Residual Triplet Attention for Siamese Tracking

Jianming Zhang[1], Xiaoyi Huang[1], Huanqing Qiu[2], Osama Alfarraj[3], Amr Tolba[3]*

[1] School of Computer and Communication Engineering, Changsha University of Science and Technology, China
[2] School of Computer Science and Engineering, Hunan University of Information Technology, China
[3] Computer Science Department, Community College, King Saud University, Saudi Arabia
jmzhang@csust.edu.cn, xiaoyihuang@stu.csust.edu.cn, xuanxuanmsb@126.com, oalfarraj@ksu.edu.sa, atolba@ksu.edu.sa

Abstract

Visual object tracking is a significant technique for various intelligent applications based on the Internet. Benefited by the application of attention mechanism, visual object tracking has made great progress. Recent popular attention mechanisms have been shown to be effective in improving the quality of the visual features, because attention mechanisms pay more attention to global information. However, most existing attention mechanisms applied in object tracking can only process the spatial or channel dimensions of feature maps independently, resulting in lack of information interaction among them. To address this issue, we propose a Siamese tracker based on our residual triplet attention. Firstly, we introduce a spatial attention module to improve the quality of the template and search region features. Secondly, we propose a residual triplet attention module (RTAM) suitable for object tracking. Feature maps have three dimensions: width, height, and channel. The first two contain spatial information, while the last one contains channel information. Treating each dimension of the feature maps equally, RTAM implements the information interaction between any two of the three dimensions simultaneously, which effectively improves the robustness and success rate of tracking. The extensive experiments on five benchmarks, including VOT2016, VOT2018, UAV123, OTB100, and GOT-10k, show that our proposed tracker achieves established performance.

Keywords: Object tracking, Spatial attention, Residual triplet attention, Siamese network

1 Introduction

With the development of internet technology, lots of software and system are running in network environments. Camera-based smart IoT systems have been widely used in smart city, intelligent transportation, environment monitoring and so on. These systems all require intelligent analysis of videos. Visual object tracking is a fundamental task in video surveillance, which has rich research value and development potential. Its aim is to predict the location of an arbitrary target in each subsequent frame given its state in the initial frame of a video sequence. Visual tracking has been widely applied in abundant practical scenarios like automatic driving [1], intelligent video surveillance [2], intelligent human-computer interaction [3], robotics [4], intelligent transportation [5], motion analysis [4], visual navigation [6] and so on. With the continuous development of deep learning, great progress has been made in the field of visual object tracking. However, tracking task is still significantly challenging and difficult, especially for real world applications [7]. Since the problems such as occlusion, illumination change, rotation, interference, and complex scenes, trackers often fail to accurately locate the target, even losing the target. In addition, real-time performance is also a deficiency of many trackers.

Most of the recent popular visual object trackers [2, 7-10] are based on Siamese network. These trackers treat the tracking task as a one-shot object matching. Their overall architecture includes two branches, i.e. the template branch and the search region branch, and the aim is to learn the similarity mapping between the template and the search region. They usually extract feature maps of the template and search region via convolutional network, then perform multi-scale matching or perform classification and regression on the feature maps to obtain tracking results. However, due to the limitations of the spatial and semantic information in the feature maps, tracking results are not accurate or robust enough.

There are three dimensions in feature maps, including width dimension (W), height dimension (H), and channel dimension (C). The first two are termed spatial dimensions, and the last one is termed channel dimension. Due to the ability to establish intra-dependencies along channels or spatial dimensions in feature maps, recently popular attention mechanisms such as SENet [11] and CBAM [12] have shown that they can effectively enhance the visual features. Therefore, we introduce a spatial attention module (SAM) to enhance the visual features, so as to improve the tracking performance.

However, existing attention mechanisms can only implement the information interaction along the C dimension, W spatial dimension, or H spatial dimension independently, while the inter-dependencies among the C dimension and the two spatial dimensions is ignored. Inspired by the triplet attention [13] in semantic segmentation, we propose a residual triplet attention module calculating correlations along each dimension of the feature tensor. In addition, we

*Corresponding Author: Huanqing Qiu; E-mail: xuanxuanmsb@126.com

DOI: https://doi.org/10.70003/160792642024072504008

add a residual structure to make it suitable for tracking. The aim of this module is to construct the information interaction between the C and H dimensions, the C and W dimensions, as well as the H and W dimensions. Both channel attention and spatial attention are cleverly implemented simultaneously. So different dimensions can guide each other to effectively improve the success rate and robustness of the Siamese tracker.

In summary, our main contributions are as follows:

1. We introduce a spatial attention module (SAM) to effectively improve the quality of template and search region features that extracted by the backbone. The module is simple yet effective by making the features more discriminative for tracking.

2. We propose a residual triplet attention module (RTAM) suitable for Siamese tracking. It allows information interacted between the C and H dimensions, the C and W dimensions, as well as the H and W dimensions of feature maps by calculating their correlations simultaneously. Therefore, mutual guidance between different dimensions can be implemented. Moreover, RTAM cleverly implements channel attention and spatial attention at the same time in this way.

3. We propose a robust Siamese tracking network with SAM and RTAM. Through extensive experiments on five challenging benchmarks, including VOT2016, VOT2018, UAV123, OTB100, and GOT-10k, our tracker obtains leading tracking performance. It is worth noting that our tracker performs better than the state-of-the-art Siamese tracker SiamCAR on VOT2018.

The rest of this paper is as follows. Section 2 focuses on the related work about Siamese tracking as well as attention mechanism. Section 3 describes the architecture of our tracker at length. Section 4 introduces the experiments and results. Section 5 concludes and looks forward to this work. The codes and data are available at https://github.com/hxy013/RTA-Tracker.

2 Related Work

Nowadays, the common object trackers are mainly divided into the correlation filter-based trackers [14-15] and the deep learning-based trackers [16-17]. However, with the continuous development of deep learning, deep learning-based trackers have gradually become popular and dominate, and the optimum performance is trackers based on Siamese networks now. Therefore, we mainly review the tracking algorithms based on Siamese.

2.1 Visual Object Tracking Based on Siamese Network

For the past few years, Siamese-based trackers have attracted a great deal of attention since great balance in accuracy and efficiency, and their ability to perform end-to-end training. SiamFC [16] introduces the Siamese structure into the visual object tracking in a pioneering way, and constructs a fully convolutional Siamese network. Because of its lightweight and intuitive structure, it achieves good real-time performance. However, it has no obvious advantage in accuracy because it uses simple multi-scale test to estimate the scale of target. But on the basis of this work, many researchers have followed this work and proposed more complex and effective Siamese trackers. SiamRPN [18] introduces the Faster-RCNN [19] in object detection into visual object tracking, divides the tracking into two subtasks, i.e. classification and regression. It proposes a candidate region proposal network, namely RPN network, through which multiple anchor frames are preset to ensure both high speed and accuracy. Next, DaSiamRPN [9], SiamRPN++ [10], SiamMask [20], and SiamDW [3] have further improved this work in different ways. Among them, DaSiamRPN adds more abundant training samples, making the tracker better able to cope with long-term tracking scenarios. SiamRPN++ introduces a deeper backbone to further enhance the feature extraction ability of the tracker. SiamMask improves the tracker with a new idea by introducing the mask branch based on the correlation between the two visual tasks of segmentation and visual tracking. SiamDW explores the influence of receptive field, step size, and padding on object tracking with a large number of experiments, which makes that existing Siamese trackers achieve better tracking results.

With the continuous development of object tracking technology, the existing Siamese trackers using predefined anchor frames for regression have gradually become a bottleneck limiting the tracker performance. Therefore, Anchor-free based mechanisms begin to be introduced into Siamese network trackers and gradually become popular. Based on SiamFC [16], SiamFC++ [21] introduces an anchor-free mechanism to eliminate a good deal of prior knowledge required by tracking model to greatly improve the performance. SiamBAN [7] proposes an adaptive frame structure, while SiamCAR [2] uses a quality evaluation branch to make its classification more accurate. On the other hand, SiamAttn [22] takes another perspective and adds the recent popular attention mechanism to the object tracking task to achieve a large performance improvement. Inspired by this, we propose a novel attention mechanism, and construct the correlation between different dimensions of object feature maps from the perspective of dimension, so as to achieve the purpose of conducting spatial attention and channel attention simultaneously.

2.2 Attention Mechanism

In human vision, people always selectively concentrate more on a part of the information they see, while correspondingly ignoring some other unimportant or uninteresting information. Later, researchers introduced this information processing mechanism into computer vision, and it became the attention mechanism as we all know.

The purpose of attention is to compute correlation, and recently, attention mechanisms have been successfully applied in various visual tasks. SENet [11] implements a kind of channel attention, where a weight vector is computed over the channel dimensions on the input feature to make the network concentrates more on those channels that are more relevant to the object information, thus improving the performance. However, it ignores the correlation of features in the spatial dimensions. Therefore, CBAM [12] combines channel attention in SENet with spatial attention. The spatial attention focuses more on the importance between different locations in the feature map, giving more weight to the region where the target is located. Meanwhile, channel attention pays more attention to the part of the channel that is more relevant to the region of the target in the feature map, and assigns higher weight to that part of the channel. However, CBAM computes two kinds of attention in sequence, i.e., the channel attention is performed first and then computes the spatial attention. There is no information interaction between the two kinds of attention, which may lose some information in the feature map. Therefore, we propose a residual triplet attention module suitable for the object tracking task, and design three branches to compute the correlation from the C and W dimension, the C and H dimension, as well as the two spatial dimensions, the W and H dimension, respectively. This module not only realizes the combination of the channel attention and the spatial attention, but also enables the information interaction between these two kinds of attention. Then, we add the module into the Siamese network framework, which achieves a good performance enhancement.

3 Methods

In this section, we introduce the proposed tracker based on our residual triplet attention in detail. The architecture includes three parts, which are feature extraction network, feature enhancement network, and specific heads for classification and regression, as shown in Figure 1.

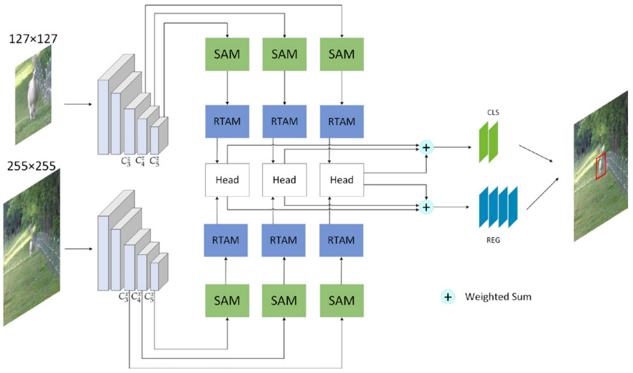


Figure 1. Illustration of our tracker framework

(Where C_i^j , $i \in \{3,4,5\}$, $j \in \{z,x\}$ represent the feature maps extracted by the backbone. SAM denotes the proposed spatial attention module, RTAM denotes the proposed residual triplet attention module, CLS denotes the classification map, and REG denotes the regression map.)

3.1 Overview

The feature extraction network utilizes ResNet-50 [23] network with shared parameters and the same structure. Since the shallow features of the backbone networks have better spatial information and the deep features have better semantic information. To make the most of the extracted features at different layers, after the template frame and search area input into the feature extraction network, the features of third, fourth as well as fifth layer of ResNet-50 are output into subsequent network as extracted template features and search region features, denoted as C_i^j , $i \in \{3,4,5\}$, $j \in \{z,x\}$. They are then fed into the SAM to improve the extracted feature representation, and make up for the limited quality of the features. Then, we input the enhanced features into the following residual triplet attention module. The module will construct the information interaction between the any two of the three dimensions in the input tensor. It can increase the weight of the regions that need attention while decreasing the weight of the background interference regions and output a refined tensor. Then, the resulting features will later be fed separately into three specific head networks for deep cross-correlation operations to fully fuse features. Each head network is divided into a classification branch and a regression branch, they output a classification map and a regression map respectively. For each branch, after the last head, the outputs of the three heads will be multiplied with weights corresponding and added. The weights are learnable and are optimized together with the network. Therefore, we can get the final results of classification branch and regression branch.

3.2 Spatial Attention Module

Considering the limitations of features extracted by traditional deep convolutional networks and the excellent global modeling capabilities of attention mechanism, especially inspired by SiamAttn [22], we introduce a Spatial Attention Module (SAM). Its structure is shown in Figure 2.

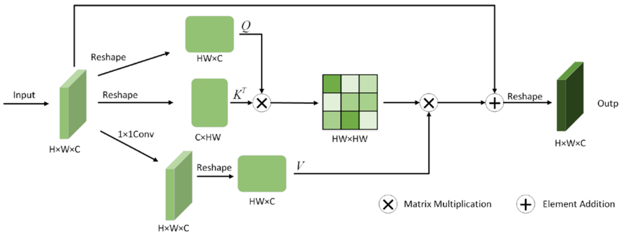


Figure 2. The detailed structure of our proposed Spatial Attention Module (SAM)

SAM uses three branches to process the input tensor. For the first and second branches, the $Input \in \mathbb{R}^{H \times W \times C}$ is firstly flattened into $Q \in \mathbb{R}^{HW \times C}$ and $K^T \in \mathbb{R}^{C \times HW}$ via Reshape operation respectively, which is followed by matrix multiplication operation. For the third branch, $Input$ is first transformed into another vector space by a 1×1 convolution, then reshaped into $V \in \mathbb{R}^{HW \times C}$, and then matrix multiplied with the result of matrix multiplication of the first two branches, we can get $M \in \mathbb{R}^{HW \times C}$, which is fed into a Softmax layer next. Finally, the result is reshaped into the size of $H \times W \times C$ and then added to the original $Input$ to obtain $Output \in \mathbb{R}^{H \times W \times C}$ of SAM. Therefore, the computational process of SAM can be expressed as equation (1).

$$SAM(Input) = Softmax(Q \times K^T)V + Input. \tag{1}$$

3.3 Residual Triplet Attention Module

Channel attention and spatial attention have gradually become popular in computer vision, such as CBAM [12] combines channel attention and spatial attention, but its disadvantage is that CBAM cannot perform channel and spatial attention operations simultaneously. Inspired by triplet attention in semantic segmentation, we propose a residual triplet attention module (RTAM) into Siamese tracking. Through this module, we can construct the information interaction between the C and H dimensions, the C and W dimensions, the H and W dimensions of the input by calculating their correlations. In addition, RTAM cleverly implements channel attention and spatial attention simultaneously. Benefited by RTAM, mutual guidance between different dimensions of the features can be implemented.

The RTAM consists of three branches, the processing flow of its first two branches is similar, as shown in Figure 3. RTAM takes in an input tensor and outputs a refined tensor of the same shape.

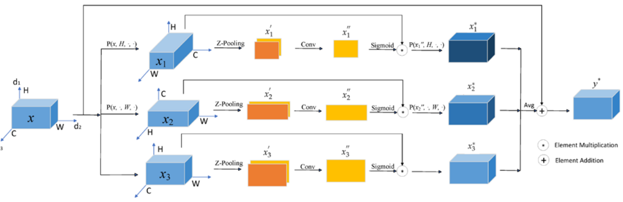


Figure 3. Architecture of RTAM

(It computes the correlation between the H and C dimensions, the W and C dimensions, as well as the H and W dimensions, respectively.)

3.3.1 Permute Operation and Z-pooling Operation

Before describing the RTAM in detail, we first define two operations. The first is the Permute operation P, which is used to keep a dimension of the input tensor $f \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ unchanged and exchange the other two dimensions. For example, P($f, d_1, \cdot, \cdot$) denotes that the first dimension of the input tensor f is unchanged, the second and third dimensions are exchanged.

The second operation is the Z-pooling, which is to perform both maximum pooling and average pooling on the third dimension d_3 of the input tensor $f \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ and concatenate the two pooling features to reduce the size of the third dimension to two, which achieves the purpose of retaining the abundant representation of the actual tensor while reducing its depth to reduce computation. Mathematically, our Z-pooling operation can be expressed by equation (2).

$$Z\text{-}pooling(f) = [MaxPool_{d_3}(f), AvgPool_{d_3}(f)]. \quad \textbf{(2)}$$

3.3.2 The Specific Structure of RTAM

In the first branch of RTAM, we compute the correlation between the first and the third dimensions, we first perform a P($x, H, \cdot, \cdot$) operation on the input $x \in \mathbb{R}^{H \times W \times C}$ to get the tensor $x_1 \in \mathbb{R}^{H \times C \times W}$, then perform a Z-pooling operation to aggregate its W dimension, and pass the result $x_1' \in \mathbb{R}^{H \times C \times 2}$ through a standard convolutional layer followed by a batch normalization layer, we can obtain $x_1'' \in \mathbb{R}^{H \times C \times 1}$, then we put it through a Sigmoid layer and multiply it as weight with the tensor x_1, and finally do a same $\mathrm{P}(x_1'', H, \cdot, \cdot)$ operation to return back to the same size as input x, and we can obtain the first branch result $x_1^* \in \mathbb{R}^{H \times W \times C}$.

For the second branch, which is used to compute the correlation between the second and third dimensions, this time we perform a $\mathrm{P}(x, \cdot, W, \cdot)$ operation on the input $x \in \mathbb{R}^{H \times W \times C}$, exchanging the H and C dimensions to obtain the tensor $x_2 \in \mathbb{R}^{C \times W \times H}$, we then do a Z-pooling aggregation on the H dimension to obtain $x_2' \in \mathbb{R}^{C \times W \times 2}$, which is fed into a standard convolutional layer followed by a batch normalization layer, then we can get $x_2'' \in \mathbb{R}^{C \times W \times 1}$. Next, it is activated by a Sigmoid layer and multiplied with the tensor x_2. Finally, we also do a $\mathrm{P}(x_2'', \cdot, W, \cdot)$ operation to return back to the size as same as input x to get the result $x_2^* \in \mathbb{R}^{H \times W \times C}$ of the second branch.

Considering that the two spatial dimensions of a three-dimensional tensor are equivalent, in other words, for the input $x \in \mathbb{R}^{H \times W \times C}$, its H dimension is equivalent to its W dimension, so that there is no difference in computing the correlation between the H dimension and the W dimension and the correlation between the W dimension and the H dimension. Therefore, we do not have to exchange the first and the second dimensions in the third branch of RTAM, and we can directly compute the correlation between the first two dimensions.

For the third branch, the input $x \in \mathbb{R}^{H \times W \times C}$ is directly subjected to a Z-pooling operation, here we aggregate the C dimension to obtain the tensor $x_3' \in \mathbb{R}^{H \times W \times 2}$, and similarly pass it through a standard convolutional layer and we can get $x_3'' \in \mathbb{R}^{H \times W \times 1}$, and then put it into a Sigmoid layer to get the attention weight of the third branch and multiplying it with x itself to get the result of the third branch $x_3^* \in \mathbb{R}^{H \times W \times C}$. Finally, the results of the three branches x_1^*, x_2^*, x_3^* are averaged together to get the result $y \in \mathbb{R}^{H \times W \times C}$. The whole process can be described as follows:

$$y = \frac{1}{3}(x_1 \sigma(\psi_1(x_1')) + x_2 \sigma(\psi_2(x_2')) + x\sigma(\psi_3(x_3'))), \quad \textbf{(3)}$$

$$x_i' = \text{Z-Pooling}(x_i), \quad \textbf{(4)}$$

$$x_1 = \mathrm{P}(x, H, \cdot, \cdot), \quad \textbf{(5)}$$

$$x_2 = \mathrm{P}(x, \cdot, W, \cdot), \tag{6}$$

$$x_3 = x, \tag{7}$$

where σ denotes the sigmoid activation operation, ψ_1, ψ_2, ψ_3 denotes the standard convolutional layers defined by kernel size k of the three branches of RTAM, respectively. The equation (3) can be described simply as follows:

$$y = \frac{1}{3}(x_1^* + x_2^* + x_3^*), \tag{8}$$

$$x_i^* = x_i \sigma(\psi_i(x_i^*)), i \in \{1,2,3\}. \tag{9}$$

It is worth noting that we also design a residual structure in order to speed up the convergence of the model and prevent the gradient from vanishing, making this module more favorable for visual tracking. We add the result y obtained through the three branches to the input x to obtain the final result y^* of RTAM.

$$y^* = y + x. \tag{10}$$

3.4 Classification and Regression Head

The head network includes two branches: classification and regression. The features from both the template and the search area will be fed into the two branches. Firstly, We copy $\varphi(x)$ and $\varphi(z)$ as $[\varphi(x)]_{cls}$, $[\varphi(x)]_{reg}$ and $[\varphi(z)]_{cls}$, $[\varphi(z)]_{reg}$ respectively. They are input into the corresponding branches for the cross-correlation operation [10], then we will get a classification map and a regression map. Each point in the correlation layer of the classification branch will output 2 channels for classifying the target and background, while for the regression branch, each point in the correlation layer will output 4 channels and will be used for the prediction of the tracking box.

$$\begin{aligned} p_{w \times h \times 2}^{cls} &= [\varphi(x)]_{cls} \star [\varphi(z)]_{cls}, \\ p_{w \times h \times 4}^{reg} &= [\varphi(x)]_{reg} \star [\varphi(z)]_{reg}, \end{aligned} \tag{11}$$

where $\star$ denotes the cross-correlation operation with $[\varphi(z)]_{cls}$, $[\varphi(z)]_{reg}$ as the kernel, $P_{w \times h \times 2}^{\mathrm{cls}}$ denotes the classification map, and $P_{w \times h \times 4}^{\mathrm{reg}}$ denotes the regression map.

3.5 Loss Function

In our tracker, the overall loss function adopted is similar to [7], as shown in equation (12).

$$L = \lambda_1 L_{cls} + \lambda_2 L_{reg}, \tag{12}$$

where $\lambda_1 = \lambda_2 = 1$, L_{cls} is the Cross Entropy Loss, defined as equation (13):

$$L_{cls}(p_{x,y}, g_{x,y}^*) = -(p_{x,y} log(g_{x,y}^*)), \tag{13}$$

where $P_{x,y}$ denotes the probability of belonging to the target area as predicted by the tracker, and $g_{x,y}^*$ denotes the truth label. L_{reg} is IoU (Inter-section over Union) loss function. We define the IoU loss function as the same as GIoU [24], which can be expressed as:

$$L_{reg} = 1 - IoU = 1 - \frac{B \cap B^*}{B \cup B^*}, \tag{14}$$

where IoU denotes the intersection and concurrency ratio of the tracker's predicted tracking frame B to the ground truth B^*, and IoU satisfies the condition $0 < IoU \leq 1$.

4 Experimental Results and Analysis

In this section, we first introduce some experimental settings about our residual triplet attention-based tracker. Then, we compare our tracker with some popular trackers on five challenging tracking benchmarks to demonstrate its great performance. Finally, we conduct ablation experiments on our two proposed modules to prove their effectiveness.

4.1 Experimental Settings

Our tracking architecture is built using PyTorch deep learning framework, the programming language is Python 3.7. We utilize ResNet-50 [23] as our backbone. During training, parameters and weights preprocessed on ImageNet are used to initialize our backbone, and the parameters of first two layers are unchanged. For the entire architecture, we utilize stochastic gradient descent on a training set including five datasets, COCO [25], VID [26], DET [26], Youtube-BoundingBoxes [27], and GOT-10k [28]. We set the batch size to 28, the optimized momentum size to 0.9, and the weight attenuation to 0.0001. There are 20 rounds of training, of which the first five rounds are warmed up during training using a learning rate that grew linearly from 0.001 to 0.005 in turn, and the next 15 rounds have an exponential decay in the learning rate from 0.005 to 0.00005. We freeze our backbone during the first ten rounds of training and fine-tune it in the last ten rounds at one-tenth the current learning rate. For each round, the network needs to train 1000,000 video frames. As shown in Table 1.

Our experiments are conducted on a virtual machine allocated by Baidu GPU Server cluster, which contains 4 Nvidia RTX 2080Ti and 500G hard disk space, and the size of the template image and search image for the input are 127×127 and 255×255, respectively. During training phase, we set the weights of both the classification loss and the regression loss to 1.0. During the testing phase, since the settings of the hyperparameters can have a large impact on the Siamese network, we used the hyperparameters that make the network perform best when testing on different benchmarks, respectively.

Table 1. The parameters and values in our experiments

Backbone	Batch_Size	Training_Epoch	Video_Per_Epoch	Template_Size	Search_Size
ResNet-50	28	20	1000000	127×127	255×255

4.2 Compare with Other Trackers

We have extensively evaluated our tracker on five famous tracking benchmarks, and compared with other mainstream trackers on these benchmarks. Our tracker attained overall optimum results. Moreover, we also compare our tracker with some popular trackers on several different tracking challenges, proving our tracker's excellent performance in the face of various tracking challenges.

4.2.1 Quantitative Comparison on Five Benchmarks

VOT2016 [29] and **VOT2018 [30]**. Both VOT2016 and VOT2018 are associated with the annual VOT Challenge and are common benchmarks for evaluating tracking performance. VOT2016 contains 60 video sequences, involving multiple tracking challenges like occlusion, illumination changes, motion changes, scale changes, and complex scenes, with a minimum frame number of 48 and a maximum frame number of 1507. VOT2018 also consists of 60 video sequences which contain 24 object categories, which are more finely labeled, with a minimum frame rate of 41 frames and a maximum frame rate of 1500 frames. Both VOT2016 and VOT2018 evaluate the trackers in terms of expected average overlap (EAO), accuracy, and robustness. We compare our tracker with some popular trackers on these two benchmarks. As shown in Table 2 and Table 3, our tracker can achieve first place on EAO, with 0.536 and 0.435, respectively. It should be noticed that the robustness of our tracker also can achieved best, 0.112 and 0.169, respectively, which shows that our tracker can face a wide range of challenges well. At the same time, in terms of accuracy our tracker is slightly behind, it can also reach an advanced level.

UAV123 [35]. UAV123 is a visual object tracking benchmark that includes aerial video collected by 123 low-altitude drone platforms, totaling approximately 110,000 frames. Many objects are characterized by fast motion, large scale change, lighting change, and occlusion, etc. At the same time, because the UAV is also in the motion state of the camera, making the benchmark more challenging. The benchmark mainly evaluates the tracker by two indicators: success rate and accuracy. We compared the proposed tracker with 7 popular trackers. As shown in Figure 4, our tracker achieves optimal results in both metrics with success rate of 0.626 and accuracy of 0.836.

Table 2. The comparison of some other trackers and our tracker on VOT2016
(The top three best results are bolded, underlined and italicized, respectively. ↑ denotes that the larger the number, the better; and ↓ denotes that the smaller the number, the better.)

	MCCT-H [31]	ECO-HC [15]	SiamRPN [18]	ECO [15]	MCCT [31]	DaSiamRPN [9]	SiamMask [20]	SiamRPN++ [10]	SiamR-CNN [32]	Ours
E↑	0.229	0.322	0.337	0.374	0.393	0.401	0.425	*0.437*	0.460	**0.536**
A↑	0.570	0.542	0.578	0.555	0.579	0.609	0.634	0.644	**0.645**	*0.634*
R↓	0.331	0.303	0.312	0.200	*0.186*	0.224	0.214	0.219	0.172	**0.112**

Table 3. The comparison of some other trackers and our tracker on VOT2018

	DaSiamRPN [9]	ATOM [33]	SiamR-CNN [32]	SiamMask [20]	SiamRPN++ [10]	SiamCAR [2]	SiamFC++ [21]	SiamKPN [34]	Ours
E↑	0.383	0.400	0.405	0.406	0.415	0.423	*0.426*	0.428	**0.435**
A↑	0.586	0.590	**0.612**	0.598	0.601	0.578	0.583	*0.596*	0.589
R↓	0.276	0.203	0.220	0.248	0.234	0.197	0.173	*0.187*	**0.169**

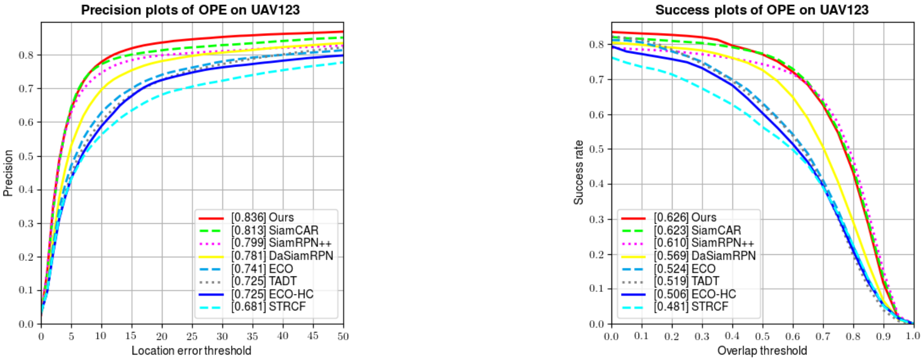


Figure 4. Precision and Success plots on UAV123

OTB100 [36]. OTB100 is also known as OTB2015, which is a famous generalized object tracking evaluation benchmark consisting of 100 video sequences that contain 22 object classes. The benchmark also evaluates trackers using two metrics, success rate as well as accuracy. We compared our tracker on the OTB100 with some popular trackers, Figure 5 shows the result. We can see that our tracker achieves second and third place in both success rate and accuracy metrics with 0.688 and 0.903, respectively, which is at an advanced level.

GOT-10k [28]. It is a large-scale object tracking evaluation benchmark with 10,000 video sequences. This dataset mainly evaluates trackers by average overlap rate (AO), success rate (SR), frame per second (FPS). It should be noticed that GOT-10k provides a unitive official training and evaluation platform for researchers. We follow the official protocol of GOT-10k to train the proposed tracker under the requirements of the protocol, and then submit our tracking results to the official platform for evaluation, and then we compare with several popular trackers, as shown in Table 4. Among them, our tracker attains the AO of 0.552, which is suboptimal, and the success rate is only second to SiamCAR [2], but the FPS of SiamCAR is significantly lower than that of our tracker. On the whole, our tracker is also very competitive on GOT-10k dataset.

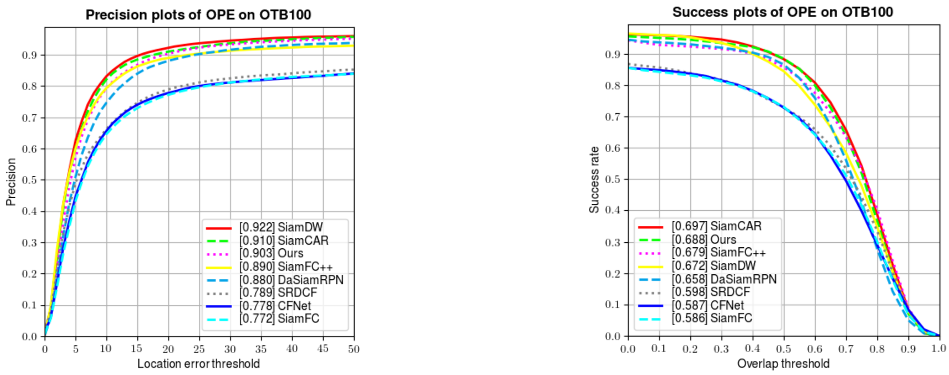


Figure 5. Precision and Success plots on OTB100

Table 4. The comparison of some other trackers and our tracker on GOT-10k

	SiamDW [3]	DaSiamRPN [9]	SiamRPN [18]	SiamRPN++ [10]	SiamMask [20]	SiamCAR [2]	Ours
AO↑	0.416	0.444	0.483	*0.517*	0.453	**0.569**	0.552
$SR_{0.5}$↑	0.475	0.536	0.581	*0.616*	0.550	**0.670**	0.662
$SR_{0.75}$↑	0.144	0.220	0.270	*0.325*	0.248	**0.415**	0.371
FPS↑	66.67	**134.40**	97.55	3.18	15.37	17.21	*53.64*

4.2.2 Qualitative Comparison of Tracking Results

In this subsection, we select 8 challenging videos on the OTB100 dataset to visualize the tracking results of our proposed tracker as well as three other popular trackers, including DaSiamRPN [9], SiamBAN [7] and SiamRPN++ [10], As shown in Figure 6. Obviously, in these challenging videos, our tracker shows much better performance.

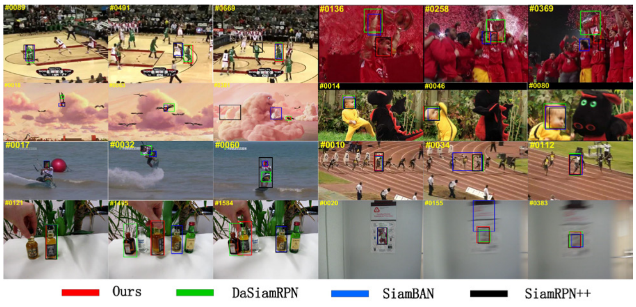
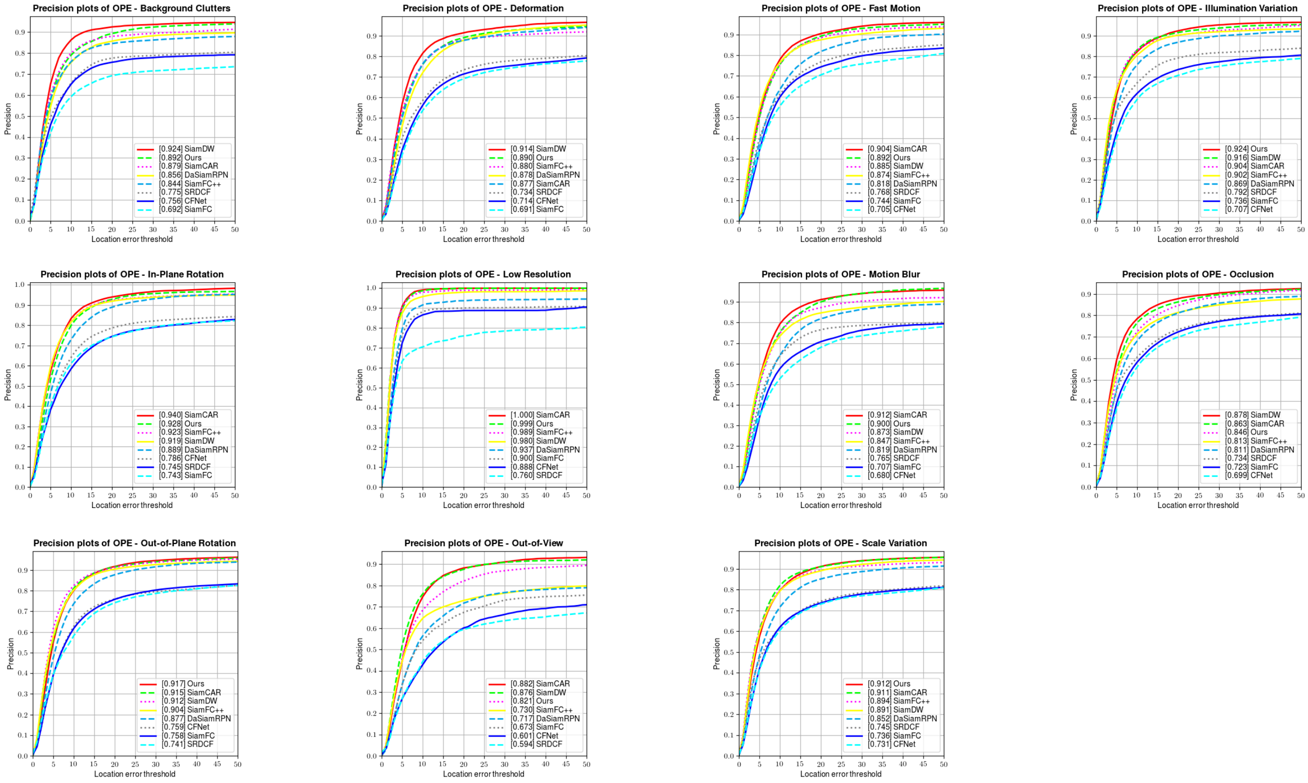


Figure 6. Comparison of tracking effectiveness of different videos on OTB100

4.2.3 Comparison on Different Challenges

In this subsection, we test our tracker on OTB100 dataset in the face of different tracking challenges. A total of 11 different challenge performances are tested on this benchmark, including low resolution, out-of-field of view, blurring, background clutter, illumination change, fast motion, deformation, occlusion, in-plane rotation, out-of-plane rotation, and scale change. Figure 7 and Figure 8 show the precisions and successes about our proposed tracker compared to other popular trackers on these 11 tracking challenges, respectively. We can see from the comparison that our proposed tracker ranks at the top of the list in terms of success rates and accuracies on all the tracking challenges, and achieves the best success rates and accuracies on several challenges. This shows that our tracker is robust enough and can cope well with a variety of different tracking challenges.


Precision plots of OPE - Background Clutters

Precision plots of OPE - Deformation

Precision plots of OPE - Fast Motion

Precision plots of OPE - Illumination Variation

Precision plots of OPE - In-Plane Rotation

Precision plots of OPE - Low Resolution

Precision plots of OPE - Motion Blur

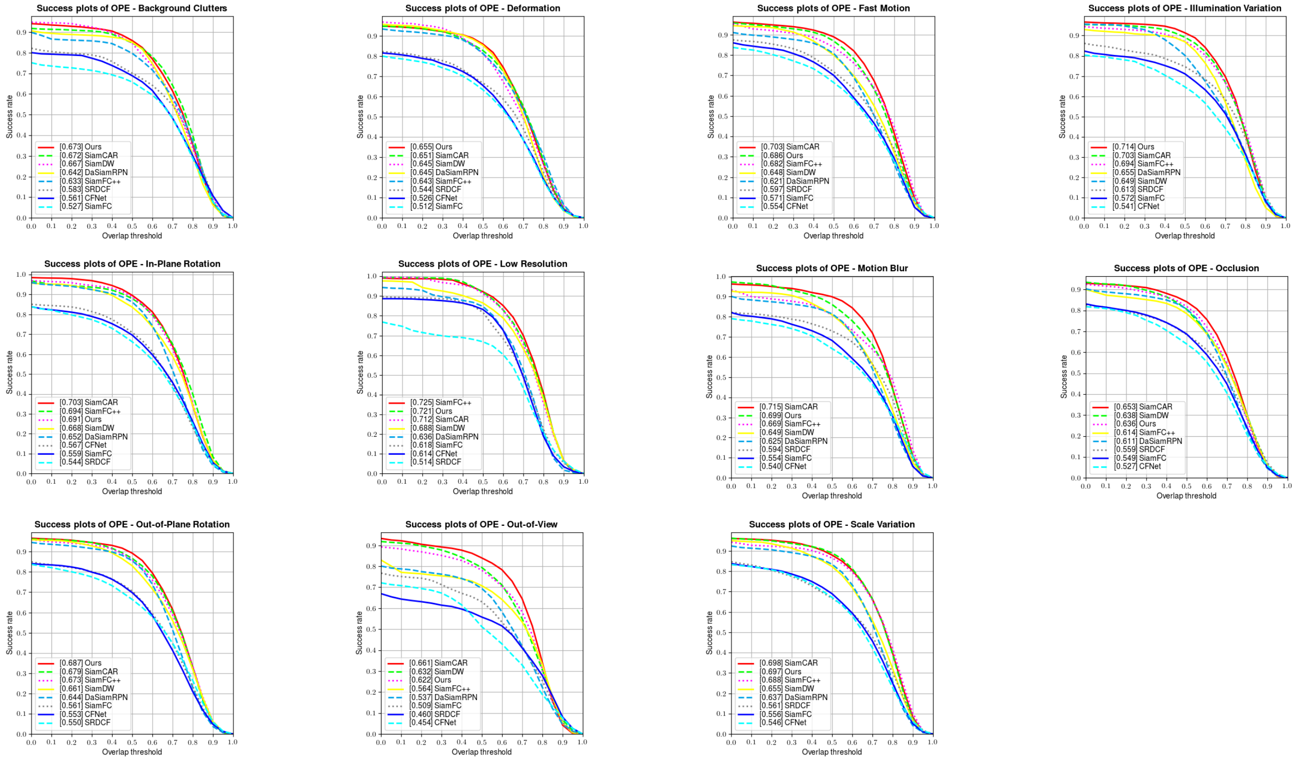
Precision plots of OPE - Occlusion

Precision plots of OPE - Out-of-Plane Rotation

Precision plots of OPE - Out-of-View

Precision plots of OPE - Scale Variation

Figure 7. Comparison of precision plots for 11 challenges on OTB100


Success plots of OPE - Background Clutters

Success plots of OPE - Deformation

Success plots of OPE - Fast Motion

Success plots of OPE - Illumination Variation

Success plots of OPE - In-Plane Rotation

Success plots of OPE - Low Resolution

Success plots of OPE - Motion Blur

Success plots of OPE - Occlusion

Success plots of OPE - Out-of-Plane Rotation

Success plots of OPE - Out-of-View

Success plots of OPE - Scale Variation

Figure 8. Comparison of success plots for 11 challenges on OTB100

Furthermore, we also compare our tracker with 8 popular trackers on VOT2018 dataset in the face of various tracking challenges, as shown in Figure 9. We can see that our tracker achieves great performance when facing different challenges, especially when facing the challenge of occlusion.

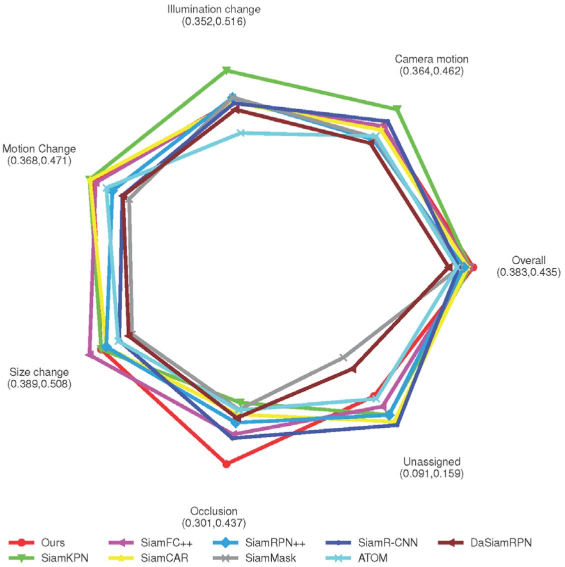


Figure 9. Comparison for different challenges on VOT2018

4.3 Ablation Study

In order to verify the validity of two proposed modules, we conducted ablation experiments on VOT2018 dataset of our proposed tracker, as shown in Table 5. To make this a fair comparison, all the trackers are trained on five datasets, including COCO [25], VID [26], DET [26], Youtube-BoudingBoxes [27] and GOT-10k [28]. Baseline denotes that there is no Spatial Attention Module or Residual Triplet Attention Module, which only includes the Siamese tracker of feature extraction network and the special head network behind it. SAM is our proposed Spatial Attention Module, and RTAM denotes our proposed Residual Triplet Attention Module. Baseline achieved an EAO of 0.366 without the addition of additional modules. With the addition of SAM and RTAM alone, it achieved a 1.0% and 1.7% improvement on the EAO, respectively, and improved the robustness of the model, which is sufficient to demonstrate the effectiveness of two proposed modules in improving the overall model performance. When we add both SAM and RTAM to the model, it achieves a 6.9% EAO improvement over Baseline, indicating that the SAM and RTAM modules together can deliver even greater performance gains. The only shortcoming is that the proposed modules do not produce effective improvement on the metric of accuracy, which may be related to the labeling of the rotated box in VOT2018, and it will be one of the directions we will work on in the next step.

Table 5. The ablation study on VOT2018

Method	E↑	A↑	R↓	Δ EAO
Baseline	0.366	0.590	0.262	
Baseline+SAM	0.376	**0.591**	0.229	+1.0%
Baseline+RTAM	0.383	0.580	0.225	+1.7%
Baseline+SAM+RTAM (Ours)	**0.435**	0.589	**0.169**	+6.9%

5 Conclusions

In this work, an improved Siamese tracker based on our residual triplet attention are proposed. By introducing SAM module, the visual features extracted by the backbone are enhanced, and the quality of the features is effectively improved. By proposing a residual triplet attention module suitable for the tracking task, it not only further improves the quality of visual features, but also solves the problem that spatial and channel attention are independent of each other, resulting in no information interaction between spatial dimensions and channel dimension from the perspective of dimension. Through experiments, we verify the effectiveness of our proposed modules. By comparing our proposed tracker with other popular trackers on five benchmarks, it is proved that our tracker has leading performance and good robustness to face different challenging tracking scenarios.

Of course, the tracker we proposed still has some shortcomings. The two improved modules we proposed belong to the category of self-attention. For the information interaction between the template and the search region, we still adopt the deep cross-correlation operation as same as common trackers, which has certain limitations. This may also be one of the reasons why our tracker does not significantly improve the accuracy of existing trackers. In the next step, we will make continuous improvements, and one of the directions is to consider introducing a cross-attention mechanism, which allows full information interaction between the template and the search region features, so as to design a tracker with better performance, higher accuracy, as well as more robustness.

Acknowledgements

This work was funded by the Postgraduate Scientific Research Innovation Fund of Changsha University of Science and Technology under Grant CXCLY2022117, and the Researchers Supporting Project Number (RSPD2024R681) King Saud University, Riyadh, Saudi Arabia. It was also supported by the Excellent Youth Project of the Science Research Project of Hunan Provincial Department of Education "Research on Key Technologies of Intelligent Emergency Platform Based on SiameseNets and Long Short-Term Memory Networks" (22B1028) and by 2023 Hunan Provincial Youth Key Teachers.

References

[1] J. M. Zhang, Y. F. He, W. T. Chen, L.-D. Kuang, B. Zheng, CorrFormer: Context-aware tracking with cross-correlation and transformer, *Computers and Electrical Engineering*, Vol. 114, Article No. 109075, March, 2024.

[2] D. Guo, J. Wang, Y. Cui, Z. Wang, S. Chen, SiamCAR: Siamese fully convolutional classification and regression for visual tracking, *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, Washington, USA, 2020, pp. 6269–6277.

[3] Z. Zhang, H. Peng, Deeper and wider Siamese networks for real-time visual tracking, *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Los Angeles, CA, USA, 2019, pp. 4591–4600.

[4] S. Javed, M. Danelljan, F. S. Khan, M. H. Khan, M. Felsberg, J. Matas, Visual object tracking with discriminative filters and Siamese networks: a survey and outlook, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 45, No. 5, pp. 6552–6574, May, 2023.

[5] S. Tang, M. Andriluka, B. Andres, B. Schiele, Multiple people tracking by lifted multicut and person re-identification, *The IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, Hawaii, USA, 2017, pp. 3539–3548.

[6] J. M. Zhang, J. Sun, J. Wang, Z. Li, X. Chen, An object tracking framework with recapture based on correlation filters and Siamese networks, *Computers & Electrical Engineering*, Vol. 98, Article No. 107730, March, 2022.

[7] Z. Chen, B. Zhong, G. Li, S. Zhang, R. Ji, Siamese box adaptive network for visual tracking, *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, Washington, USA, 2020, pp. 6668–6677.

[8] J. M. Zhang, X. Jin, J. Sun, J. Wang, A. K. Sangaiah, Spatial and semantic convolutional features for robust visual object tracking, *Multimedia Tools and Applications*, Vol. 79, No. 21-22, pp. 15095–15115, June, 2020.

[9] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, W. Hu, Distractor-aware Siamese networks for visual object tracking, *The European Conference on Computer Vision (ECCV)*, Munich, Germany, 2018, pp. 101–117.

[10] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, J. Yan, SiamRPN++: Evolution of Siamese visual tracking with very deep networks, *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Los Angeles, CA, USA, 2019, pp. 4282–4291.

[11] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, *The IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 7132–7141.

[12] S. Woo, J. Park, J. Y. Lee, I. S. Kweon, CBAM: Convolutional block attention module, *The European Conference on Computer Vision (ECCV)*, Munich, Germany, 2018, pp. 3–19.

[13] D. Misra, T. Nalamada, A. U. Arasanipalai, Q. Hou, Rotate to attend: Convolutional triplet attention module, *The IEEE/CVF Winter Conference on Applications of Computer Vision*, Waikoloa, HI, USA, 2021, pp. 3139–3148.

[14] J. M. Zhang, W. Feng, T. Yuan, J. Wang, A. K. Sangaiah, SCSTCF: spatial-channel selection and temporal regularized correlation filters for visual tracking, *Applied Soft Computing*, Vol. 118, Article No. 108485, March, 2022.

[15] M. Danelljan, G. Bhat, F. S. Khan, M. Felsberg, Eco: Efficient convolution operators for tracking, *The IEEE Conference on Computer Vision and Pattern Recognition*, Hawaii, HI, USA, 2017, pp. 6638–6646.

[16] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, P. H. Torr, Fully-convolutional Siamese networks for object tracking, *The 14th European Conference Computer Vision (ECCV) Workshops*, Amsterdam, The Netherlands, 2016, pp. 850–865.

[17] J. M. Zhang, H. Huang, X. Jin, L. D. Kuang, J. Zhang, Siamese visual tracking based on criss-cross attention and improved head network, *Multimedia Tools and Applications*, Vol. 83, No. 1, pp. 1589–1615, January, 2024.

[18] B. Li, J. Yan, W. Wu, Z. Zhu, X. Hu, High performance visual tracking with Siamese region proposal network, *The IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 8971–8980.

[19] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, *Advances in Neural Information Processing Systems* 28, Montreal, Quebec, Canada, 2015, pp. 91-99.

[20] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, P. H. Torr, Fast online object tracking and segmentation: A unifying approach, *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Los Angeles, CA, USA, 2019, pp. 1328–1338.

[21] Y. Xu, Z. Wang, Z. Li, Y. Yuan, G. Yu, SiamFC++: Towards robust and accurate visual tracking with target estimation guidelines, *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, New York, NY, USA, 2020, pp. 12549–12556.

[22] Y. Yu, Y. Xiong, W. Huang, M. R. Scott, Deformable Siamese attention networks for visual object tracking, *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, Washington, USA, 2020, pp. 6728–6737.

[23] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, *The IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016, pp. 770–778.

[24] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, S. Savarese, Generalized intersection over union: A metric and a loss for bounding box regression, *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Los Angeles, CA, USA, 2019, pp. 658–666.

[25] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, *The 13th European Conference Computer Vision (ECCV)*, Zurich, Switzerland, 2014, pp. 740–755.

[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, F. F. Li, ImageNet large scale visual recognition challenge, *International Journal of Computer Vision*, Vol. 115, No. 3, pp. 211-252, December, 2015.

[27] E. Real, J. Shlens, S. Mazzocchi, X. Pan, V. Vanhoucke, Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video, *The IEEE Conference on Computer Vision and Pattern Recognition*, Hawaii, HI, USA, 2017, pp. 5296–5305.

[28] L. Huang, X. Zhao, K. Huang, Got-10k: A large high-diversity benchmark for generic object tracking in the wild, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 43, No. 5, pp. 1562–1577, May, 2021.

[29] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Čehovin, T. Vojíř, G. Häger, A. Lukežič, G. Fernández, A. Gupta, A. Petrosino, A. Memarmoghadam, A. Garcia-Martin, A. S. Montero, A. Vedaldi, A. Robinson, A. J. Ma, A. Varfolomieiev, A. Alatan, A. Erdem, B. Ghanem, B. Liu, B. Han, B. Martinez, C.-M. Chang, C. Xu, C. Sun, D. Kim, D. Chen, D. Du, D. Mishra, D.-Y. Yeung, E. Gundogdu, E. Erdem, F. Khan, F. Porikli, F. Zhao, F. Bunyak, F. Battistone, G. Zhu, G. Roffo, G. R. K. S. Subrahmanyam, G. Bastos, G. Seetharaman, H. Medeiros, H. Li, H. Qi, H. Bischof, H. Possegger, H. Lu, H. Lee, H. Nam, H. J. Chang, I. Drummond, J. Valmadre, J.-C. Jeong, J.-I. Cho, J.-Y. Lee, J. Zhu, J. Feng, J. Gao, J. Y. Choi, J. Xiao, J.-W. Kim, J. Jeong, J. F. Henriques, J. Lang, J. Choi, J. M. Martinez, J. Xing, J. Gao, K. Palaniappan, K. Lebeda, K. Gao, K. Mikolajczyk, L. Qin, L. Wang, L. Wen, L. Bertinetto, M. K. Rapuru, M. Poostchi, M. Maresca, M. Danelljan, M. Mueller, M. Zhang, M. Arens, M. Valstar, M. Tang, M. Baek, M. H. Khan, N. Wang, N. Fan, N. Al-Shakarji, O. Miksik, O. Akin, P. Moallem, P. Senna, P. H. S. Torr, P. C. Yuen, Q. Huang, R. Martin-Nieto, R. Pelapur, R. Bowden, R. Laganière, R. Stolkin, R. Walsh, S. B. Krah, S. Li, S. Zhang, S. Yao, S. Hadfield, S. Melzi, S. Lyu, S. Li, S. Becker, S. Golodetz, S. Kakanuru, S. Choi, T. Hu, T. Mauthner, T. Zhang, T. Pridmore, V. Santopietro, W. Hu, W. Li, W. Hübner, X. Lan, X. Wang, X. Li, Y. Li, Y. Demiris, Y. Wang, Y. Qi, Z. Yuan, Z. Cai, Z. Xu, Z. He, Z, Chi, The visual object tracking VOT2016 challenge results, *The 14th European Conference Computer Vision (ECCV) Workshops*, Amsterdam, Netherlands, 2016, pp. 777–823.

[30] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Č. Zajc, T. Vojíř, G. Bhat, A. Lukežič, A. Eldesokey, G. Fernández, Á. García-Martín, Á. Iglesias-Arias, A. A. Alatan, A. González-García, A. Petrosino, A. Memarmoghadam, A. Vedaldi, A. Muhič, A. He, A. Smeulders, A. G. Perera, B. Li, B. Chen, C. Kim, C. Xu, C. Xiong, C. Tian, C. Luo, C. Sun, C. Hao, D. Kim, D. Mishra, D. Chen, D. Wang, D. Wee, E. Gavves, E. Gundogdu, E. Velasco-Salido, F. S. Khan, F. Yang, F. Zhao, F. Li, F. Battistone, G. De Ath, G. R. K. S. Subrahmanyam, G. Bastos, H. Ling, H. K. Galoogahi, H. Lee, H. Li, H. Zhao, H. Fan, H. Zhang, H. Possegger, H. Li, H. Lu, H. Zhi, H. Li, H. Lee, H. J. Chang, I. Drummond, J. Valmadre, J. S. Martin, J. Chahl, J. Y. Choi, J. Li, J. Wang, J. Qi, J. Sung, J. Johnander, J. Henriques, J. Choi, J. van de Weijer, J. R. Herranz, J. M. Martínez, J. Kittler, J. Zhuang, J. Gao, K. Grm, L. Zhang, L. Wang, L. Yang, L. Rout, L. Si, L. Bertinetto, L. Chu, M. Che, M. E. Maresca, M. Danelljan, M.-H. Yang, M. Abdelpakey, M. Shehata, M. Kang, N. Lee, N. Wang, O. Miksik, P. Moallem, P. Vicente-Moñivar, P. Senna, P. Li, P. Torr, P. M. Raju, Q. Ruihe, Q. Wang, Q. Zhou, Q. Guo, R. Martín-Nieto, R. K. Gorthi, R. Tao, R. Bowden, R. Everson, R. Wang, S. Yun, S. Choi, S. Vivas, S. Bai, S. Huang, S. Wu, S. Hadfield, S. Wang, S. Golodetz, T. Ming, T. Xu, T. Zhang, T. Fischer, V. Santopietro, V. Štruc, W. Wei, W. Zuo, W. Feng, W. Wu, W. Zou, W. Hu, W. Zhou, W. Zeng, X. Zhang, X. Wu, X.-J. Wu, X. Tian, Y. Li, Y. Lu, Y. W. Law, Y. Wu, Y. Demiris, Y. Yang, Y. Jiao, Y. Li, Y. Zhang, Y. Sun, Z. Zhang, Z. Zhu, Z.-H. Feng, Z. Wang, Z. He, The sixth visual object tracking vot2018 challenge results, *The European Conference on Computer Vision (ECCV) Workshops*, Munich, Germany, 2018, pp. 3–53.

[31] N. Wang, W. Zhou, Q. Tian, R. Hong, M. Wang, H. Li, Multi-cue correlation filters for robust visual tracking, *The IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 4844–4853.

[32] P. Voigtlaender, J. Luiten, P. H. Torr, B. Leibe, Siam r-cnn: Visual tracking by re-detection, *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, Washington, USA, 2020, pp. 6578–6588.

[33] M. Danelljan, G. Bhat, F. S. Khan, M. Felsberg, Atom: Accurate tracking by overlap maximization, *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Los Angeles, CA, USA, 2019, pp. 4660–4669.

[34] Q. Li, Z. Qin, W. Zhang, W. Zheng, *Siamese keypoint prediction network for visual object tracking*, arXiv preprint arXiv:2006.04078, June, 2020. https://arxiv.org/abs/2006.04078

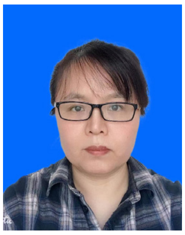
[35] M. Mueller, N. Smith, B. Ghanem, A benchmark and simulator for UAV tracking, *The 14th European Conference on Computer Vision (ECCV) workshops*, Amsterdam, The Netherlands, 2016, pp. 445–461.

[36] Y. Wu, J. Lim, M.-H. Yang, Object Tracking Benchmark, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 37, No. 9, pp. 1834–1848, September, 2015.

Biographies

Jianming Zhang received the B.S. degree from Zhejiang University, in 1996, the M.S. degree from the National University of Defense Technology, China, in 2001, and the Ph.D. degree from Hunan University, China, in 2010. He is currently a Full Professor and Ph.D. supervisor with the School of Computer and Communication Engineering, Changsha University of Science and Technology, China. He has published more than 120 research articles. His research interests include computer vision, pattern recognition, image processing and applications, data management and analysis, and mobile computing. He was listed in the World's Top 2% Scientists released by Stanford University for citation impact in single year from 2020 to 2022. He is a member of IEEE and a Distinguished Member of CCF.

Xiaoyi Huang received the B.S. degree in Changsha University of Science and Technology in 2021. He is currently pursuing the master's degree in electronic information engineering at Changsha University of Science and Technology, Changsha, China. His current research interests include Visual Object Tracking.



Huanqing Qiu was born in 1984 in Puyang, Henan, China. She obtained a master's degree from Hunan University in China. Now, she works at the School of Computer Science and Engineering, Hunan University of Information Technology, China. Her research interests include artificial intelligence, deep learning, and big data analysis.

Osama Alfarraj received the master's and Ph.D. degrees in information and communication technology (ICT) from Griffith University, in 2008 and 2013, respectively. He has served as a consultant and a member of Saudi National Team for Measuring E-Government, Saudi Arabia, for two years. He is currently a professor of computer sciences with King Saud University, Riyadh, Saudi Arabia. His current research interests include eSystems (eGov, eHealth, and ecommerce), cloud computing, and big data.

Amr Tolba received the M.Sc. and Ph.D. degrees from the Department of Mathematics and Computer Science, Faculty of Science, Menoufia University, Egypt, in 2002 and 2006, respectively. He is currently a full professor of computer science at King Saud University (KSU), Saudi Arabia. He has authored or coauthored over 180 scientific articles in top ranked (ISI) international journals and conference proceedings. His main research interests include artificial intelligence (AI), the Internet of Things (IoT), data science, and cloud computing.