications. 2022; 34(8):6359–6376. https://doi.org/10.1007/s00521-021-06771-4

34. Bertinetto L, Valmadre J, Henriques JF, Vedaldi A, Torr PH. Fully-convolutional siamese networks for object tracking. In: European conference on computer vision. Springer; 2016. p.850–865.

35. Liu S, Wang S, Liu X, Lin CT, Lv Z. Fuzzy detection aided real-time and robust visual tracking under-complex environments. IEEE Transactions on Fuzzy Systems. 2020; 29(1):90–102. https://doi.org/10.1109/TFUZZ.2020.3006520

36. Yang K, He Z, Pei W, Zhou Z, Li X, Yuan D, et al. SiamCorners: Siamese corner networks for visual tracking. IEEE Transactions on Multimedia. 2021; 24:1956–1967.

37. Zhang J, Liu Y, Liu H, Wang J, Zhang Y. Distractor-aware visual tracking using hierarchical correlation-filters adaptive selection. Applied Intelligence. 2022; 52(6):6129–6147. https://doi.org/10.1007/s10489-021-02694-8

38. Liu Y, Yan H, Liu Q, Zhang W, Huang J. ECO++: Adaptive deep feature fusion target tracking method in complexscene. Digital Communications and Networks. 2022;. https://doi.org/10.1016/j.dcan.2022.10.020

39. Xu Y, Wang Z, Li Z, Yuan Y, Yu G. Siamfc++: Towards robust and accurate visual tracking with target-testimation guidelines. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34; 2020. p. 12549–12556.

40. Zhang Z, Peng H, Fu J, Li B, Hu W. Ocean: Object-aware anchor-free tracking. In: European Conference on Computer Vision. Springer; 2020. p.771–787.

41. Zhang J, Sun J, Wang J, Li Z, Chen X. An object tracking framework with recapture based on correlationfilters and Siamese networks. Computers & Electrical Engineering. 2022; 98:107730. https://doi.org/10.1016/j.compeleceng.2022.107730

42. Zhang J, Feng W, Yuan T, Wang J, Sangaiah AK. SCSTCF: spatial-channel selection and temporal regularizedcorrelation filters for visual tracking. Applied Soft Computing. 2022; 118:108485. https://doi.org/10.1016/j.asoc.2022.108485

43. Zhao D, Xiao L, Fu H, Wu T, Xu X, Dai B. Augmenting cascaded correlation filters with spatial–temporal-saliency for visual tracking. Information Sciences. 2019; 470:78–93. https://doi.org/10.1016/j.ins.2018.08.053

44. Zhang J, He Y, Feng W, Wang J, Xiong NN. Learning background-aware and spatial-temporal regularizedcorrelation filters for visual tracking. Applied Intelligence. 2022; p. 1–16.

45. Mueller M, Smith N, Ghanem B. Context-aware correlation filter tracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 1396–1404.

46. Danelljan M, Hager G, Shahbaz Khan F, Felsberg M. Learning spatially regularized correlation filters for visualtracking. In: Proceedings of the IEEE international conference on computer vision; 2015. p. 4310–4318.

47. Wang G, Luo C, Sun X, Xiong Z, Zeng W. Tracking by instance detection: A meta-learning approach. In: Proceedings of the IEEE/CVF conference on computer vision andpattern recognition; 2020. p. 6288–6297.

48. Yang T, Xu P, Hu R, Chai H, Chan AB. ROAM: Recurrently optimizing tracking model. In: Proceedings of the IEEE/CVF conference on computer vision andpattern recognition; 2020. p. 6718–6727.

49. Li P, Chen B, Ouyang W, Wang D, Yang X, Lu H. Gradnet: Gradient-guided network for visual object tracking. In: Proceedings of the IEEE/CVF International conference on computer vision; 2019. p. 6162–6171.

50. Liu Q, Yuan D, Fan N, Gao P, Li X, He Z. Learning dual-level deep representation for thermal infrared-tracking. IEEE Transactions on Multimedia. 2022;.

51. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T,et al. An Image is Worth 16x16 Words: Transformers for Image Recognition atScale. In: International Conference on Learning Representations;2021. Available from: https://openreview.net/forum?id=YicbFdNTTy.

52. Wang N, Song Y, Ma C, Zhou W, Liu W, Li H. Unsupervised deep tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision andPattern Recognition; 2019. p. 1308–1317.

53. Chen Z, Zhong B, Li G, Zhang S, Ji R. Siamese box adaptive network for visual tracking. In: Proceedings of the IEEE/CVF conference on computer vision andpattern recognition; 2020. p. 6668–6677.

54. Li B, Wu W, Wang Q, Zhang F, Xing J, Yan J. Siamrpn++: Evolution of siamese visual tracking with very deepnetworks. In: Proceedings of the IEEE/CVF Conference on Computer Vision andPattern Recognition; 2019. p. 4282–4291.

55. Dunnhofer M, Martinel N, Micheloni C. A Distilled Model for Tracking and Tracker Fusion. arXiv preprint arXiv:200704108. 2020;.

56. Xie F, Wang N, Yao Y, Yang W, Zhang K, Liu B. Hierarchical representations with discriminative meta-filters in dualpath network for tracking. In: Chinese Conference on Pattern Recognition and Computer Vision(PRCV). Springer; 2020. p. 303–315.

57. Yang T, Chan AB. Learning dynamic memory networks for object tracking. In: Proceedings of the European conference on computer vision (ECCV);2018. p. 152–167.

58. Nam H, Han B. Learning multi-domain convolutional neural networks for visualtracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 4293–4302.

59. Lukezic A, Vojir T, Čehovin Zajc L, Matas J, Kristan M. Discriminative correlation filter with channel and spatialreliability. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 6309–6318.

60. Wang Q, Zhang L, Bertinetto L, Hu W, Torr PH. Fast online object tracking and segmentation: A unifying approach. In: Proceedings of the IEEE/CVF conference on Computer Vision andPattern Recognition; 2019. p. 1328–1338.

61. Xu T, Feng ZH, Wu XJ, Kittler J. AFAT: adaptive failure-aware tracker for robust visual objecttracking. arXiv preprint arXiv:200513708. 2020;.

62. Yan B, Zhang X, Wang D, Lu H, Yang X. Alpha-refine: Boosting tracking performance by precise bounding boxestimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision andPattern Recognition; 2021. p. 5289–5298.

63. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 770–778.

64. Howard A, Sandler M, Chu G, Chen LC, Chen B, Tan M, et al. Searching for mobilenetv3. In: Proceedings of the IEEE/CVF international conference on computer vision; 2019. p. 1314–1324.