IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 8, 2021, accepted March 12, 2021, date of publication March 17, 2021, date of current version March 25, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3066294

Triplet Network Template for Siamese Trackers

TAO SHI[1], DONGHUI WANG[2], AND HONGGE REN[3]

[1]Tianjin Key Laboratory for Control Theory and Applications in Complicated Systems, School of Electrical and Electronic Engineering, Tianjin University of Technology, Tianjin 300384, China
[2]College of Electrical Engineering, North China University of Science and Technology, Tangshan 063210, China
[3]School of Control and Mechanical Engineering, Tianjin Chengjian University, Tianjin 300384, China

Corresponding author: Tao Shi (st99@email.tjut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61203343, and in part by the Natural Science Foundation of Hebei Province under Grant F2018209289.

ABSTRACT Siamese network based trackers describe object tracking as a similarity matching problem and these trackers achieve state-of-the-art performance on multiple benchmarks. However, due to the non-update of the appearance template and the change of the object appearance, the tracking drift problem often occurs, especially in the background clutter scene. Effective appearance template update methods can improve tracker performance, but most trackers use simple linear interpolation to update the template or do not update the initial template at all. Through experiments, we find that the channel response of the search region with the adjacent frame appearance template is often better than that with the initial frame appearance template. So we add the results of the previous frame prediction as a new template branch to the Siamese network to form a Triplet network. We applied the Triplet network to the SiamFC and SiamCAR, called TripFC and TripCAR. We tested on four challenging benchmarks (GOT-10K, OTB2013, OTB2015, UAV123). The experiments show that our method is powerful and effective, it can be easily embedded into the Siamese trackers. TripFC has a good effect on solving the problem of tracking drift. If necessary we can publish the code to facilitate research in this area.

INDEX TERMS Siamese trackers, tracking drift, triplet network, visual tracking.

I. INTRODUCTION

As an important part of computer vision, visual object tracking technology has been greatly developed with the rapid development of artificial intelligence technology and the continuous progress of hardware facilities in recent years. Visual object tracking has always been a challenging work in video surveillance [1], video understanding [2], autonomous driving [4], to robotics [3], navigation, positioning, and other fields. It often faces the influence of object occlusion, disappearance, morphological change, and other factors [5].

In recent years, visual object tracking has been a basic topic, and many deep learning-based trackers have achieved state-of-the-art performance on multiple benchmarks. The current popular visual tracking methods revolve around the Siamese network based architecture, such as SiamCAR [7], SiamFC++ [8], SiamBAN [9], which have excellent performance recently. The Siamese network describes the tracking problem as a object matching problem. First, the object template and the search region are extracted by convolution neural network with the same weight, and use the cross-correlation operation for position encoding. Then, the object position and size are predicted by classification and border regression. However, the object template of these Siamese network trackers is often the initial frame object template, and the template is not updated in the tracking process. When the object changes in shape or occlusion, the tracking drift often occurs.

In order to solve the above problems, template updating is generally used to adapt to the current object state. Template updating using linear interpolation with fixed learning rates is a simple method. It assumes that the object shape conversion rate in the video is constant, and replaces the template at a fixed time [10]. Some recent template update methods with excellent performance all use complex strategies to update the template, such as using a neural network to determine whether the template needs to be updated and how to update it [25]. Regardless of the simple or complex update strategy, the update is performed on the initial template. These update methods will change the object information in the initial template. With the continuous update of the template, the object template will often lose the information of the initial state of

The associate editor coordinating the review of this manuscript and approving it for publication was Maurizio Tucci.

the target. This template update method will make the tracker perform better in a short time after the template update, but when tracking for a long time, a large amount of useless information will be obtained to affect the tracking effect.

In this paper, we use the Triplet network [11] to solve the problem of template updating. The initial template contains complete object information, so the integrity of the initial template should be maintained during the tracking process and should not be changed at will. Therefore, we add a supplementary template branch on the basis of the Siamese network to implement the update of the template. The Siamese network tracker contains two branches: a object template branch which takes the object template as input, and a search branch which takes the search region as input. Compared with Siamese network, Triplet network have one more input branch than Siamese network. Through experi-ments, we find that the channel response of the search region with the adjacent frame is often better than the search region with the initial frame. Therefore, we add the results of the previous frame prediction as a supplementary template branch to the Siamese network to form Triplet network. We applied the Triplet network to the SiamFC [12] (TripFC) and SiamCAR [7] (TripCAR), Experiments on multiple bench-marks with good results, TripFC has a good effect on solving the problem of tracking drift.

Our main contributions are as follows:

1) We introduce the Triplet network into the Siamese net-work tracker. Add the adjacent frame result as a supplementary template branch to the Siamese network to form the Triplet network.
2) Our Triplet network does not require training and can be directly applied to the Siamese network tracker. Easy to use.
3) Our network is simple and effective. The experiment shows that TripFC has a good effect on solving the problem of tracking drift and improves the results of SiamFC with an absolute gain of 2.2% in terms of success score on OTB2013.

II. RELATE WORKS

In this section, we mainly review the tracking framework and template update method.

A. TRACKING FRAMEWORK

Most existing trackers are based on tracking-by-detection or template matching method. The tracker based on tracking-by-detection treats object location as a classification problem, in which the decision boundary is implemented by the online learning classifier. In order to solve the drift problem in the tracking process, Zhong *et al.* [48] proposed a probabilistic method in a new type of weakly supervised learning scenario to judge the position of the object and the accuracy of each tracker. And online evaluation and heuristic training are used to make tracking faster and more effective. In order to make full use of the object motion model, Zhong *et al.* [49] use

FIGURE 1. Through comparison, it can be seen that the Triplet network can suppress the drift phenomenon in the SiamFC, and the coincidence between the prediction bounding box and the ground-truth bounding box is high.

the data-driven motion model learned by deep recursive reinforcement learning to coarsely locate the object, and use the appearance model to perform fine positioning based on the coarse positioning. Guo *et al.* [50] proposed a fast compression tracking scheme through structural regularization and online data-driven sampling. In order to solve the motion blur, [51] proposed a GAN network to improve the robustness of the tracker to motion blur. In order to solve the optimization complexity brought by spatial regularization, [52] introduced selective spatial regularization.

The Siamese network describes the tracking problem as an object matching problem. The Siamese network tracker contains two branches, one is the template branch and the other is the search region branch. The input image is subjected to feature extraction through a convolutional neural network (CNN) to obtain a feature map, and then the feature map of the two branches are subjected to a cross-correlation operation to obtain response map. Although SiamFC is not the first to use the Siamese network for object tracking, it is appearance breaks the dominance of correlation filtering. The object template and search region are extracted through a fully-convolutional network, and the obtained feature maps are subjected to cross-correlation operations, achieve dense and effective sliding window evaluation.

Inspired by the region proposal network for object detection, the SiamRPN [13] tracker performs the region proposal extraction after the Siamese network outputs. SiamRPN defines the tracking task as a one-time detection task, which consists of a Siamese Sub-network and a Classification-Regression Sub-network. The template branch uses a Siamese network, and the detection branch uses a regional proposal network (RPN) [14].

As the development of object tracking, AlexNet [15] is not enough to meet the requirements of tracker. In order to solve this problem, SiamRPN++ [16] introduces the deep network into the Siamese network tracker, and uses random translation to solve the problem of translation invariance. SiamRPN++ uses ResNet [17] as the backbone of the Siamese network,

proposes depth-wise Cross Correlation Layer, and adds multiple layers of information fusion to the tracker according to different feature information between different layers of the deep network.

The above Siamese trackers are anchor-based trackers, SiamCAR uses the anchor-free strategy to convert the regression output of the network into the distance between the point on the search patch and the four edges of the selected ground-truth box on the feature map. Determine the best object center point by observing the classification score map and the centrality score map. Then extract the distance between the best target center point and the four sides of the box to get the prediction box.

However, since the size of the object feature region needs to be determined in advance, the cross-correlation method either retains a lot of unfavorable background information or loses a lot of foreground information. To solve this problem, SiamGAT [18] proposes a Graph Attention Module (GAM) to achieve part-to-part matching of information embedding. Compared with traditional methods based on cross correlation, GAM [19] can greatly eliminate their shortcomings and effectively transfer object information from the template patch to the search region.

B. OBJECT TEMPLATE UPDATE

Object template update is an important part of object tracking and plays an important role in improving tracking accuracy. Generally, linear interpolation [20]–[24] is used to update the object template, but this template update method is often inaccurate. The object often appears occluded, disappeared, blurred, and light changes. A simple template update method may make the new template unable to fully express the object and reduce the tracking accuracy. DSiam [53] can effectively learn the changes of target appearance variation online and suppress the background from the previous frame through the fast conversion learning model, use element-based multi-layer fusion to adaptively integrate the network output. Unlike traditional trackers, it can be trained on video sequences.

Recently, Yang *et al.* [25] introduced Long short time memory (LSTM) to control template updates. Pass the feature map that obtained through CNN to the LSTM [28] that controls the reading and writing of the memory stack. LSTM returns a residual template, which is convolved with the object feature map to obtain a response map. Use the same interpolation method as SiamFC to get the final bounding box, and write this result to the memory stack.

Zhang *et al.* [26] proposed an update component called UpdateNet, which can be easily embedded in the Siamese tracker to achieve adaptive update of the template. Update-Net takes the initial frame template, the previous cumulative template, and the current frame template as the input, outputs the updated cumulative template. Use the distance to the ground-truth object template of the next frame to train the UpdateNet. Experiments on four standard tracking benchmarks show that UpdateNet is universal, can be embedded in all Siamese trackers, and can effectively update template to improve tracking performance.

Dai *et al.* [27] proposed an offline training Meta-Updater to solve the problem of template update. Meta-Updater can effectively integrate geometric, discriminative, and appearance information, use cascaded LSTM to mine sequence information, and finally learn a binary output to guide the update. Geometric information refers to a series of bounding boxes in consecutive frames that contain important object motion information, such as speed and scale changes, that is, the coordinate vector of the bounding box. The discriminant information is that the maximum value of the response graph is also discriminant information, but it is unstable, so CNN is used to fully mine the information of the response graph.

Compared with the above complex template updating method, our template updating method does not require offline training. The newly added supplementary template branch uses the backbone network of the Siamese network tracker to turn the Siamese network into a Triplet network. This method is simple in structure and can be directly embedded into the Siamese network tracker.

III. METHOD

In this section, we will introduce our Triplet network tracker in detail. TripFC is improved on the basis of SiamFC and TripCAR is improved on the basis of SiamCAR. The overall frameworks is shown in the Fig.2, we introduce a new supplementary template branch in the Siamese network to form a Triplet network.

A. OBJECT TEMPLATE

Object tracking task can be regarded as a similarity matching problem. Specifically, the Siamese network is trained offline and evaluated online to locate the template image in a larger search image. These two branches perform the same transformation on the same Siamese backbone to embed them in the feature map of subsequent tasks.

Some recent Siamese network trackers all use the initial frame as the object template, but in the tracking process, the information contained in the initial frame template is often insufficient to support the tracking of subsequent frames. The object often has a series of problems such as occlusion, light transformation, and deformation. In order to solve these problems and improve tracking performance, many trackers have added template update methods, but these methods are often simple linear interpolation methods, which are not enough to cope with object changes.

The initial frame template, which depicts the most basic information of the object, plays a dominant role in the tracking process, so the initial frame template cannot be abandoned. Through experiments, we found that the response map of the adjacent frame object template is better than the response map of the initial frame object template. As shown in the Fig.3, the adjacent frame response map can better distinguish the foreground and the background. Therefore, on the basis of the Siamese network, we add a supple-

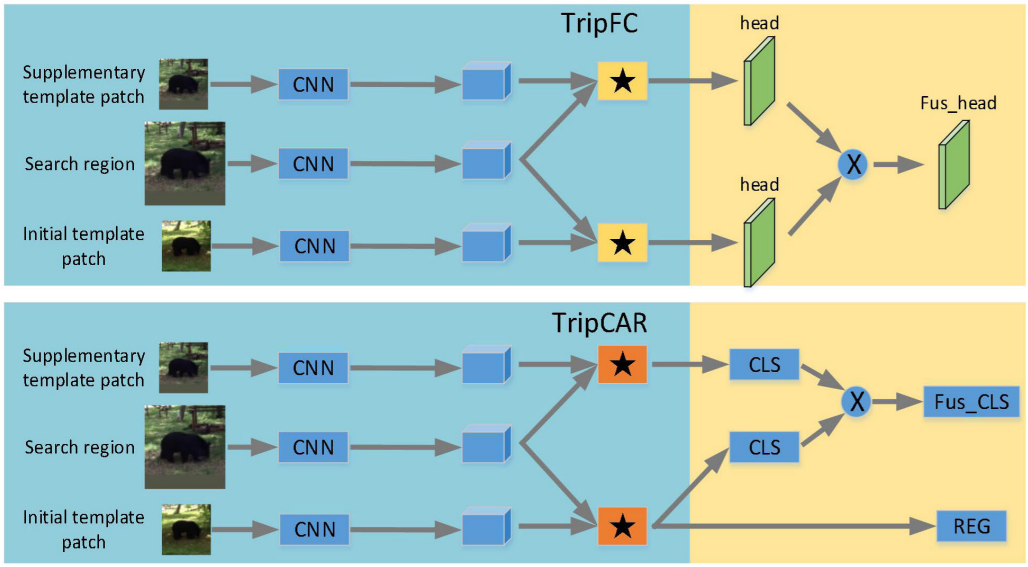


FIGURE 2. Overview of our tracking framework. The left side is the Triplet network, which contains three branches: a supplementary template branch, a search region branch, and an initial template branch. The right half is information fusion, which fusing two response maps into one response map. In the figure, $\star$ denotes Cross Correlation Layer, and X denotes information fusion.

mentary object template branch to form a Triplet network. That is, use the current prediction result of the tracker as a template to predict the search region of the next frame. During the tracking process, the supplementary template will continue to change with the tracker results, so that the supplementary template and the search region are adjacent in sequence.

B. TRIPLET NETWORK FOR SIAMFC

Based on SiamFC, by adding a supplementary object template branch, we proposed TripFC. In the Triplet network, we use a fully-convolutional network without padding as the feature extraction network. Compared with the Siamese network, the Triplet network consists of three branches: an initial object branch that takes the initial object template patch Z as the input, a search branch that takes the search region X as the input, and a supplementary object template branch that takes prediction object patch P of the previous frame as the input. The backbone model in three branches share the same CNN architecture. Z, X, and P through the same transformation to obtain $\varphi(Z)$, $\varphi(X)$,and $\varphi(P)$ to embed them in the feature space of subsequent tasks.

Z is the initial object template patch, collected in the initial frame, and Z remains unchanged during the entire tracking process. P is the supplementary object template patch, that is, the result of the previous frame tracker prediction, which will be continuously updated with the tracking process. In order to embed the information of these branches, we use the feature map of the template patch and the feature map of the search region to perform cross-correlation operations. Like SiamFC, a single-channel compressed response map R is generated. Since we have two template patch branches and a search region branch, our response map:

$$\begin{aligned} R_1 &= \varphi(Z) \star \varphi(X) \\ R_2 &= \varphi(P) \star \varphi(X) \end{aligned} \tag{1}$$

where $\star$ denotes the cross-correlation operation. R_1 and R_2 have the same size. Based on the response map, each position in R can be mapped back to the input search region. SiamFC uses positive sample scoring to predict the object position. In order to use the information of the two response maps, we define R_{fus}:

$$R_{\text{fus}} = R_1 + \lambda_{\text{R}} R_2 \tag{2}$$

where λ_{R} denotes the weight of R_2.

C. TRIPLET NETWORK FOR SIAMCAR

In order to prove the wide practicability of the Triple network, we conducted experiments on the state-of-the-art Siamese network trackers. Recently, state-of-the-art Siamese network trackers all use classification branch and regression branch to predict the bounding box of the target, such as SiamCAR. Based on SiamCAR, by adding a supplementary object template branch, we proposed TripCAR.

Like TripFC, the backbone network of TripCAR also contains three branches, which includes initial template branch Z, search region branch X, and supplementary template branch P. Unlike TripFC, TripCAR uses depth-wise correlation to produce multi-channel response map, which can retain more semantic information:

$$\begin{aligned} R_1 &= \varphi(Z) \star \varphi(X) \\ R_2 &= \varphi(P) \star \varphi(X) \end{aligned} \tag{3}$$

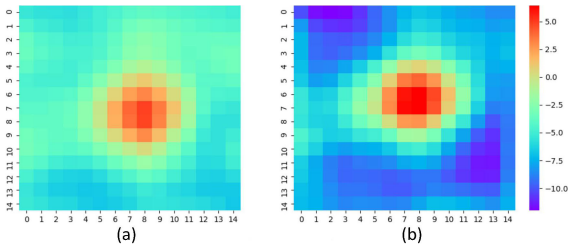


FIGURE 3. (a) The cross-correlation response map of the initial frame object template and the search region. (b) The cross-correlation response map of the adjacent frame object template and the search region.

where $\star$ denotes the depth-wise correlation. R_1 and R_2 have the same size. The generated response map R has the same number of channels as $\varphi(X)$.

Use classification branch to predict the location of the object, and use regression branch to predict the size of the object bounding box. The classification branch contains two sub-branches: a classification sub-branch and a center-ness sub-branch. Through experiments, we found that the application of the regression branch will reduce the effect. So our response map:

$$\begin{aligned} R^1_{\text{cls}} &= \varphi_{\text{cls}}(R_1) \\ R^1_{\text{cen}} &= \varphi_{\text{cen}}(R_1) \\ R^1_{\text{reg}} &= \varphi_{\text{reg}}(R_1) \\ R^2_{\text{cls}} &= \varphi_{\text{cls}}(R_2) \\ R^2_{\text{cen}} &= \varphi_{\text{cen}}(R_2) \end{aligned} \tag{4}$$

where φ denotes the information extraction operation. So we can get the final response map:

$$\begin{aligned} A_{\text{cls}} &= R^1_{\text{cls}} + \lambda_{\text{cls}} R^2_{\text{cls}} \\ A_{\text{cen}} &= R^1_{\text{cen}} + \lambda_{\text{cen}} R^2_{\text{cen}} \\ A_{\text{reg}} &= R^1_{\text{reg}} \end{aligned} \tag{5}$$

where λ_{cls} denotes the weight of R_{cls}, λ_{cen} denotes the weight of R_{cen}.

D. TRACKING PHASE

The purpose of tracking is to find the bounding box of the object in the current frame. In TripFC, we use multiple anchor ratios to predict the bounding box. Crop and resize the search region according to different anchor ratios, cat them together and send them to the feature extraction network. Empirically, we found that the anchor ratios adopt $[-2, -0.5, -1]$ delivers stable tracking results. We take the response map with the largest response value in the response graph as the final response map A:

$$\begin{aligned} j &= \arg\max_i(\text{amax}(R^i_{\text{mix}})) \\ A &= R^j \end{aligned} \tag{6}$$

where i denotes the sequence number of different response map, j denotes the sequence number of the response map with the largest response value, amax denotes take the maximum

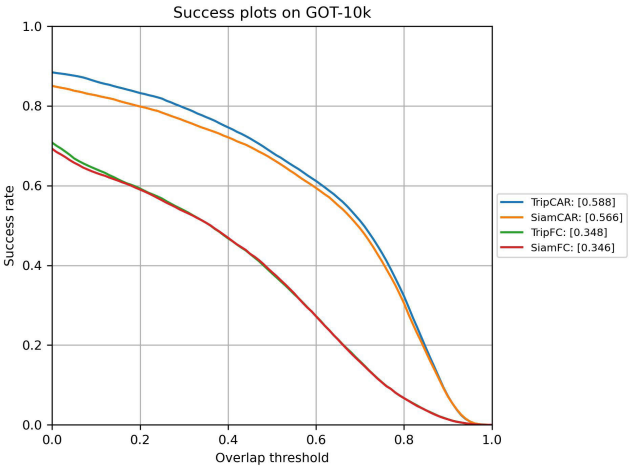


FIGURE 4. Compared with SiamFC and SiamCAR on GOT-10K, TripCAR has achieved better results in performance.

value in the response map. Then the position of the object is predicted as:

$$q = \arg\max_{x,y}(A) \tag{7}$$

where q is the central position of the object.

TripCAR uses anchor-free method to predict the object bounding box. Compared with anchor-based, anchor free reduces more hyperparameter settings. Then the position of the object is predicted as:

$$\begin{aligned} p &= \arg\max_{x,y}(A_{\text{cls}}) \\ q &= p \times A_{\text{cen}p} \end{aligned} \tag{8}$$

where q is the central position of the object.

IV. EXPERIMENTS

A. IMPLEMENTATION DETAILS

The proposed TripFC and TripCAR are implemented in Python with Pytorch. For comparison, the input size of the template patch and search region is set to be the same as SiamFC and SiamCAR. The template patch uses 127 pixels, and the search region uses 255 pixels.

In TripFC, we use the same Alexnet as SiamFC as the backbone of the Triplet network. The Alexnet is pretrained on ImageNet. We use GOT-10K [29] training set to train our network. GOT-10K contains 87 movement modes of 560 kinds of moving objects, and provides 10,000 video clips containing 1,500,000 manually labeled bounding box. During the training process, the batch size is set as 8 and totally 50 epochs are performed by using stochastic gradient descent (SGD) with an initial learning rate 0.001. The learning rate is adjusted according to exponential decay. For the first 20 epochs, the parameters of the backbone network are frozen while training cross-correlation layer. In the last 30 epochs, the entire network trains together.

In TripCAR, we use the tracking model provided by Siam-CAR. No more training to ensure the accuracy of the experiment.

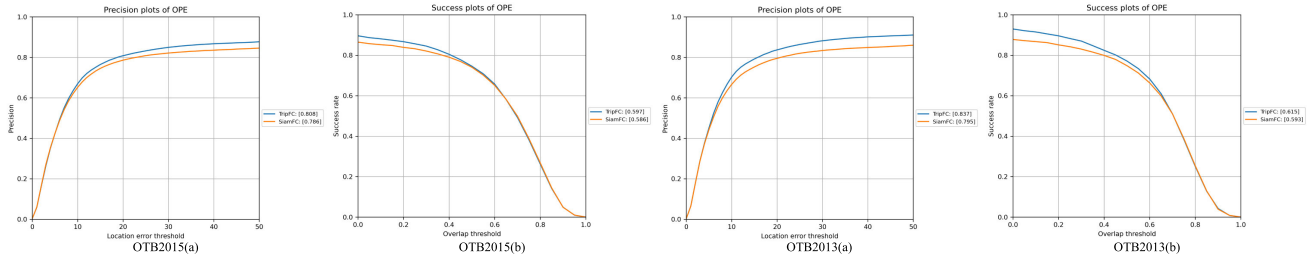


FIGURE 5. Compared with SiamFC on OTB2015 and OTB2013, TripFC has achieved better results in performance. The upper part is the precision plots and success plots of OTB2015. The lower part is the precision plots and success plots of OTB2013.

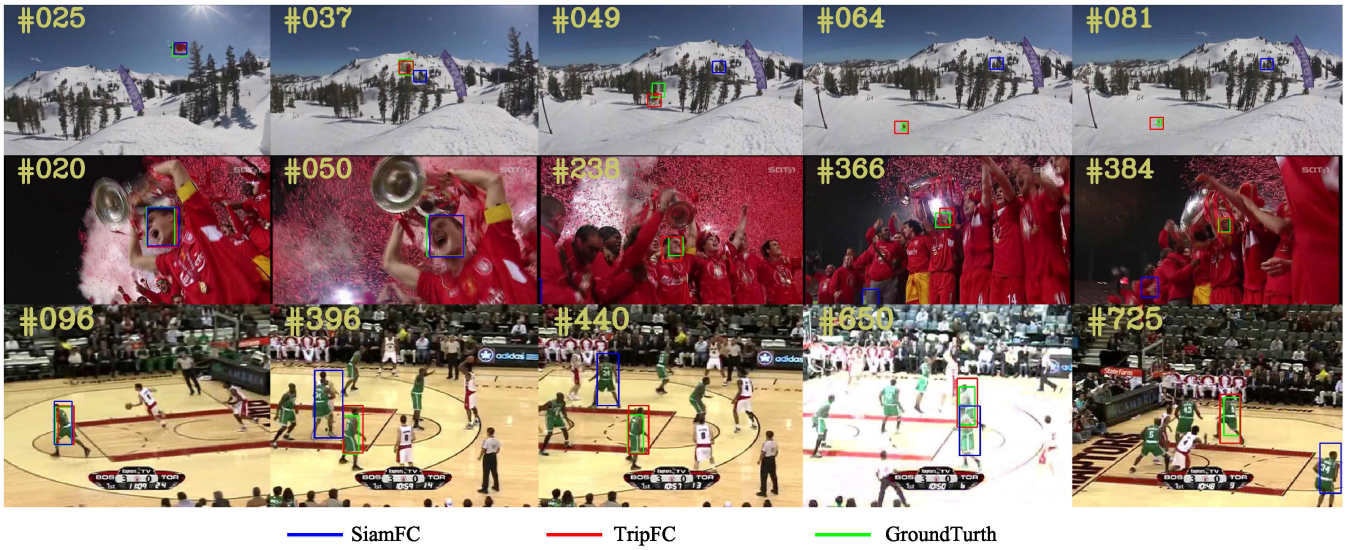


FIGURE 6. Compared with SiamFC on Skiing, Soccer, and Basketball. TripFC has achieved better results in performance and effectively solved the tracking drift problem.

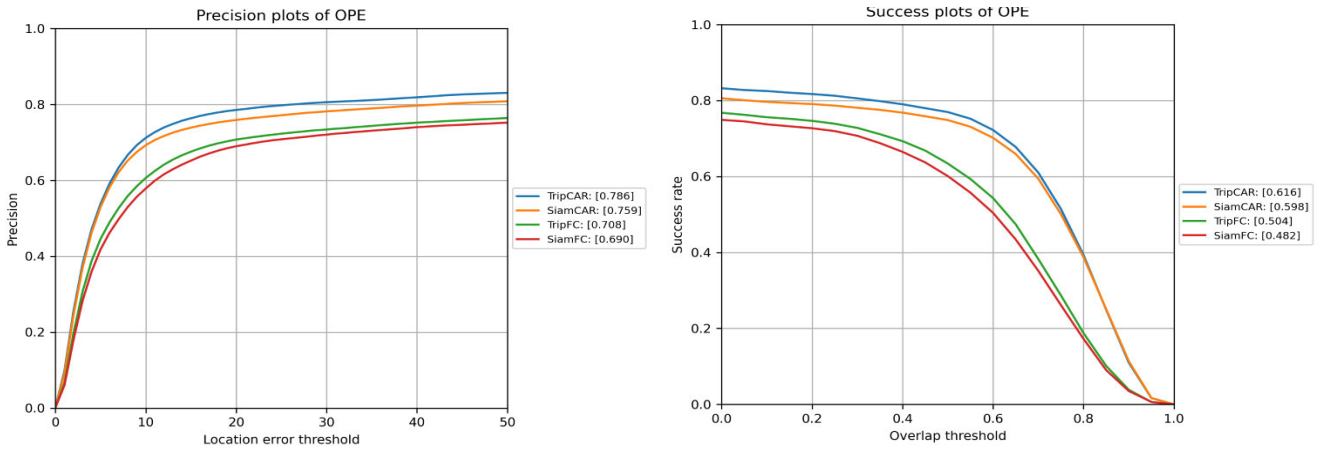


FIGURE 7. Compared with SiamFC and SiamCAR on UAV123. TripFC and TripCAR have achieved remarkable results.

To ensure the validity of the experiment, we used the GOT-10K test set, UAV123 [43], OTB2013 [30], and OTB2015 [5] for testing.

During the test, we adopted an offline tracking strategy. The object in the initial frame are collected as the initial template patch, so that the initial template branch of the Triplet network can be calculated and fixed in advance during the entire tracking period. Use the search region in the current frame as the input of the search region branch. Use the prediction result of the previous frame as a supplementary template patch and send it to the supplementary template branch. The anchor ratios we adopt $[-2, -0.5, 1]$.

B. RESULTS ON GOT-10K

GOT-10K training set and test set are not overlapping. By studying the influence of video number, target category,

TABLE 1. The effect of different λ on GOT-10K. ↑ means bigger is better.

Tracker	λ_{cls}	λ_{cen}	λ_{reg}	GOT-10K		
				AO↑	$SR_{0.5}$↑	$SR_{0.75}$↑
SiamCAR	0.0	0.0	0.0	0.566	0.667	0.410
TripCAR	0.1	0.1	0.1	0.577	0.676	0.421
TripCAR	0.2	0.2	0.2	0.577	0.673	0.416
TripCAR	0.4	0.4	0.4	0.561	0.648	0.399
TripCAR	1.0	1.0	1.0	0.542	0.622	0.381
TripCAR	0.3	0.3	0.3	0.581	0.676	0.423
TripCAR	0.3	0.3	0.0	**0.588**	**0.683**	**0.431**
TripCAR	0.3	0.0	0.0	0.579	0.679	0.426

TABLE 2. Comparisons on GOT-10K. ↑ means bigger is better.

Tracker	GOT-10K		
	AO↑	$SR_{0.5}$↑	$SR_{0.75}$↑
SiamFC	0.346	0.353	0.098
TripFC	0.348	0.380	0.107
SiamRPN-R18	0.483	0.581	0.270
SPM	0.513	0.593	0.359
SiamRPN++	0.517	0.616	0.325
ATOM	0.556	0.634	0.402
SiamCAR	0.566	0.667	0.410
TripCAR	**0.588**	**0.683**	**0.431**

motion category and repetition time, the final test set contains 180 videos, 84 types of moving objects and 32 types of motion. Except for the person class, all object classes between the training video and the test video are non-overlapping.

Each tracker conducts 3 experiments and averages the score to ensure a reliable evaluation.

We tested on the designated test set and sent the results to the official website for evaluation. The evaluation indicators provided include average overlap (AO: the average of overlap rates between tracking results and ground-truths over all frames) and success rate (SR: success rate, the percentage of successfully tracked frames where overlap rates are above a threshold). The $SR_{0.5}$ represents the rate of successfully tracked frames whose overlap exceeds 0.5, while $SR_{0.75}$ represents the rate of successfully tracked frames whose overlap exceeds 0.75.

We evaluated TripFC and TripCAR. The comparison proves that our TripCAR has been successfully improved in SiamCAR. Fig. 4 shows that our tracker is better than Siam-CAR, and Table 1 shows the comparison details of different parameters. As shown in Table 1, for the supplementary template patch, it performs best when coefficient adopts [0.3, 0.3, 0.0]. Our TripCAR improves the scores by 2.2%, 1.6%, and 2.1% relatively for AO, $SR_{0.5}$ and $SR_{0.75}$. As shown in Table 2, we compare our trackers with state-of-the-art trackers including SiamRPN [13], SPM [44], SiamRPN++ [16], and ATOM [45].

TABLE 3. Comparison between the proposed trackers. ↑ means bigger is better.

Tracker	OTB2015	
	Precision↑	Success↑
LCT	0.761	0.561
CF2	0.837	0.561
HDT	0.847	0.564
Staple	0.783	0.581
CFNet	0.777	0.586
SiamFC	0.786	0.586
SINF	0.788	0.592
SRDCF	0.789	0.598
TripFC	0.808	0.597

TABLE 4. The effect of different λ on OTB100. ↑ means bigger is better.

Tracker	λ_R	OTB2015	
		Precision↑	Success↑
SiamFC		0.786	0.586
TripFC	0.15	0.800	0.591
TripFC	0.20	**0.808**	**0.597**
TripFC	0.25	0.802	0.594
TripFC	0.30	0.802	0.596

Through experiments, we found that when the weight of the supplementary template is set to 1 or a larger number, the tracking performance will be worse. This shows that there is a lot of uncertainty in the predicted results of the tracker, and the importance of the initial template to the tracker.

C. RESULTS ON OTB2015

OTB2015 contains 100 challenging video sequences. The sequence is manually labeled with 9 attributes to represent challenging aspects of visual tracking. It includes illumination changes, scale changes, occlusion, deformation, motion, blur, fast motion, in-plane rotation, out-of-plane rotation, out-of-view, background clutter, and low resolution.

We evaluated TripFC with different parameters and compared with SiamFC. The success plots of OPE and precision plots of OPE for each tracker are evaluated, as shown in the Fig.5, the tracking performance has been significantly improved. As shown in Table 3, we compare the proposal trackers including LCT [32], CF2 [33], HDT [34], staple [35], CFNet [36], SiamFC, SINF [37], and SRDCF [38] on the OTB2015 benchmarks. Table 4 shows the comparison details with different parameters. We can get that the addition of the supplementary template branch improves the performance of the tracker, the precision is increased by 2.2%, and the success is increased by 1.1%.

D. RESULTS ON OTB2013

OTB2013 contains 50 challenging video sequences, the content of which is contained in OTB2015. The test sequences are

TABLE 5. Comparison between the proposed trackers. ↑ means bigger is better.

Tracker	OTB2013	
	Precision↑	Success↑
KCF	0.740	0.514
DSST	0.740	0.554
MEEM	0.830	0.566
SAMF	0.785	0.579
CFNet	0.785	0.589
SiamFC	0.795	0.593
CNN-SVM	0.852	0.597
Staple	0.793	0.600
HDT	0.889	0.603
CF2	0.891	0.605
CSR-DCF	0.891	0.605
TripFC	0.837	0.615

TABLE 6. The effect of different λ on OTB2013. ↑ means bigger is better.

Tracker	λ_R	OTB2013	
		Precision↑	Success↑
SiamFC		0.795	0.593
TripFC	0.1	0.808	0.595
TripFC	0.2	**0.837**	**0.615**
TripFC	0.3	0.821	0.604
TripFC	0.4	0.825	0.601

manually tagged with 9 attributes to represent the challenging aspects, including illumination variation, scale, variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, out-of-view, back-ground clutters and low resolution. We evaluated TripFC with different on OTB2013 and compared it with SiamFC.

As shown in the Fig. 5, the precision is increased by 4.2%, and the success is increased by 2.2%. As shown in Table 5, we compare the proposal trackers including KCF [21], DSST [39], MEEM [40],SAMF [41], CFNet, CNN-SVM [31], Staple, HDT, CF2, and SCR-DCF [42]. Table 6 shows the comparison details under different parameters.

By comparison, we found that TripFC performed well on the Basketball, Skiing, and Soccer test sequences. As shown in Fig. 6, SiamFC has serious drift on these sequences, and our TripFC can avoid this phenomenon well.

E. RESULTS ON UAV123

UAV123 is a dataset of special scenes, which are all shot with drones. It has 91 videos, including 123 short sequences. The objects in the dataset mainly suffer from fast motion, large scale variation, large illumination variation, and occlusions, which make the tracking challenging.

We evaluated TripFC and TripCAR, as shown in the Fig. 7, Our trackers achieved good results. Our method significantly

TABLE 7. Comparison between the state-of-the-art trackers. ↑ means bigger is better.

Tracker	UAV123	
	Precision↑	Success↑
Staple	0.614	0.450
SRDCF	0.627	0.463
SiamFC	0.690	0.482
TripFC	0.708	0.504
ECO	0.688	0.525
SiamRPN	0.710	0.577
DaSiamRPN	0.724	0.569
SiamRPN++	0.752	0.610
SiamCAR	0.759	0.598
TripCAR	**0.786**	**0.616**

improves SiamFC with an absolute gain of 1.8% and 2.2%, in terms of precision and success. Compared with SiamCAR, TripCAR has improved precision and success by 2.7% and 1.6%. As shown in Table 7, we compared our trackers with 6 state-of-the-art trackers including Staple, SRDCF, ECO [46], SiamRPN, DaSiamRPN [47], Siam-RPN++. The results show that our method is effective.

V. CONCLUSION

In this paper, we use a Triplet network to improve the performance of the Siamese network tracker. We used this method to improve SiamFC and SiamCAR. Experiments show that our method is effective. This method does not require training and can be directly applied to the Siamese network tracker to improve the performance of the tracker. We only need to add a supplementary template branch, and apply the result of the previous frame network prediction as a supplementary template patch to the tracker. The supplementary template patch and the template patch go through the same process, and the two results obtained are merged. Evaluated on GOT-10K, UAV123, OTB2015, and OTB2013, the results show that the proposed update method does significantly improve the performance of the tracker. This structure is very simple and can be easily integrated into all trackers. Since the current framework is relatively simple, it can be easily modified for further improvement in the future.

REFERENCES

[1] S. Tang, M. Andriluka, B. Andres, and B. Schiele, "Multiple people tracking by lifted multicut and person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 3701–3710.

[2] B. Renoust, D.-D. Le, and S. Satoh, "Visual analytics of political networks from face-tracking of news video," *IEEE Trans. Multimedia*, vol. 18, no. 11, pp. 2184–2195, Nov. 2016.

[3] Z. Chen, S. Li, N. Zhang, Y. Hao, and X. Zhang, "Eye-to-hand robotic visual tracking based on template matching on FPGAs," *IEEE Access*, vol. 7, pp. 88870–88880, 2019.

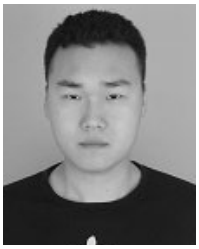
[4] C. Wu, H. Sun, H. Wang, K. Fu, G. Xu, W. Zhang, and X. Sun, "Online multi-object tracking via combining discriminative correlation filters with making decision," *IEEE Access*, vol. 6, pp. 43499–43512, Jun. 2018.

[5] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.
[6] Z. Zhu, W. Wu, W. Zou, and J. Yan, "End-to-end flow correlation tracking with spatial-temporal attention," 2017, *arXiv:1711.01124*. [Online]. Available: http://arxiv.org/abs/1711.01124
[7] D. Guo, J. Wang, Y. Cui, Z. Wang, and S. Chen, "SiamCAR: Siamese fully convolutional classification and regression for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 6268–6276.
[8] Y. D. Xu, Z. Y. Wang, Z. X. Li, Y. Yuan, and G. Yu, "SiamFC++: Towards robust and accurate visual tracking with target estimation guidelines," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 12549–12556.
[9] Z. Chen, B. Zhong, G. Li, S. Zhang, and R. Ji, "Siamese box adaptive network for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 6667–6676.
[10] D. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, San Francisco, CA, USA, Jun. 2010, pp. 2544–2550.
[11] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," 2014, *arXiv:1412.6622*. [Online]. Available: http://arxiv.org/abs/1412.6622
[12] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *Proc. ECCV Workshop*, 2016, pp. 850–865.
[13] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 8971–8980.
[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. NISP Workshop*, 2015, pp. 91–99.
[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
[16] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN++: Evolution of siamese visual tracking with very deep networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 4277–4286.
[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.
[18] D. Guo, Y. Shao, Y. Cui, Z. Wang, L. Zhang, and C. Shen, "Graph attention tracking," 2020, *arXiv:2011.11204*. [Online]. Available: http://arxiv.org/abs/2011.11204
[19] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*. [Online]. Available: http://arxiv.org/abs/1710.10903
[20] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 4310–4318.
[21] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.
[22] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," 2016, *arXiv:1608.03773*. [Online]. Available: http://arxiv.org/abs/1608.03773
[23] J. Choi, H. J. Chang, T. Fischer, S. Yun, K. Lee, J. Jeong, Y. Demiris, and J. Y. Choi, "Context-aware deep feature compression for high-speed visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, Jun. 2018, pp. 479–488.
[24] H. K. Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 1144–1152.
[25] T. Yang and A. B. Chan, "Learning dynamic memorynetworks for object tracking," in *Proc. ECCV*, 2018, pp. 153–169.
[26] L. Zhang, A. Gonzalez-Garcia, J. V. D. Weijer, M. Danelljan, and F. S. Khan, "Learning the model update for siamese trackers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct. 2019, pp. 4009–4018.
[27] K. Dai, Y. Zhang, D. Wang, J. Li, H. Lu, and X. Yang, "High-performance long-term tracking with meta-updater," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 6297–6306.
[28] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," 2015, *arXiv:1506.04214*. [Online]. Available: http://arxiv.org/abs/1506.04214
[29] L. Huang, X. Zhao, and K. Huang, "GOT-10k: A large high-diversity benchmark for generic object tracking in the wild," 2018, *arXiv:1810.11981*. [Online]. Available: http://arxiv.org/abs/1810.11981
[30] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Columbus, OH, USA, Jun. 2013, pp. 2411–2418.
[31] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *Proc. ICML*, Lille, France, Jun. 2015, pp. 597–606.
[32] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Long-term correlation tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 5388–5396.
[33] E. Gundogdu and A. A. Alatan, "Good features to correlate for visual tracking," 2017, *arXiv:1704.06326*. [Online]. Available: http://arxiv.org/abs/1704.06326
[34] Y. Qi, S. Zhang, and L. Qin, "Hedged deep tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 5, pp. 1116–1130, May 2015.
[35] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 1401–1409.
[36] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr, "End-to-end representation learning for correlation filter based tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 5000–5008.
[37] R. Tao, E. Gavves, and A. W. M. Smeulders, "Siamese instance search for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 1420–1429.
[38] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Boston, MA, USA, Dec. 2015, pp. 4310–4318.
[39] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf.*, Nottingham, Britain, Sep. 2014, pp. 1–11.
[40] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: Robust tracking via multiple experts using entropy minimization," in *Proc. ECCV Workshop*, Zurich, Switzerland, Sep. 2014, pp. 188–203.
[41] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proc. ECCV*, Zurich, Switzerland, Sep. 2014, pp. 254–265.
[42] A. Lukezic, T. Vojir, L. C. Zajc, J. Matas, and M. Kristan, "Discriminative correlation filter with channel and spatial reliability," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 6309–6318.
[43] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for UAV tracking," in *Proc. ECCV*, 2016, pp. 445–461.
[44] G. Wang, C. Luo, Z. Xiong, and W. Zeng, "SPM-tracker: Series-parallel matching for real-time visual object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 3638–3647.
[45] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ATOM: Accurate tracking by overlap maximization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 4655–4664.
[46] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 6931–6939.
[47] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "Distractor-aware siamese networks for visual object tracking," in *Proc. ECCV*, 2018, pp. 103–119.
[48] B. Zhong, H. Yao, S. Chen, R. Ji, X. Yuan, S. Liu, and W. Gao, "Visual tracking via weakly supervised learning from multiple imperfect oracles," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1395–1401.

[49] B. Zhong, B. Bai, J. Li, Y. Zhang, and Y. Fu, "Hierarchical tracking by reinforcement learning-based searching and coarse-to-fine verifying," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2331–2341, May 2019.
[50] Q. Guo, W. Feng, C. Zhou, C.-M. Pun, and B. Wu, "Structure-regularized compressive tracking with online data-driven sampling," *IEEE Trans. Image Process.*, vol. 26, no. 12, pp. 5692–5705, Dec. 2017.
[51] Q. Guo, W. Feng, R. Gao, Y. Liu, and S. Wang, "Exploring the effects of blur and deblurring to visual object tracking," *IEEE Trans. Image Process.*, vol. 30, pp. 1812–1824, 2021.
[52] Q. Guo, R. Han, W. Feng, Z. Chen, and L. Wan, "Selective spatial regularization by reinforcement learned decision making for object tracking," *IEEE Trans. Image Process.*, vol. 29, pp. 2999–3013, 2020.
[53] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, and S. Wang, "Learning dynamic siamese network for visual object tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 1781–1789.

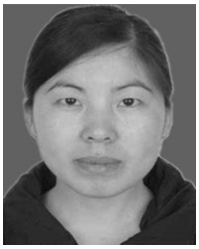
TAO SHI received the Ph.D. degree in control science and engineering from the University of Science and Technology Beijing, Beijing, China, in 2015.

He is currently an Associate Professor with the School of Electrical and Electronic Engineering, Tianjin University of Technology. His research interests include brain-like intelligent robots, robot vision, and biologically inspired intelligent computing.



DONGHUI WANG received the B.E. degree in electrical engineering and automation from the North China University of Science and Technology, Qinhuangdao, Hebei, China, in 2019, where he is currently pursuing the joint M.E. degree.

His research interests include computer vision, deep learning, and object tracking.



HONGGE REN received the B.E. degree in measurement and control technology and instrument from the North China University of Science and Technology, Qinhuangdao, Hebei, China, in 2003, and the M.E. and Ph.D. degrees from the Beijing University of Technology, Beijing, China, in 2007 and 2011, respectively.

She is currently an Associate Professor with the School of Control and Mechanical Engineering, Tianjin Chengjian University. Her research interests include cognitive robots, computer vision, and deep learning.