Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 6681391, 19 pages
https://doi.org/10.1155/2020/6681391

Research Article

MFCFSiam: A Correlation-Filter-Guided Siamese Network with Multifeature for Visual Tracking

Chenpu Li, Qianjian Xing, Zhenguo Ma, and Ke Zang

College of Biomedical Engineering & Instrument Science, Zhejiang University, Hangzhou 310027, China

Correspondence should be addressed to Zhenguo Ma; 850501@zju.edu.cn

Received 21 October 2020; Revised 12 November 2020; Accepted 12 December 2020; Published 24 December 2020

Academic Editor: Amr Tolba

With the development of deep learning, trackers based on convolutional neural networks (CNNs) have made significant achievements in visual tracking over the years. The fully connected Siamese network (SiamFC) is a typical representation of those trackers. SiamFC designs a two-branch architecture of a CNN and models' visual tracking as a general similarity-learning problem. However, the feature maps it uses for visual tracking are only from the last layer of the CNN. Those features contain high-level semantic information but lack sufficiently detailed texture information. This means that the SiamFC tracker tends to drift when there are other same-category objects or when the contrast between the target and the background is very low. Focusing on addressing this problem, we design a novel tracking algorithm that combines a correlation filter tracker and the SiamFC tracker into one framework. In this framework, the correlation filter tracker can use the Histograms of Oriented Gradients (HOG) and color name (CN) features to guide the SiamFC tracker. This framework also contains an evaluation criterion which we design to evaluate the tracking result of the two trackers. If this criterion finds the SiamFC tracker fails in some cases, our framework will use the tracking result from the correlation filter tracker to correct the SiamFC. In this way, the defects of SiamFC's high-level semantic features are remedied by the HOG and CN features. So, our algorithm provides a framework which combines two trackers together and makes them complement each other in visual tracking. And to the best of our knowledge, our algorithm is also the first one which designs an evaluation criterion using correlation filter and zero padding to evaluate the tracking result. Comprehensive experiments are conducted on the Online Tracking Benchmark (OTB), Temple Color (TC128), Benchmark for UAV Tracking (UAV-123), and Visual Object Tracking (VOT) Benchmark. The results show that our algorithm achieves quite a competitive performance when compared with the baseline tracker and several other state-of-the-art trackers.

1. Introduction

Visual tracking is a very fundamental and important research topic in computer vision. It is widely used in video surveillance [1], automonitoring [2], motion-based recognition [3], and many other fields. The main purpose of visual tracking is to solve the problem of target recognition and localization in a series of video image frames. Typically, given the labeled bounding box of the target in the first frame of a video, an ideal tracker should come up with this target's accurate position coordinates and mark it with a properly sized bounding box in each following frame of the video [4]. However, this seemingly simple task involves many difficulties that will lead to tracking failure if not properly addressed, such as obstacle occlusion [5, 6] illumination changing [7–9], deformation [10], size scale variations [11], and complex background clutter [12, 13].

To solve the problems listed above, a large variety of tracking approaches have been proposed over the years. Roughly, those approaches can be divided into two main categories: discriminative methods and generative methods. Discriminative methods [14–18] usually model the visual tracking task as a binary classification problem and train a robust classifier to distinguish the target from the background in every video frame. For example, in [11, 19, 20], all those three trackers use a support vector machine (SVM) as their main component in the visual tracking framework, and the SVM is a typical and classic discriminative

model which is widely used in machine-learning-related tasks. In [21], the authors design a tracker which combines local sparse descriptors into a boosting-based strong classifier using a discriminative appearance model. However, the main purpose of generative methods is to build up several appearance models of the target as templates and then search the video frame to find which region is most similar to the target's templates; this region is marked as the final tracking result. In [22], a consistent low-rank sparse tracker (CLRST) is designed on the basis of particle filter framework. The particle filter is a classic and typical generative model which is widely used in visual tracking. What is more, in the tracker from [23], the particle filter is used to build a redetection model, and this model is combined with a kernel correlation filter tracker to make it more robust. Other examples of generative models in visual tracking include trackers based on matrix decomposition [21, 22, 24] and those based on subspace learning [11, 19, 20]. In [24], an incremental nonnegative matrix factorization (INMF) method is used to address the visual tracking task. In [21], the authors combine the holistic and part-based representations with nonnegative matrix factorization (NMF) and model the target by a nonnegative combination of nonnegative components. In [11], the authors design a tracker which can efficiently adapt online information of the target's appearance by learning a low-dimensional subspace representation incrementally. Both the generative model and discriminative model are used in the tracking framework proposed by [19]. In [20], a subspace learning algorithm is used to impose joint row-wise sparsity structure on the target subspace. By this method, distractive information can be adaptively excluded.

Both the generative tracking models and the discriminative tracking models share the same key step: extracting powerful features from the target to represent it as distinctly as possible; those features are then used as references for tracking. Some traditional features include Histograms of Oriented Gradients (HOG), scale-invariant feature transform (SIFT), and color name (CN) [25, 26]. Most recently, with the boom in convolutional neural networks (CNNs) [27], many computer vision-related tasks have benefited from this and have achieved state-of-the-art performance [28–33]. CNN is a typical deep-learning architecture. Trained with a large set of image data, the CNN can learn to capture different levels of features owing to its multiple layers of convolution filters. Each filter can act as a specific feature pattern extractor, and combining them results in very powerful feature models. Recently, researchers have begun to integrate CNN into a visual tracking framework to try to explore the potential of deep features in this field [34–37]. In [34], the authors design a tracker using a single CNN to learn effective feature representations of the target object in an online manner. The tracker in [35] pretrains a CNN on a large set of videos to make sure the CNN can learn a generic target representation of the target. In [36], the authors adopt a tree structure to manage multiple target appearance models. They use multiple CNNs to estimate target states and determine the desirable paths for model update during tracking. Typically, Bertinetto et al. utilize the SiamFC [38] architecture and treat visual tracking as a general similarity-learning model that achieves the end-to-end workflow of tracking. This architecture achieves state-of-the-art performance and runs at about 80 fps on graphics processing units (GPUs), showing its significant potential in visual tracking.

However, in SiamFC's tracking process, only the features from the last CNN layer are used for visual tracking. The advantage of those high-level features is that they contain semantic information of the target that is very robust to appearance deformation. However, a drawback is that the semantic information lacks enough detailed texture information to distinguish the target from other same-category objects. That is to say, if there are other objects of the same type as the target in the search region, those objects would distract the tracker and cause the tracking to fail. What is more, when the contrast between the target and the background is very low—for example, as shown in Figure 1—the SiamFC also tends to drift. We can see that in the first 3 columns, when there exists other same-category objects in the search region, the score maps of SiamFC will produce large response on all these objects and this may lead the tracker to failure. And the fourth column shows that when the contrast between the target and the background is very low, the score maps of SiamFC will produce large response on the background. In other words, the deep features of this sequence even cannot provide effective information to distinguish the target from the background.

During our research, we found that the defect above is better addressed by correlation filter-based trackers. Some traditional handcrafted features such as HOG and CN are used in those correlation filter trackers, and their performances showed that these two features are very robust and effective when dealing with some complicated tracking environments. In our opinion, human's eyes are powerful trackers and we believe that the effective way to design a robust tracker is to follow the tracking logic of human eyes. When we use our eyes to track object, the main information we use contain two aspects: the target's contour information and color information. The HOG feature can represent the distribution of gradient and edge information in each local part of an object; thus, it is an ideal tool to describe the target's contour information. And the CN features are an ideal tool to describe the target's color information. What is more, the calculation of the HOG and CN features is very fast so they meet the requirement of real-time tracking. So, we believe that the HOG and CN features are the ideal instruments to compensate for SiamFC's shortcomings. Those features can express the detailed texture information of a target, and the information is usually the more visible traits that distinguish the right target from other objects in the search region. What is more, in each frame of a correlation filter tracker, a large number of samples are produced by cyclic sampling to train a robust classifier. This procedure guarantees that the correlation filter tracker is discriminative enough to distinguish the right target from other same-category distractors or a complex background. One another characteristic is that the search region's size in the correlation filter tracker is usually smaller than that in the SiamFC tracker. The benefit is that a smaller search region contains fewer objects so the tracker will not be likely to drift.

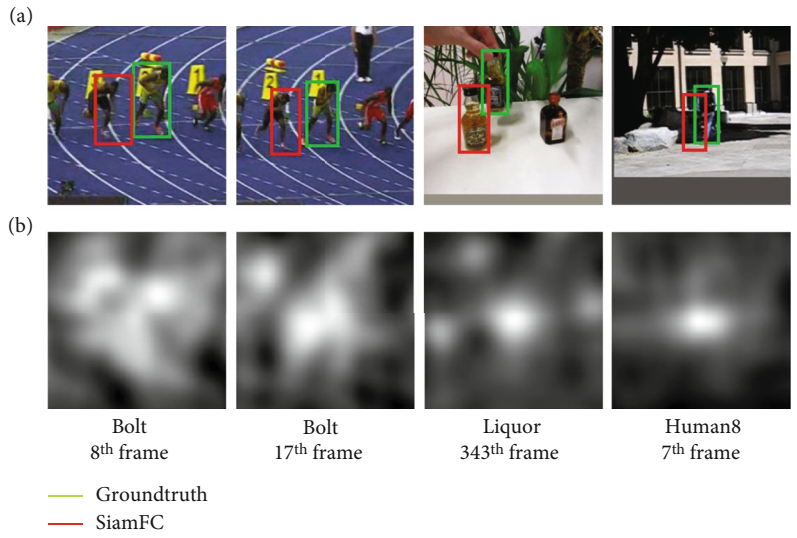

FIGURE 1: Several typical sequences of SiamFC's tracking failure from OTB100 dataset. (a) The original search region of SiamFC. (b) Their corresponded score maps, and the whiter areas on the score maps represent higher responses.

However, a smaller search region is more likely to lose the target if it moves fast.

So, to a certain extent, the advantages of the SiamFC tracker and CF trackers are complementary to each other. This provides the motivation for our research question: can we design a framework to make a correlation filter tracker and a SiamFC tracker work together? In this paper, we design a tracking framework that uses a correlation filter tracker to guide the tracking process of a SiamFC tracker. This framework is called a correlation-filter-guided Siamese network with multiple features (MFCFSiam). The main contributions of our work are summarized as follows:

(1) We design an effective criterion based on a correlation filter and the zero-padding method to judge which tracking results are more credible between a SiamFC tracker and a CF tracker. To the best of our knowledge, our algorithm is the first one which designs an evaluation criterion using correlation filter and zero padding to evaluate the tracking result

(2) We design a novel tracking framework that combines the advantages of a SiamFC tracker and correlation filter trackers. This framework works on the basis of the evaluation criterion in (1) and can effectively utilize both the semantic features from CNN and detailed texture features from traditional handcrafted feature extractors such as HOG and CN. Each kind of feature can make up for the disadvantages of other features, so the tracker can be more robust when faced with complicated tracking environments. Our framework shows an example of how to combine two trackers that share mutual advantages and make them complement each other in visual tracking

(3) We conducted a number of experiments to evaluate our proposed tracker on the dataset of Online Tracking Benchmark (OTB), Temple Color (TC128), the Benchmark for UAV Tracking (UAV-123), and the Visual Object Tracking (VOT) Benchmark. These datasets are very classic and typical in visual tracking, and the experiment results showed that our tracker achieved competitive performance when compared with baseline trackers and other state-of-the-art trackers

The rest of the paper is organized as follows. In Section 2, we introduce some related works in visual tracking. Then, we present our tracking framework in Section 3. In Section 4, we evaluate the performance of our tracker on the mainstream dataset and compare it with other representative trackers. Section 5 presents a summary of our work.

2. Related Work

2.1. Correlation Filter-Based Trackers. Recently, correlation filter-based trackers have attracted a great deal of attention due to their computational efficiency and competitive performance. These trackers [39] mainly focus on constructing a robust yet efficient appearance model of the target, which is called a correlation filter. Then, they sample several candidates around the search region and use them as inputs of the correlation filter. The filter will output each candidate's correlation score, and the one that gets the maximum response score is labeled as the final tracking result. Bolme et al. [40] first designed a correlation filter- (CF-) based tracker with a minimum output sum of squared error (MOSSE) filter, using raw pixels of images as inputs to train the correlation filters without any feature extraction. Henriques et al. [41] designed a CSK tracker using ridge regression and kernel tricks, but only utilized the gray features when training the filter, which limited the tracker's accuracy. Then, Danelljan et al. [25] integrated the color attribute into the CF tracker and improved its performance. In the KCF/DCF tracker [42] designed by Henriques et al., the feature

representation was extended into multichannel HOG and efficiently incorporated those features into the Fourier domain. It also proposed a kernel ridge regression model to accelerate its processing speed.

However, all the trackers listed above showed poor performance when the targets' size scale changed significantly. LCT [43] solved this problem by decomposing the tracking task into translation and scale estimation. Danelljan et al. [44] trained two kinds of filters to tackle the target's fast scale estimation—one for translation and one for scale estimation—and this DSST tracker enhanced the tracking performance significantly and showed a generic method to address the problem of scale estimation in visual tracking, while the tracker in [45] designs a metric learning function to solve the target scale problem. GFS-DCF [46] introduces a channel selection mechanism into CF-based trackers. This tracker is equipped with deep neural network features and the ability of joint feature selection and filter learning. The TRBACF [47] tracker designs a temporal regularization strategy which can efficiently adjust the model to adapt to the change of the tracking scenes, and this makes it more robust to complex environments. The ARCF [48] filter focuses on addressing the boundary effect problem in a correlation filter and adds restrictions to the alteration rate in response maps, so aberrances in detection can be largely suppressed, which makes the tracker more robust and accurate. The TFCR [49] designs a target-focusing loss function to alleviate the influence of background noise on the response map and improves the tracking accuracy.

2.2. CNN-Based Trackers. CNN-based trackers can be categorized into two main types. One uses the CNN as a single component in visual tracking and as a feature extractor to provide powerful features. For example, in HCF [10] and HDT [50], CNN was used to extract features instead of conventional handcrafted features. DeepSRDCF [51] employed the features extracted from shallow layers of CNN in a spatially regularized DCF tracking framework. All the above methods have one characteristic in common, that is, the CNN they used was always pretrained in some other task, such as image classification or target detection. In other words, they did not model the visual tracking as an end-to-end task and did not train the CNN specifically for visual tracking, so CNN's advantage in end-to-end tasks was not realized.

The other method is to model the visual tracking as an end-to-end task and train the CNN especially for tracking. Bertinetto et al. [38] considered visual tracking as a similarity-learning problem and designed a fully convolutional Siamese network (SiamFC) to evaluate the similarity between the target and the candidate search region. To some extent, this framework realized an end-to-end workflow specifically for tracking problems and achieved quite competitive performance. Following SiamFC, CFNet [52] added a correlation filter layer into the SiamFC network to extract features that are consistent with the CF layer. Guo et al. [53] designed the DSiam tracker, which added a component that combined two general transformations to represent target appearance and suppress noise.

With the development of the visual detection task, some researchers have tried to adopt the experience in visual detection to address the visual tracking problem. The SiamRPN [54] tracker introduces a region proposal network (RPN) from visual detection into visual tracking and designs a regression branch for a bounding box on the basis of SiamFC. So, this tracker's ability at target-scale estimation is obviously improved, but as its template is fixed during the tracking process, it is not so robust when the target's appearance changes quickly. The D3S [55] tracker addresses this defect by setting up two models to encode the target. One model is adaptive and discriminative while the other model is invariant to a broad range of transformations. The SiamAttn [56] tracker also focuses on improving the tracker's robustness to large appearance variations. It introduces an attention mechanism into SiamFC to improve the network's feature-learning capability and achieves more stable and accurate tracking. Cascaded RPN (C-RPN) [57] is another tracker that uses RPN to address the visual tracking problem. It focuses on solving the data imbalance during training and designs a hard negative sampling method to train the network. SiamRCNN [58] adapts Faster RCNN [59] on the basis of Siamese architecture and designs a tracker using the Tracking by Re-Detection framework. It uses Faster RCNN to generate region proposals and determines if those proposals are the same as the template target.

What is more, the RCT (real-time complementary tracker) in [60] is produced by combining SiamFC and CF-based trackers together in a series connection. In this tracker, the SiamFC is used to locate the target coarsely, and then, in the second stage, the derived coarse location is refined by CF-based trackers for higher accuracy. Actually, the design of our MFCFSiam tracker is mostly inspired by RCT. But we combine the SiamFC and CF-based trackers in a parallel connection, not a series connection. The CF-based tracker in our framework is used to guide the SiamFC by HOG and CN features. The disadvantage of RCT is that as the trackers are combined in a series connection, so if the SiamFC goes wrong in the first stage, then the CF-based trackers will be sure to lose the target. However, in our MFCFSiam, this disadvantage does not exist. The parallel connection can guarantee that the two trackers work independently, and the evaluation criterion we design can make the two trackers cooperate with each other more effectively. Next, we will introduce our MFCFSiam tracker in detail.

3. Proposed Algorithm

The tracking algorithm we propose in this paper is to design a framework to remedy the defects of SiamFC by combining it with a correlation filter tracker. This correlation filter can use the detailed texture information of the target such as HOG and CN features to guide the SiamFC tracker. We design a criterion to evaluate the validity of the two trackers' tracking results. If the evaluation shows that the correlation filter tracker's result is more reliable, our framework will use this result to adjust the SiamFC tracker. The overview of our algorithm's workflow is shown in Figure 2. Details will be described in the following sections.

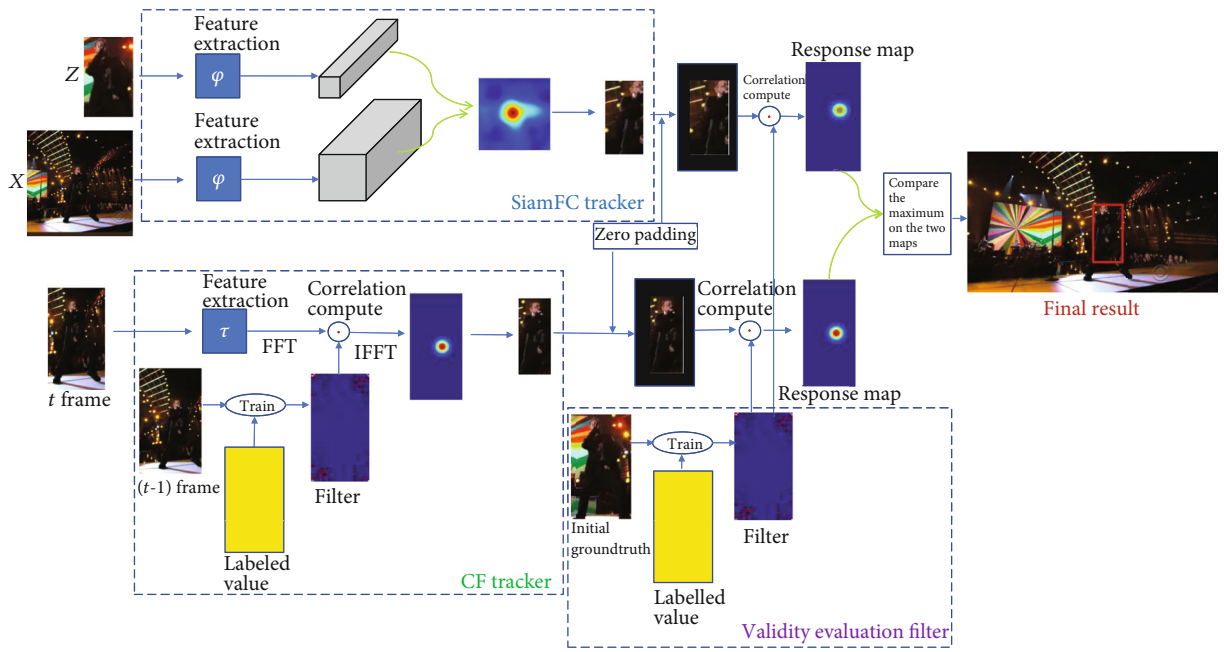

Figure 2: The basic workflow of our proposed tracking framework. The CF tracker based on HOG and CN features is used to guide the SiamFC tracker. The SiamFC tracker and the CF tracker produce their own tracking results. The validity evaluation filter uses the initial groundtruth in the first frame to generate a robust filter, and this filter is used to evaluate the two tracking results' validity. Correlation computation is conducted between this filter and each of the two results, and two response maps are produced. Finally, the result that has the bigger maximum on its response map is considered to be the final result, and this result is also used to update the SiamFC tracker.

3.1. The SiamFC Tracker Using Deep Features. The main architecture of SiamFC is made up of two branches: one branch is used to process the initial groundtruth annotated in the first frame of the image sequence—i.e., the initial template of the target, denoted as exemplar x, is used as the reference to judge whether an image pitch is the target or not—while the other branch is used to process the search region cropped from the frame that is to be searched, denoted as instance z.

The sizes of images input into the exemplar branch and instance branch are set to 127×127 pixels and 255×255 pixels, respectively. Inputs of the two branches will pass through the CNN and meet in the cross-correlation layers. In the cross-correlation layers, the feature maps from the exemplar branch are used as a slide window to compute similarity scores with all the subregions of the feature maps from the instance branch. Each similarity score is achieved by calculating

$$\text{Similarity}(z, x(i)) = \sum_{(m,n)\in S} \varphi(Z)_{(m,n)} \times \varphi(x(i))_{(m,n)}, \quad (1)$$

where $x(i)$ is the ith image pitch of exemplar x with the same size as z, φ represents the process of feature extraction, and $\varphi(z)_{(m,n)}$ is the pixel value vector with location coordinates (m, n) on z's feature maps. S denotes the whole area of each feature map. Thus, this correlation calculation procedure will generate a score map, and each pixel value on the score map records the similarity between an image pitch of instance x and exemplar z. Then, the maximal value of the whole score map is considered to be the target.

Regarding the training of the backbone convolutional neural network, an offline pretraining method with logistic loss function is used. The training data are pairs of images that are input into the instance branch and exemplar branch, respectively. Then, we choose the response score map from the last layer of the CNN and label each pixel with -1 or $+1$ according to the pixel's distance from the map's center; this binary-labeled score map is used as the groundtruth. The loss function is defined as

$$l(y, t) = \log\ (1 + \exp\ (-yt)), \quad (2)$$

where y is the groundtruth and t is the real-valued score map. Then, the final loss of the score map is defined as the mean of each pixel's loss:

$$L(y, t) = \frac{1}{|D|} \sum_{u\in D} l(y[u], t[u]), \quad (3)$$

where D is the whole area of the score map and $u \in D$ represents each position on it. Thus, the parameters of the CNN θ can be obtained by using the Stochastic Gradient Descent (SGD):

$$\arg\ \min L(y, t(z, x, \theta)). \quad (4)$$

3.2. The Guide Correlation Filter Tracker Using HOG and CN Features. As discussed in Section 1, features used in SiamFC

tracker are high-level semantic features from the last layer of CNN. Those features will not be robust enough if the search region contains other same-category distractors. What is more, another factor can make this situation worse: the large search area used in SiamFC. To some extent, a large search area can ensure that the right target is included in it even the target moves very quickly; that is the advantage. However, a large search area also makes it easier to introduce other distractors that may lead the tracker to drift. The location and size of the current frame's search region are determined by the tracking results of the previous frame. Once the tracker drifts to another distractor in one frame, it will be hard for the tracker to get back to the right target in the following frames.

To make a distinction between same-category objects, handcraft features such as HOG and CN are more effective because they always contain detailed texture information on the objects. As a correlation filter is an excellent model that can utilize HOG and CN features effectively in visual tracking, this motivates us to use a correlation tracker to guide the SiamFC tracker.

The features used in the correlation filter tracker in our framework are HOG and CN features. The HOG features are extracted by calculating the gradient information of an image. CN feature is another classic handcrafted feature that describes the color attributes of an image in a new space. Both the HOG and CN features can be calculated very efficiently. For an image X, its HOG and CN features can be concatenated and denoted as a multichannel feature map $x = [x_1, x_2, x_3 \cdots x_d]$, where each x_d in x represents a matrix with the size of M*N. Circular shift sampling is adopted along the M and N dimensions to generate a large number of training samples. The label value of each sample is generated by a Gaussian distribution according to its Euclidean distance to the target's coordinates. The label value map can be denoted as a M*N matrix y. In the training stage, the correlation filter f can be obtained by minimizing the cost function of ridge regression model in

$$\min \left\| y - \sum_{d=1}^{D} x^d * f^d \right\|^2 + \lambda \sum_{d=1}^{D} \left\| f^d \right\|^2, \tag{5}$$

where $*$ denotes the circular convolution and λ is the regularization parameter to control the model overfitting.

Equation (5) can be solved efficiently in each individual channel by FFT in the Fourier domain. The d-th channel of the filter f can be denoted as follows:

$$F^d = \frac{\bar{Y} \odot X^d}{\lambda + \sum_{k=1}^{D} \bar{X}^k \odot X^k}, \tag{6}$$

where $\odot$ represents element-wise multiplication; the capital letters represent the Fourier transformation of corresponding quantities, and the bar represents the complex conjugation.

In the detection stage, an image patch z is cropped from the new frame according to the tracking results of the previous frame and patch z is considered as the search region. Then, the target's location coordinates of the new frame are achieved by using the filter generated in Equation (6) to process patch z using Equation (7):

$$R = \mathscr{F}^{-1} \left(\frac{\sum_{d=1}^{D} A^d \odot Z^d}{B + \lambda} \right), \tag{7}$$

where R denotes the response score map, and the maximum value on R is considered to be the target's location in the new frame. A and B in Equation (7) represent the numerator and denominator in Equation (6), respectively. As the tracking process continues, both A and B are updated iteratively in each frame by the linear interpolation method, as shown in Equation (9):

$$A_t^d = (1 - \eta) A_{t-1}^d + \eta \overline{Y_t} \odot X_t^d, \tag{8}$$

$$B_t = (1 - \eta) B_{t-1} + \eta \sum_{k=1}^{D} \overline{X_t}^k \bigodot X_t^k, \tag{9}$$

where η represents the learning rate. The linear interpolation update strategy can make the tracker more robust to the target's appearance changes.

3.3. The Evaluation Criterion Based on Correlation Filter and Zero Padding. As discussed in Section 1, the high-level semantic information and large search area used in SiamFC can improve the tracking accuracy. On the other hand, they can lead the tracker to drift more easily. Therefore, we introduced a correlation filter tracker in Section 3.2 to remedy the defect. This correlation filter uses HOG and CN features to do visual tracking; those features contain detailed texture information so that the tracker is more robust when meeting other same-category distractors. In our proposed tracking framework, we used the tracking result from this correlation filter tracker to guide the SiamFC tracker. We designed a criterion to evaluate the validity of the tracking results from the SiamFC tracker and correlation filter tracker. If the evaluation shows that the results from the correlation filter tracker are more reliable, our framework will use it to replace the SiamFC's results. Correspondingly, the search region of the next frame in SiamFC tracker is also adjusted on the basis of the correlation filter tracker. In this way, the SiamFC's defects can be remedied.

The initial groundtruth in the first frame is the only template we can use when visual tracking begins. As the tracking goes on, the target's appearance can be influenced by illumination, occlusion, and many other factors, so the initial groundtruth is also the most reliable reference we can use. We chose the initial groundtruth to train a robust regression model, i.e., another correlation filter, to validate the tracking results. When the visual tracking begins, the initial groundtruth of the image sequence is input into the HOG and CN feature extractors. As discussed in Section 3.2, the output can be denoted as a multichannel feature map $t = [t_1, t_2, t_3 \cdots t_d]$, where each t_d in t represents a matrix with the size of M*N. We still adopted circular shift sampling along the M and N dimensions to generate the samples for the

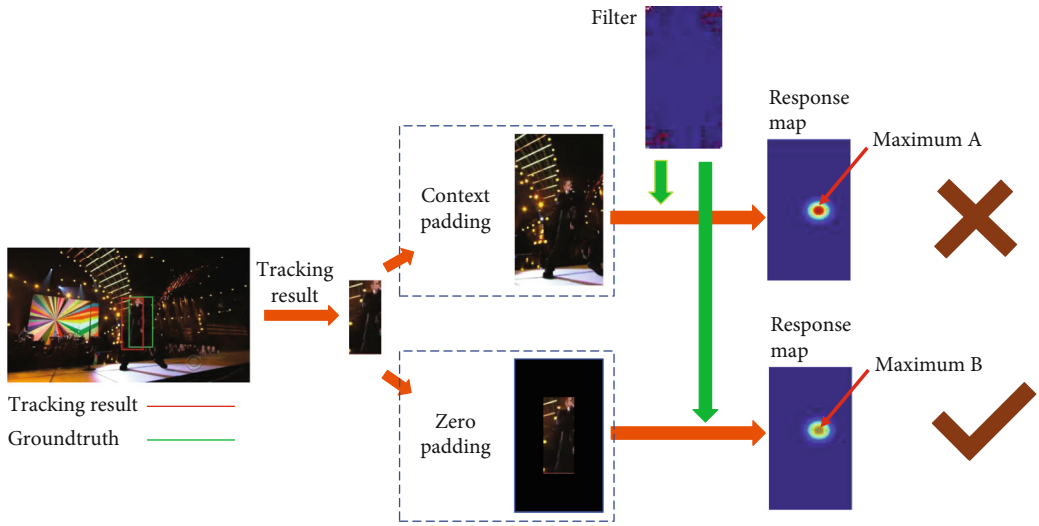

Figure 3: The comparison between the context padding and the zero padding in our proposed tracking framework. The tracking result is obviously not ideal. However, after the context padding, the whole target is still included in the padded bounding box, and it can still generate a significant response in the response map (maximum A). After the zero padding, the target outside the result bounding box is padded as zero. So, the maximum on the response map is compressed (maximum B). On the response map, a redder pixel represents a larger value, so maximum A is bigger than maximum B.

validation regression model. The label value of each sample is produced by a Gaussian distribution according to its Euclidean distance to the initial groundtruth's coordinates. For more details on training the regression model, see Section 3.2.

However, to make the correlation filter more robust and control overfitting, context padding is adopted before inputting the initial groundtruth into the feature extractor. When we use the correlation filter to process the tracking results from SiamFC tracker and CF tracker, the two results should also be padded. The traditional context padding used in CF-based trackers is to find the target's exact location so it can always produce a high response to the filter only if the target is contained in the image patch, whether the target's location is in the center or not. However, our purpose is to evaluate the validation of the two tracking results, so we must make sure that the more reliable tracking result is the one that contains the right target in the center. On the other hand, if the target is contained in the result bounding box, but not in the center, its response to the filter should be compressed.

As shown in Figure 3, the zero-padding method uses 0 to pad around the result bounding box. If the target is not in the center, some parts of the target will be outside the bounding box and the pixels' value in these parts will be set to zero. Then, we can use the correlation filter to process the padded result bounding boxes from the SiamFC and CF trackers. After this procedure, we could get their corresponding response maps. Then, we compared the maximum of each map, and the one that had the larger maximum was considered to be the more reliable tracking result. If this result belongs to the CF tracker, we will use it to adjust the SiamFC tracker.

4. Experiments

We conducted comprehensive experiments on the dataset of Online Tracking Benchmark (OTB) and Temple Color (TC128) to evaluate the effectiveness of our proposed tracking framework. All the experiments were implemented using Google's TensorFlow library. The platform we used to run the experiments is a Dell Alienware DESKTOP-N7K2SPB with a 3.70 Ghz Intel Core i7-8700K CPU and a NVIDIA GeForce GTX 1080Ti GPU. The operating system is 64-bit Windows 10 Professional. Our MFCFSiam tracker can realize real-time tracking with a speed of 16 FPS.

4.1. Benchmark and Evaluation Metric. Both the OTB [13] and TC128 are classic benchmarks designed especially to evaluate the trackers' performance in visual tracking. OTB has three subsets: OTB100 (OTB2015), OTB50, and OTB2013. OTB100 consists of 100 fully labeled video sequences that contain several different tracking scenarios such as scale variation (SV), low resolution (LR), illumination variation (IV), motion blur (MB), out-of-plane rotation (OPR), out of view (OV), background cluttered (BC), deformation (DEF), fast motion (FM), in-plane rotation (IPR), and occlusion (OCC). OTB50 and OTB2013 both consist of 50 video sequences, which are selected from OTB100. TC128 [61] is another famous benchmark used in visual tracking evaluation. It consists of 128 sequences of color images, which contain all kinds of complicated tracking environments. Both OTB and TC128 adopt the one-pass evaluation (OPE) protocol to evaluate the trackers' performance, which means using the tracker to process each image sequence from the beginning to the end only one time and then recording the tracking results to evaluate the tracker's performance.

We followed the metric introduced in OTB and TC128 to evaluate the trackers' performance. This metric contains a success plot and a precision plot, which are based on the Center Location Error (CLE) and Intersection Over Union (IOU), respectively. CLE compares the Euclidean distance

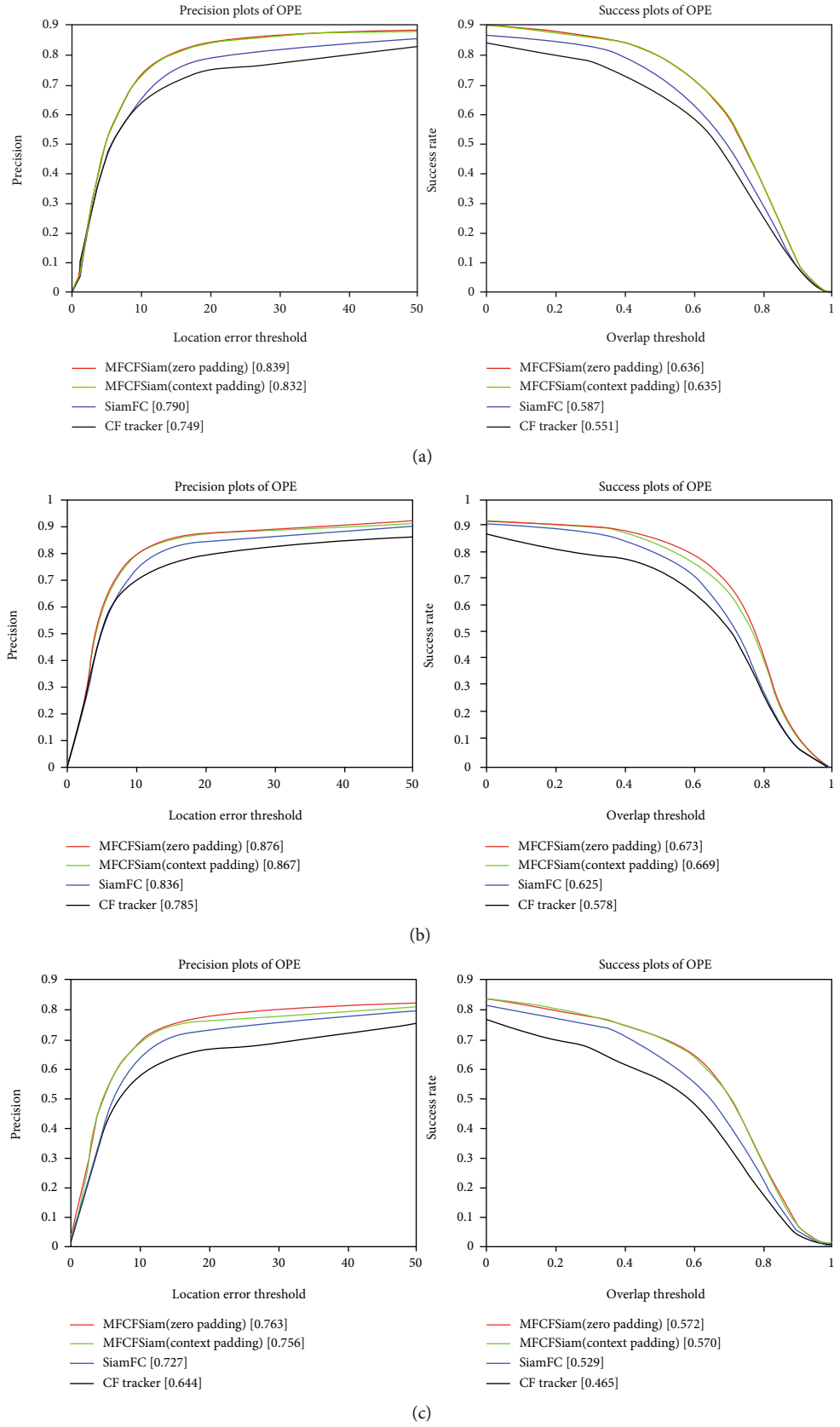

Figure 4: Comparison of the four trackers' performance on OTB dataset. Three plot pairs are results of (a) OTB100, (b) OTB2013, and (c) OTB50. This picture is best viewed on high-resolution displays.

Table 1: The four trackers' average precision values and average AUC values on the OTB dataset.

		OTB100	OTB2013	OTB50
Precision	MFCFSiam (zero padding)	*0.839*	*0.876*	*0.763*
	MFCFSiam (context padding)	**0.832**	**0.867**	**0.756**
	SiamFC	0.790	0.836	0.727
	CF tracker	0.749	0.785	0.644
AUC	MFCFSiam (zero padding)	*0.636*	*0.673*	**0.570**
	MFCFSiam (context padding)	**0.635**	**0.669**	*0.572*
	SiamFC	0.587	0.625	0.529
	CF tracker	0.551	0.578	0.465

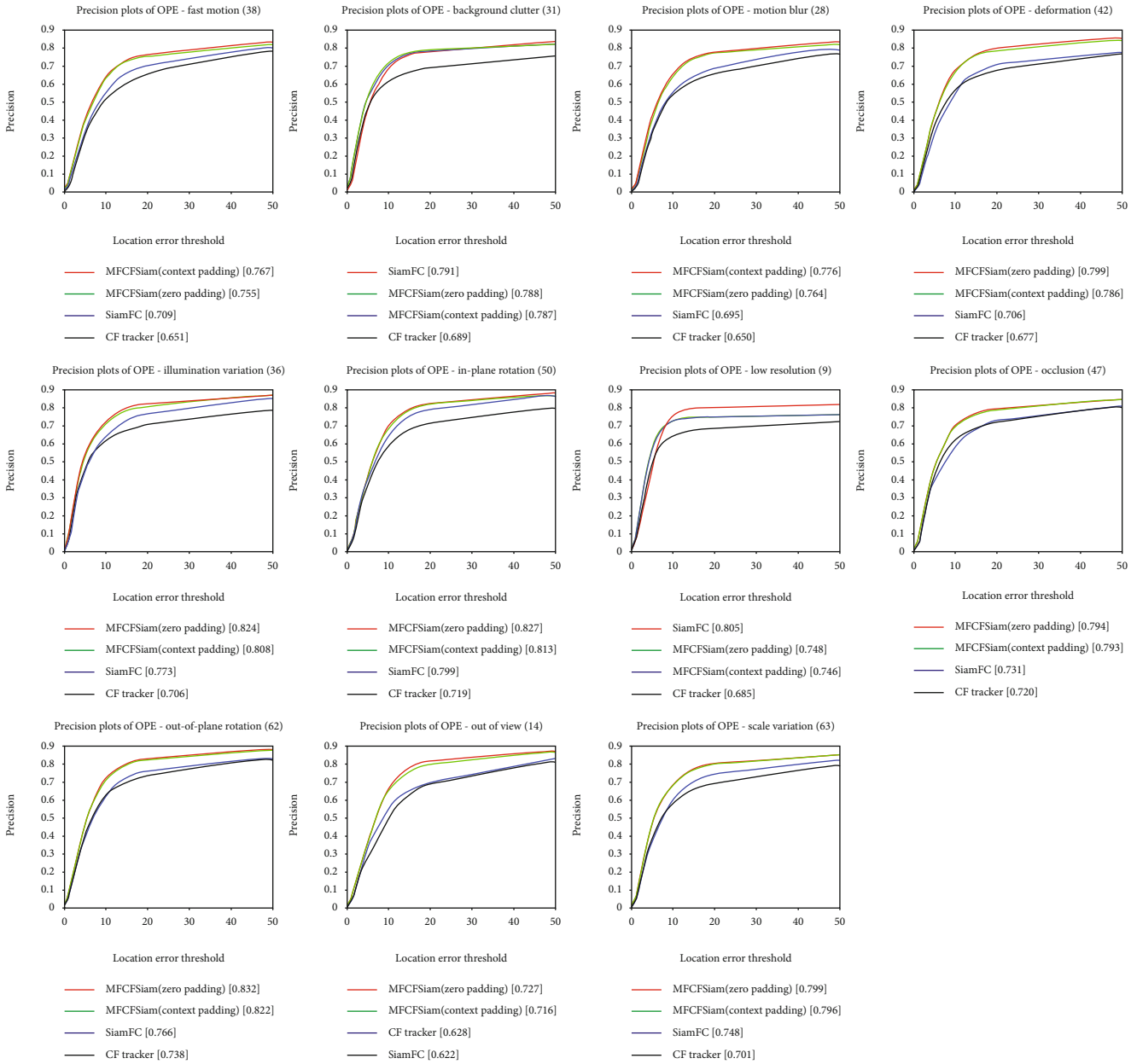

Figure 5: Comparison of MFCFSiam (zero padding) and MFCFSiam (context padding) and two baseline trackers using a precision-plot metric under the 11 tracking scenarios. This picture is best viewed on high-resolution displays.

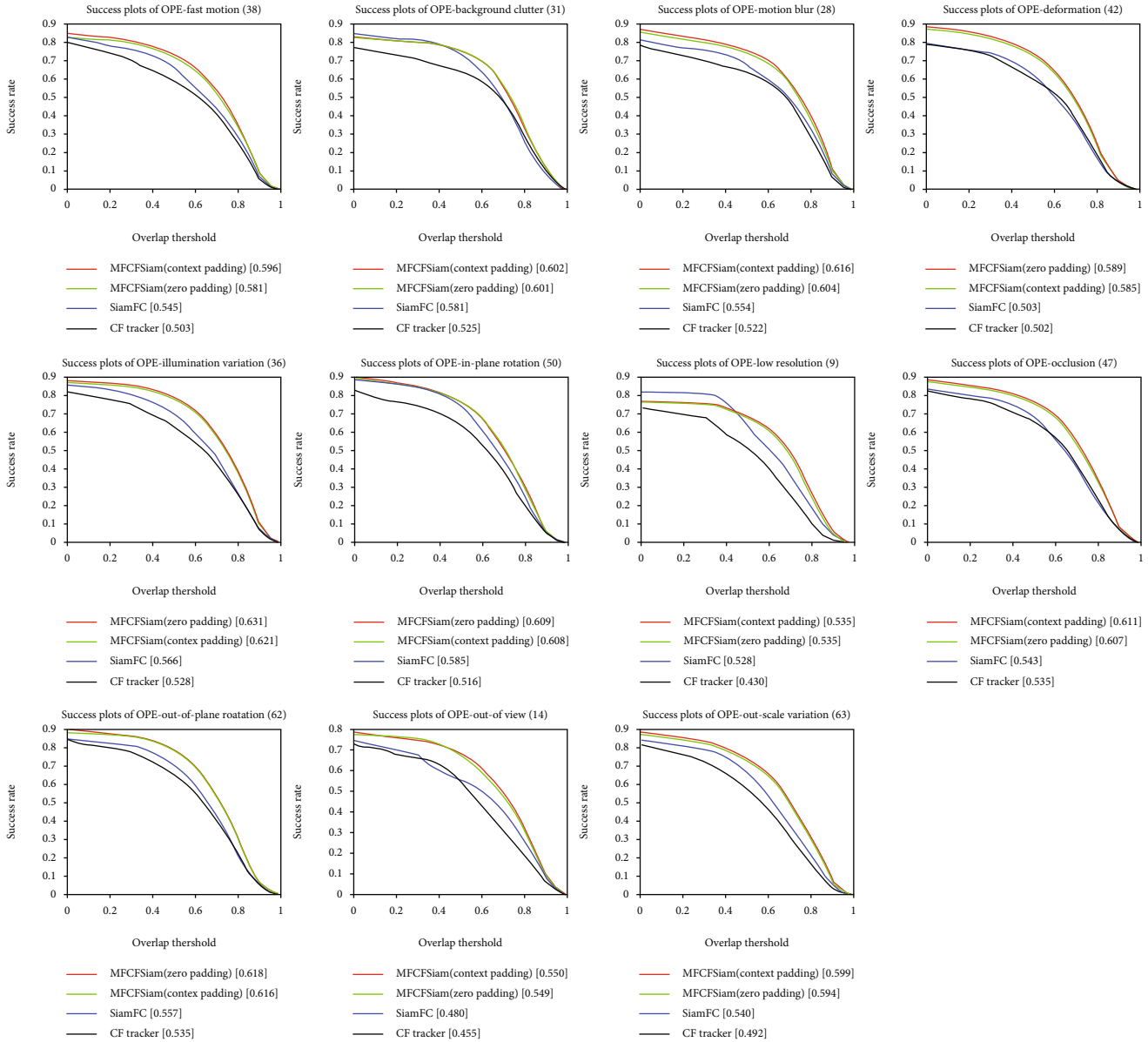

Figure 6: Comparison of MFCFSiam (zero padding) and MFCFSiam (context padding) and two baseline trackers using a success-plot metric under the 11 tracking scenarios. This picture is best viewed on high-resolution displays.

between the center locations of the tracking result provided by the tracker and the corresponding groundtruth of each frame with a given threshold to determine whether the tracking is successful. A smaller Euclidean distance in CLE denotes better tracking. IOU is defined as

$$\mathrm{IOU} = \frac{\mathrm{area}\left(R_T \cap R_G\right)}{\mathrm{area}\left(R_T \cup R_G\right)}, \tag{10}$$

where $\cap$ and $\cup$ are the intersection area and union area between the tracked results (R_T) and groundtruth (R_G), respectively. A parameter called the area under curve (AUC) value is used to represent the tracker's performance in IOU. A bigger AUC value denotes better tracking. So, the precision plot represents the percentage of successfully tracked frames based on the CLE, and the success plot represents the percentage of successfully tracked frames based on the IOU.

4.2. Ablation Experiment. In this section, an ablation experiment is conducted to evaluate the correctness of the tracking strategy proposed in this paper. Four trackers are used in this section: MFCFSiamFC (zero padding) tracker, MFCFSiamFC (context padding) tracker, SiamFC tracker, and CF tracker. The MFCFSiamFC (zero padding) and MFCFSiamFC (context padding) trackers are used to analyze the difference between zero padding and context padding proposed in the validity evaluation criterion in Section 3.3. The SiamFC tracker and CF tracker are used as the baseline trackers. We will run each of these two trackers separately

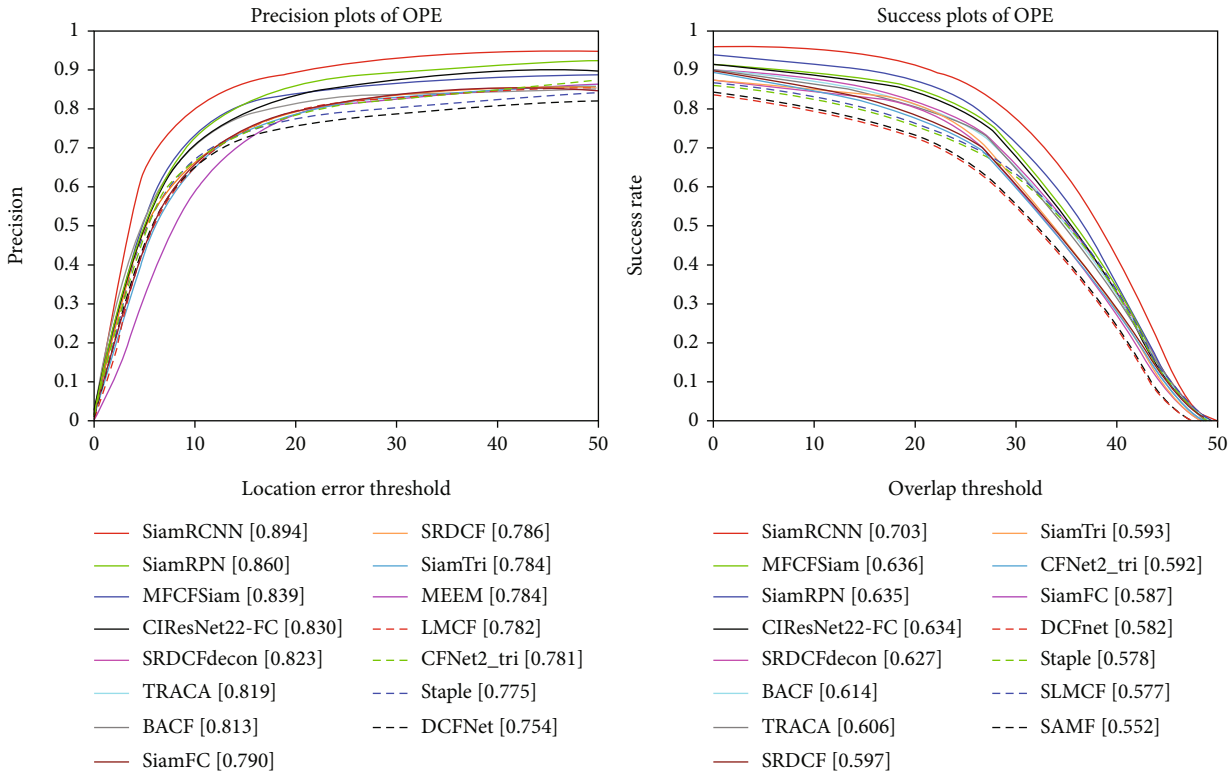

Figure 7: Comparison between our MFCFSiam tracker and 13 several state-of-the-art trackers on OTB100 dataset. This picture is best viewed on high-resolution displays.

on the OTB dataset; then, we can compare the baseline trackers' tracking results with our MFCFSiamFC tracker.

4.2.1. Overall Performance. Figure 4 shows the overall performance of all four trackers on the OTB dataset in terms of the precision based on CLE and success based on IOU. And the four trackers' average precision values and average AUC values on the OTB dataset are summarized in Table 1. The best performance is marked with italic, and the second-best performance is marked with bold. We see that both the MFCFSiamFC (zero padding) tracker and MFCFSiamFC (context padding) tracker outperform the baseline trackers (SiamFC tracker and CF tracker), no matter which dataset is used. This proves the correctness and effectiveness of our tracking theory. The CF tracker alone does not show very good performance; it ranks last in all the six plots in Figure 4 and the SiamFC ranks the third. However, our MFCFSiam tracker, which combines the two trackers into one single framework, has showed obvious better performance in all the six plots. When the CF tracker using HOG and CN features is regard as the guide for the SiamFC tracker, which uses the feature maps from the last layer of the CNN, the advantages of both detailed texture information and high-level semantic information are combined together effectively. That is the reason why our MFCFSiam tracker can make those improvements. What is more, the MFCFSiamFC (zero padding) tracker ranks first in five (OTB100, OTB2013, and precision plot of OTB50) out of the six plots in Figure 4, which proves that the zero-padding method is more robust than the context-padding method in the validity evaluation criterion.

4.2.2. Scenario-Based Performance. A tracker's performance can be influenced by many factors such as deformation, scale variation, and illumination. To evaluate the tracker as a whole, the OTB dataset divides all the video sequences into 11 kinds of tracking scenarios. Each scenario represents one crucial factor that may influence the tracker's performance, i.e., scale variation (SV), low resolution (LR), illumination variation (IV), motion blur (MB), out-of-plane rotation (OPR), out-of-view (OV), background cluttered (BC), deformation (DEF), fast motion (FM), in-plane rotation (IPR), and occlusion (OCC).

We further evaluated and compared the four trackers' performances under the 11 annotated tracking attributes on OTB100 separately. Figures 5 and 6 show the results. We found that in the success plots, our MFCFSiamFC tracker outperformed the two baseline trackers in all 11 different tracking attributes, and in the precision plots, our MFCFSiamFC tracker outperformed the two baseline trackers in nine out of the 11 different tracking attributes (except for BC and LR). The results show that our tracking strategy has made obvious improvements. Because the targets in BC tracking attribute usually have complicated background and the background usually have abundant and complicated texture information, so sometimes our tracker may drift to the background a little. As for the LR tracking attribute, the poor resolution of the targets may lead that

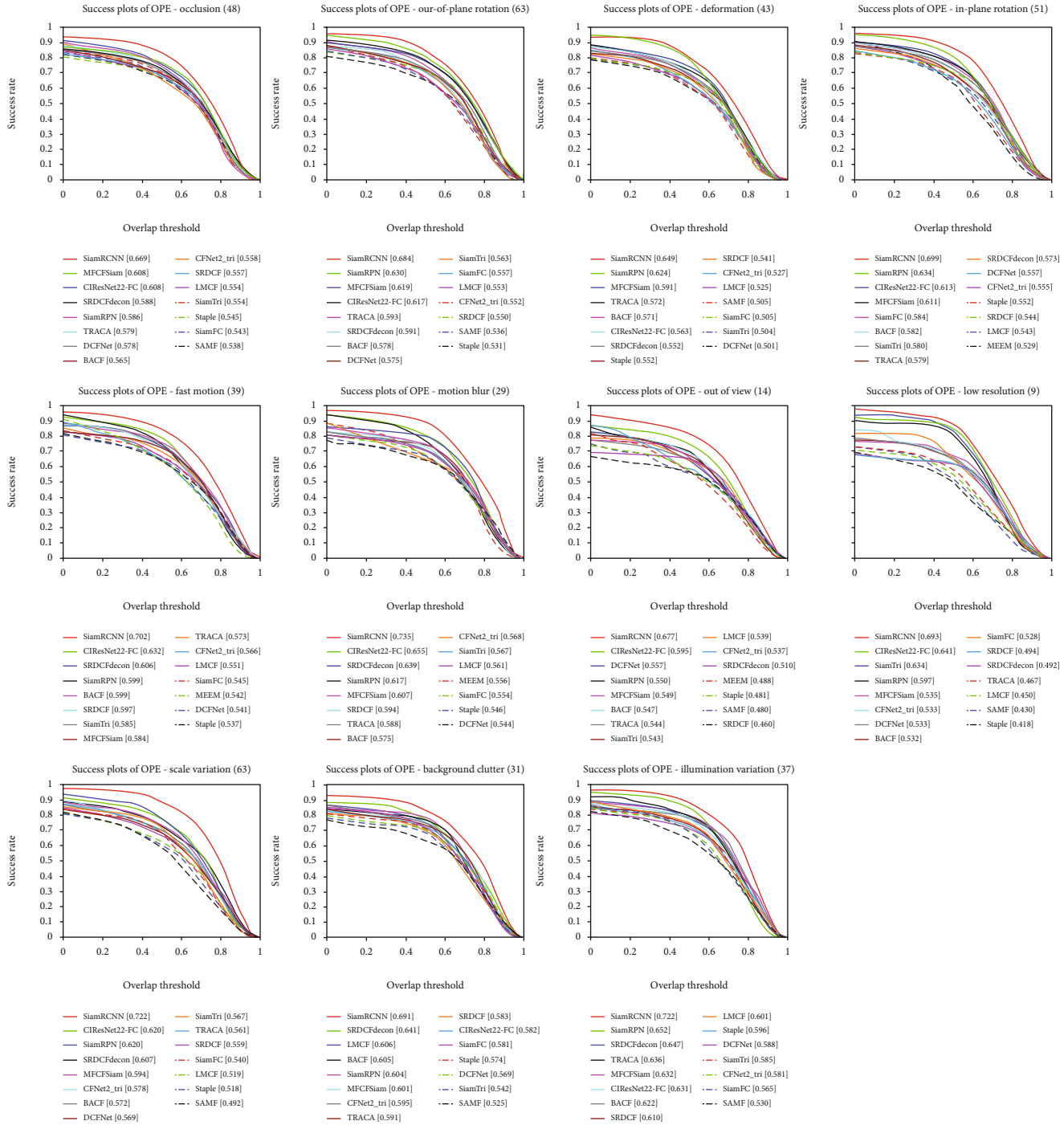

Figure 8: Tracking result of the 15 compared trackers in the success plot of OTB100 in 11 different scenarios. This picture is best viewed on high-resolution displays.

the texture information of the targets and the background are mixed up so that the tracker may drift to the background a little. We think these are the reason that our MFCFSiam tracker's performance is not so good as the SiamFC tracker in the precision plots of BC and LR attributes. Anyway, our tracker has outperformed the baseline trackers in 20 out of the total 22 plots in Figures 5 and 6, This can strongly prove the correctness and effectiveness of our tracking theory. What is more, in the precision plots, the MFCFSiamFC (zero padding) tracker's performance is better than the MFCFSiamFC (context padding) tracker in nine out of the 11 attributes (except FM and MB). In the success plots, the MFCFSiamFC (zero padding) tracker's performance is better than the MFCFSiamFC (context padding) tracker in four out of the 11 attributes (except FM, BC, MB, LR, OCC, OV, and SV). So, in general, the zero padding is more robust than the context padding in the validity evaluation criterion.

Table 2: MFCFSiam's precision ranking in each scenario of all 15 trackers.

	FM	BC	MB	DEF	IV	IPR	LR	OCC	OPR	OV	SV
Precision ranking	8	5	5	3	4	4	5	2	3	5	5

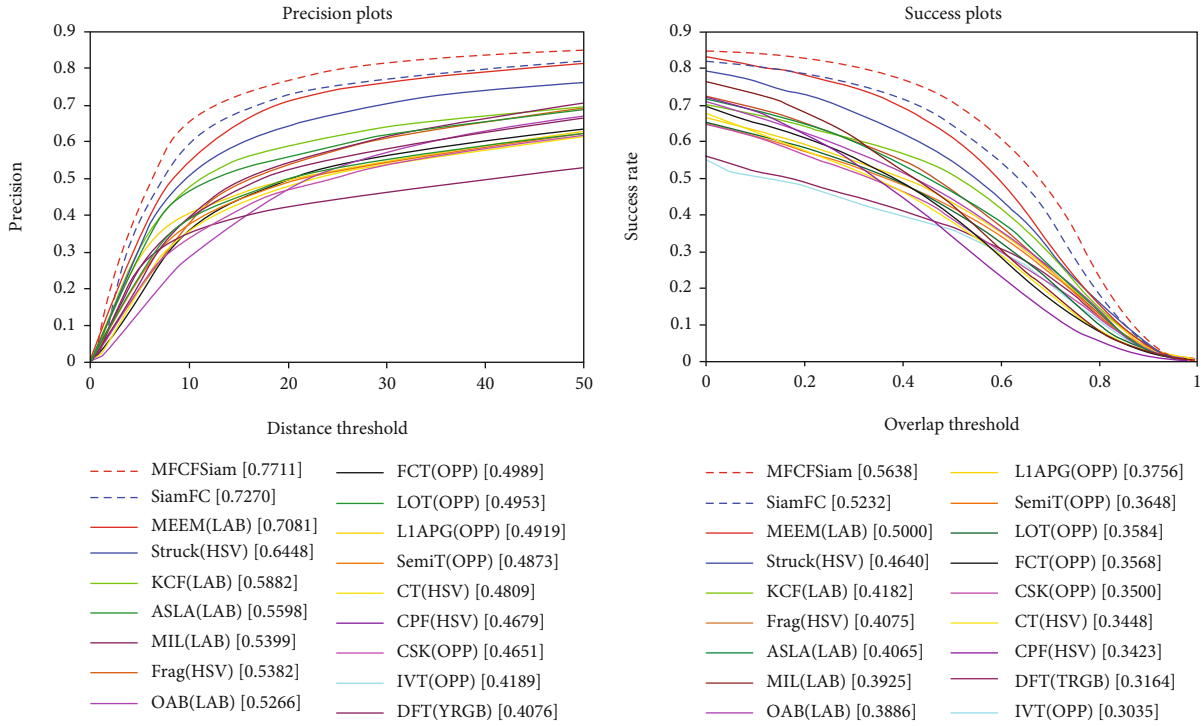

Figure 9: Tracking results of our MFCFSiam tracker and 17 other trackers on the TC128 dataset. This picture is best viewed on high-resolution displays.

4.3. Comparison with State-of-the-Art Trackers on OTB100. To further evaluate the performances of our MFCFSiam tracker, we selected several state-of-the-art trackers that are designed on different tracking strategies. They are all classic and representative in the visual tracking community. Those trackers are as follows: Siam RCNN tracker [58], a fully convolutional Siamese (SiamFC) tracker [38], a Siamese network using triple loss (SiamFC_tri) tracker [62], SiamRPN tracker [54], a context-aware deep feature compression (TRACA) tracker [63], a correlation filter network using triple loss (CFNet2_tri) tracker [62], a cropping-inside residual fully convolutional network (CIResNet-22FC) tracker [64], a staple tracker [65], a spatially regularized discriminative correlation filter (SRDCF) tracker [51], a background-aware correlation filter (BACF) tracker [66], a discriminative correlation filter network (DCFNet) tracker [67], a large-margin correlation filter (LMCF) tracker [68], and scale adaptive kernel correlation filter tracker (SAMF) [69]. The selection of these trackers provided a horizontal comparison to comprehensively evaluate our MFCFSiam tracker. To be concise, we have only listed the results of the precision and success plots on OTB100 because it contains all the sequences that OTB50 and OTB2013 have, so these results were persuasive enough. What is more, the MFCFSiam we adopt here is the one using zero-padding method.

4.3.1. Overall Performances. We compared our MFCFSiam tracker's overall tracking performance on the OTB100 dataset with all the state-of-the-art trackers listed above. Figure 7 shows the results of the precision plot and success plot. Our MFCFSiam tracker achieved the third-best performance (0.839) in the precision plot, inferior only to SiamRCNN (0.894) and SiamRPN (0.860), outperforming all 12 other state-of-the-art trackers. In the success plot, MFCFSiam achieved the second-best performance (0.636), only inferior to the MDNet tracker (0.678) and outperforming all the other 12 state-of-the-art trackers. So, this comparison shows that our MFCFSiam tracker's overall performance on OTB100 is quite competitive and proves the effectiveness of our tracking strategy.

4.3.2. Scenario-Based Performance. We also compared our MFCFSiam tracker's performance with that of the 13 state-of-the-art trackers in 11 different scenarios on OTB100. Figure 8 shows the detailed results of all trackers' performance in the success plot. MFCFSiam's rankings in all 11 scenarios are summarized in Table 2. We can see that our MFCFSiam ranks in the top five in 10 out the 11 scenarios, except for the fast motion (FM) scenario. This proves that our MFCFSiam tracker can handle many complicated tracking environments and shows competitive performance when compared with the state-of-the-art trackers.

Table 3: Precision and area under the curve (AUC) values of the MFCFSiam and eight other trackers on the TC128 dataset (the best and second-best scores are marked with italics and bold, respectively).

	MFCFSiam	MCPF	SRDCF	DeepSRDCF	Staple	BACF	SRDCFdecon	HDT	CNT
Precision	*77.11%*	**76.9%**	69.6%	74.0%	66.8%	66.0%	72.9%	68.6%	44.9%
AUC	*56.38%*	**55.2%**	51.6%	54.1%	50.9%	49.6%	54.3%	48.0%	33.5%

Table 4: Precision and area under the curve (AUC) values of the MFCFSiam and twelve other trackers on the UAV-123 dataset (the best and second-best scores are marked with italics and bold, respectively).

	MFCFSiam	ECO	SRDCF	MEEM	SiamFC	CNT	MUSter	ALSA	DSST	BACF	SAMF	OAB	CFNet
Precision	**71.2%**	*74.1%*	67.6%	62.7%	69.9%	52.4%	72.9%	57.1%	58.6%	65. 4%	59.2%	49.5%	65.1%
AUC	**49.0%**	*52.5%*	46.4%	39.2%	45.7%	36.9%	54.3%	40.7%	35.6%	45.7%	39.6%	33.1%	43.6%

4.4. Comparison with State-of-the-Art Trackers on TC128. Besides from the 100 image sequences of OTB dataset, we also evaluated our MFCFSiam tracker's performance using the TC128 dataset. This dataset has 128 sequences of color images, more than the OTB dataset, and contains some more complicated tracking environments. So, using TC128 helped us to more comprehensively examine MFCFSiam's properties. We also adopted many other classic trackers whose tracking results could be downloaded from the TC128's homepage as comparisons. All those trackers' tracking results of precision plot and success plot are shown in Figure 9. We see that our MFCFSiam tracker (0.7711 and 0.5638) outperformed all the other trackers.

Some other state-of-the-art trackers have also published their average precision values and AUC values for the TC128 dataset. We also summarize these data and compare them with our MFCFSiam tracker. These trackers are as follows: convolutional networks without a training (CNT) tracker [70], a hedged deep tracker (HDT) [50], an adaptive decontamination of spatially regularized discriminative correlation filter (SRDCFdecon) tracker [71], a multitask correlation particle filter (MCPF) [72], a spatially regularized discriminative correlation filter (SRDCF) tracker [51], a background-aware correlation filter (BACF) tracker [66], and convolutional features for correlation filter (DeepSRDCF) tracker [73]. Detailed results are shown in Table 3; the best and second-best performances are marked with italics and bold, respectively. We can see that the MFCFSiam tracker's precision value and AUC value rank first out of the nine trackers.

4.5. Comparison with State-of-the-Art Trackers on UAV-123. UAV-123 [74] is another typical dataset which is widely used for visual tracking. It consists of 123 color image sequences collected by the low-altitude UAV (unmanned aerial vehicle). Compared with the OTB and TC128, UAV-123 contains more long-term image sequences, and the number of frames it contains in total is more than 110 K. The typical characteristic of UAV-123 is that the background in the images is always not so complex, but most sequences contain many changes of view angle, which makes the tracking task quite challenging. So, we believe that the UAV-123 dataset can provide another approach to test the effectiveness of our tracking framework.

Table 5: All the 10 trackers' performance on VOT2018 dataset. The best, second-best, and third-best performances are marked with italics, bold, and bold-italics, respectively.

Tracker	Accuracy ↑	Robustness ↓	EAO ↑
MFCFSiam	**0.589**	0.263	**0.386**
SiamRPN++ [76]	*0.600*	0.234	*0.414*
DeepSTRCF [77]	0.523	0.215	0.345
SiamRPN	***0.586***	0.276	***0.383***
SiamVGG [78]	0.531	0.286	0.348
SA_Siam_R [79]	0.566	0.258	0.337
DSiam [53]	0.512	0.646	0.196
MBSiam	0.529	0.443	0.231
UPDT [80]	0.536	**0.184**	0.378
DRT [81]	0.519	***0.201***	0.356
RCO	0.507	*0.155*	0.376
CPT [82]	0.506	0.239	0.339

The evaluation metric UAV-123 uses are the same as OTB. In this section, we compare our MFCFSiam tracker with 12 classical state-of-the-art trackers, including the efficient convolution operator (ECO) tracker [75], the discriminative correlation filter network (DCFNet) tracker [67], convolutional networks without a training (CNT) tracker [70], and the background-aware correlation filter (BACF) tracker [66]. We summarize all these 13 trackers' performance scores in Table 4 to compare them with our MFCFSiam tracker. The best and second-best scores are marked with italics and bold, respectively. From Table 4, we can find that, among all the 13 compared trackers, our MFCFSiam tracker is only outperformed by the ECO tracker and ranks the second in both plots (71.2% and 49.0%). Compared with the baseline tracker (e.g., SiamFC), our MFCFSiam tracker has made obvious improvements in both the precision plot and the success plot. This can further demonstrate the effectiveness of the tracking framework we proposed.

4.6. Comparison with State-of-the-Art Trackers on VOT2018. VOT (Visual Object Tracking) is another typical dataset designed for visual tracking evaluation. It contains 60 sequences of color images. The property of those images is

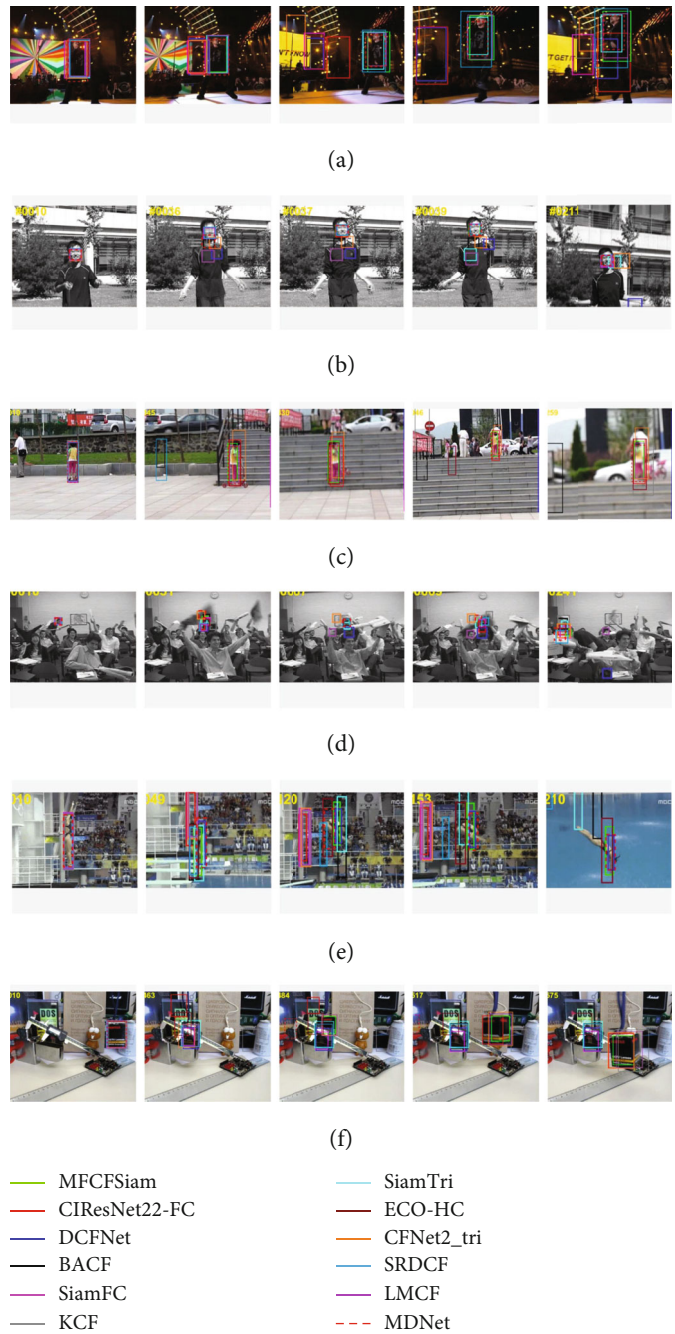

Figure 10: Qualitative tracking results of our MFCFSiam tracker and 11 state-of-the-art trackers on several typical sequences of the OTB. (a–f) The six rows of sequences are (a) singer 2, (b) jumping, (c) girl 2, (d) freeman 4, (e) diving, and (f) box. The color of each tracker is listed at the bottom of the figure.

similar to those of OTB and TC128, but VOT dataset uses an evaluation protocol which is different from the OTB. In the VOT challenge protocol, the tracker will be reinitialized whenever a tracking failure is observed. Three metrics are used to represent the performance of the tracker: accuracy, robustness, and expected average overlap (EAO) score. Accuracy denotes the average overlap between the tracking result and the groundtruth bounding box. Robustness represents how many times tracking failures occur during the tracking process, so a smaller robustness value means a better tracker. EAO represents the average overlap with no reinitialization following a failure. So, using VOT dataset can provide another new approach to show the effectiveness of our tracking framework. In this section, we use the VOT2018 dataset to evaluate the trackers' performance.

We compare our MFCFSiam tracker with eleven other state-of-the-art trackers, and the calculated metrics to represent those trackers' performances are summarized in Table 5. We can see from the table that our MFCFSiam tracker's accuracy (0.589) ranks second, only smaller than the SiamRPN++

(0.600) and outperformed all the other 10 trackers. Our tracker's EAO (0.386) also ranks the second and only inferior to the SiamRPN++ (0.414), outperforming all the other 10 trackers. So, we can see that on the VOT2018 dataset, our MFCFSiam tracker still shows quite competitive performance. This can further demonstrate the effectiveness of the tracking framework we proposed.

4.7. Qualitative Experiments. In this section, we visualize the tracking results of our MFCFSiam tracker and 11 other state-of-the-art trackers on the OTB dataset. Details are shown in Figure 10. The six rows of image sequences are (a–f) singer 2, jumping, girl 2, freeman 4, diving, and box. The color of each tracker's bounding box is listed at the bottom of Figure 10.

The singer 2 sequence is a typical example that contains deformation and background clutter (BC); both the target and the background color are very dark, and the contrast between them is very low, so this makes the trackers tend to drift to the background. Figure 10 shows that our MFCFSiam tracker can constantly capture the right target. The jumping sequence is a typical sequence containing fast motion (FM) and motion blur (MB); the target moves very fast, and his face is blurred when he is jumping, so many trackers drift to the target's body, while our MFCFSiam tracker is among the few trackers that locate the right target. The girl 2 sequence contains many people distractors in the scene, and those distractors usually walk past the right target and block her, so some trackers drift to the background or focus on other distractors because of the occlusion. As shown in Figure 10, the MFCFSiam tracker can always locate the target girl. The freeman 4 sequence is a typical sequence that contains occlusion (OCC); the size of the target is so small that it is very easily blocked. What is more, the occlusion in this sequence is very frequent. Our tracker's performance is good both in precision and scale. The diving sequence is a typical sequence that contains background clutter (BC) and fast motion (FM); the audience in the background can lead the trackers to drift. Before the diver jumped into the river, some trackers had already failed, as shown in the third image; while the diver was entering the river, as shown in the fifth image, only three trackers still held the right target, including the MFCFSiam tracker. The box sequence is another sequence containing typical occlusion (OCC), and the occlusion in this sequence lasts for a few seconds, as shown in the second and third images. When the box appears again later, as shown in the 4th image, many trackers drifted but the MFCFSiam tracker still captured the right target. In conclusion, all six sequences offer evidence of the robustness and effectiveness of our MFCFSiam tracker in challenging tracking scenarios.

5. Conclusions

In this paper, we proposed a novel tracking framework to explore the potential of combining the SiamFC tracker with other CF-based trackers, using the detailed texture features such as the HOG and CN to guide the high-level semantic features in CNN. We also designed an evaluation criterion that uses a correlation filter and the zero-padding method to evaluate the validity of the tracking results. Comparative experiments with other state-of-the-art trackers were conducted on the OTB, TC128, UAV-123, and VOT2018 dataset to verify the effectiveness of our strategy. In the future, our work will mainly focus on keeping optimize the evaluation criterion of tracking results. We believe this can provide a meaningful tool for combining more different trackers.

Data Availability

The data used to support the findings of this study are available from those websites: http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html, https://cemse.kaust.edu.sa/ivul/uav123, and https://www.dabi.temple.edu/~hbling/publication/TColor-128.pdf.

Conflicts of Interest

The authors declare no conflict of interest.

Authors' Contributions

C.L. constructed the tracking framework, performed the experiments, and wrote the original manuscript. Q.X., K.Z, and Z.M. analyzed and interpreted the experiment results and provided suggestions about the experiments and revision of this manuscript.

References

[1] X. Wang, "Intelligent multi-camera video surveillance: a review," *Pattern Recognition Letters*, vol. 34, no. 1, pp. 3–19, 2013.

[2] B. Maurin, O. Masoud, and N. P. Papanikolopoulos, "Tracking all traffic: computer vision algorithms for monitoring vehicles, individuals, and crowds," *IEEE robotics & automation magazine*, vol. 12, no. 1, pp. 29–36, 2005.

[3] J. Lien, E. M. Olson, P. M. Amihood, and I. Poupyrev, *RF-Based Micro-Motion Tracking for Gesture Tracking and Recognition*, 2019.

[4] N. Wang, J. Shi, D.-Y. Yeung, and J. Jia, "Understanding and diagnosing visual tracking systems," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3101–3109, Santiago, Chile, 2015.

[5] X. Dong, J. Shen, D. Yu, W. Wang, J. Liu, and H. Huang, "Occlusion-aware real-time object tracking," *IEEE Transactions on Multimedia*, vol. 19, no. 4, pp. 763–771, 2016.

[6] K. Li, F.-Z. He, and H.-P. Yu, "Robust visual tracking based on convolutional features with illumination and occlusion handing," *Journal of Computer Science and Technology*, vol. 33, no. 1, pp. 223–236, 2018.

[7] H. Alismail, B. Browning, and S. Lucey, "Robust tracking in low light and sudden illumination changes," in *2016 Fourth International Conference on 3D Vision (3DV)*, pp. 389–398, Stanford, CA, USA, 2016.

[8] L. Chen, F. Zhou, Y. Shen, X. Tian, H. Ling, and Y. Chen, "Illumination insensitive efficient second-order minimization for planar object tracking," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4429–4436, Singapore, Singapore, 2017.

[9] S. Liu, G. Liu, and H. Zhou, "A robust parallel object tracking method for illumination variations," *Mobile Networks and Applications*, vol. 24, no. 1, pp. 5–17, 2019.

[10] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proceedings of the IEEE international conference on computer vision*, pp. 3074–3082, Santiago, Chile, 2015.

[11] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International journal of computer vision*, vol. 77, no. 1-3, pp. 125–141, 2008.

[12] Z. Zhu, W. Wu, W. Zou, and J. Yan, "End-to-end flow correlation tracking with spatial-temporal attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 548–557, Salt Lake City, UT, USA, 2018.

[13] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: a benchmark," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2411–2418, Portland, OR, USA, 2013.

[14] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: robust tracking via multiple experts using entropy minimization," in *European conference on computer vision*, pp. 188–203, Springer, 2014.

[15] S. Zhang, X. Yu, Y. Sui, S. Zhao, and L. Zhang, "Object tracking with multi-view support vector machines," *IEEE Transactions on Multimedia*, vol. 17, no. 3, pp. 265–278, 2015.

[16] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 8, pp. 1619–1632, 2010.

[17] H. Song, Y. Zheng, and K. Zhang, "Robust visual tracking via self-similarity learning," *Electronics Letters*, vol. 53, no. 1, pp. 20–22, 2016.

[18] B. Ma, J. Shen, Y. Liu, H. Hu, L. Shao, and X. Li, "Visual tracking using strong classifier and structural local sparse descriptors," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1818–1828, 2015.

[19] Y. Xie, W. Zhang, Y. Qu, and Y. Zhang, "Discriminative subspace learning with sparse representation view-based model for robust visual tracking," *Pattern Recognition*, vol. 47, no. 3, pp. 1383–1394, 2014.

[20] Y. Sui, S. Zhang, and L. Zhang, "Robust visual tracking via sparsity-induced subspace learning," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 4686–4700, 2015.

[21] Y. Wu, B. Shen, and H. Ling, "Visual tracking via online nonnegative matrix factorization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 3, pp. 374–383, 2013.

[22] T. Zhang, S. Liu, N. Ahuja, M.-H. Yang, and B. Ghanem, "Robust visual tracking via consistent low-rank sparse learning," *International Journal of Computer Vision*, vol. 111, no. 2, pp. 171–190, 2015.

[23] D. Yuan, X. Lu, D. Li, Y. Liang, and X. Zhang, "Particle filter re-detection for visual tracking via correlation filters," *Multimedia Tools and Applications*, vol. 78, no. 11, pp. 14277–14301, 2019.

[24] H. Zhang, S. Hu, X. Zhang, and L. Luo, "Visual tracking via constrained incremental non-negative matrix factorization," *IEEE signal processing letters*, vol. 22, no. 9, pp. 1350–1353, 2015.

[25] M. Danelljan, F. Shahbaz Khan, M. Felsberg, and J. Van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1090–1097, Columbus, OH, USA, 2014.

[26] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2009.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[28] L. Zhang, L. Lin, X. Liang, and K. He, "Is faster R-CNN doing well for pedestrian detection?," in *European conference on computer vision*, pp. 443–457, Springer, 2016.

[29] H. Jiang and E. Learned-Miller, "Face detection with the faster R-CNN," in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pp. 650–657, Washington, DC, USA, 2017.

[30] J. Li, X. Liang, S. Shen, T. Xu, J. Feng, and S. Yan, "Scale-aware fast R-CNN for pedestrian detection," *IEEE transactions on Multimedia*, vol. 20, no. 4, pp. 985–996, 2017.

[31] F. Milletari, S.-A. Ahmadi, C. Kroll et al., "Hough-CNN: deep learning for segmentation of deep brain regions in MRI and ultrasound," *Computer Vision and Image Understanding*, vol. 164, pp. 92–102, 2017.

[32] Z. Cai and N. Vasconcelos, "Cascade r-cnn: delving into high quality object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6154–6162, Salt Lake City, UT, USA, 2018.

[33] M. Duan, K. Li, C. Yang, and K. Li, "A hybrid deep learning CNN–ELM for age and gender classification," *Neurocomputing*, vol. 275, pp. 448–461, 2018.

[34] H. Li, Y. Li, and F. Porikli, "Deeptrack: learning discriminative feature representations online for robust visual tracking," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1834–1848, 2015.

[35] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4293–4302, Las Vegas, NV, USA, 2016.

[36] H. Nam, M. Baek, and B. Han, "Modeling and propagating cnns in a tree structure for visual tracking," 2016, http://arxiv.org/abs/1608.07242.

[37] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," *International conference on machine learning*, pp. 597–606, 2015.

[38] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *European conference on computer vision*, pp. 850–865, Springer, 2016.

[39] Z. Chi, H. Li, H. Lu, and M.-H. Yang, "Dual deep network for visual tracking," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 2005–2015, 2017.

[40] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2544–2550, San Francisco, CA, USA, 2010.

[41] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *European conference on computer vision*, pp. 702–715, Springer, 2012.

[42] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 3, pp. 583–596, 2014.

[43] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Long-term correlation tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5388–5396, Boston, MA, USA, 2015.

[44] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *British Machine Vision Conference*, Nottingham, 2014.

[45] D. Yuan, W. Kang, and Z. He, "Robust visual tracking with correlation filters and metric learning," *Knowledge-Based Systems*, no. article 105697, 2020.

[46] T. Xu, Z.-H. Feng, X.-J. Wu, and J. Kittler, "Joint group feature selection and discriminative filter learning for robust visual object tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 7950–7960, Seoul, Korea, 2019.

[47] D. Yuan, X. Shu, and Z. He, "TRBACF: learning temporal regularized correlation filters for high performance online visual object tracking," *Journal of Visual Communication and Image Representation*, vol. 72, article 102882, 2020.

[48] Z. Huang, C. Fu, Y. Li, F. Lin, and P. Lu, "Learning aberrance repressed correlation filters for real-time uav tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2891–2900, Seoul, Korea, 2019.

[49] D. Yuan, N. Fan, and Z. He, "Learning target-focusing convolutional regression model for visual object tracking," *Knowledge-Based Systems*, no. article 105526, 2020.

[50] Y. Qi, S. Zhang, L. Qin et al., "Hedged deep tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4303–4311, Las Vegas, NA, USA, 2016.

[51] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proceedings of the IEEE international conference on computer vision*, pp. 4310–4318, Santiago, Chile, 2015.

[52] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. Torr, "End-to-end representation learning for correlation filter based tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2805–2813, Honolulu, HI, USA, 2017.

[53] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, and S. Wang, "Learning dynamic siamese network for visual object tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1763–1771, Venice Italy, 2017.

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

[55] A. Lukezic, J. Matas, and M. Kristan, "D3S-A discriminative single shot segmentation tracker," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7133–7142, 2020.

[56] Y. Yu, Y. Xiong, W. Huang, and M. R. Scott, "Deformable Siamese attention networks for visual object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6728–6737, 2020.

[57] H. Fan and H. Ling, "Siamese cascaded region proposal networks for real-time visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7952–7961, Long Beach Canada, 2019.

[58] P. Voigtlaender, J. Luiten, P. H. Torr, and B. Leibe, "Siam r-cnn: visual tracking by re-detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6578–6588, 2020.

[59] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, pp. 91–99, 2015.

[60] D. Li, F. Porikli, G. Wen, and Y. Kuai, "When correlation filters meet siamese networks for real-time complementary tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 2, pp. 509–519, 2019.

[61] P. Liang, E. Blasch, and H. Ling, "Encoding color information for visual tracking: algorithms and benchmark," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5630–5644, 2015.

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

[63] J. Choi, H. Jin Chang, T. Fischer et al., "Context-aware deep feature compression for high-speed visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 479–488, Salt Lake City, UT, USA, 2018.

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

[65] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. Torr, "Staple: complementary learners for real-time tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1401–1409, Las Vegas, NA, USA, 2016.

[66] H. Kiani Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1135–1143, Venice Italy, 2017.

[67] Q. Wang, J. Gao, J. Xing, M. Zhang, and W. Hu, "Dcfnet: discriminant correlation filters network for visual tracking," 2017, http://arxiv.org/abs/1704.04057.

[68] M. Wang, Y. Liu, and Z. Huang, "Large margin object tracking with circulant feature maps," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4021–4029, 2017.

[69] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *European conference on computer vision*, pp. 254–265, Springer, 2014.

[70] K. Zhang, Q. Liu, Y. Wu, and M.-H. Yang, "Robust visual tracking via convolutional networks without training," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1779–1792, 2016.

[71] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Adaptive decontamination of the training set: a unified formulation for discriminative visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1430–1438, Las Vegas, NA, USA, 2016.

[72] T. Zhang, C. Xu, and M.-H. Yang, "Multi-task correlation particle filter for robust object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4335–4343, Honolulu, HI, USA, 2017.

[73] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 58–66, Santiago, Chile, 2015.

[74] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for UAV tracking," in *European Conference on Computer Vision (ECCV16)*, Amsterdam, The Netherlands, 2016.

[75] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg, "Eco: efficient convolution operators for tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6638–6646, Honolulu, HI, USA, 2017.

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

[77] F. Li, C. Tian, W. Zuo, L. Zhang, and M.-H. Yang, "Learning spatial-temporal regularized correlation filters for visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4904–4913, Salt Lake City, UT, USA, 2018.

[78] Y. Li and X. Zhang, "SiamVGG: visual tracking using deeper siamese networks," 2019, http://arxiv.org/abs/1902.02804.

[79] A. He, C. Luo, X. Tian, and W. Zeng, "A twofold Siamese network for real-time object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4834–4843, Salt Lake City, UT, USA, 2018.

[80] G. Bhat, J. Johnander, M. Danelljan, F. Shahbaz Khan, and M. Felsberg, "Unveiling the power of deep tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 483–498, Munich, Germany, 2018.

[81] C. Sun, D. Wang, H. Lu, and M.-H. Yang, "Correlation tracking via joint discrimination and reliability learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 489–497, Salt Lake City, UT, USA, 2018.

[82] M. Che, R. Wang, Y. Lu, Y. Li, H. Zhi, and C. Xiong, "Channel pruning for visual tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, 2018.