Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Attention shake siamese network with auxiliary relocation branch for visual object tracking

Jun Wang[a], Weibin Liu[a,*], Weiwei Xing[b], Liqiang Wang[c], Shunli Zhang[b]

[a] *Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China*
[b] *School of Software Engineering, Beijing Jiaotong University, Beijing 100044, China*
[c] *Department of Computer Science, University of Central Florida, Orlando, FL 32816, USA*

A R T I C L E   I N F O

Article history:
Received 10 October 2019
Revised 30 January 2020
Accepted 29 February 2020
Available online 12 March 2020

Communicated by Dr. Shen Jianbing Shen

2010 MSC:
00-01
99-00

Keywords:
Visual object tracking
Siamese network
Attention shake layer
Auxiliary relocation branch
Switch function

A B S T R A C T

Siamese network is highly regarded in the visual object tracking filed because of its unique advantages of pairwise input and pairwise training. It can measure the similarity between two image patches, which coincides with the principle of the matching-based tracking algorithm. In this paper, a variant Siamese network based tracker is proposed to introduce attention module into traditional Siamese network, and relocate the object with some auxiliary relocation methods, when the proposed tracker runs under an untrusted state. Firstly, a novel attention shake layer is proposed to replace the max pooling layer in Siamese network. This layer could introduce and train two different kinds of attention modules at the same time, which means the proposed attention shake layer could also help to improve the expression power of Siamese network without increasing the depth of the network. Secondly, an auxiliary relocation branch is proposed to assist in object relocation and tracking. According to the prior assumptions of visual object tracking, some weights are involved in the auxiliary relocation branch, such as structure similarity weight, motion similarity weight, motion smoothness weight and object saliency weight. Thirdly, a novel response map based switch function is proposed to monitor the tracking process and control the effect of auxiliary relocation branch. Furthermore, in order to discuss the effect of pooling layer in Siamese network, 9 pooling and attention architectures are proposed and discussed in this paper. Some empirical results are shown in the experiment part. Comparing with the state-of-the-art trackers, the proposed tracker could achieve comparable performance in multiple benchmarks.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

Visual object tracking is an enduring and fundamental research direction, because of its strong application requirements, such as supervision, auto driving, etc. Besides it is also the foundation of some other artificial intelligence research, like video based crowd behavior analysis and anomaly detection, etc. According to the definition given by Smeulders et al. [1], visual object tracking is an online semi-supervised learning problem. The only training sample is from the state of object at the first frame. How to construct the appearance model of the tracking object accurately, how to online update the appearance model to adapt the changes of object in the video and how to monitor tracking process to relocate the object when failures occur are the key points to be solved in visual object tracking. Before deep learning based tracking methods, Correlation Filter (CF) based tracking methods [2–4] have attracted much attention of the researchers because of their rapidity and simplicity. However, the tracking performance of this kind tracking methods is limited by their expression power. Using the deep features from a well pre-trained network [5,6] may improve the representation ability to a certain extent. However, these deep feature extraction methods may also cause the representation inaccuracy of some objects, because the well pre-trained networks usually are not trained for tracking tasks, but for classification tasks. Using deep networks as classifiers in visual object tracking directly [7] may provide suitable appearance model. However, updating the parameters in deep networks is very time consuming, and that is also the reason why Multi-Domain Network (MDNet) based tracker[7] could not run in real time.

In addition, Siamese network, which is a pairwise input and pairwise training network, is also very popular in the visual object tracking field [8,9]. It could measure the similarity between two image patches without knowing the category labels of these two image patches. Normally, Siamese network is treated as a matching function and used to measure the similarity between templet images and instance patches obtained from the frames [9]. This kind

[*] Corresponding author.
E-mail address: wbliu@bjtu.edu.cn (W. Liu).

https://doi.org/10.1016/j.neucom.2020.02.120
0925-2312/© 2020 Elsevier B.V. All rights reserved.

of methods is simple and easy to understand, but considering the instance patches that need to be measured, these methods is very time consuming. Besides, it fails to consider the update of templets. In order to improve the tracking speed and break the limitation in traditional Siamese network that the two input patches must be of the same size, the idea of CF based trackers is introduced into Siamese network through fully-convolutional operations [10]. The output of exemplar image is used as a kernel in correlation filter. Thus, the tracking results can be obtained by finding the peak response score in response map. Due to the way of training and the backbone network, the expression power of the proposed tracker in [10] is very limited. In addition to replacing the ALexNet with some deeper network architectures, many variants [11–13] have been proposed to improve the expression power of Siamese network as well. Such as replacing the training loss with a triplet loss [12], combining Siamese network with Region Proposal Network (RPN) [13], etc. Considering the Siamese network based trackers and their variants, we find that most of the trackers focus on how to improve the expression power by a deeper backbone network. However, this may result in GPU consumption during the training process. It may also cause the over fitting, which means the pre-trained Siamese trackers are very dependent on the distribution of training samples and testing sequences. Besides, all the Siamese network based trackers fail to monitor the tracking process and do not have relocation algorithm, which means these trackers may provide some results with low response/confidence scores, and it is hard for them to relocate the objects when failure occurs. Furthermore, they do not consider the prior assumptions of visual object tracking. For example, normally, the motion and deformation of object in the tracking sequences are smooth, which means the location or state of the object between two adjacent frames does not change much. These prior knowledge sometimes helps refine tracking results and relocates objects.

Motivated by the above discussion, we try to design a Siamese network based tracker named as AS-Siamfc (Attention Shake Siamfc, AS-Siamfc), And it mainly focuses on the following three aspects: how to improve the expression power of AlexNet based Siamese network without increasing the depth or the layers of the network, how to introduce the prior knowledge of visual object tracking into the Siamese network based tracker to refine and improve the tracking results, and how to monitor the tracking process and detect the tracking failure. In order to solve the three problems above, firstly, a novel Attention Shake (AS) layer is proposed in this paper to replace the max pooling layer and improve the expression power of Siamese network. Different from the other attention methods, the proposed AS layer combines two different attention modules with a shake-shake framework [14]. The shake-shake framework replaces the standard summation of parallel branches in a multi-branch network with a stochastic affine combination [14]. This also helps to train the two different attention modules automatically and avoid over-fitting in the training process at the same time. Due to the AS layer, the proposed tracker pays more attention on real objects rather than the hard negative samples, which also means the AS layer helps to improve the expression power of Siamese network. Secondly, considering the prior assumptions of visual object tracking, an auxiliary relocation branch is proposed to refine the location of object when the proposed tracker runs under untrusted state. In the auxiliary relocation branch, there are some weights to meet the prior assumptions of visual object tracking, such as structure similarity weight, motion similarity weight, motion smoothness weight and object saliency weight. The additional prior knowledge could refine the tracking results and bring benefits, especially when the failure occurs or a tracker runs under untrusted state. Contrarily, it may also bring the noise and interferes with the tracking results. Hence, monitoring tracking process and detecting failure become very important. Thirdly, in order to monitor tracking process, we propose a response map based switch function. When the value of switch function is below a certain threshold, we believe the tracker runs under untrusted state, and the weights in auxiliary relocation branch will help this tracker to refine tracking results. Moreover, noticing that few works discuss the impact of pooling layer on Siamese network, we also propose 9 kinds of pooling and attention architectures and show some empirical results in the experiment part.

The main contributions of this paper can be summarized as follows:

- A novel AS layer is proposed to improve the expression power without increasing the depth or the layers of AlexNet based Siamese network. By combining two different attention modules with a shake-shake framework, The proposed AS layer could train these two different attention modules at the same time.
- An auxiliary relocation branch is proposed to refine the tracking results and introduce some prior knowledge of visual object tracking into Siamese network based trackers. This auxiliary relocation branch involves some weights, such as structure similarity weight, motion similarity weight, motion smoothness weight and object saliency weight, to meet the prior assumptions of visual object tracking and relocate the object when the tracker runs under an untrusted state.
- A switch function is proposed to monitor the tracking process and determine whether the weights in auxiliary relocation branch affect the tracking results. When the switch function score is over a certain threshold, we believe the tracker runs under a trusted state, and auxiliary relocation branch is not required to assist in tracking results. Otherwise, auxiliary relocation branch helps to refine the tracking results.
- The impact of pooling layer on Siamese network is discussed in this paper. 9 pooling and attention architectures are proposed in this paper and relative empirical results are shown in the experiment part.

The organization of this paper is as follows: Section 1 shows the motivations and contributions of the proposed tracker. Then, some related works are discussed in Section 2. Section 3 illustrates the detailed information and process of the proposed tracker. The analyses of the contributions in this paper and the experiments on widely used benchmarks are shown in Section 4. Finally, Section 5 concludes this paper.

2. Related works

Usually, according to the number of objects which need to be tracked, visual object tracking can be divided into multi-object tracking and single object tracking. For multi-object tracking [15–18], there are multiple objects to be tracked, however, these objects are usually known in advance, such as pedestrian tracking, vehicle tracking. Thus, multi-object tracking is always formulated as a data association problem. For single object tracking [1,19–23] the object to be tracked is single, but unknown in advance. Thus, single object tracking is formulated as an online learning problem. In this paper, we mainly focus on the single object tracking. However, some methods in single object tracking can be applied in the multi-object tracking as well. For instance, Shen et al. [17] try to apply minimum output sum of squared error filter which is widely used in single object tracking in multi-object tracking. For single object tacking, Survey [19] and [1] divided tracking methods into two categories and three components. The two categories are generative methods [24–26] and discriminative methods [2,27,28]. Generative tracking methods regard tracking as a templet matching problem and build the model of joint probability. While discriminative tracking methods treat tracking as a classification problem and build the model of posterior probability. For both of these two kinds of tracking methods, large numbers of manually designed features and traditional machine learning methods are used in visual object tracking, such as patch based methods [29–32], sparse representation based methods [22,33,34], Support Vector Machine (SVM) based methods [35,36] Correlation Filter (CF) based methods [2–4], etc. Huang et al. [32] represent object by a part space with two online learned probabilities to capture the structure of object. Ma et al. do a lot research on sparse representation based tracking methods [20–23]. Based on the globally linear approximation, a discriminative visual dictionary and a nonlinear classifier in sparse coding manner is proposed for tracking in [20]. Besides, Ma et al. [21] also presented a joint blur state estimation and multi-task reverse sparse learning framework. While, the three components are appearance model, motion model and update model respectively. Among these three components, appearance model, which is used to describe object and distinguish object from background, plays an important role in visual object tracking for both generative and discriminative methods.

Recently, along with the development of correlation filter and deep learning, large numbers of CF based methods [37–39], deep network based methods [7,13,40,41] and their hybrid methods [5,10,42,43] are proposed to construct the appearance model of object in visual object tracking. Normally, deep learning based trackers can be divided into two kinds: the deep feature based trackers [5,6] and the deep classification based trackers [7,44]. For deep feature based trackers, well pre-trained networks are applied to extract the features of objects and construct the appearance model, then the extracted deep features are introduced into the traditional tracking frameworks. For deep classification based tracker, the deep networks are used as classifiers. The hyperparameter optimization is also one of the drawbacks in deep network based trackers. Dong et al. [45] proposed a continuous Deep Q-Learning based action-prediction network for hyperparameter optimization. In order to evaluate the proposed trackers reasonably and enhance the comparability among different trackers, many benchmarks are applied in visual object tracking, such as OTB50 [46], OTB100 [47], TC128 [48], UAV123 [49], GOT-10K [50], Lasot [51], etc. The presentation of these data sets also provides a large and diverse training samples for deep network based trackers. Since the proposed tracker in this paper is based on the combination of CF based methods and deep network based methods, we mainly discuss the CF based trackers, deep network based trackers and the combination of these two kind of trackers in this section.

CF based trackers: David et al. [52] firstly introduce Correlation filters into the visual object tracking field. The correlation filter in [52] is trained by the state of object given at the first frame with a loss function to minimum the output sum of squared error. Thus the location of object has the largest correlation response score. In order to increase training samples and improve the robustness of correlation filter. Henriques et al. [37] propose a cyclic matrix to train the correlation filter. Instead of the sing-channel feature used in [37], they also propose a way to integrate multi-channel features into correlation filtering framework with a kernel method in [2]. This Kernelized Correlation Filter based tracker is also known as KCF. Martin et al. [39] apply two correlation filers to track the object. One is the translation filter which is used to obtain the location of object. The other is the scale filter which is proposed to estimate the scale of object and help the tracker to cope with the scale changing challenge. Chao et al. [38] find that the correlation between temporal context improves the accuracy and reliability for translation estimation, and train two different correlation filters from one frame. One is the filter of object, the other is the temporal context which is a correlation filter of surrounding context with spatial weights. Besides, an online re-detection module is proposed in [38] to monitor the tracking process in case of tracking failure. Generally, the CF based trackers are very effective and fast (more than 100 fps) which leave room for improvement, such as introducing some complex modules into the CF based trackers. Dong et al. [53] propose a two-stage classifier with kernelized circulant structure for occlusion-aware. Besides, a classifier pool is built to save classifiers with noisy updates and to redetect object when object is in occlusion.

Deep network based trackers: Along with the successful application of deep learning in other research fields, such as classification, object detection, image caption, etc., many deep network based tracking methods are proposed [7,13,40,44,54,55]. - Nam et al. [7] propose a tracker based on Convolutional Neural Network (CNN) which is trained for classification tasks. This network is composed of two parts: the shared layers and domain-specific layers. The shared layers contain the generic object representations. While, the domain-specific layers which are updated online show representations of individual sequences. Considering that online updating deep network is very time consuming, David et al. [44] apply a simple feed-forward network without online training and updating to improve the tracking speed to 100 fps. Instead of treating the deep network as a classifier, this network is trained to regress the state of bounding-box of object. In addition to the trackers mentioned above, another kind of widely used tracking methods is based on Siamese network [8]. Siamese network with pairwise inputs could measure the similarity of two image patches without knowing the category labels [56]. Therefore, Tao et al. [9] propose a tracker based on templet matching and violence search. In this tracker, Siamese network is applied to measure the similarity between templet and instance images. Li et al. [13] and Zhu et al. [57] try to combine the Siamese network with a Region Proposal Network (RPN) to make the tracking task an end-to-end learning process. In order to improve the expression power of Siamese RPN network, Li et al. [40] replace the backbone network of Siamese with a deeper network. While, Fan et al. [55] design the Siamese network by cascading the RPN networks. Furthermore, Wang et al.[54] try to add an image segmentation branch to the Siamese network to improve the tracking success rate of the proposed tracker.

Combination of CF based trackers and deep network based trackers: There are two ways to combine CF based trackers and deep network based trackers. One way is treating deep networks as feature extractors and applying the deep features directly to train correlation filters without training [5,6,42]. Ma et al. [5] analyse the impact of convolutional features from different layers on visual object tacking and train three correlation filters of different layers. While, Danelljan et al. [42] try to fuse the convolution features from different layers with an interpolation operator and apply this fusion feature to train correlation filter. According to the discussion above, these combinatorial methods are simple and direct. However, the expression power of the deep features may be reduced, since the well-trained deep network is trained for classification task rather than tracking task. The other way is Siamese network based combination methods [10–12,58,59]. These methods make full use of the pairwise inputs and pairwise training of Siamese network. They are usually separated into two parts: the training process and the tracking process. And the network parameters which are more suitable for tracking problem are trained by pairwise training methods. Bertinetto et al. [10] propose a tracker named as SiameseFC which is similar to CF based tracker by using the fully convolutional Siamese network. Dong et al. [12] try to replace the training loss with a triplet loss by considering the relationship between positive instances and negative instances. A novel design principle of Siamese network is proposed in [11] to replace the backbone network with a deeper and wider one. Thus, the expression power of Siamese network is also improved. In order to achieve the online learning of Siamese network based trackers, Guo et al. [58] propose a dynamic Siamese network with a fast general transformation learning model which enables online learning. Dong et al. [59] increase the two shared sub-branches in Siamese network to four sub-branches to take advantage of the underlying structure of data and relationship.

There are also some similar works [43,60,61], which try to introduce attention module into the Siamese network. He et al. [43] propose a twofold Siamese network for visual object tracking, this network is composed of two branches: the semantic branch and appearance branch. In the semantic branch, a channel attention based module is proposed to obtain the semantic information of object. While, Wang et al. [60] add three kinds of attention modules into the Siamese network: the residual attention, channel attention and general attention to improve the tracking performance. However, all these three attention modules are added in the exemplar branch of the Siamese network, and Siamese network is used as feature extractor. Shen et al. [61] obtain attention weights with proposed Attention Net(A-Net). Unlike traditional Siamese network, the feature maps of different layers in Siamese network are feeded into A-Net to calculate the attentive feature maps for cross correlation. Different from these Three trackers mentioned above, our proposed attention shake Siamese network could train two different kinds of attention modules at the same time, and the parameters of these two attention modules could be trained dynamically. This could help to obtain the benefits of these two attention modules and avoid over fitting problems. Furthermore, our proposed tracker can monitor the tracking process and introduce the prior knowledge of visual object tracking to refine the tracking results when the tracker runs under untrusted state. The specific details of the proposed tracker are presented in the following section.

3. Our approach

In this section, we present the architecture and details of our proposed Attention Shake based Siamfc (AS-Siamfc) tracker. Firstly, we present the architecture of the proposed tracker. Then, the following three subsections mainly focus on the details of proposed tracker respectively, such as attention shake layer, auxiliary relocation branch and switch function. Finally, we show the training and tracking procedures of AS-Siamfc.

3.1. Architecture of the proposed tracker

The Siamese network based trackers treat visual object tracking as a cross-correlation problem and compute the response map from Siamese network based deep model. They usually have two branches for the pairwise input. One branch is to learn the presentation of object z' in a semantic embedding space $\Phi()$, and the other branch shows the presentation of the search area x'. Thus the response map can be calculated by Eq. (1).

$$f(z', x') = \Phi(z') \circledast \Phi(x') + \mathbf{b} \tag{1}$$

where $\mathbf{b}$ is bias term and $\circ$*ledast* denotes the cross-correlation operation. The goal is to match the maximum value in response map to the object location.

In this subsection, we describe the architecture of the proposed tracker. As shown in Fig. 1, the architecture of AS-Siamfc can be mainly divided into three parts: attention shake network, relocation branch and switch function. Moreover, at the right side of Fig. 1, a weight based fusion method controlled by switch function is proposed to introduce the prior knowledge of visual object tracking into the proposed tracker and calculate the final response map, R_T. $\odot$ and $\circ$*ledast* in Fig. 1 denote element-wise product and cross-correlation operation respectively.

For the part of attention shake network Fig. 1(a), a novel attention shake layer is proposed to replace the max pooling layer in the AlexNet based Siamese network. This proposed attention layer could combine two different attention modules and improve the expression power. For the part of relocation branch Fig. 1(b), many weight maps, such as structure similarity weight, motion similarity weight, motion smoothness weight and object saliency weight, are applied to introduce some prior knowledge into AS-Siamfc. These types of prior knowledge could refine and relocate object when the proposed tracker runs under an untrusted state. For the switch function part Fig. 1(c), by observing the relationship between the score of response map of AS network and the success rate of AS-Siamfc, we design a switch function to monitor the tracking process online and control the effect of auxiliary relocation branch on tracking results.

The whole process can be summarized as follows: firstly, we feed the exemplar image, I_z (also templet obtained from the first image) and instance image, I_x (also candidate search image which is larger than exemplar image and represents the search area) into the proposed AS network and obtain the response map. Then, the switch function is used to monitor the tracking process according to the response map. If the tracker runs under a trusted state, the tracking results could be obtained according to the response maps directly. Otherwise, the response map of AS network is updated by the weight map of auxiliary relocation branch through element-wise product to refine and relocate the tracking results.

3.2. Attention shake in siamese network

The expression power of Siamese network directly affects the performance of tracking and attention modules are proved to be effective in classification tasks. Thus, we try to introduce attention modules into Siamese network to improve the expression power. In this section, a novel attention shake layer is proposed to replace the max pooling layer in Siamese network. Pooling layer in deep network helps to reduce the dimension of convolutional features, which is like a process of feature selection. Max pooling layer selects the maximum impact within an area (the max value). This could reduce the error of estimated mean which is caused by error of parameters in convolution layers and retain more useful information. While, average pooling layer considers the average effect of all elements in a certain area (the mean value). Thus, the average pooling layer pays more attention to the integrity of information, and helps to reduce the estimated variance caused by the constraints of neighborhood size. Considering the analyses above, the proposed AS layer can have the advantages of both max pooling layer and average pooling layer at the same time.

As shown in Fig. 2, the AS layer can be mainly divided into two parts: the attention part and the shake part. In the attention part, there are two modified Squeeze and Excitation block (SE block) [62] based attention modules: the max-attention module (left side of Fig. 2(a)) and the average-attention module (right side of Fig. 2(a)). Different from the traditional SE block, the modified SE blocks in this paper is applied to further refine the feature maps of max pooling and average pooling. The architectures of max-attention module and average-attention module can be found in Fig. 2(a). The max pooling and average pooling are used as the spatial attention in AS layer. After the max pooling layer (or average pooling layer), another global pooling layer is used to transfer the feature map of max pooling (or average pooling) from $((H-3)/2+1)*((H-3)/2+1)*C$ to $1*1*C$, where *H, W* and *C* denote the height, width and channel of convolution feature map, *X*. Then two fully convolutional layers are used to reduce and then increase the number of channels with a penalty coefficient, *r*, respectively. Finally, the channel attention weight can be calculated

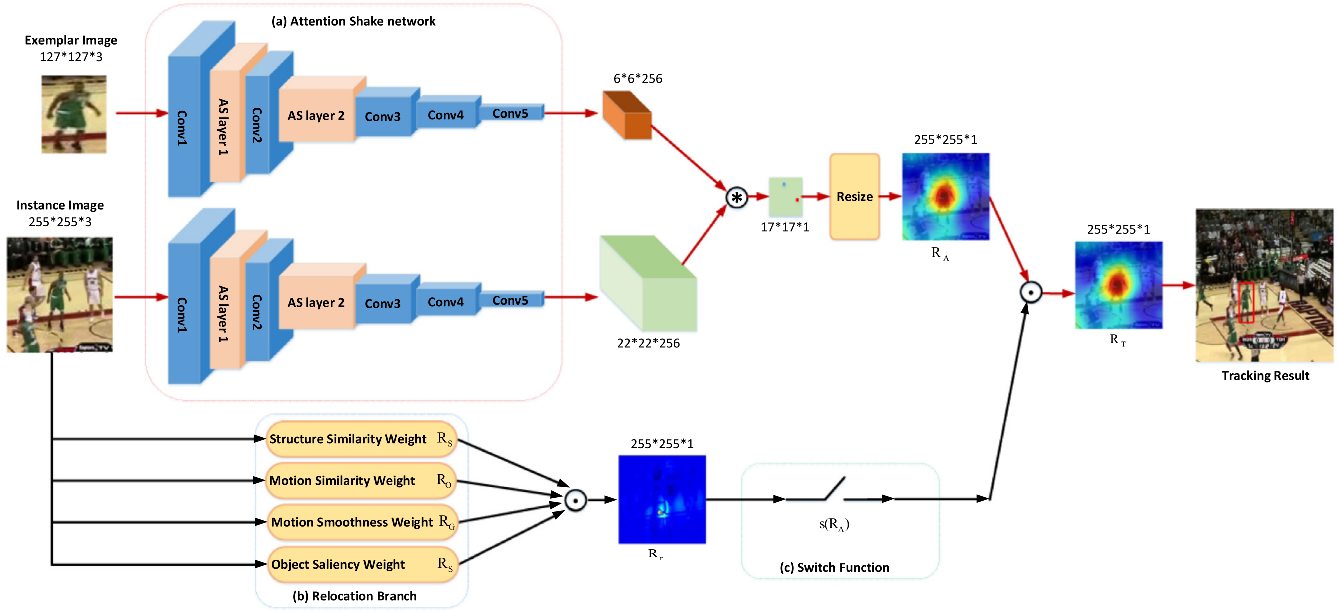


Fig. 1. The main framework of the proposed AS-Siamfc. It basically consists of 3 parts: (a) the attention shake network with proposed attention shake layer. (b) the auxiliary relocation branch which contains structure similarity weight, motion similarity weight, motion smoothness weight and object saliency weight. (c) the switch function which is used to control the four weights in relocation branch.-

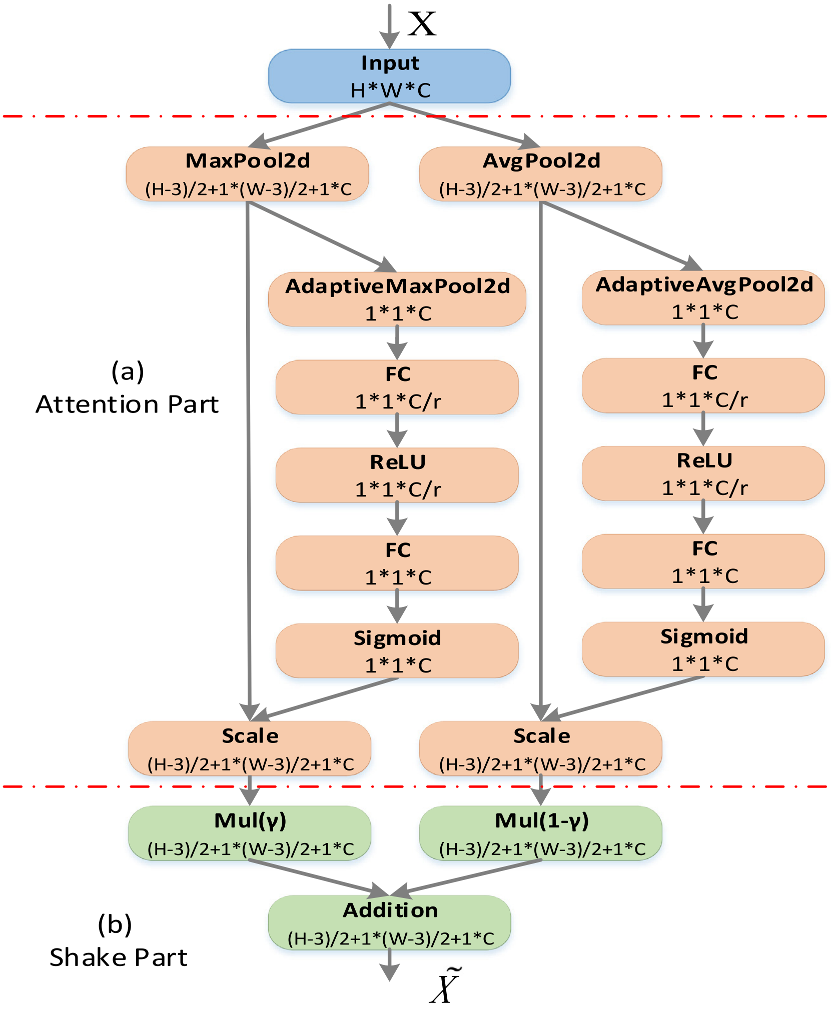


Fig. 2. The architecture of the proposed attention shake layer. It can be divided into two parts (a) the attention part and (b) the shake part.

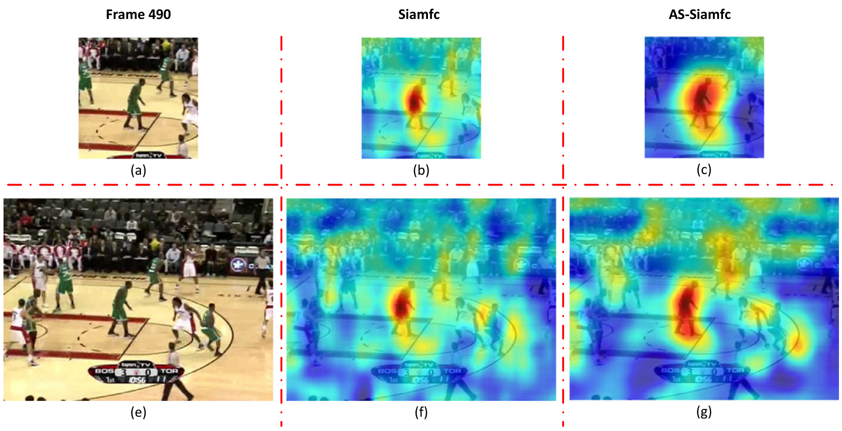


Fig. 3. The response maps of instance image and the whole image, using Siamfc and AS-Siamfc. (a) and (e) show the instance image of Frame 490 and the whole image respectively. (b) and (f) are the response map of Siamfc. (c) and (g) are the response map of AS-Siamfc.

by a sigmoid function. The feature map of max-attention, $AT_{\max}$, and average-attention, AT_{avg}, can be obtained by Eqs. (2) and (3).

$$AT_{max}(X) = sig(W_{m1}(W_{m0}(maxpool(X)))) \otimes maxpool(X) \tag{2}$$

$$AT_{avg}(X) = sig(W_{a1}(W_{a0}(avgpool(X)))) \otimes avgpool(X) \tag{3}$$

where W_{m0} and W_{a0} are the operations of the first fully connected layers of max-attention and average-attention, which try to reduce the channels from C to C/r. While, W_{m1} and W_{m2} are the operations of the second fully connected layers of max-attention and average-attention, which try to rise raise the channels from C/r back to C. $sig()$ denotes the sigmoid function, and $\otimes$ is the channel-wise product.

In order to make the proposed attention shake network contain the advantages of both max-attention and average-attention, the shake-shake model [14] is introduced into the AS layer. As shown in Fig. 2(b), the shake part in AS layer combines the feature map of max-attention and average-attention by a weighted sum. One benefit of the shake part is that the weight coefficient is dynamic in the train process, which could cause the attention part to adjust its parameters dynamically and prevent over fitting problems. The feature map of the AS layer can be computed by Eq. (4).

$$M_{AS}(X) = \hat{X} = \gamma * AT_{max}(X) + (1-\gamma) * AT_{avg}(X) \tag{4}$$

where $M_{AS}(X)$ denotes the feature map of AS layer, and γ denotes the weight coefficient. In the training process, γ varies according to uniform distribution from 0 to 1. In the tracking process, γ is set to be a fixed scalar, like 0.5. Since the proposed attention shake network is based on the Siamese network, the response map of attention shake network can be calculated by Eq. (5).

$$\begin{aligned} f(I_z, I_x) &= g(\varphi(I_z), \varphi(I_x)) \\ &= \varphi(I_z) \circledast \varphi(I_x) + \mathbf{b} \end{aligned} \tag{5}$$

where $\varphi()$ denotes the attention shake network, $\circ$*ledast* denotes the cross-correlation operation and **b** denotes a bias term. I_z and I_x are the exemplar image and instance image respectively.

Fig. 3 shows the response maps of Siamfc [10] and the proposed AS-Siamfc respectively. Fig. 3(b) and (c) are the response maps of instance image. In order to show the effect of attention shake layer persuasively, the response maps of the whole frame are shown in Fig. 3(f) and (g). By comparing the response maps between Siamfc and the proposed AS-Siamfc, we find that Siamfc only focus on the center part of object and cannot cover the whole object. While, AS-Siamfc could focus on the whole object. Furthermore, AS-Siamfc makes the object area of the response map redder and the background area bluer, which means the proposed method could increase the discrimination between object and background, and the attention shake layer could improve the expression power of Siamese network.

3.3. Auxiliary relocation branch

For tracking tasks, especially for some specific scenarios, there are some prior assumptions about the tracking object. For example, we always assume that the motion of tracking object is smooth, which means the state of object between two adjacent frames does not vary much. And people tend to choose some conspicuous objects as tracking objects. Moreover, the sequential relationship of object can also help to refine the tracking results. Thus, the purpose of the auxiliary relocation branch is to introduce some prior knowledge into AS-Siamfc, and relocate objects when the tracker runs under untrusted state. The auxiliary relocation branch along with the switch function (mentioned below) can be viewed as the failure detection and relocation part of AS-Siamfc. According to the prior assumptions mentioned above, some weight maps are introduced into the auxiliary relocation branch, such as structure similarity weight, motion similarity weight, motion smoothness weight and object saliency weight. The detail procedure is shown in Fig. 4.

From Fig. 4, we can see that the auxiliary relocation branch can be divided into four sub-branches: motion similarity weight sub-branch, motion smoothness weight sub-branch, object saliency weight sub-branch and structure similarity weight sub-branch.Firstly, we calculate the weight maps of these four sub-branches respectively. Then, in order to merge the weight maps obtained from the four sub-branches, we normalize the weight maps of these four sub-branches. Finally, the response maps are merged by an element-wise function. Thus, we obtain the weight map of auxiliary relocation branch. Noticing that the auxiliary relocation only works on the instance images.

For motion similarity weight sub-branch, we apply Lucas-Kanade method(LK) [63] to calculate the optical flow, $Op(i, i-1)$, between instance image i and instance image $i-1$, and we also compute the optical flow, $Op(i-1, i-2)$, between instance images $i-1$ and $i-2$. Then, according to the state of object at instance image $i-1$, we select an area of optical flow $Op(i-1, i-2)$, and the Histograms of Oriented Optical Flow feature (HOF) [64] of this area is viewed as the motion characteristic of object. Similarly, we also extract the HOF of the optical flow $Op(i, i-1)$. Thus, the motion similarity weight can be calculated by the cross-correlation between the HOF of the selected area and optical flow $Op(i, i-1)$,

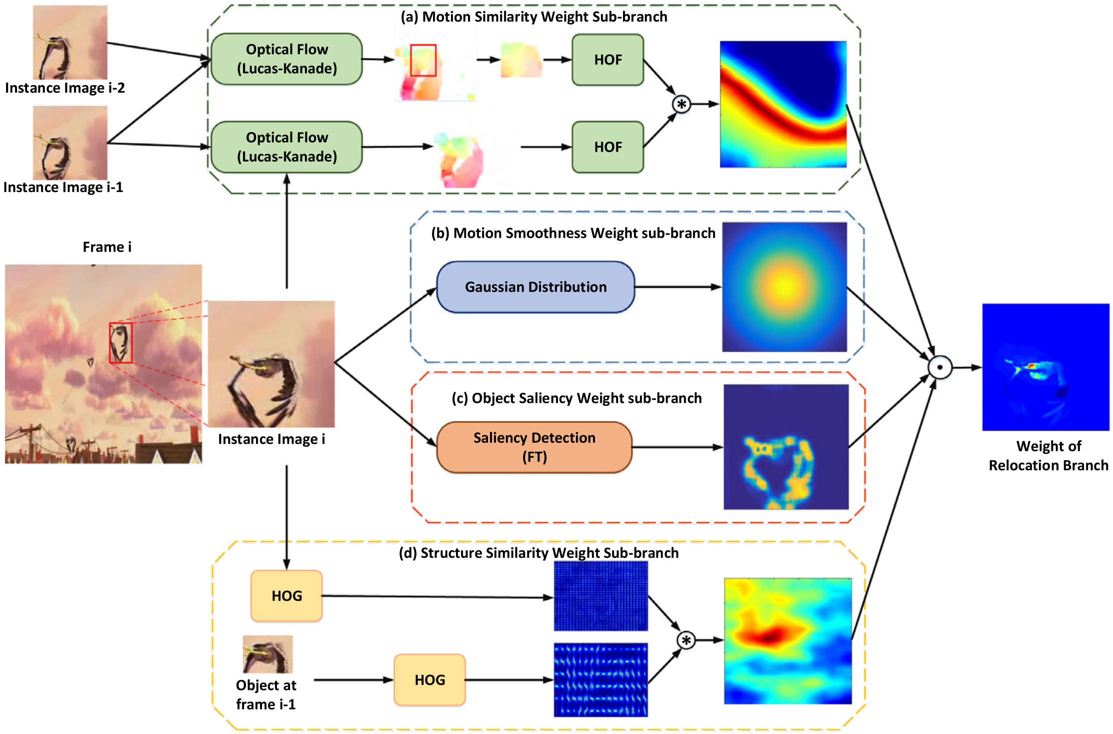

(a) Motion Similarity Weight Sub-branch
Instance Image i-2
Instance Image i-1
Optical Flow (Lucas-Kanade)
Optical Flow (Lucas-Kanade)
HOF
HOF
Frame i
Instance Image i
(b) Motion Smoothness Weight sub-branch
Gaussian Distribution
(c) Object Saliency Weight sub-branch
Saliency Detection (FT)
Weight of Relocation Branch
(d) Structure Similarity Weight Sub-branch
HOG
Object at frame i-1
HOG

Fig. 4. The procedure of the relocation branch. It can be divided into 4 sub-branches: (a) the motion similarity weight sub-branch, (b) the motion smoothness weight sub-branch, (c) the object saliency weight sub-branch and (d) the structure similarity weight sub-branch.

as shown in Eq. (6).

$$R_O = HOF(select(Op(i-1,i-2))) \circledast HOF(Op(i,i-1)) + \mathbf{b_o} \quad (6)$$

where R_O is the weight map of motion similarity weight sub-branch, and it is the motion similarity weight used in this paper. $select(Op(i-1,i-2))$ represents the optical flow of object calculated by instance image $i-1$ and instance image $i-2$. $HOF()$ means the extraction of HOF feature. $\mathbf{b_o}$ denotes the bias term of motion similarity weight sub-branch.

For object saliency weight sub-branch, the Frequency Tuned salient region detection method (FT) [65] is applied to compute the weight map of object saliency weight sub-branch, R_S, which could be obtained by Eq. (7).

$$R_S(x,y) = \|I_\mu - I_{whc}(x,y)\| \quad (7)$$

where I_μ is the average value of all pixels in the instance image, and I_{whc} denotes the smooth image of instance image after Gaussian filtering. Thus, $I_{whc}(x, y)$ is the corresponding score of I_{whc} at (x, y).

For motion smoothness weight sub-branch, the traditional two-dimensional Gaussian distribution function which is centered at (x_c, y_c) is used to construct the weight map of motion smoothness weight sub-branch. (x_c, y_c) can be obtained by the location of object center in the previous frame. The two-dimensional Gaussian distribution function is shown in Eq. (8).

$$R_G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-x_c)^2+(y-y_c)^2}{2\sigma^2}} \quad (8)$$

where σ is standard deviation of Gaussian distribution function, and R_G is the weight map of motion smoothness weight sub-branch.

For structure similarity weight, The Histograms of Oriented Gradients feature (HOG) [66] is applied to describe the structure information of object. Firstly, we extract the HOG feature of both the instance image i, $I_x(i)$ and the object image of instance image $i-1$, $I_o(i-1)$. The object image of instance image $i-1$ can be obtained by the state of object at frame $i-1$. Thus, the structure similarity weight can be calculated by the cross-correlation between the HOG feature of $I_o(i-1)$ and $I_x(i)$, as shown in Eq. (9)

$$R_{St} = HOG(I_o(i-1)) \circledast HOG(I_x(i)) + \mathbf{b_{St}} \quad (9)$$

where R_{St} is the weight map of structure similarity weight sub-branch, and it is the structure similarity weight used in this paper. $HOG()$ means the extraction of HOG feature, and $\mathbf{b_{St}}$ denotes the bias term of structure similarity weight sub-branch. Thus the weight map of auxiliary relocation branch can be calculated by Eq. (10) through element-wise product.

$$R_r = R_O \odot R_G \odot R_S \odot R_{St} \quad (10)$$

Fig. 5 shows the weight maps of object saliency weight, motion similarity weight, motion smoothness weight, structure similarity weight and the total weight map of auxiliary relocation branch respectively. From Fig. 5, we find that the object saliency weight in Fig. 5(b) focuses on detecting the entire object, especially when the difference between object and background is obvious. The motion similarity weight in Fig. 5(c) pays more attention to the areas which have similar movement of object. And this makes the response map of optical flow more suitable for tracking the moving rigid objects. While, the motion smoothness weight in Fig. 5(d) estimates the probabilities of the locations of objects in instance image. It is consistent with the assumption that the motion of objet is smooth. Similar to the motion similarity weight, the structure similarity weight in Fig. 5(e) pays more attention to the areas which have similar structure of object. This may help the tracker handle some tracking challenges, such as illumination variation, color variation, etc. More analyses about these three sub-branches can be found in Section 4.2.2. From Fig. 5(f), we find that the weight map of auxiliary relocation branch can not only outline the object, but also estimate the location of object center accurately. Thus, we believe the auxiliary relocation branch could refine and relocate the objects when the proposed tracker runs under untrusted state.

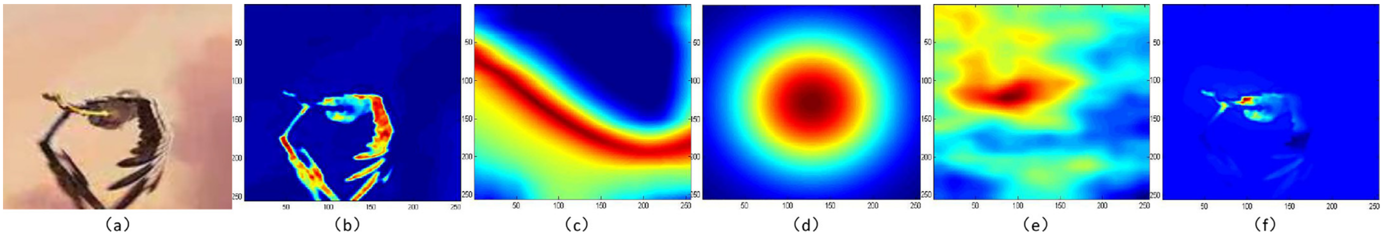


Fig. 5. The visualization of the four weight sub-branches in auxiliary relocation branch. (a) is the original image, (b) is the object saliency weight, (c) is the motion similarity weight, (d) is the motion smoothness weight, (e) is the structure similarity weight, and (f) is the weight of auxiliary relocation branch.

3.4. Switch function

Considering that introducing prior knowledge into AS-Siamfc may also bring noises, and sometimes, the prior knowledge itself is the noise which could impact tracking performance, we should monitor the tracking process of the proposed tracker, and ensure that the prior knowledge presented in auxiliary relocation branch affects the response map of AS Siamese network and further refine and relocate the objects only when the tracker runs under the untrusted state. Thus, in this section, a novel switch function which is based on the response map of AS Siamese network is proposed to monitor the tracking process and control the effect of auxiliary relocation branch on the tracking performance. The proposed switch function is shown in Eq. (11).

$$s(R_A) = \varepsilon\left(\frac{max(R_A) - avg(R_A)}{max(R_A) - min(R_A)} - s_t\right) \tag{11}$$

where R_A represents the response map of AS Siamese network. Thus, $max(R_A)$, $avg(R_A)$ and $min(R_A)$ are the maximum, average and minimum values of R_A. $\frac{max(R_A)-avg(R_A)}{max(R_A)-min(R_A)}$ is the confidence percentage, which is used to assess the reliability of tracking process. $\varepsilon()$ denotes the unit step function and s_t is the threshold. When confidence percentage is over s_t, the switch function score is 1. otherwise, the switch function score is 0. Thus, the final response map of AS-Siamfc can be calculated by Eq. (12), which is controlled by the switch function.

$$R_T = s(R_A)R_A + (1 - s(R_A))R_r \odot R_A \tag{12}$$

Where R_T is the final response map of AS-Siamfc. R_r and R_A are the weight map of auxiliary relocation branch and the response map of AS network respectively. From Eq. (12), we can see that when confidence percentage is under the threshold s_t, we believe that the tracker runs under an untrusted state, and the weight map of auxiliary relocation branch helps to refine and relocate the object. Thus, the final response map R_T is computed by an element-wise product, $R_T = R_r \odot R_A$. Otherwise, we think the tracker runs under a trusted state, and the final response map is equal to the response map of AS network, $R_T = R_A$. Another benefit of the proposed function is that we do not need to calculate the response map of auxiliary relocation branch in every frame. Instead, we only compute the response maps of auxiliary relocation branch, when the tracker runs under an untrusted state. This may also increase the tracking speed and reducing the amount of calculation.

Fig. 6 shows the relationship among response scores of R_A, scores of switch function, precision scores and success scores in two different sequences. The horizontal axes are the indexes of frames, and the vertical axes are the corresponding scores. Fig. 6(a) and (e) show plots of the maximum, minimum and average values of R_A of every frame. From these plots, we find that the average and minimum values of R_A in every frame do not change much. While, the maximum values vary with the frames. In both Fig. 6(a) and (e), we can see significant decline in the maximum values of R_A. Fig. 6(b) and (f) are the values of confidence percentage in switch function, which can be computed by $\frac{max(R_A)-avg(R_A)}{max(R_A)-min(R_A)}$. From these plots we can also see the significant decline which is consistent with the decline in Fig. 6(a) and (e). By comparing the plots, Fig. 6(a) and (b), Fig. 6(e) and (f), the confidence percentage represents the ratio of the distance between the peak value and mean value of R_A to the distance between the peak value and the valley value of R_A, and it can be applied to monitor tracking process and measure the confidence of the tracking results. When the confidence percentage is under a certain threshold, the maximum value of R_A is low, which means the tracker is hard to tell the object from the background and the tracker runs under an untrusted state. By comparing the precision and success scores of Siamfc and AS-Siamfc in Fig. 6(c) and (d), Fig. 6(g) and (h), we find that the precision and success scores of AS-Siamfc with switch function and relocation branch become smoother and higher, which also means the switch function and auxiliary relocation branch could monitor the tracking process, detect failure and relocate the object when the tracker runs under an untrusted state.

3.5. Training and tracking

Similar to Siamfc [10], the proposed AS-Siamfc can be divided into the offline training process and online tracking process. During the training process, we try to optimize the parameters of the proposed AS network by reducing the loss of the whole data set. While, in tracking process, the pre-trained AS network is used to calculate R_A and obtain the final response map R_T along with the weight map of auxiliary relocation branch R_r. The state of object can be estimated by searching the index of peak value in R_T.

For training process, we adopt the logistic loss as the loss function, and train the proposed AS network on positive and negative pairs. The obtaining of positive and negative pairs is similar to Siamfc [10]. The loss function of a single response map is shown in Eq. 13.

$$L(l_y, v_{x,z}) = \frac{1}{|D|}\sum_{u\in D} log(1 + e^{-l_y[u]v_{x,z}[u]}) \tag{13}$$

where l_y is the set which contains all the labels of a response map, and $v_{x,\,z}$ is the set which contains all the real values of a response map. Thus, $l_y[u]$ and $v_{x,\,z}[u]$ represent the u th label and real value of a response map. The loss function of a response map is the mean value of logistic losses of all elements in the response map. D is the set of index in a response map, and $|D|$ is the number of indexes in D. For each index u in a response map, the label $l_y[u]$ can be obtained by Eq. 14.

$$l_y[u] = \mathbb{I}(||u - c|| - r \leqslant 0) - \mathbb{I}(||u - c|| - r > 0) \tag{14}$$

where c is the center of object, and r is the radius. $\mathbb{I}(*)$ denotes the indicate function. When $*$ is true, $\mathbb{I}(*) = 1$, otherwise, $\mathbb{I}(*) = 0$. Thus, when the distance between u and c is longer than the radius r, the label $l_y[u] = -1$, otherwise, $l_y[u] = 1$. Furthermore, the parameter θ of AS network can be optimized by minimizing the mean value of all response maps in the data set with Stochastic

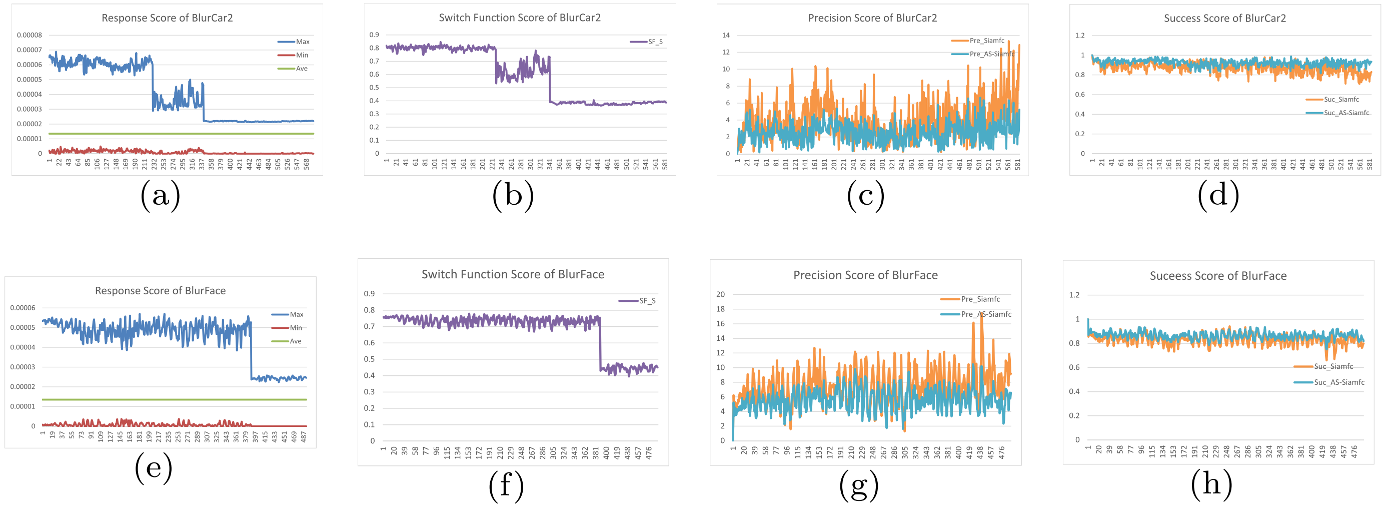


Fig. 6. Visualization of the response scores, scores of the switch function, precision scores and success scores on 2 sequences.(a) and (e) are the response scores of proposed attention shake network. (b) and (f) show the values of confidence percentage in switch function. (c) and (g) are the precision scores of Siamfc and AS-Siamfc. (d) and (h) are the success scores of Siamfc and AS-Siamfc.

Gradient Descent (SGD). The equation is shown in Eq. (15).

$$\theta = \arg\min_{\theta} \frac{\sum_{(I_z, I_x, l_y)\in Da} L(l_y, f(I_z, I_x; \theta))}{|Da|} \tag{15}$$

where Da and $|Da|$ represent the data set and the number of data set which is used to train the proposed AS network. (I_z, I_x, l_y) is a training sample in the data set. I_z, I_x and I_y are the exemplar image, instance image and label of training data respectively.

For tracking process, we try to estimate the state of object through the response map R_A which is obtained by the pre-trained AS network along with the weight map of auxiliary relocation branch R_r. The pseudo code is shown in Algorithm 1.

Algorithm 1 Pseudo code of AS-Siamfc.

```
Input: The exemplar image, $I_z$; The initial object state, $X_G$; The pre-trained AS network, Threshold, $s_t$; Number of frames, $N_f$;
Output: The states of object $X_t$, $t \in \{1, 2, ., N_f\}$;
Calculating the feature map of exemplar image by feeding $I_z$ into The AS network;
Initializing the object state, $X_1 = X_G$;
for $t = 2$; $t < N_f$; $t++$ do
    Calculating the instance image $I_x$, according to $X_{t-1}$;
    Feeding the instance image $I_x$ into AS network and computing the response map by Eq. 5;
    Resizing the response map to $(255 * 255 * 1)$ and obtaining $R_A$.
    Computing the score of switch function $S(R_A)$, by Eq. 11;
    if $S(R_A) > 0$ then
        Estimating the location of object through $R_A$;
        Updating the state of object $X_t$;
    else
        Calculating the response map of auxiliary relocation branch, $R_r$, by Eqs. 6, 7, 8, 9, 10;
        computing the final response map, $R_T$, by Eq. 12;
        Estimating the location of object through $R_T$;
        Updating the state of object $X_t$;
    end if
end for
return $X_t$, $t \in \{1, 2, ., N_f\}$;
```

The tracking process can be summarized as follows: firstly, we feed the exemplar image, I_z (also templet obtained from the first image) and instance image, I_x (also candidate search image which is larger than exemplar image and represents the search area) into the proposed AS network to compute the response map R_A by Eq. (5). Then, we calculate the score of the proposed switch function $S(R_A)$. If $S(R_A)$ is larger than 0, we believe that the tracker runs under a trusted state. Therefore, we could estimate the state of object only by the response map R_A. If not, the tracker runs under an untrusted state. The weight map of auxiliary relocation branch R_r which is obtained by Eqs. (6)–(10) helps to refine and relocate object with R_A. Thus, the final response map R_T is calculated by an element-wise product, Eq. (12). Finally, we estimate the state of object by the final response map R_T and update the instance image of the next frame by the state of object.

4. Experiments

This section provides some experimental results of the proposed AS-Siamfc tracker. Generally, there are three subsections in this section: The implementation subsection which describes the settings and parameters of the experiments; The basic experiments which discuss and analyze the effectiveness and availability of the proposed AS network and auxiliary relocation branch respectively; The Experiments on widely used benchmark which shows some quantitative and qualitative comparison experiments on some widely used benchmarks.

4.1. Implementation

In this subsection, we show some details about the settings and implementation of the proposed AS-Siamfc tracker. All the experiments run on a remote server with 64G memory and one GeForce GTX Titan X. The proposed AS network is trained on GOT-10K [50] benchmark which contains 10,000 video sequences and 1.5 million manually labeled boxes. Unlike some other training data sets, the tracking objects in this data set belong to more than 560 categories, which is helpful to improve the classification ability of AS-Siamfc. During the training process, the weight coefficient γ in attention shake layer varies randomly from 0 to 1. While, in the tracking process, γ is set to be 0.5. Moreover, the widely used benchmarks, OTB2013 [46], OTB100 [47], and OTB50, along with their evaluation criteria are applied in this paper to test the performance of the proposed AS-Siamfc tracker. OTB50 is composed of 50 hard-to-track sequences selected from the OTB100. Besides,

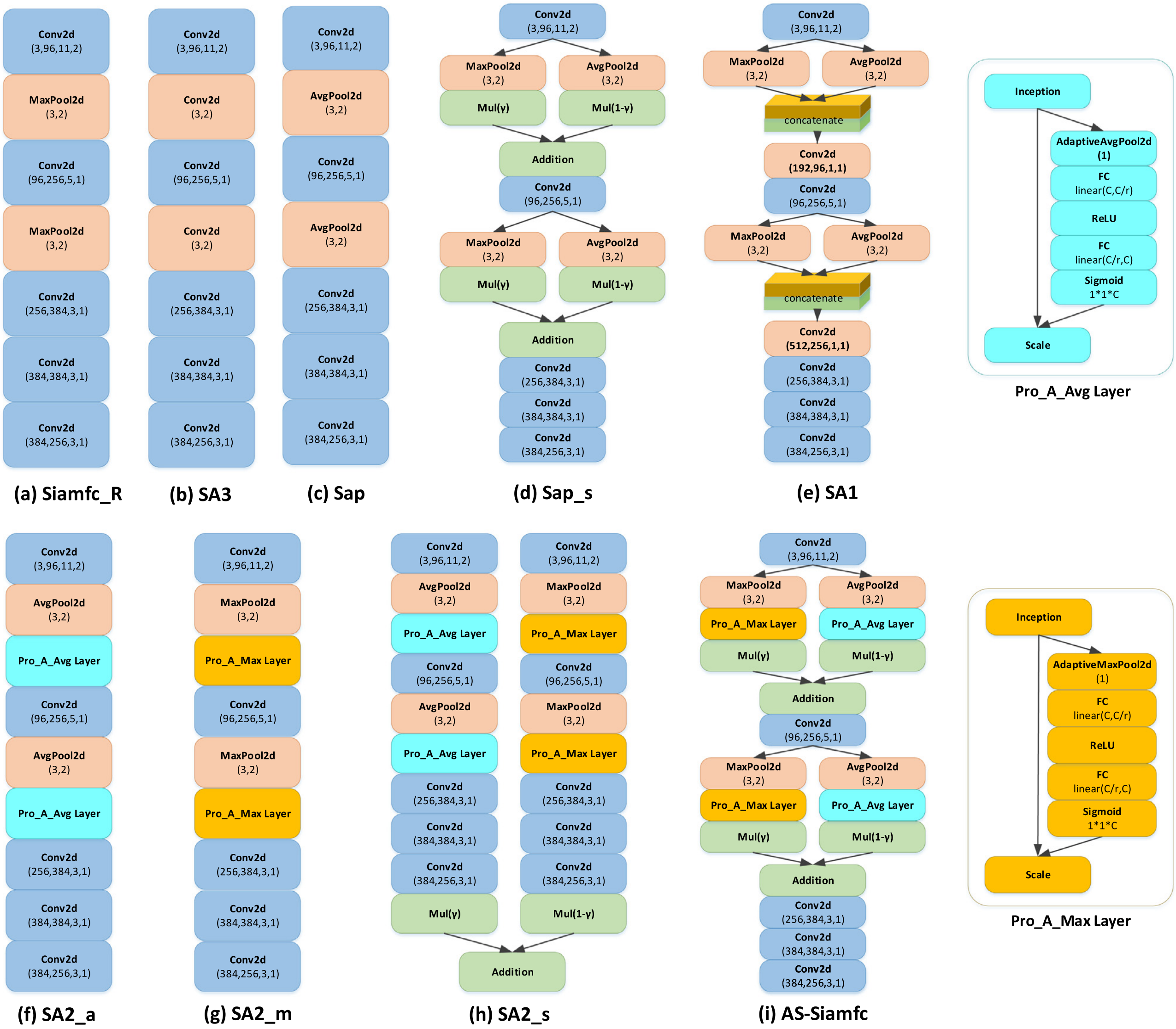


Fig. 7. 9 kinds of attention shake network architectures for comparison.

some state-of-the-art trackers are used for the comparison experiments, such as Siamfc [10], SAMF [67], DSST [39], Struck [27], TLD [68], CSK [37], ASLA [69], OAB [70] and IVT [71].

The proposed tracker could track objects in real time. The average tracking speed of AS-Siamfc on OTB100 data set is 70.625 fps, which is slightly slower than Siamfc [10], which is 84 fps. We believe the reasons are as follows: Firstly, the proposed tracker is based on the Siamfc tracker. Since the tracking speed of SimaFC is more than 80 fps, the proposed tracker is easy to track in real time. Secondly, computing the weight map of auxiliary relocation branch are very time consuming, especially the motion similarity weight. This makes the proposed tracker run slower than Siamfc tracker. However, we can reduce this time consumption by calculating the weight maps of instance images rather than the whole frame. Because of the proposed switch function, we only calculate the weight maps of auxiliary relocation branch when the proposed tracker runs under untrusted state. This also reduces the time consumption and increases the tracking speed of AS-Siamfc.

4.2. Basic experiments

In order to illustrate the feasibility and effectiveness of our proposed AS network, auxiliary relocation branch and switch function, some basic experiments and analyses are set and provided in this section. These basic experiments and analyses can be divided into two subsections: some analyses of the attention shake method and some analyses of the auxiliary relocation branch. For the subsection of attention shake method, we compare the proposed AS network with some other possible attention shake based networks. For the subsection of auxiliary relocation branch, we present the tracking comparisons between the trackers with and without auxiliary relocation branch.

4.2.1. Some analyses of the attention shake method

In this section, we show some results and analyses of different attention shake methods. In order to discuss the influence of pooling layer on Siamese network, the results of the Siamese networks with different pooling layers are also shown in this subsection. Firstly, we design 9 kinds of Siamese network architectures for comparison. These 9 kinds of designed networks contain different pooling layers and attention shake methods. Secondly, we present the precision and success plots of these 9 networks in OTB100.

As shown in Fig. 7, there are 9 kinds of Siamese network architectures which are designed for comparison. The Pro_A_Avg layer and Pro_A_Max layer at the right side of Fig. 7 are the proposed

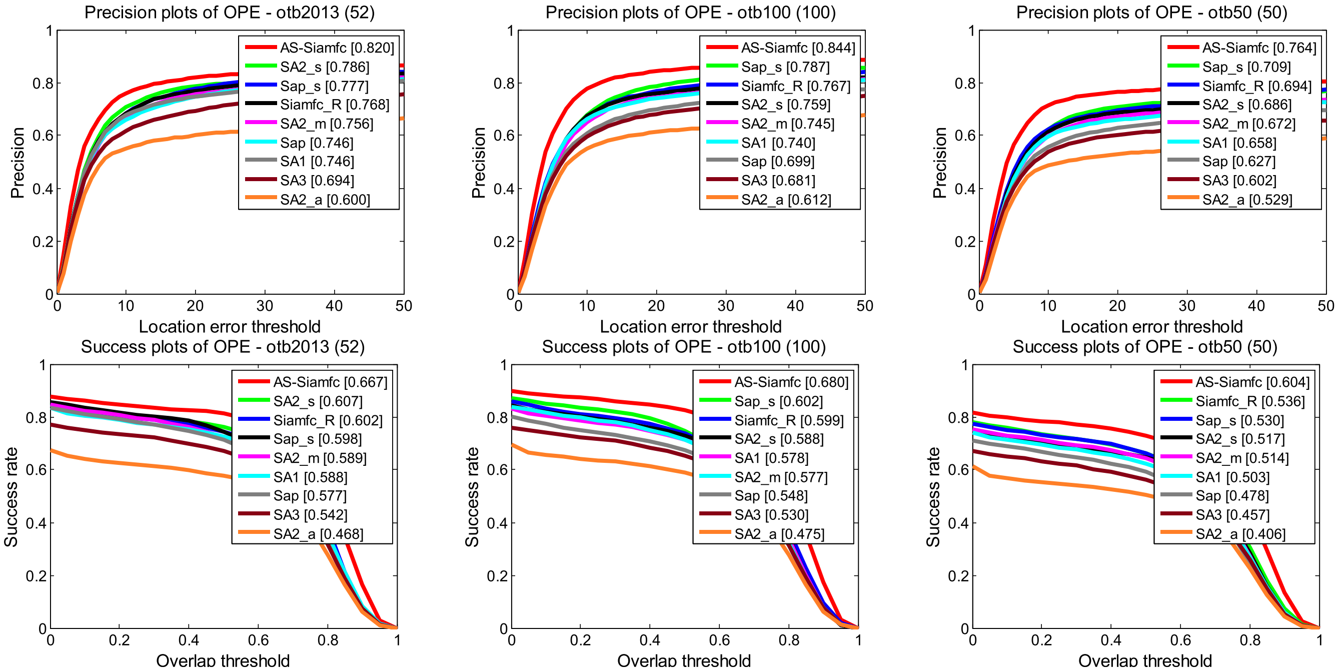


Fig. 8. The precision and success plots of OPE in OTB2013, OTB100 and OTB50.

max-attention module and average-attention module respectively. Fig. 7(a) is the backbone architecture of Siamfc network, and AS-Siamfc in Fig. 7(i) is the proposed AS-Siamfc network. The others are some possible variants of this network. In order to discuss the influences of pooling layers in Siamese network, we also construct two networks SA3 in Fig. 7(b) and Sap in Fig. 7(c) by replacing the max pooling layers in Fig. 7(a) with convolutional layers and average pooling layers respectively. Besides, we also combine the max pooling layers and average pooling layers with a shake module and construct the Sap_s network in Fig. 7(d). In order to analyse the proposed max-attention module and average-attention module Separately, we construct the SA2_a network which only contains average-attention module and SA2_m network which only contains max-attention module in Fig. 7(f) and (g) respectively. we also design a spacial attention model, SA1, by merging the feature maps of max pooling layer and average pooling layer in Fig. 7(e). Noticing that SA3, Sap_s, SA2_a and SA2_m can also be viewed as some modified attention methods, we can show the comparisons of different attention methods. SA2_s in Fig. 7(h)and AS-Siamfc in Fig. 7(i) show two possible network architectures of attention shake network.

Fig. 8 shows the precision and success plots of the 9 network architectures above in OTB2013, OTB100 and OTB50 data set. In order to ensure the comparability of the experiment and better reflect the influence of different networks on the tracking results, all the 9 network architectures are applied in the tracking framework with auxiliary relocation branch. Especially, Siamfc_R is the Siamfc tracker with auxiliary relocation branch, and it is also trained in GOT-10K data set. As shown in Fig. 8, we can see that comparing with the other 8 network architectures, AS-Siamfc shows the best performance of both precision and success plots in all the three data sets. Compared with Siamfc_R, the proposed AS-Siamfc has an average increase of 6.63% and 7.13% in terms of precision plots and success plots. Comparing SA2_s, Sap_s, AS-Siamfc with the other network architectures, we find that the network architectures with shake module are more likely to have good tracking performance. This also illustrates the effectiveness and rationality of attention shake module and the proposed AS-Siamfc tracker.

Table 1 shows the precision and success scores of 11 tracking challenges in OTB100 data set. In Table 1, each tracker is evaluated by the precision and success score, and these two sets of scores are divided into two rows in the Table. The precision scores are in the row above. While, the success scores are in the following row. IV, SV, OCC, DEF, MB, FM, IPR, OPR, OV, BC and LR in Table 1 represent illumination variation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, out-of-view, background clutters and low resolution respectively. By comparing the precision and success scores of Siamfc_R, SA3, Sap and Sap_s, we can get some empirical conclusions about the influence of pooling layers on Siamese network. It is that comparing with Sap and SA3, the Siamese network with max pooling layers can obtain relatively better results. However, the performance of the designed Sap_s which contains the shake module are better than Siamfc_R in the tracking challenge of out-of-plane, scale variation, deformation, motion blur, in-plane rotation, etc. This also presents the shake module could improve the expression power of Siamese network. Generally, the attention shake based backbone network architectures, SA2_s and AS-Siamfc can rank in the top three of the 9 network architectures in all tracking challenges. This illustrates the effectiveness of attention shake based network architectures. Comparing with the other network architectures, the proposed AS-Siamfc tracker shows a better performance in all tracking challenges.

In order to analysis the effectiveness of the shake part in attention shake layer, we also designed four different training and tracking methods for the proposed tracker. The first one is training with random weight coefficients of shake part, but tracking with a fixed weight coefficient. It is also known as AS-Siamfc_W which is applied in this paper. The second one is training with random weight coefficients of shake part, and tracking with random weight coefficients as well (AS-Siamfc_RR). In contrast, the third one is training with a fixed weight coefficient, but tracking with random weights (AS-Siamfc_SR), and the last one, AS-Siamfc_SS, is both training and tracking with a fixed weight coefficient. Fig. 9 shows the performance of these four training and tracking methods. To ensure the experiment is clear and fair, all these trackers run without

Table 1
The precision and success scores of 11 tracking challenges in OTB100. For each tracker, the precision scores are in the first row, and the success scores are in the following row.

		IV	SV	OCC	DEF	MB	FM	IPR	OPR	OV	BC	LR
SA1	Pre	0.713	0.720	0.630	0.702	0.675	0.698	0.738	0.721	0.561	0.681	0.708
	Suc	0.562	0.556	0.506	0.537	0.556	0.569	0.566	0.551	0.451	0.534	0.535
Sap	Pre	0.670	0.674	0.648	0.671	0.680	0.670	0.696	0.697	0.477	0.642	0.631
	Suc	0.523	0.525	0.516	0.524	0.556	0.545	0.537	0.537	0.374	0.503	0.460
Sap_s	Pre	0.750	0.773	0.714	0.770	0.743	0.742	0.761	0.783	0.617	0.727	0.671
	Suc	0.571	0.585	0.556	0.578	0.598	0.592	0.572	0.590	0.487	0.554	0.479
SA2_a	Pre	0.529	0.586	0.520	0.635	0.606	0.562	0.588	0.602	0.403	0.461	0.490
	Suc	0.417	0.453	0.409	0.482	0.496	0.455	0.457	0.459	0.309	0.362	0.354
SA2_m	Pre	0.675	0.747	0.672	0.723	0.687	0.730	0.750	0.753	0.610	0.618	0.697
	Suc	0.528	0.574	0.531	0.553	0.568	0.583	0.571	0.569	0.476	0.476	0.526
SA2_s	Pre	0.702	0.743	0.684	0.735	0.693	0.711	0.788	0.748	0.603	0.667	0.686
	Suc	0.541	0.569	0.543	0.561	0.564	0.566	0.572	0.571	0.464	0.518	0.498
SA3	Pre	0.601	0.633	0.658	0.663	0.608	0.643	0.684	0.698	0.453	0.645	0.587
	Suc	0.474	0.489	0.517	0.512	0.490	0.512	0.519	0.530	0.358	0.497	0.434
Siamfc_R	Pre	0.745	0.761	0.685	0.715	0.733	0.750	0.771	0.757	0.626	0.662	0.700
	Suc	0.584	0.590	0.542	0.545	0.603	0.602	0.594	0.577	0.500	0.514	0.537
Siamfc	Pre	0.746	0.751	0.667	0.694	0.721	0.732	0.752	0.733	0.621	0.669	0.751
	Suc	0.567	0.562	0.514	0.509	0.570	0.567	0.560	0.539	0.481	0.500	0.541
AS-Siamfc	Pre	0.791	0.837	0.784	0.824	0.810	0.812	0.835	0.833	0.708	0.753	0.705
	Suc	0.631	0.675	0.633	0.657	0.685	0.677	0.668	0.669	0.569	0.599	0.561
AS-Siamfc_W	Pre	0.760	0.806	0.739	0.797	0.786	0.774	0.800	0.798	0.664	0.726	0.693
	Suc	0.563	0.602	0.566	0.589	0.625	0.610	0.597	0.599	0.500	0.599	0.493

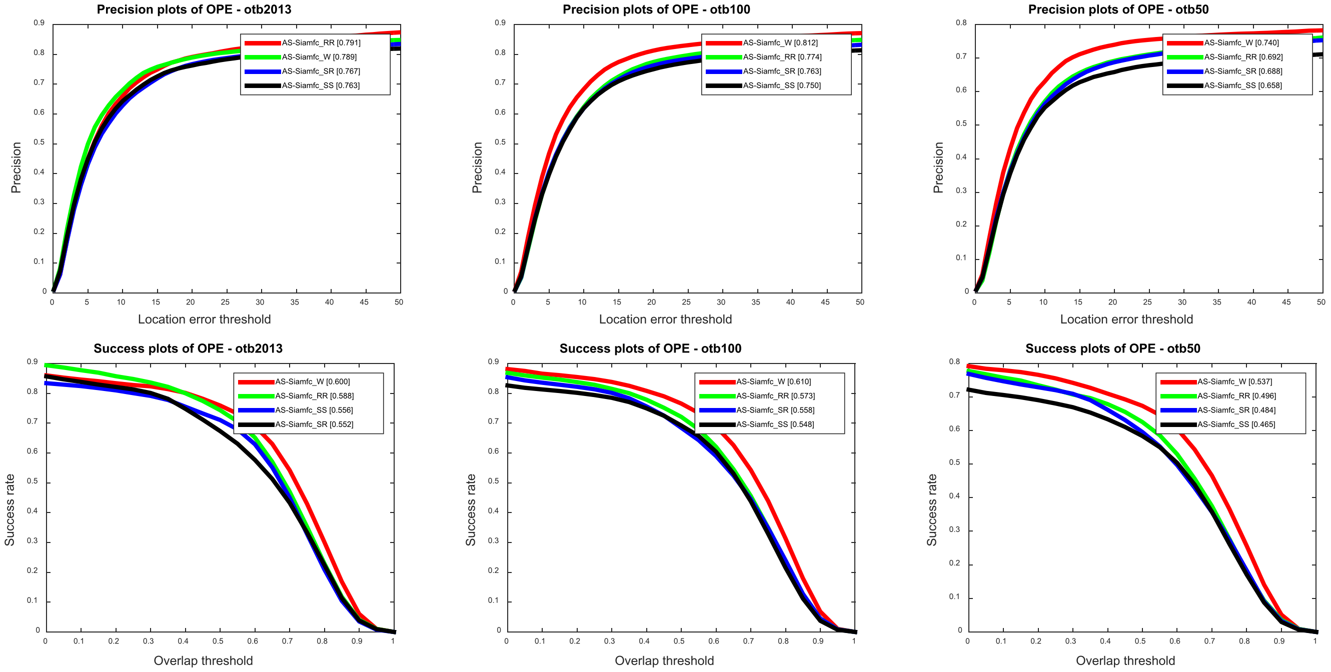


Fig. 9. The precision and success plots of OPE in OTB2013, OTB100 and OTB50.

auxiliary relocation branch. From Fig. 9, we can see that the precision and success scores of AS-Siamfc_W and AS-Siamfc_RR are higher than those of AS-Siamfc_SR and AS-Siamfc_SS, which means the trackers which train with random weight coefficients have better performance. Thus, the shake part in attention shake layer may help to improve the performance of AS-Siamfc. Though, AS-Siamfc_RR may also get high precision and success scores. Sometimes, it is even better than AS-Siamfc_W, such as the precision in OTB2013. However, AS-Siamfc_RR seems less robust and AS-Siamfc_W shows the better performance in most cases. Thus, the method that training with random weight coefficients and tracking with a fixed weight coefficient is applied in the proposed tracker.

4.2.2. Some analyses of the auxiliary relocation branch

In this section, we discuss the impacts of the prior assumptions of visual object tracking on Siamese network based trackers. Furthermore, we also present some analyses and results of the proposed auxiliary relocation branch and switch function. According to the prior assumptions that the movement of the object is smooth, the object is obvious in a certain range and the movement and structure of the object are consistent to a certain extent, we apply motion smoothness weight, object saliency weight, motion similarity weight and structure similarity weight to fit the prior assumptions above. All these methods are merged into the proposed auxiliary relocation branch.

Fig. 10 shows the effects of saliency, optical flow, HOG feature and the final response map of AS-Siamfc qualitatively. The first row of each sub-figure represents the original image. The second row of each sub-figure is the object saliency. The third and fourth rows of each sub-figure represent the optical flow and HOG feature of object respectively. While, the last row shows the final response map of AS-Siamfc. From Fig. 10, we can see that each method in

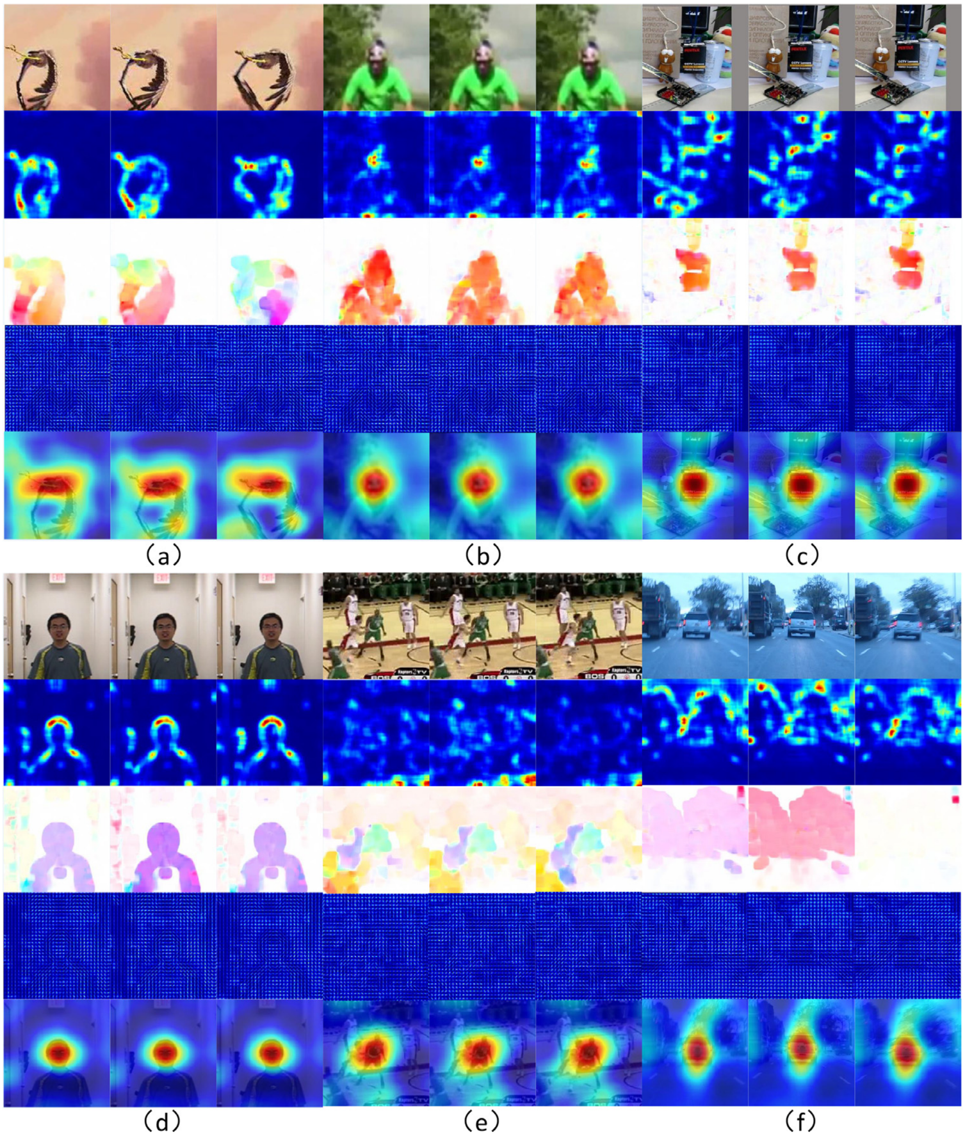


Fig. 10. The saliency, optical flow, HOG feature and response map of proposed tracker for 6 sequences in OTB50 and OTB100.

auxiliary relocation branch has its own applicable video sequences and scenarios. As shown in Fig. 10(a), both saliency detection and optical flow could outline the object in some simple sequences or the difference between object and background is large. Since the helmet in Fig. 10(b) is more prominent than the background, the saliency detection method is more suitable to locate the object than optical flow. On the contrary, Fig. 10(c) shows that optical flow is good at dealing with background clutter sequences which saliency detection method cannot handle. Fig. 10(d) and (f) show the sequences which are not applicable to saliency detection method and optical flow. However, the structure (the HOG feature) of objects in these two sequences does not change much. Fig. 10(e) shows a complex sequence. In this sequence, the motion of object is not regular, the object is not very salient and there are many other people who have the similar structure with the object in this sequence. Thus, in this sequence, the saliency detection method, optical flow and HOG feature is hard to estimate the location of the object accurately. However, with the help of motion smoothness weight we can still estimate the state of object.

In order to further explore the impact of the proposed auxiliary relocation branch and switch function, Fig. 11 shows the precision and success plots of trackers with and without auxiliary relocation branch and switch function. Siamfc_R and Siamfc represent the Siamfc trackers with and without auxiliary relocation branch respectively. While, AS-Siamfc and AS-Siamfc_W are the proposed trackers with and without auxiliary relocation branch respectively. All these trackers are tested in OTB2013, OTB100 and OTB50 data set. From Fig. 11, we find that the propose AS-Siamfc tracker (with auxiliary relocation branch) achieves the best tracking performance among all these four trackers. By comparing the precision and success plots of AS-Siamfc and AS-Siamfc_W, we find that the precision score and success score of AS-Siamfc are increased by 2.87% and 6.80% on average. Comparing the precision and success plots of Siamfc_R and Siamfc, we can also find the slight improvements. These experiments validate the effectiveness and availability of the proposed auxiliary relocation branch.

Table 1 also provides the precision and success scores of AS-Siamfc, AS-Siamfc_W, Siamfc and Siamfc_R under 11 tracking challenges in OTB100. By comparing the precision and success scores of these four trackers, we can see that in most cases, the performances of the trackers with auxiliary relocation branch are better than those trackers without auxiliary relocation branch. Thus, we can say that the proposed auxiliary relocation branch along with the switch function could monitor the tracking process, refine and

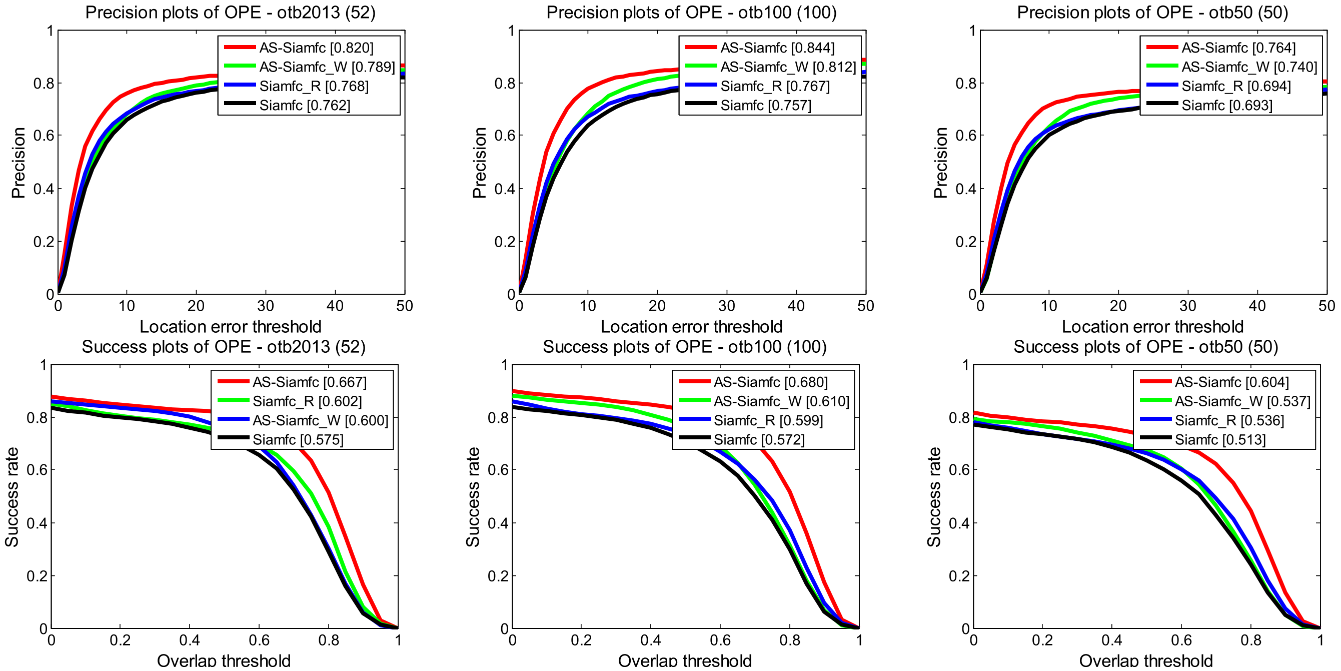


Fig. 11. The precision and success plots of OPE in OTB2013, OTB100 and OTB50.

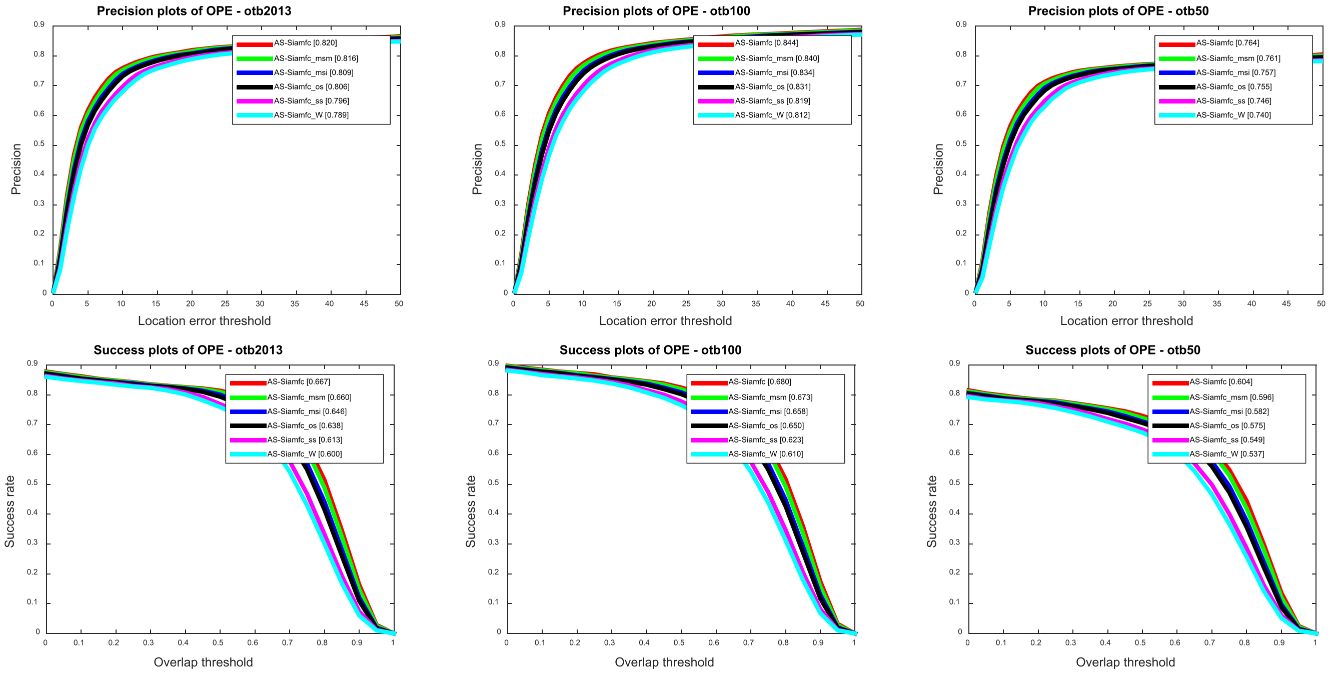


Fig. 12. The precision and success plots of OPE in OTB2013, OTB100 and OTB50.

relocate objects when the tracker runs under untrusted state and improve tracking performance.

In order to discuss the effectiveness of the four sub-branches in auxiliary relocation branch specifically, the performances of AS-Siamfc without motion smoothness weight (AS-Siamfc_msm), AS-Siamfc without motion similarity (AS-Siamfc_msi), AS-Siamfc without object saliency weight (AS-Siamfc_os) and AS-Siamfc without structure similarity (AS-Siamfc_ss) are shown in the Fig. 12 along with the performance without the whole auxiliary relocation branch (AS-Siamfc_W) and the proposed tracker AS-Siamfc. From all the precision and success plots in Fig. 12, we can see that AS-Siamfc_ss shows the worst performance except AS-Siamfc_W, and AS-Siamfc_msm gets the highest precision and success score except AS-Siamfc, which indicates that the structure similarity weight plays a relatively important role in the auxiliary relocation branch.

Furthermore, we also show the effectiveness of the threshold of switch function in OTB100 data set in Fig. 13. The first two figures in Fig. 13 are the precision and success plots of OPE respectively. The numbers in the brackets in legend indicate the corresponding thresholds applied in AS-Siamfc. Notice that the thresholds of AS-Siamfc and AS-Siamfc_W are 0.6 and 0. The third figure is

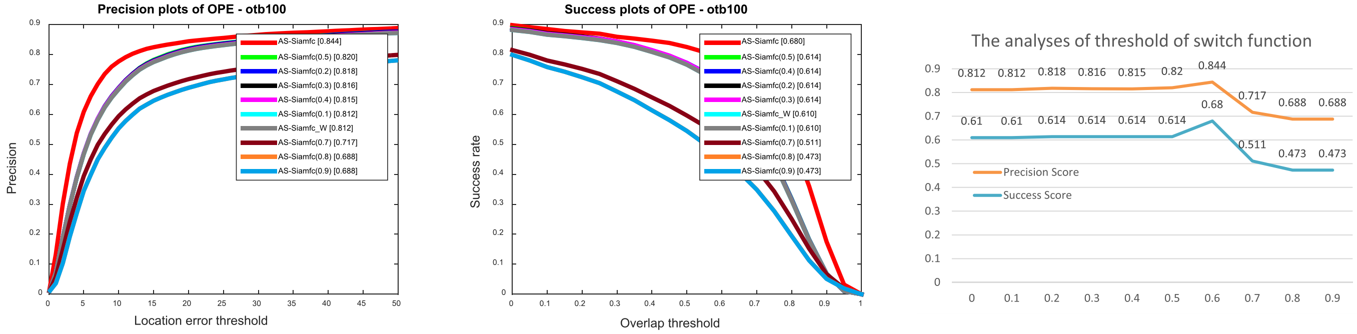


Fig. 13. The analyses of the threshold of switch function in OTB100.

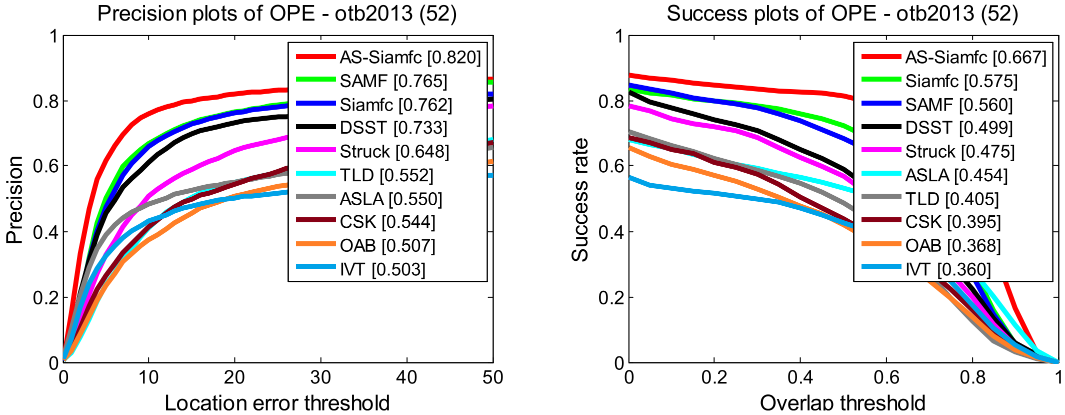


Fig. 14. The precision and success plots of OPE in OTB2013.

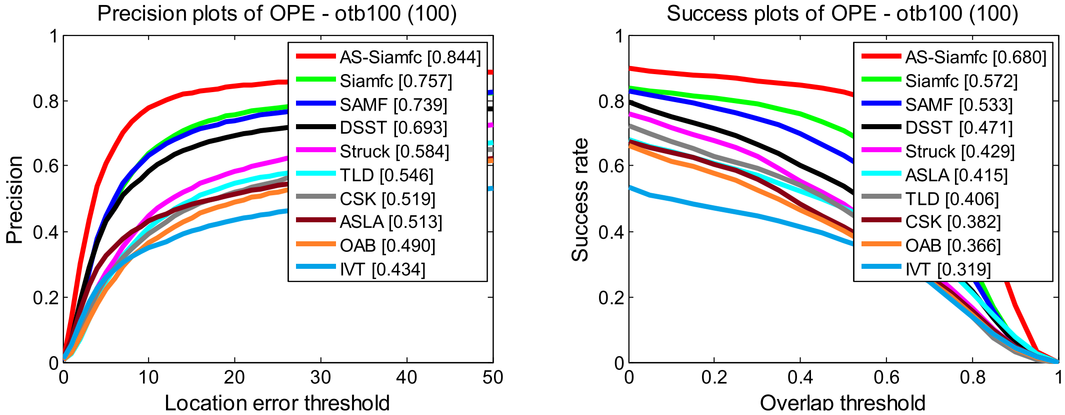


Fig. 15. The precision and success plots of OPE in OTB100.

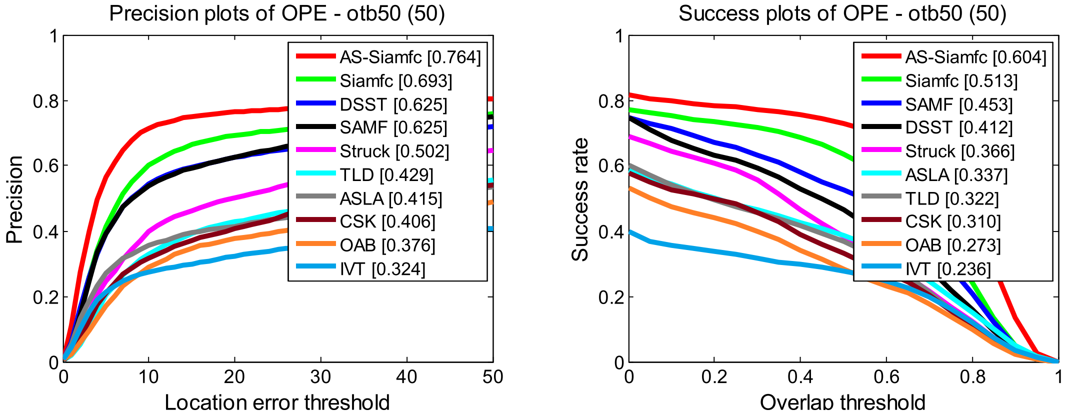


Fig. 16. The precision and success plots of OPE in OTB50.

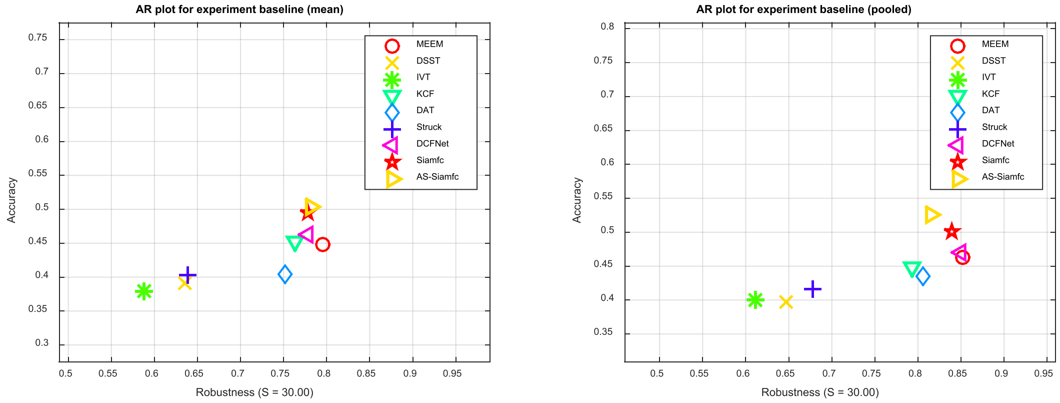


Fig. 17. The AR plot for experiment baseline of VOT2018.

the trend map of precision and success score. The abscissa of this figure is thresholds, while, the ordinate is the corresponding precision and success scores. From Fig. 13, we can see that when the threshold is less than 0.5, the plots of precision and success score rise steadily, and reach the peaks at 0.6. Then, the plots begin to plunge. The precision and scores may even lower than AS-Siamfc_W, when the threshold is over 0.7. We believe the reason is when the threshold is over 0.7, the performance of AS-Siamfc is more dependent on auxiliary relocation branch than on AS network. Thus, the auxiliary relocation branch may bring some noise and wrong knowledge, which leads to the error of AS-Siamfc and reduces the tracking performance.

4.3. Experiments on widely used benchmark

In order to compare the proposed AS-Siamfc tracker with some state-of-the-art trackers, we show some quantitative and qualitative experiments on the widely used benchmarks, OTB2013 [46], OTB100 [47], OTB50 and VOT2018 [72] in this section. Firstly, we show the precision and success plots of One-Pass Evaluation (OPE) in benchmark OTB2013, OTB100 and OTB50, along with the Accuracy-Robustness plots (AR plots) in VOT2018. The performances of the proposed tracker under 10 tracking challenges are also analysed in this section. Secondly, we provide the tracking bounding boxes of 10 sequences to show the qualitative analyses of the proposed AS-Siamfc tracker.

4.3.1. Quantitative analyses

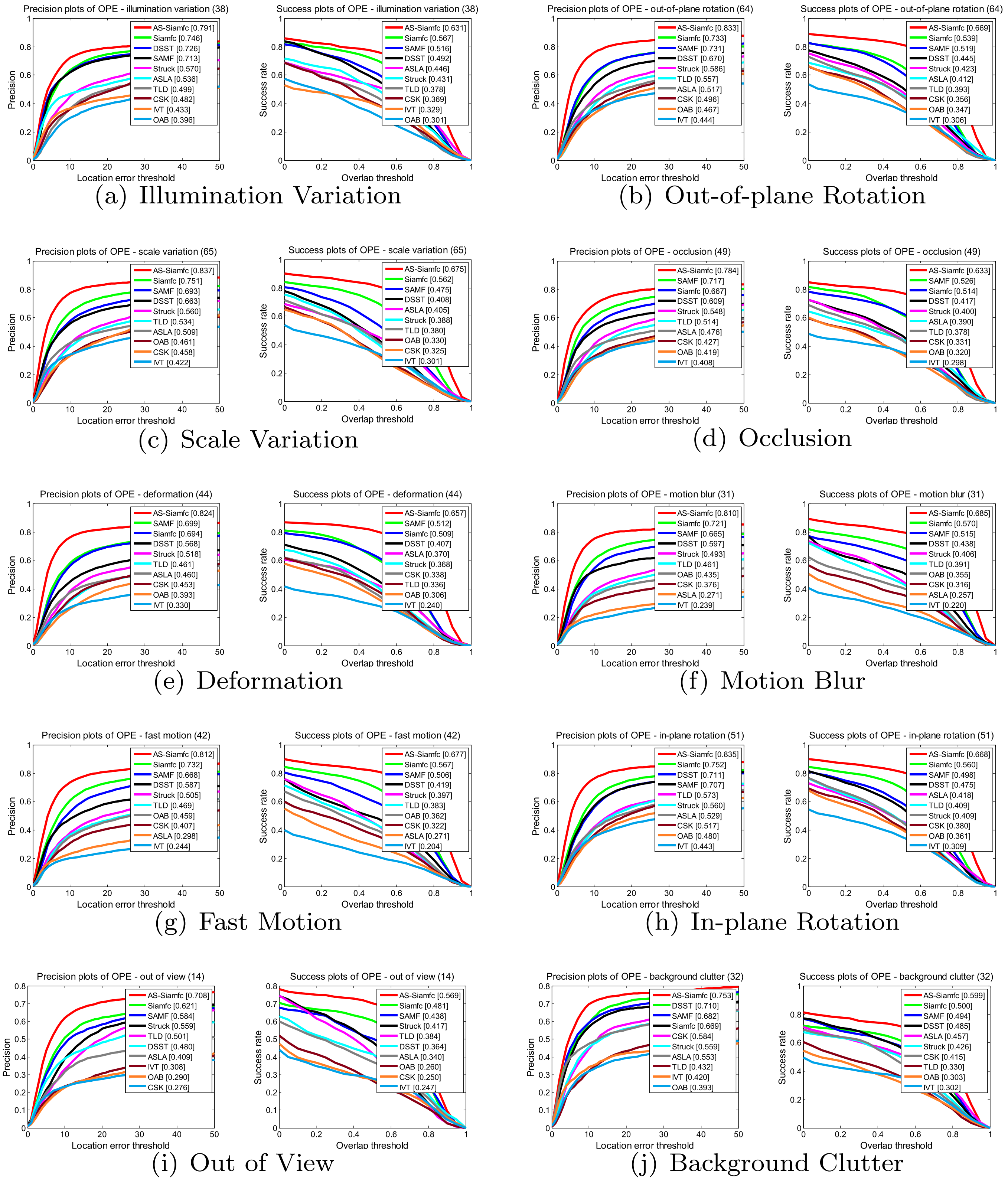
We provide some quantitative analyses in this subsection. We also selected some state-of-the-art trackers including some Siamese network based trackers for comparison which is conducted on the widely used benchmark, OTB2013, OTB100 and OTB50. Two metrics, precision and Area Under Curve (AUC), are used to rank these trackers. Firstly, we show the comparison results of OTB2013, OTB100 and OTB50 respectively. Then, we analyse the performance of the proposed AS-Siamfc tracker under 10 tracking challenges.

Experiments on OTB2013 data set: OTB2013 is one of the widely used benchmark with 52 fully annotated sequences. In order to facilitate the comparison test, the author also provides two evaluation criteria and a toolkit. Fig. 14 shows the precision and success plots of One-Pass Evaluation (OPE) in OTB2013 data set. From Fig. 14, we can see that our proposed AS-Siamfc tracker achieves the best tracking performance against the other comparative trackers at the average speed of 70.625 fps. The precision score and success score are 0.820 and 0.667 respectively. Comparing with Siamfc tracker, the performance of the proposed AS-Siamfc tracker exceeds the performance of Siamfc tracker 0.058 and 0.092 on the precision score and success score respectively.

Experiments on OTB100 data set: In order to increase the number of sequences of OTB2013 data set and to evaluate the visual object trackers more accurately, Wu et al. [47] add some fully annotated sequences into OTB2013 data set to construct the OTB100 data set. Thus, OTB100 data set expends OTB2013 data set from 52 sequences to 100 sequences. Similarly, the evaluation criteria and toolkits in OTB2013 data set are also applicable in OTB100. Fig. 15 shows the precision and success plots of OPE in OTB100 data set. From Fig. 15, we can see that the precision score and success score of the proposed AS-Siamfc are 0.844 and 0.680, which are also the highest scores in Fig. 15. The precision score and success score of the proposed tracker are 0.087 and 0.108 larger than that of Siamfc. By comparing the scores of AS-Siamfc tracker in OTB2013 and OTB100, we find that the scores of AS-Siamfc in OTB100 are average 1.85% higher than that in OTB2013. We believe the reason is that the number of sequences in OTB2013 data set is relatively small, which will have a great impact on the overall precision and success scores if the tracking performances of a certain sequence are not good. On the contrary, there are more sequences in OTB100 and the distribution of the sequences in OTB100 is relatively uniform. Thus, the tracking results of a single video sequence have little influence on the overall precision and success scores. This also illustrates the effectiveness and applicability of the proposed AS-Siamfc tracker.

Experiments on OTB50 data set: OTB50 is composed of 50 hard-to-track sequences selected from the OTB100. It is one of the widely used benchmark with 50 fully annotated sequences. The toolkit proposed in OTB2013 [46] can also applied in OTB50 data set. Fig. 16 shows the precision and success plots of OPE in OTB50 data set. As shown in Fig. 16, our performance of the proposed tracker is better than the other state-of-the-art trackers. The precision score and success score are 0.764 and 0.604 respectively. Comparing with Siamfc tracker, the performance of the proposed AS-Siamfc tracker exceeds the performance of Siamfc tracker 0.071 and 0.091 on the precision score and success score respectively.

Experiments on VOT2018 data set: VOT2018[72] is also one of the widely used benchmarks, which contents 60 sequences, including many tiny, similar tracking objects. Fig. 17 shows the mean AR plot and pooled AR plot for experiment baseline of VOT2018 data set respectively. According to the definition of AR plot in VOT2018 [72], the trackers which locate at the upper right quarter of AR plot perform better than the ones locate at the lower left quarter. As shown in Fig. 17, though the proposed AS-Siamfc tracker has a relatively low score on robustness, its accuracy score is the highest among the trackers for comparison. Generally, when comparing with some state-of-the-art trackers, AS-Siamfc could give a



(a) Illumination Variation

(b) Out-of-plane Rotation

(c) Scale Variation

(d) Occlusion

(e) Deformation

(f) Motion Blur

(g) Fast Motion

(h) In-plane Rotation

(i) Out of View

(j) Background Clutter

Fig. 18. The precision and success plots of 10 tracking challenges in OTB100, using proposed tracker and 8 state-of-the-art trackers.

comparable performance in VOT2018, which illustrates the efficiency of the proposed tracker.

Experiments of 10 tracking challenges in OTB100: In order to analyse the applicability of the proposed AS-Siamfc tracker in different sequences in detail. The sequences in OTB100 data set are divided into 11 tracking challenges, including illumination variation, out-of-plane rotation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out of view, background clutter and low resolution. Fig. 18 shows the precision and success plots of AS-Siamfc under these challenges along with some state-of-the-art trackers. In order to make the figure neat, we select 10 challenges in Fig. 18 instead of all 11 challenges.

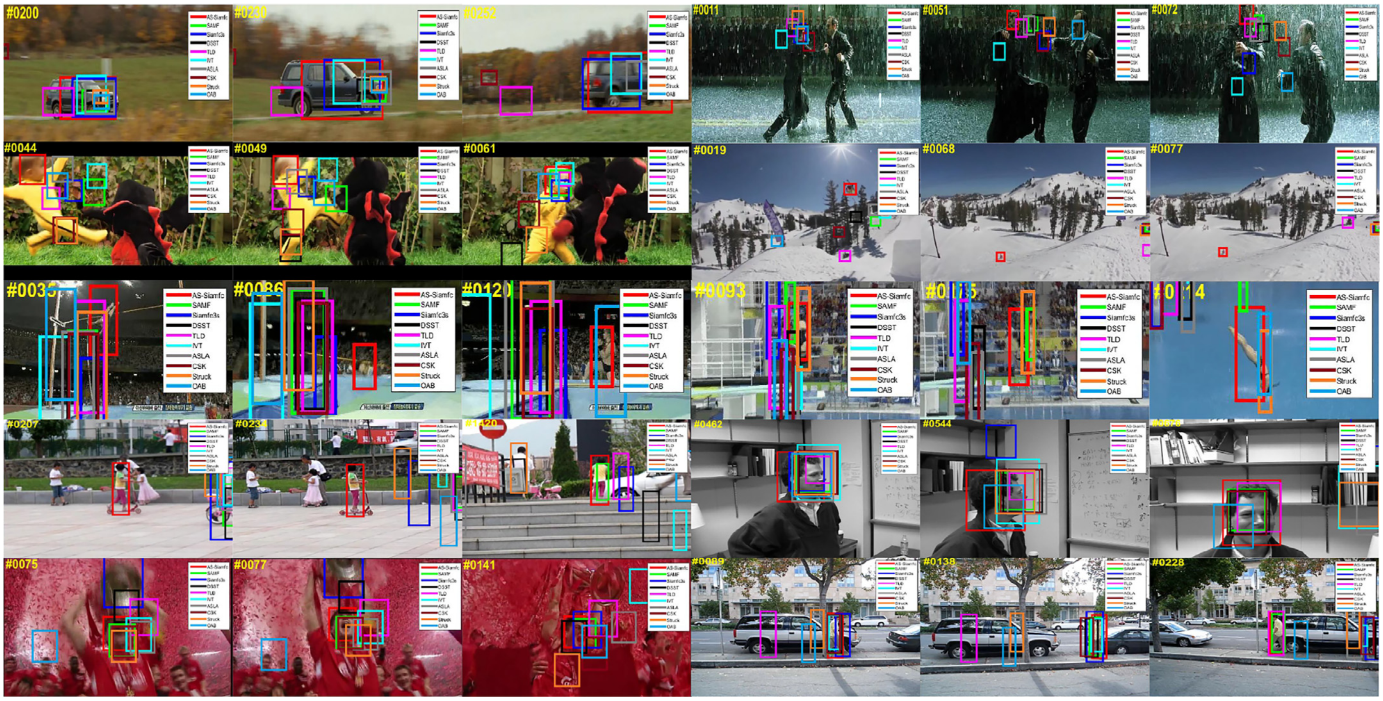


Fig. 19. Qualitative results of 10 typical video sequences in OTB2013, OTB100 and OTB50, using the proposed tracker and 9 state-of-the-art trackers.

From Fig. 18, we can see that the proposed AS-SimaFC tracker performs the best than the other trackers under all the 10 tracking challenges. By comparing the performances of the proposed tracker and Simafc tacker, we find that the proposed tracker shows better performances under the challenges of out-of-plane rotation, scale variation, deformation and fast motion,etc. especially the deformation. From Fig. 18(d), we can see that our proposed tracker shows a relatively good performance in the tracking challenge of occlusion, even though AS-Siamfc does not deliberately design a module to deal with this challenge. We believe the reason is the proposed switch function could indirectly detect the occurrence of occlusion to some extent, and using auxiliary relocation branch to refine the tracking results, since when occlusion occurs, the response map of AS Siamese network may get a low response score, which also means the tracker is running under an untrusted state. Indeed, there are also many occlusion aware methods in visual object tracking and motion segmentation [53,73,74]. How to integrate these occlusion aware methods in the proposed tracking framework to further improve the tracking performance will be our future work. In Fig. 18(e), the precision score and success score of AS-Siamfc are 0.13 and 0.148 higher than that of Siamfc and are 0.125 and 0.145 higher than that of SAMF tracker [67], which is also the second-ranking tracker in Fig. 18(e). We believe this also proves that the proposed AS network and auxiliary relocation branch with switch function could improve the expression power of Siamese network and provide good tracking performances.

4.3.2. Qualitative analyses

In addition to the above-mentioned quantitative analyses experiments, we also show some tracking bounding boxes of the sequences in OTB2013, OTB100 and OTB50 for qualitative analyses in this subsection.

As shown in Fig. 19, we select 10 typical video sequences in OTB2013, OTB100 and OTB50. The names of these 10 sequences are CarScale, Matrix, DragonBaby, Skiing, Jump, Diving, Girl2, FleetFace, Soccer and David3 in order from left to right and from top to bottom. These 10 sequences basically contain all the 11 tracking challenges (one sequence may contain multiple challenges). However, in order to show the advantages of the proposed AS-Siamfc tacker better, these sequences focus more on the challenges of deformation, scale variation and fast motion. From the sequence CarScale, we can see that AS-Siamfc could estimate the state of object better, when the car runs from far to near and gradually larger. In the sequences of Diving and Jump, the athletes have obvious and fast deformations. Even so, the proposed tracker can still track the athletes well. Meanwhile, the sequences of Matrix, DragonBaby and Skiing show the challenge of fast motion, since the people in the fighting or skiing always move fast. The proposed tracker can also provide a relatively accurate tracking performance. Generally, all the quantitative and qualitative experiments can prove the applicability and effectiveness of the proposed AS-Siamfc tracker.

5. Conclusions

In this paper, we proposed a novel Attention shake based tracker which is based on a modified Siamese network. The proposed tracker which is named as AS-Siamfc tracks the object in real-time, whose average tracking speed is 70.625 fps. Moreover, AS-Siamfc tracker can improve the expression power of Siamese network by merging two different attention modules into Siamese network, introducing the prior knowledge into the proposed tracking framework and monitoring the tracking process. Firstly, in order to improve the expression power of Siamese network, a novel attention shake layer which combines two different attention modules with shake-shake framework is proposed in the architecture of the backbone network. Secondly, the auxiliary relocation branch which contains structure similarity weight, motion similarity weight, motion smoothness weight and object saliency weight is proposed to introduce the prior knowledge into the tracking framework and relocate the objects when the tracker runs under untrusted state. Finally, a novel switch function which is based on the response map of AS network is proposed to monitor the tracking process and detect the tracking failure. The qualitative and quantitative experiments as well as the basic experiments show the effectiveness and applicability of the proposed tracking algorithm.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Jun Wang: Software, Methodology, Writing - original draft, Investigation, Conceptualization. **Weibin Liu:** Supervision, Project administration. **Weiwei Xing:** Funding acquisition, Project administration. **Liqiang Wang:** Resources, Conceptualization. **Shunli Zhang:** Funding acquisition, Conceptualization.

Acknowledgment

This research is partially supported by National Natural Science Foundation of China (No. 61876018, No. 61601021, No. 61976017), Beijing Natural Science Foundation (L172022).

References

[1] A.W. Smeulders, D.M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, M. Shah, Visual tracking: an experimental survey, IEEE Trans. Pattern Anal. Mach. Intell. 36 (7) (2014) 1442–1468.

[2] J.F. Henriques, R. Caseiro, P. Martins, J. Batista, High-speed tracking with kernelized correlation filters, IEEE Trans. Pattern Anal. Mach. Intell. 37 (3) (2015) 583–596.

[3] M. Tang, J. Feng, Multi-kernel correlation filter for visual tracking, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, pp. 3038–3046, doi:10.1109/ICCV.2015.348.

[4] M. Danelljan, G. Hger, F.S. Khan, M. Felsberg, Learning spatially regularized correlation filters for visual tracking, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, pp. 4310–4318, doi:10.1109/ICCV.2015.490.

[5] C. Ma, J.-B. Huang, X. Yang, M.-H. Yang, Hierarchical convolutional features for visual tracking, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 3074–3082.

[6] M. Danelljan, G. Hger, F.S. Khan, M. Felsberg, Convolutional features for correlation filter based visual tracking, in: Proceedings of the IEEE International Conference on Computer Vision Workshop (ICCVW), 2015, pp. 621–629, doi:10.1109/ICCVW.2015.84.

[7] H. Nam, B. Han, Learning multi-domain convolutional neural networks for visual tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 4293–4302, doi:10.1109/CVPR.2016.465.

[8] S. Chopra, R. Hadsell, Y. LeCun, et al., Learning a similarity metric discriminatively, with application to face verification, in: Proceedings of the CVPR (1), 2005, pp. 539–546.

[9] R. Tao, E. Gavves, A.W. Smeulders, Siamese instance search for tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1420–1429.

[10] L. Bertinetto, J. Valmadre, J.F. Henriques, A. Vedaldi, P.H. Torr, Fully-convolutional siamese networks for object tracking, in: Proceedings of the European Conference on Computer Vision, Springer, 2016, pp. 850–865.

[11] Z. Zhang, H. Peng, Deeper and wider siamese networks for real-time visual tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 4591–4600.

[12] X. Dong, J. Shen, Triplet loss in siamese network for object tracking, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 459–474.

[13] B. Li, J. Yan, W. Wu, Z. Zhu, X. Hu, High performance visual tracking with siamese region proposal network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8971–8980.

[14] X. Gastaldi, Shake-shake regularization, arXiv preprint arXiv:1705.07485 (2017).

[15] X. Zhou, Y. Li, B. He, Entropy distribution and coverage rate-based birth intensity estimation in GM-PHD filter for multi-target visual tracking, Signal Process. 94 (2014) 650–660.

[16] L. Wen, Z. Lei, S. Lyu, S.Z. Li, M.H. Yang, Exploiting hierarchical dense structures on hypergraphs for multi-object tracking, IEEE Trans. Pattern Anal. Mach. Intell. 38 (10) (2016) 1983–1996, doi:10.1109/TPAMI.2015.2509979.

[17] J. Shen, D. Yu, L. Deng, X. Dong, Fast online tracking with detection refinement, IEEE Trans. Intell. Transp. Syst. 19 (1) (2017) 162–173.

[18] J. Shen, Z. Liang, J. Liu, H. Sun, L. Shao, D. Tao, Multiobject tracking by submodular optimization, IEEE Trans Cybern 49 (6) (2018) 1990–2001.

[19] A. Yilmaz, O. Javed, M. Shah, Object tracking: a survey, Acm Comput. Surv. (CSUR) 38 (4) (2006) 13.

[20] B. Ma, H. Hu, J. Shen, Y. Zhang, L. Shao, F. Porikli, Robust object tracking by nonlinear learning, IEEE Trans. Neural Netw. Learn. Syst. 29 (10) (2017) 4769–4781.

[21] B. Ma, L. Huang, J. Shen, L. Shao, M.-H. Yang, F. Porikli, Visual tracking under motion blur, IEEE Trans. Image Process. 25 (12) (2016) 5867–5876.

[22] B. Ma, H. Hu, J. Shen, Y. Liu, L. Shao, Generalized pooling for robust object tracking, IEEE Trans. Image Process. 25 (9) (2016) 4199–4208.

[23] B. Ma, J. Shen, Y. Liu, H. Hu, L. Shao, X. Li, Visual tracking using strong classifier and structural local sparse descriptors, IEEE Trans. Multimedia 17 (10) (2015) 1818–1828.

[24] R.-S.L. David Ross Jongwoo Lim, M.-H. Yang, Incremntal learning for robust visual tracking, Int. J. Comput. Vis. 77 (1) (2008) 125–141.

[25] A. Abdel-Hadi, Real-time object tracking using color-based kalman particle filter, in: Proceedings of the International Conference on Computer Engineering and Systems (ICCES), IEEE, 2010, pp. 337–341.

[26] Z. Han, T. Xu, Z. Chen, An improved color-based tracking by particle filter, in: Proceedings of the International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE), IEEE, 2011, pp. 2512–2515.

[27] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S.L. Hicks, P.H. Torr, Struck: structured output tracking with kernels, IEEE Trans. Pattern Anal. Mach. Intell. 38 (10) (2016) 2096–2109.

[28] J. Zhang, S. Ma, S. Sclaroff, Meem: robust tracking via multiple experts using entropy minimization, in: Proceedings of the European Conference on Computer Vision, Springer, 2014, pp. 188–203.

[29] W. Chen, K. Zhang, Q. Liu, Robust visual tracking via patch based kernel correlation filters with adaptive multiple feature ensemble, Neurocomputing 214 (2016) 607–617.

[30] F. Yang, H. Lu, M.-H. Yang, Robust superpixel tracking, IEEE Trans. Image Process. 23 (4) (2014) 1639–1651.

[31] Y. Li, Y. Su, Y. Liu, Fast two-cycle curve evolution with narrow perception of background for object tracking and contour refinement, Signal Process. Image Commun. 44 (2016) 29–43.

[32] L. Huang, B. Ma, J. Shen, H. He, L. Shao, F. Porikli, Visual tracking by sampling in part space, IEEE Trans. Image Process. 26 (12) (2017) 5800–5810.

[33] X. Mei, H. Ling, Robust visual tracking and vehicle classification via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (11) (2011) 2259–2272, doi:10.1109/TPAMI.2011.66.

[34] W. Hu, W. Li, X. Zhang, S. Maybank, Single and multiple object tracking using a multi-feature joint sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 37 (4) (2015) 816–833, doi:10.1109/TPAMI.2014.2353628.

[35] S. Zhang, Y. Sui, X. Yu, S. Zhao, L. Zhang, Hybrid support vector machines for robust object tracking, Pattern Recognit. 48 (8) (2015) 2474–2488.

[36] Y. Yin, D. Xu, X. Wang, M. Bai, Online state-based structured SVM combined with incremental PCA for robust visual tracking, IEEE Trans. Cybern. 45 (9) (2015) 1988–2000.

[37] J.F. Henriques, R. Caseiro, P. Martins, J. Batista, Exploiting the circulant structure of tracking-by-detection with kernels, in: Proceedings of the European Conference on Computer Vision, Springer, 2012, pp. 702–715.

[38] C. Ma, X. Yang, C. Zhang, M.-H. Yang, Long-term correlation tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5388–5396.

[39] M. Danelljan, G. Häger, F. Khan, M. Felsberg, Accurate scale estimation for robust visual tracking, in: Proceedings of the British Machine Vision Conference, Nottingham, September 1–5, 2014, BMVA Press, 2014.

[40] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, J. Yan, Siamrpn++: evolution of siamese visual tracking with very deep networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 4282–4291.

[41] M. Danelljan, G. Bhat, F.S. Khan, M. Felsberg, Atom: accurate tracking by overlap maximization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 4660–4669.

[42] M. Danelljan, A. Robinson, F.S. Khan, M. Felsberg, Beyond correlation filters: learning continuous convolution operators for visual tracking, in: Proceedings of the European Conference on Computer Vision, Springer, 2016, pp. 472–488.

[43] A. He, C. Luo, X. Tian, W. Zeng, A twofold siamese network for real-time object tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4834–4843.

[44] D. Held, S. Thrun, S. Savarese, Learning to track at 100 fps with deep regression networks, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 749–765.

[45] X. Dong, J. Shen, W. Wang, Y. Liu, L. Shao, F. Porikli, Hyperparameter optimization for tracking with continuous deep q-learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 518–527.

[46] Y. Wu, J. Lim, M.H. Yang, Online object tracking: a benchmark, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2411–2418, doi:10.1109/CVPR.2013.312.

[47] Y. Wu, J. Lim, M.H. Yang, Object tracking benchmark, IEEE Trans. Pattern Anal. Mach. Intell. 37 (9) (2015) 1834–1848, doi:10.1109/TPAMI.2014.2388226.

[48] P. Liang, E. Blasch, H. Ling, Encoding color information for visual tracking: algorithms and benchmark, IEEE Trans. Image Process. 24 (12) (2015) 5630–5644.

[49] M. Mueller, N. Smith, B. Ghanem, A benchmark and simulator for UAV tracking, in: Proceedings of the European Conference on Computer Vision, Springer, 2016, pp. 445–461.

[50] L. Huang, X. Zhao, K. Huang, Got-10k: a large high-diversity benchmark for generic object tracking in the wild, IEEE Trans. Pattern Anal. Mach. Intell. (2019).

[51] H. Fan, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, H. Bai, Y. Xu, C. Liao, H. Ling, Lasot: A high-quality benchmark for large-scale single object tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 5374–5383.

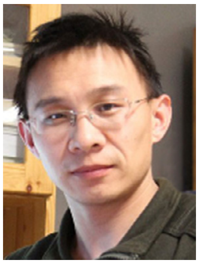
[52] D.S. Bolme, J.R. Beveridge, B.A. Draper, Y.M. Lui, Visual object tracking using adaptive correlation filters, in: Proceedings of the IEEE Computer Conference on Computer Vision and Pattern Recognition, 2010, pp. 2544–2550, doi:10.1109/CVPR.2010.5539960.
[53] X. Dong, J. Shen, D. Yu, W. Wang, J. Liu, H. Huang, Occlusion-aware real-time object tracking, IEEE Trans. Multimedia 19 (4) (2016) 763–771.
[54] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, P.H. Torr, Fast online object tracking and segmentation: a unifying approach, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 1328–1338.
[55] H. Fan, H. Ling, Siamese cascaded region proposal networks for real-time visual tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 7952–7961.
[56] S. Zagoruyko, N. Komodakis, Learning to compare image patches via convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4353–4361.
[57] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, W. Hu, Distractor-aware siamese networks for visual object tracking, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 101–117.
[58] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, S. Wang, Learning dynamic siamese network for visual object tracking, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1763–1771.
[59] X. Dong, J. Shen, D. Wu, K. Guo, X. Jin, F. Porikli, Quadruplet network with one-shot learning for fast visual object tracking, IEEE Trans. Image Process. 28 (7) (2019) 3516–3527.
[60] Q. Wang, Z. Teng, J. Xing, J. Gao, W. Hu, S. Maybank, Learning attentions: residual attentional siamese network for high performance online visual tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4854–4863.
[61] J. Shen, X. Tang, X. Dong, L. Shao, Visual object tracking by hierarchical attention siamese network, IEEE Trans. Cybern. (2019).
[62] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141.
[63] J.-Y. Bouguet, et al., Pyramidal implementation of the affine Lucas Kanade feature tracker description of the algorithm, Intel Corp. 5 (1–10) (2001) 4.
[64] R. Chaudhry, A. Ravichandran, G. Hager, R. Vidal, Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 1932–1939.
[65] R. Achanta, S. Hemami, F. Estrada, S. Süsstrunk, Frequency-tuned salient region detection, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2009), in: CONF, 2009, pp. 1597–1604.
[66] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), vol. 1, IEEE, 2005, pp. 886–893.
[67] Y. Li, J. Zhu, A scale adaptive kernel correlation filter tracker with feature integration., in: Proceedings of the ECCV Workshops (2), 2014, pp. 254–265.
[68] Z. Kalal, J. Matas, K. Mikolajczyk, Pn learning: Bootstrapping binary classifiers by structural constraints, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 49–56.
[69] H. Lu, X. Jia, M.H. Yang, Visual tracking via adaptive structural local sparse appearance model, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1822–1829.
[70] H. Grabner, M. Grabner, H. Bischof, Real-time tracking via on-line boosting., in: Proceedings of the BMVC, 1, 2006, p. 6.
[71] D.A. Ross, J. Lim, R.-S. Lin, M.-H. Yang, Incremental learning for robust visual tracking, Int. J. Comput. Vis. 77 (1) (2008) 125–141.
[72] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Cehovin Zajc, T. Vojir, G. Bhat, A. Lukezic, A. Eldesokey, et al., The sixth visual object tracking vot2018 challenge results, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018. 0–0
[73] W. Wang, J. Shen, F. Porikli, R. Yang, Semi-supervised video object segmentation with super-trajectories, IEEE Trans. Pattern Anal. Mach. Intell. 41 (4) (2018) 985–998.
[74] J. Shen, J. Peng, L. Shao, Submodular trajectories for better motion segmentation in videos, IEEE Trans. Image Process. 27 (6) (2018) 2688–2700.

Jun Wang received the M.S. degree in Pattern Recognition and Intelligent Systems from Hebei University, China, in 2015. Currently, he is a Ph.D. candidate of Institute of Information Science at Beijing Jiaotong University, China. His research interests include image processing, computer vision, visual object tracking and pattern recognition.

Weibin Liu received the Ph.D. degree in Signal and Information Processing from Institute of Information Science at Beijing Jiaotong University, China, in 2001. During 2001–2005, he was a researcher in Information Technology Division at Fujitsu Research and Development Center Co., LTD. Since 2005, he has been with the Institute of Information Science at Beijing Jiaotong University, where currently he is a professor in Digital Media Research Group. He was also a visiting researcher in Center for Human Modeling and Simulation at University of Pennsylvania, PA, USA during 2009–2010. His research interests include computer vision, computer graphics, image processing, virtual human and virtual environment, and pattern recognition. He is a member of the IEEE, ACM, IEICE and CCF.

Weiwei Xing received her B.S. degree in Computer Science and Technology and Ph.D. degree in Signal and Information Processing from Beijing Jiaotong University, in 2001 and 2006 respectively. During 2011–2012, she was a visiting scholar at University of Pennsylvania. Currently, she is an associate professor at School of Software Engineering, Beijing Jiaotong University. Her research interests mainly include intelligent information processing and artificial intelligence.

Liqiang (Eric) Wang is an associate professor in the Department of Computer Science at the University of Central Florida. He is the director of Big Data Computing Lab. He was a faculty member (2006–2015) in the Department of Computer Science at the University of Wyoming. He received Ph.D. in Computer Science from Stony Brook University in 2006. He was a visiting Research Scientist in IBM T.J. Watson Research Center during 2012–2013. His research focuses on integrating deep learning, parallel computing, and program analysis, which includes the following aspects: (1) improving the robustness, accuracy, speed, and scalability of deep learning; (2) optimizing performance, scalability, resilience, and resource management of big data processing, especially on Cloud, GPU, and multicore platforms; (3) using hybrid program analysis to detect and avoid programming errors, execution anomaly, as well as performance defects in large-scale parallel computing systems. He received NSF CAREER Award in 2011 and Castagne Faculty Fellowship (2013–2015).



Shunli Zhang received the B.S. and M.S. degrees from Shandong University in 2008 and 2011 respectively, and Ph.D degree from Tsinghua University in 2016. He is currently a faculty in Beijing Jiaotong University. His research interests include image processing, pattern recognition and computer vision.