remote sensing

MDPI

Article

Online Semantic Subspace Learning with Siamese Network for UAV Tracking

Yufei Zha [1,2,†], Min Wu [2,*,†], Zhuling Qiu [2], Jingxian Sun [1], Peng Zhang [1] and Wei Huang [3]

[1] National Engineering Laboratory for Integrated Aero-Space-Ground-Ocean Big Data Application Technology, School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China; yufeizha@nwpu.edu.cn (Y.Z.); jingxiansun@mail.nwpu.edu.cn (J.S.); zh0036ng@nwpu.edu.cn (P.Z.)

[2] Aeronautics Engineering College, Air Force Engineering University, Xi'an 710038, China; mengshu@mails.ccnu.edu.cn

[3] School of Computer and information Engineering, Jiangxi Normal University, Nanchang 330006, China; n060101@e.ntu.edu.sg

* Correspondence: jmyuan@mail.nwpu.edu.cn

† These authors contributed equally to this work.

Received: 25 December 2019; Accepted: 14 January 2020; Published: 18 January 2020

Abstract: In urban environment monitoring, visual tracking on unmanned aerial vehicles (UAVs) can produce more applications owing to the inherent advantages, but it also brings new challenges for existing visual tracking approaches (such as complex background clutters, rotation, fast motion, small objects, and realtime issues due to camera motion and viewpoint changes). Based on the Siamese network, tracking can be conducted efficiently in recent UAV datasets. Unfortunately, the learned convolutional neural network (CNN) features are not discriminative when identifying the target from the background/clutter, In particular for the distractor, and cannot capture the appearance variations temporally. Additionally, occlusion and disappearance are also reasons for tracking failure. In this paper, a semantic subspace module is designed to be integrated into the Siamese network tracker to encode the local fine-grained details of the target for UAV tracking. More specifically, the target's semantic subspace is learned online to adapt to the target in the temporal domain. Additionally, the pixel-wise response of the semantic subspace can be used to detect occlusion and disappearance of the target, and this enables reasonable updating to relieve model drifting. Substantial experiments conducted on challenging UAV benchmarks illustrate that the proposed method can obtain competitive results in both accuracy and efficiency when they are applied to UAV videos.

Keywords: UAV tracking; semantic subspace; siamese network; occlusion detection

1. Introduction

Unmanned aerial vehicles (UAVs) with visual calculating capabilities (e.g., detection, tracking, navigation , semantic segmentation , and remote sensing) have played an important role in urban environment monitoring, and this has accelerated with the numerous amount of low-cost UAVs [1–5]. Different from traditional static vision systems, visual tracking on UAVs can produce more applications owing to the inherent advantages (such as being easy to deploy, large view scope, and maneuverability). However, it also brings new challenges for existing visual tracking approaches, such as complex background clutters, rotation, fast motion, small objects, and realtime issues due to camera motion and viewpoint changes. To tackle the above problems, in this paper we propose a robust and fast tracker for UAV tracking tasks under capricious scenarios.

Recently, Siamese network-based trackers have led to great performances in some popular UAV tracking databases and competitions [6–8]. More specifically, the semantic similarity of the image pair is learned offline on the external massive video dataset ILSVRV2015 [9] through different backbone architectures (such as AlexNet [10], VGGNet [11], and ResNet [12]). Unlike the traditional hand-crafted features (such as HOG [13,14] and CN [15]), these CNN features include high-level semantic information and are able to verify the target from the background/clutter. At the same time, the transfer capabilities of the features across datasets enable the tracker to locate the unseen target. When the tracking is on-the-fly, only a single forward network needs to be conducted without any backward propagation for the high speed.

Despite Siamese network-based trackers having achieved such significant progress, they still have some limitations [16–19]. (1) The output score only measures the similarity of the input pair, but some instance-specific details of the target itself are lost, which is shown in Figure 1a. The network is usually trained to learn the discriminative ability of the categories in the offline dataset. This enables the less sensitive CNN features to be able to identify the object that has similar attributions or semantic information with the real target. (2) The fixed CNN features can adapt to the temporal variations of the target. It is not realistic to update the whole model to fit the target online, due to the large number of parameters and rare training data. The varied targets illustrated in Figure 1b during the tracking procedure confuses the tracker's adaptability, which leads to tracking failure. (3) Lack of occlusion/disappearance detection mechanism. Occlusion and disappearance, shown in Figure 1c are challenges for the tracker that is a forward loop and considers a region as the target in any frame. When the target has occluded or disappeared in the scene, the false result severely disrupts the tracking accuracy.

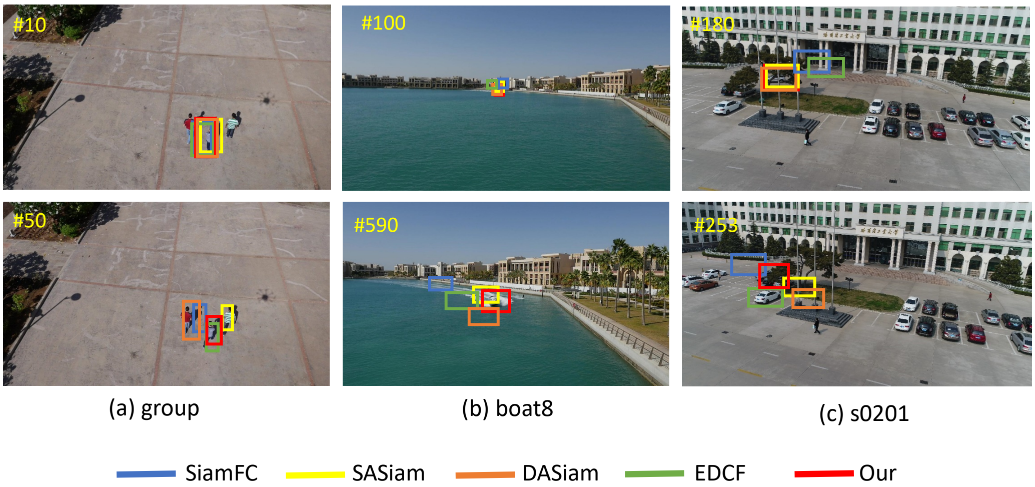


Figure 1. Comparison of the proposed method with the state-of-the-art Siamese network-based trackers (SiamFC [20], SASiam [21], DASiam [22], and EDCF [19]) on the group and *boat8* sequences from the UAV123 [6], and *s0201* sequence from UAVDT [7]. These methods that have been adopted with the Siamese network deal with various challenges (such as the distractor, object variations, and partial occlusions). The proposed method performs favorably against these state-of-the-art trackers.

In order to address the above problems, various methods have been proposed to enhance the discriminative ability of CNN features learned by the Siamese network [18,19,21]. Wang et al. [19] adopted UNet [23] to decode the local fine-grained details of the target during the training of the Siamese network. Unlike how the Encoder–Decoder architecture made the learned CNN features more discriminative than that obtained by only the Siamese network, SASiam [21] introduced two independent Siamese networks to describe the target. One branch was used to learn the similarity between the model and search region, and the other branch was utilized to learn the appearance representation of the target. Unfortunately, these methods trained the model offline on the image recognition and detection dataset [9], while the tracking target is unseen in the training dataset. On the other hand, Guo et al. [18] proposed a dynamic module that is integrated into the Siamese network for enhancing the object and suppressing the background online, but this may not work when the occlusion happens.

Unlike the existing methods, a semantic subspace of the target was designed to learn the principle information online by the shared Siamese network for the UAV tracking in this study. More specifically, the logistical loss was employed to train a filter to encode the fine-grained details of the target, and these are helpful to identify the distractor from the background/clutter. Unlike the traditional linear dimension reduction, the input was derived from the shared Siamese network, which was used to verify the target and background so the high-level semantic subspace helped to obtain the discriminative features. Additionally, the pixel-wise subspace response could be used to detect occlusion or disappearance of the target, and this enabled reasonable updates to relieve model shifting.

We conducted experiments on UAVDT [7] and UAV123 [6] datasets, and verify the reliability of the algorithm through numerous experiments. The performances of our method have achieved 49.9% and 72.0% in the AUC and DP on the UAVDT dataset [7], which is competitive with state-of-the-art methods. The main contributions of this paper are:

- A semantic subspace module is designed to be integrated into the Siamese network tracker to encode the fine-grained details for UAV tracking.
- Online learning of the semantic subspace is conducted and optimized when the tracking is on-the-fly.
- Occlusion/disappearance detection is implemented in terms of the semantic subspace response to make the model updates reasonable.

Before the proposed method and optimization are introduced in Section 3, the related works have firstly been illustrated in Section 2. The experiments and conclusion are in Sections 4 and 6, respectively.

2. Related Works

In this section, we will discuss the closely related tracking methods with our work. A comprehensive review on visual tracking can be found in the literature [24].

2.1. Siamese Network Tracking

In the visual tracking community, similarity learning with convolutional neural networks (CNN) has attracted lots of attention, because its powerful representation can deal with the intra-class variation effectively [20,25]. The CNN features of both the template and the search region are extracted simultaneously by the Siamese network, and they are correlated to obtain a response map, whose maximum indicates the location of the target.

Bertinetto et al. [20] proposed a logistical loss to train the network on a dataset for object detection in videos for similarity learning, which was then used to find the target in the search region online. Then, the traditional pyramid-like operation for scale estimation was replaced by the SiamRPN [16] tracker, which employed a region proposal sub-network to refine the target states by regression. Unlike Wang et al. [26] who utilized the prior attention, spatial residual attention, and channel attention to enhance the features' discriminative ability, Guo et al. [18] added an online dynamic network to adapt to appearance variation and suppress the background in terms of previous frames. Recently, a residual module [27] and spatial aware sampling strategy [17] was designed for a deeper and wider backbone network to achieve better results.

These trackers have successfully enhanced the SiamFC [20] tracker, but unfortunately, less attention is paid to the exploiting of the potential feature-level characteristics in the temporal domain. The motivation of this study is to extract a low-rank semantic principle for the verification process when tracking is on-the-fly.

2.2. Subspace Tracking

The subspace representation [28] is an effective method for visual tracking. The incremental visual tracking (IVT) [28] pursuits the basis of the target by online principal component analysis (PCA), and the reconstructed error is utilized to locate the target in the current frame. It is beneficial for dealing with illumination and clutter, but sensitive to some more complicated situations (e.g., partial occlusion) because of the linear ability of the model. To address this problem, sparse representation that is robust when used for image corruptions, In particular when there is an occlusion [29], was employed to describe the target in the visual tracking field.The L_1 tracker [30] composed the model of a linear combination of the dictionary templates, but was expensive for calculation. Xiao et al. [31] proposed to determine the coefficients of the representation by the L_2 regularized least square method and achieved satisfying results without loss of accuracy. Additionally, the manifold is a non-linear low-dimensional representation, and is popularized by locally linear embedding [32]. Ma et al. introduced a manifold regularized to make better use of the unlabeled data in the correlation filter [33] and convolutional neural networks [34].

Unlike how these methods construct the subspace of the target by considering both the unlabeled and labeled data simultaneously, our method works by learning a non-linear low-dimensional representation of the target by a convolutional layer online on the semantic features.

3. Materials and Method

The similarity between the target and the template can be learned offline by the Siamese network that is a Y shape. Unfortunately, this embedding feature focuses on the consistency of the input pair and neglects the intrinsic structure of the target, and this makes the tracker be influenced easily by the complicated background/clutter, In particular for similar distractors. Thus, it is critical to obtain the local fine-grained details for target representation. In this study, we propose online semantic subspace learning with a shared Siamese network for UAV tracking. The pipeline of the proposed method is shown in Figure 2.

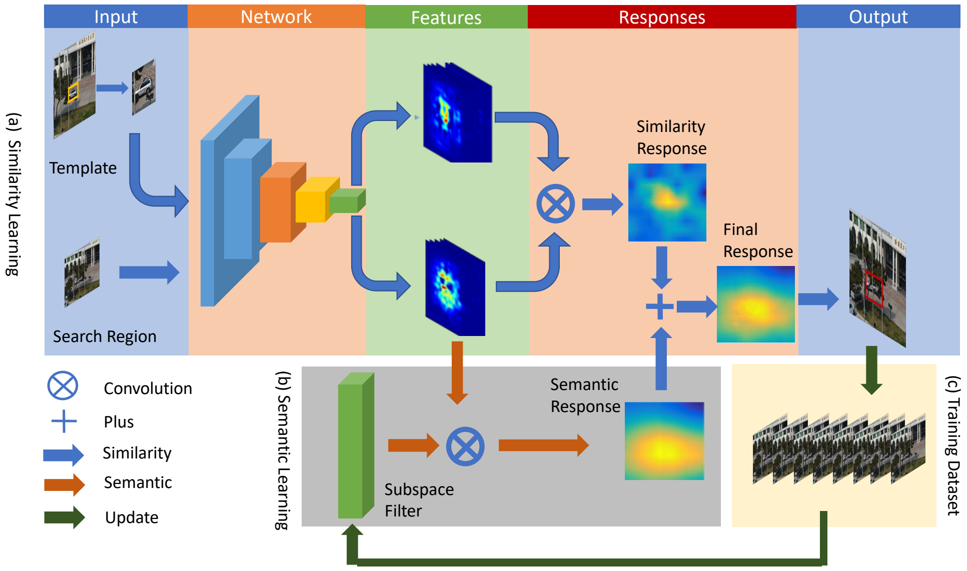


Figure 2. The pipeline of the proposed method. (**a**) The similarity response that is achieved by convoluting the CNN features driven from the Siamese network, where the inputs are the initial template and search region. (**b**) The semantic response that is obtained by convoluting the subspace filter and the search region. The final response is combined with these two responses (similarity and semantic responses) to indicate the location of the target. (**c**) The tracking results are then fed into the training dataset that is used to update the subspace filter online. Here, the blue, orange, and green lines represent similarity learning, semantic learning, and updating flows, respectively.

In the proposed method, there are two branches—one is the traditional Siamese network [20] shown in Figure 2a, and the other one is the semantic subspace learning, illustrated in Figure 2b.

The input of the Siamese network is a patch pair (i.e., initial template and the search region) and the output (similarity response) is achieved by convoluting their respective CNN features derived from the shared network. On the other hand, the semantic response is obtained by convoluting the CNN features of the search region and the learned subspace filter online, and this will be described in detail in Section 3.2. The final response is combined with these two responses (similarity response and semantic response) and is used to indicate the location (the red bounding box) of the target in the current frame. Figure 2c illustrates the training dataset, which was gathered from the tracking results. In particular, the pixel-wise semantic response can be used for occlusion/disappearance detecting, and this benefits the evaluation of the qualities of the training dataset, which is a basic part of learning the subspace filter.

3.1. Similarity Learning by Siamese Network

The similarity between the template and search region is measured by the CNN features trained offline on the public dataset. We adopted the fully convolutional neural network [20] that is composed of five convolutional layers to extract the CNN feature. During the offline training, the final layer, followed by the convolutional layers, is the loss function that aims to learn the similarity measurement, and the formula is written as follows [20]:

$$\mathcal{L}(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} [\frac{1}{|D|} \sum_{u \in D} \log(1 + \exp(-y(u)v(u)))], \tag{1}$$

where $\boldsymbol{\theta}$ is the network model. Here, u is the position which is ranged by the search region D, and the y and v are the label and the obtained response, respectively, that is convoluted by the CNN features of the template and search region. The goal of training is to increase the score of the target and decrease the score of the background. In the offline training phase, when the u is a part of the target, it is regarded as positive, and the corresponding y is set as $+1$. Then, the v is larger if the $\mathcal{L}(\boldsymbol{\theta})$ is getting smaller. In contrast, when the u is a part of the background, it is regarded as negative, and the corresponding y is set as -1. In this situation, if the $\mathcal{L}(\boldsymbol{\theta})$ is smaller, the v must be smaller.

Given the template z and the search region x, the extracted CNN features by the shared network can be denoted as $f(z)$ and $f(x)$. The value of v in Equation (1) can be calculated as [20]:

$$v = f(x) \otimes f(z), \tag{2}$$

where $\otimes$ is the convolution operation. The loss function in Equation (1) enables v to approach the label y.

3.2. Semantic Learning by Subspace Filter

Semantic learning is used to capture the local fine-grained detail of the target through a subspace filter. The input is the CNN feature extracted by the shared network described in Section 3.1, so it can represent the target with the high-level semantic layer, whereas the subspace filter is designed to encode the fine-grained detail of the target online to adapt the appearance variation temporally when tracking is on-the-fly.

3.2.1. Online Learning

In this study, the subspace filter is followed by the shared network illustrated in Figure 3 to learn the low-dimensional representation for the high-level semantics of the target. The high-level semantics features that are extracted by the shared network are utilized to learn the subspace filter.

Then, the generated and labels are both fed into the logistical loss function to minimize the loss so as to optimize the model. The loss function can be denoted as:

$$\mathcal{L}(\boldsymbol{\omega}) = \frac{1}{N}\sum_{i=1}^{N} \log\left(exp(-y_i(\boldsymbol{\omega} \otimes f(\boldsymbol{x}_i)) + 1)\right) + \frac{1}{2}\boldsymbol{\omega}^T * \boldsymbol{\omega}, \tag{3}$$

where $\boldsymbol{\omega}$ is the parameter of the subspace filter, and the $f(\boldsymbol{x}_i)$ denotes the extracted CNN features of the sample $\boldsymbol{x}_i$. Differently to the offline training, the label y_i is generated in terms of the tracking results in the online learning phase. More specifically, the value of the response y_i is set to $+1$, when the corresponding location in the tracking result represents the target. The other value of the response y_i is set to -1. N is the number of the samples randomly selected to be trained. The $\otimes$ and T are the convolution and matrix transpose operations, respectively.

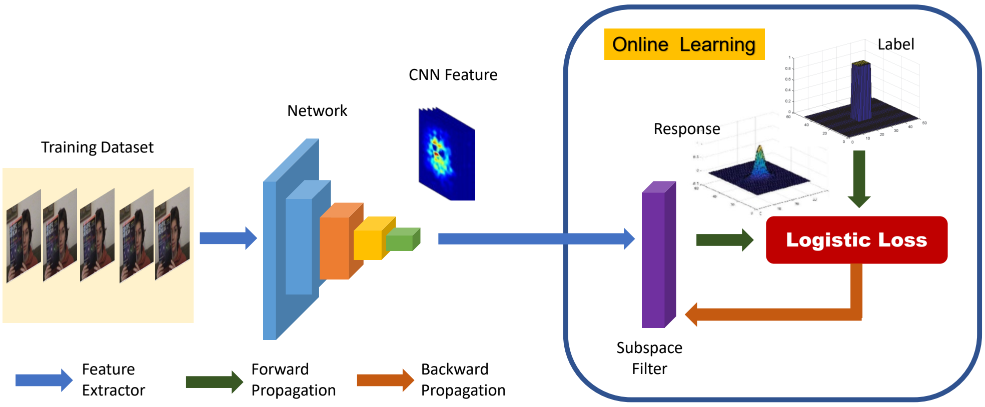


Figure 3. Online learning of semantic subspace filter. The training data are fed into the network to extract the CNN feature for the high-level semantics of the target. These features are then used as the input of the subspace filter that is trained under the supervised signal (e.g., the label) online. The logistic loss is employed as the last layer of the network to make the generated response close to the expected one, which benefits identification of the target from the background. The blue, green, and orange lines represent feature extraction, forward propagation, and backward propagation, respectively.

The training dataset is important for obtaining the subspace filter. This benefits the enhancement of the discriminative ability of the learned features because the subspace filter is driven from these samples. The false-positive sample will influence the model to decrease the discriminative ability of the learned filter. During the tracking procedure, only the tracking result whose response is greater than a threshold is gathered to the training dataset for semantic subspace learning.

3.2.2. Optimization

The parameters of the subspace filter can be optimized by minimizing the loss function based on the training dataset. The loss function in Equation (3) can be decomposed as two parts: the penalty item $\mathcal{L}_1$, and the regularization item $\mathcal{L}_2$.

$$\mathcal{L}(\omega) = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_1(\omega) + \frac{1}{2}\mathcal{L}_2(\omega). \tag{4}$$

Here,

$$\mathcal{L}_1 = \log(e^{-y_i(\omega \otimes f(x_i))} + 1). \tag{5}$$

$$\mathcal{L}_2 = \boldsymbol{\omega}^T * \boldsymbol{\omega}. \tag{6}$$

The derivative of the penalty item $\mathcal{L}_1$ with respect to the variable ω is calculated as follows:

$$\begin{aligned}\frac{\partial \mathcal{L}_1}{\partial \omega} &= \frac{\partial \log(e^{-y_i(\omega \otimes f(x_i))}+1)}{\partial \omega} \\ &= \frac{1}{e^{-y_i(\omega \otimes f(x_i))}+1} \otimes \frac{\partial e^{-y_i(\omega \otimes f(x_i))}}{\partial \omega} \\ &= \frac{-y_i * f(x_i e^{-y_i(\omega \otimes f(x_i))}}{e^{-y_i(\omega \otimes f x_i)}+1}.\end{aligned} \tag{7}$$

The derivative of the regularization item $\mathcal{L}_2$ with respect to the variable ω is as follows:

$$\frac{\partial \mathcal{L}_2}{\partial \omega} = \frac{\partial \omega^T * \omega}{\partial \omega} = 2\omega. \tag{8}$$

The final derivative of the loss function $\mathcal{L}$ is as follows:

$$\frac{\partial \mathcal{L}}{\partial \omega} = \frac{1}{N}\sum_{i=1}^{N} \frac{\partial \mathcal{L}_1}{\partial \omega} + \frac{1}{2}\frac{\partial \mathcal{L}_2}{\partial \omega}. \tag{9}$$

This equation can be expanded as:

$$\frac{\partial \mathcal{L}}{\partial \omega} = \frac{1}{N}\sum_{i=1}^{N} \frac{-y_i * f(x_i)e^{-y_i(\omega \otimes f(x_i))}}{e^{-y_i(\omega * f(x_i))}+1} + \omega. \tag{10}$$

The update of the subspace filter is as follows :

$$\omega' = \omega + \eta * \frac{\partial \mathcal{L}}{\partial \omega}, \tag{11}$$

where ω' is the updated filter, and η is a learning rate. Here, $*$ is a multiplication operation.

The online learning of the semantic subspace filter is summarized in Algorithm 1 as follows:

Algorithm 1 Online semantic learning by the subspace filter.

```
Input:
dataset: $\delta = \{f(x_1), f(x_2), \cdots, f(x_n)\}$;
labels: $\{y_1, y_2, \cdots, y_n\}$;
Subspace filter: $\omega$;
Given the maximum iterations: $iter$.
Output:
Updated subspace filter: $\omega'$;
While( $i < iter$)
    (1)Forward propogation: convoluting these features $\{f(x_1), f(x_2), \cdots, f(x_n)\}$ with the subspace filter parameter $w$ to generate the response $v_i$.
    (2) Backward propagation: minimizing the loss function in terms of the obtained response $v_i$ and the label $y_i$.
    (3) Update: the filter parameter with the learning rate $\eta$ as Equation (11).
end
```

3.2.3. Occlusion Detection

The pixel-wise semantic response indicates the probability that the pixel in the same position belongs to the target, which is illustrated in Figure 4. The CNN features in Figure 4c are extracted from the network shown in Figure 4b with respect to the input patch shown in Figure 4a. Then, these features are convoluted with the learned subspace filter in Figure 4d to achieve the semantic response shown in Figure 4e.

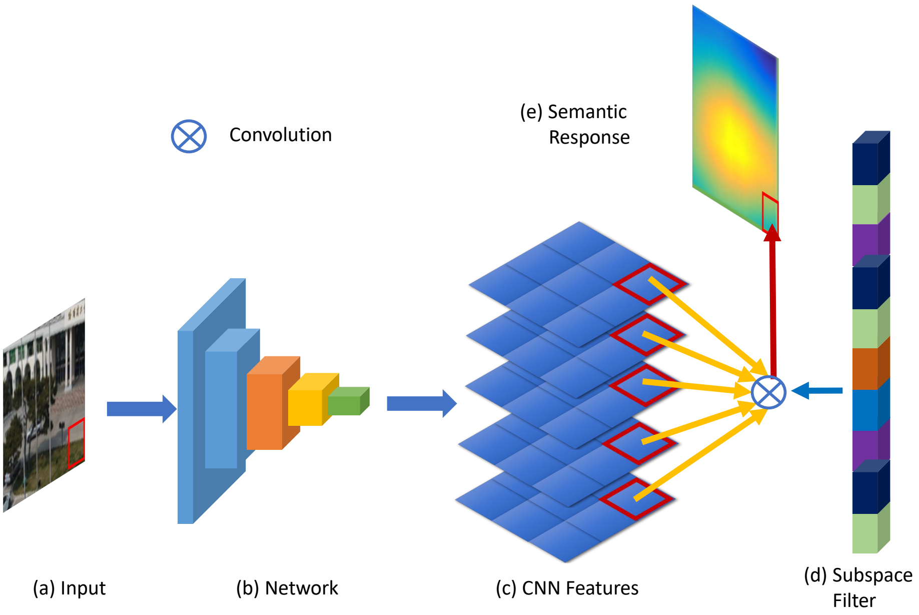


Figure 4. Semantic response. The input in (**a**) is fed into the network (**b**) to obtain the CNN features shown in (**c**). These CNN features are then convoluted by the learned subspace filter in (**d**) to generate the semantic response shown in (**e**). Here, the red bounding box illustrates the corresponding location of one single pixel in the whole operating flow.

The higher the value of the semantic response is, the more likely that the corresponding pixel is one part of the target. If target occlusion/disappearance happens, the semantic response of the corresponding location will be lower than the location where the target has neither occlusion/disappearance. Thus, it is beneficial to indicate which part is the target or not because of the pixel-level response.

3.3. Locating Target by the Combined Response

After obtaining the similarity and semantic responses, they are then combined so as to locate the target accurately, which is shown in this Section. The semantic response represents the probability that each pixel in the search region belongs to the target by the subspace filter, while the similarity response is used to measure the search region with the target patch-by-patch. Thus, the pixel-wise semantic response by the subspace filter and the patch-wise similarity response by the Siamese network are complementary, which improves the tracker.

When tracking is on-the-fly, in the search region, the response v_{sim} obtained by the Siamese network and v_{sem} obtained by the subspace filter can be combined to achieve the final response, v:

$$v = v_{sim} + \lambda v_{sem}, \tag{12}$$

where λ is a super-parameter determined by the experiments to balance these two responses. Scale estimation is performed similarly to pyramid-like searching [35]. The maximum value of the final response is used to locate the target in the current frame. If the maximum value is greater than a threshold, this result will be inserted into the training dataset, while if the maximum value is smaller than another threshold, online updating is performed.

The proposed algorithm can be described as Algorithm 2.

Algorithm 2 Online semantic subspace learning with siamese network tracker.

```
Input:
Video sequence: $\{\boldsymbol{I}_i | i = 1, 2, \ldots, N\}$;
Initial state of the target: $\{p_1, q_1, w_1, h_1\}$;
Given the parameters: $r_{max}, r_{min}, \lambda$;
Output:
Predicted state of the targer in $t$-th frame: $\{p_t, q_t, w_t, h_t\}$;
For   $t = 1 : N$
    If   $t = 1$   Then
    (1) Get the search region $\boldsymbol{x}_1$ according to the inital state $\{p_1, q_1, w_1, h_1\}$ in the frame $\boldsymbol{I}_1$;
    (2) Extract the CNN feature $f(\boldsymbol{x}_1)$ by the Siamese network;
    (3) Construct the training dataset: $\delta = f(\boldsymbol{x}_1)$;
    Else
    (4) Extract the CNN feature of search region $f(\boldsymbol{z}_t)$;
    (5) Calculate the similarity response: $\boldsymbol{v}_{sim} = f(\boldsymbol{x}_t) \otimes f(\boldsymbol{z}_t)$;
    (6) Calculate the semantic response: $\boldsymbol{v}_{sem} = \boldsymbol{\omega} \otimes f(\boldsymbol{z}_t)$;
    (7) Locate the target according to the combined response $\boldsymbol{v}$ in Equation (12);
    (8) Estimate the scale of the target with pyramid-like searching [35];
        If   $max(r) > r_{max}$   Then
        (9) Update the training dataset $\delta = [\delta, f(\boldsymbol{x}_t)]$;
        EndIf
        If   $max(r) < r_{min}$   Then
        (10) Learn subspace filter $\boldsymbol{\omega}$ by the Algorithm 1
        EndIf
    EndIf
EndFor
```

4. Results

In this Section, the proposed method is verified on the two public UAV datasets: UAVDT [7] and UAV123 [6]. Firstly, we describe the implementation details and evaluation criteria in detail. Next, we will compare our approach with state-of-art trackers. For a fair comparison, these trackers were conducted by the available codes provided by the authors with the default parameters.

4.1. Implementation Details

Our method was implemented with Matlab 2017a, and all the experiments were run on a PC equipped with Intel i7 7700 CPU, 32GB RAM, and a single NVIDIA GTX 1070Ti GPU.

The shared network is composed of five convolutional layers. The first two convolutional layers are followed by a pooling layer that uses max-pooling. Except for the fifth convolutional layer, there is a ReLU non-linear activation behind each convolutional layer. During training, batch normalization is used before each ReLU layer to reduce the risk of overfitting. Like the work [20], the network is a pre-trained Alexnet network [10], which is trained offline to measure the similarity between the template and the search region based on the ILSVRC2015 [9], while a filter was followed by the Alexnet network to learn the target's semantic subspace online according to the dataset gathered when tracking is on-the-fly.

When the subspace filter updated, the learning rate η was set to 0.001, and the maximum number of training dataset was set to 10. The score was mapped to 0∼1, and the weight λ was set to 1. When the maximum response of the tracking result was greater than $r_{max} = 0.8$, this result was selected as a sample, which was inserted into the dataset δ. If the maximum response is smaller that the $r_{max} = 0.3$, then the online learning is conducted according to the training dataset.

The test databases were public unmanned aerial vehicle databases: UAVDT [7] and UAV123 [6]. UAVDT [7] is a recently announced database which contains 50 video sequences with nine different attributes. The UAV123 [6] is a database of specialized scenes shot by UAV, which contains 123 videos. In this study, we were able to test the performance of our algorithm in UAV scenarios in these two databases.

4.2. Evaluation Criteria

In this experiment, we employed the criteria in the work [36] to analyze and evaluate the performance of trackers, and only one-pass evaluation (OPE) was adopted as the protocol. Through comparing the labeled states of the target, the results obtained by the proposed algorithm were used to achieve the center position errors and overlaps, which are the basis for the precision plots and success plots. More specifically, the precision plots are the ratio of the frames lower than the predefined threshold and the total number of frames. When the threshold is set to 20, the value of the precision curve is defined as the distance precision rate (DP), which is a rank criterion for precision. The value of the overlap ratio is between 0 and 1 and is calculated by the ratio of the intersection and union between the result and ground truth. This is then used to generate the success plot curve by the ratio of the frames whose overlaps are greater than the predefined threshold.

4.3. Results of UAVDT Dataset

From 10 hours of raw videos, the UAVDT [7] database selected about 80,000 representative frames that were fully annotated with bounding boxes, as well as up to 14 kinds of attributes (e.g., weather condition, flying altitude, camera view, vehicle category, and occlusion) for the single object tracking task.

In this study, we compare the proposed tracker with the most advanced trackers to achieve a comprehensive evaluation. These trackers can be classified as follows:

- Siamese network based trackers: SiamFC [20], EDCF [19], SiamRPN [16], DSiam [22], and SA-Siam [21];
- CNN network based trackers: HCF [37], MDNet [38], and CREST [39];
- Correlaton filter based trackers: SRDCF [40], ASRCF [41], CSRDCF [42], LADCF [43], and BACF [44].

4.3.1. Comparisons to State-of-the-Art Trackers

In the UAVDT [7] dataset, our algorithm is evaluated by three evaluation criteria: AUC, DP, and tracking speed, which are illustrated in Figure 5 and Table 1. In the DP, the scores of all trackers on 20 pixels are 74.0% (SiamRPN), 70.2% (DASiam), 70.2% (ECO), 70.0% (ASRCF), 68.6% (BACF), 68.2% (EDCF), 68.1% (SiamFC), 70.0% (SRDCF), 68.6% (SASiam), 68.2% (LADCF), 68.1% (CSRDCF), 64.9% (CRESR), 62.9% (STRCF) and 60.2% (HCF), respectively. In the AUC, the scores of all trackers are 56.5% (SiamRPN), 48.8% (DASiam), 48.8% (ECO), 46.0% (ASRCF), 46.9% (BACF), 52.0% (EDCF), 52.6% (SiamFC), 45.0% (SRDCF), 38.1% (SASiam), 46.8% (LADCF), 37.9% (CSRDCF), 40.7% (CRESR), 45.2% (STRCF), and 34.5% (HCF), respectively. The results of our method are 49.9% and 72.0% in AUC and DP, respectively, and the tracking speed reached 52 frames per second (FPS). Overall, the proposed algorithm achieved competitive performances in the UAVDT [7], In particular for Siamese network-based algorithms.

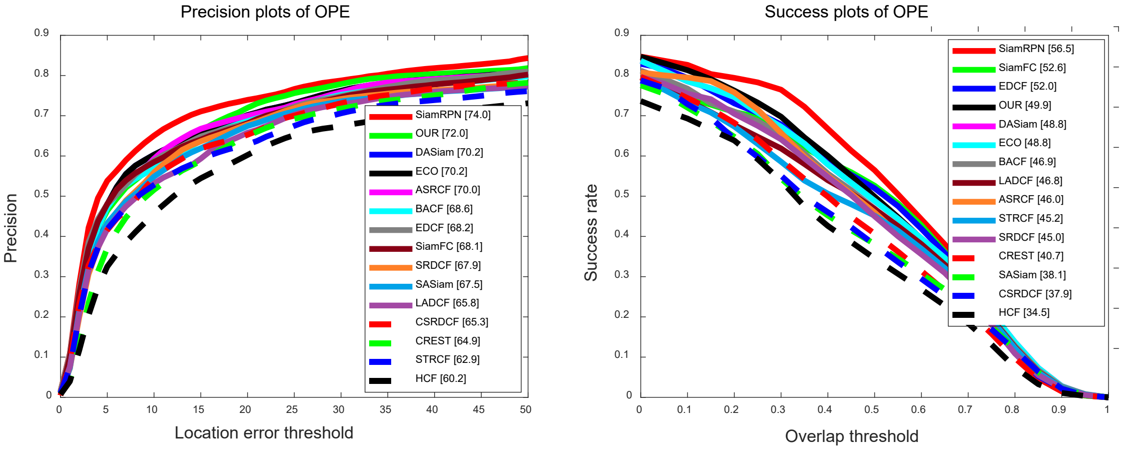


Figure 5. Distance precision and overlap success plots on the UAVDT [7] dataset. These are quantitative results on the 50 benchmark sequences using OPE [45]. The legend of distance precision contains threshold scores at 20 pixels, while the legend of overlap success contains area-under-the-curve scores for each tracker. The proposed algorithm performs favorably against state-of-the art trackers.

Table 1. The comparison results of our method and neural network-based trackers in the UAVDT [7], where AUC and DP were used as evaluation criteria. The red, blue, and green colors represent the first, second, and third scores.

	Trackers	AUC(%)	DP(%)
Siamese Network	SiamFC [20]	52.6	68.1
	EDCF [19]	52.0	70.2
	SiamRPN [16]	56.5	74
	DASiam [22]	48.8	70.2
	SASiam [21]	38.1	67.5
CNN Network	MDNet [38]	46.4	72.5
	HCF [37]	44.7	68.1
	ECO [46]	48.8	70.2
	C-COT [47]	38.1	64.9
	ASRCF [41]	43.7	70.0
Handcrafted Feature	BACF [44]	43.2	68.6
	SRDCF [40]	42.3	67.9
	LADCF [43]	42.2	65.8
	CSRDCF [42]	38.1	55.3
	Staple [48]	39.5	69.7
	KCF [13]	29.0	57.0
	Our	49.9	72.0

Comparison with the Siamese network-based trackers. To analyze the proposed method, we evaluated the five related trackers: SiamFC [20], EDCF [19], SiamRPN [16], DASiam [22], and SASiam [21], and they were developed based on the Siamese network in recent years. SiamFC [20] uses the Siamese network to verify the candidates and target, but the similarity metric is short of identifying the target from the background, especially for the distractor. In replacement of pyramid-like scale searching, SiamRPN [16] directly estimates the scale of the target only once to improve the efficiency. In order to improve the discriminative ability of the CNN features, UNet [19], classification-based network [21], and the online updating strategy [22] were utilized to enhance the traditional Siamese network.

The results shown in Table 1 indicate that the tracker proposed in this paper can achieve excellent scores in terms of two criteria and real-time tracking speed. Compared with the SiamFC [20], our method improves the success rate by 3.9%. The reason for this is that our online method learns the semantic subspace details of the target based on the traditional similarity measurement between the template and search region.

Comparison with CNN network-based trackers. Figure 5 and Table 1 show the comparative results with the state-of-the-art CNN network-based trackers: HCF [37], ECO [46], ASRCF [41], C-COT [47], and MDNet [38]. HCF [37] performs the correlation filter on multi-layer CNN features, but the filter will be polluted when the tracking fails, and this reduces the accuracy level. The core idea of MDNet [38] is to update the network with the tracking results, which slows down the speed. C-COT [47] proposes a continuous convolution operation, which uses an implicit interpolation model on continuous space. The ECO [46] algorithm balances the tracking performance and speed by the sparse update strategy and achieves excellent results in most fashionable datasets. Although both of them show good performance, the speed is only about 1 FPS, and this limits them in the practical system.

Overall, our method outperforms the competitive results with state-of-the-art trackers, can run in real-time, and obtains similar performance, and this makes a good balance between the performance and speed.

Comparison with the handcrafted features-based correlation filter trackers. To analyze the proposed method, we selected some classic and advanced tracking algorithms, such as KCF [13] and Staple [48] for evaluation. KCF [13] is a classic correlation filtering algorithm and builds samples by the cyclic shift in the spatial domain and optimizes the filter in the frequency domain for fast calculation. Owing to the rapid speed, various variants are developed to improve tracking performance. For example, Staple [48] fuses the multi-feature to enhance the robustness of the filter. In particular, we evaluate the five trackers: SRDCF [40], BACF [44], LADCF [43], and CSRDCF [42], which are developed based on the regularized correlation filter in recent years.

The results shown in Table 1 indicate that the tracker proposed in this paper can achieve excellent scores in terms of two criteria. Compared with the latest work, LADCF [43], our method improves the success rate by 7.7% and the precision by 6.2%. Compared with other correlation filters, our method also achieves state-of-the-art results.

4.3.2. Ablation Study

The Siamese network focuses on learning the similarity information between the template and the search region to determine the target. Based on this, we propose to learn the semantic subspace of the target online, and this module is integrated into the Siamese network. The subspace focuses on the fine-grained details of the target, and this helps to suppress the background/clutter. Additionally, the generated pixel-wise response benefits occlusion or disappearance detection of the target to improve accuracy.

In this section, we will analyze each component's contributions to the final performance. To be concise, we denote the algorithm that only concludes the Siamese module as $OSSL_p$, which locates the target by the learned similarity information. $OSSL_q$ is the algorithm with a separate semantic subspace for online learning, and $OSSL_m$ denotes the algorithm that contains the occlusion detection based on $OSSL_q$. In these two methods, the semantic subspace of the target is explored to encode the fine-grained details, and online learning can adapt to the target temporally.

Figure 6 shows the results of trackers with different components. Compared with $OSSL_p$, the AUC score and DP score of $OSSL_q$ increased by 3.3% and 2.7%, respectively. The performance of $OSSL_m$ exceeds $OSSL_q$ by 2.2% in AUC and 2.4% in DP. The AUC and DP scores of our tracker are 49.9% and 72.0% and excels the $OSSL_p$ 2.6% (AUC) and 4% (DP), respectively.

The experimental results show that the single module (Siamese module or subspace module) can follow the target effectively, but the performance is not satisfactory. When the semantic subspace module integrates into the Siamese module, the performance can be improved more. We think the reason is that the semantic information is complementary to the similarity information obtained by the Siamese network.

Figure 7 shows the results of the qualitative comparison. We chose to analyze sequence *S1603* due to its challenge of a similar distractor. It can be seen that $OSSL_p$ lost the target when the distractor appeared, while the $OSSL_q$ tracker failed after the 896th frame due to the increasing distractors. The reason may be that the fixed regularization matrix could not fit the target, and this degraded the discriminating ability of the filter so as not to identify the distractor. While $OSSL_m$ can follow the target longer than the $OSSL_q$, it fails in the 1071th frame. Only our tracker can still track the target accurately in the whole video, even when many distractors appear when the tracking is on-the-fly.

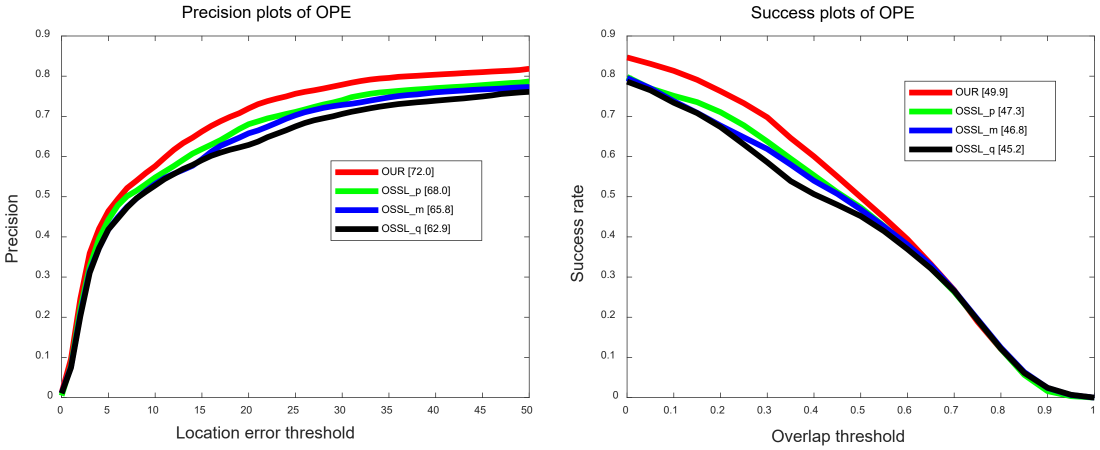


Figure 6. Distance precision and success rate of different functional parts on the UAVDT [7] dataset. Quantitative results on the 50 benchmark sequences using OPE [45]. The legend of distance precision contains threshold scores at 20 pixels, while the legend of overlapping success contains the area-under-the-curve score for each tracker.

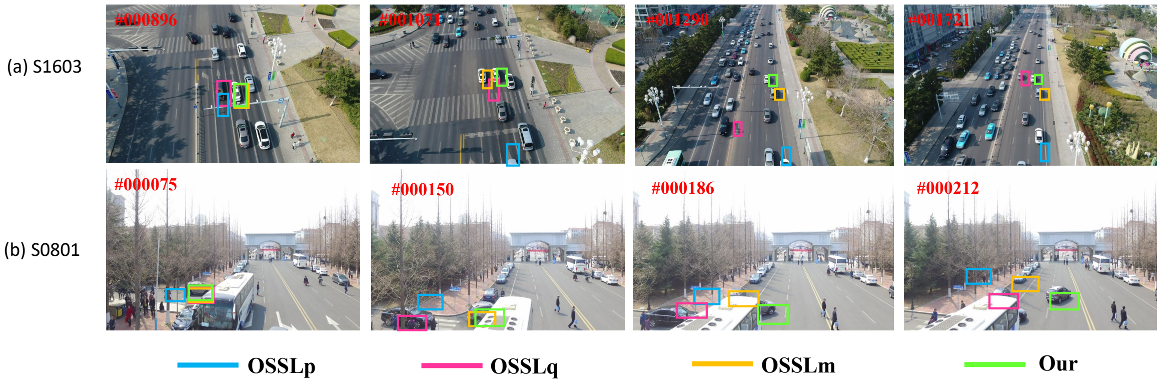


Figure 7. Quantitative analysis results of ablation analysis on challenging image sequences (*S1603* and *S0801*).

The challenge of occlusion appears in the sequence *S0801*, where the target is occluded heavily from the 186th frame. This makes the trackers ($OSSL_p$ and $OSSL_q$) move away from the real target, but due to the tracker with temporal regularization, it can still track accurately. Comparably, our tracker and $OSSL_m$ can still locate the target after the 186th frame, but $OSSL_m$ fails at the 212th frame since it does not constrain the filter temporally.

4.3.3. Attribute Analysis

We chose 9 attributes in the UAVDT [7] dataset to analyze the tracker's performance: background clutter (29), camera motion (30), object motion (32), small object (23), illumination variations (28), object blur (23), scale variations (29), long-term tracking (6), and large occlusion (20) (the number of videos per attribute is appended to the end of each difficulty attribute). Figures 8 and 9 show the results of a one-pass evaluation of these challenging attributes for visual object tracking. From the results, the proposed tracker in this paper performs well under these scenarios.

Different from other algorithms, our online method learns the subspace in the high-level semantics to preserve the fine-grained details of the target, and the generated response can be combined by the similarity response obtained by the Siamese network. In particular, Table 2 shows that our tracker outperforms the baseline algorithm SiamFC [20] by about 7.9% (DP) and 5.6% (AUC) in the case of large occlusion, while excels SiamFC [20] by about 7.9% in DP and 4.1% in AUC in the case of background/clutter.

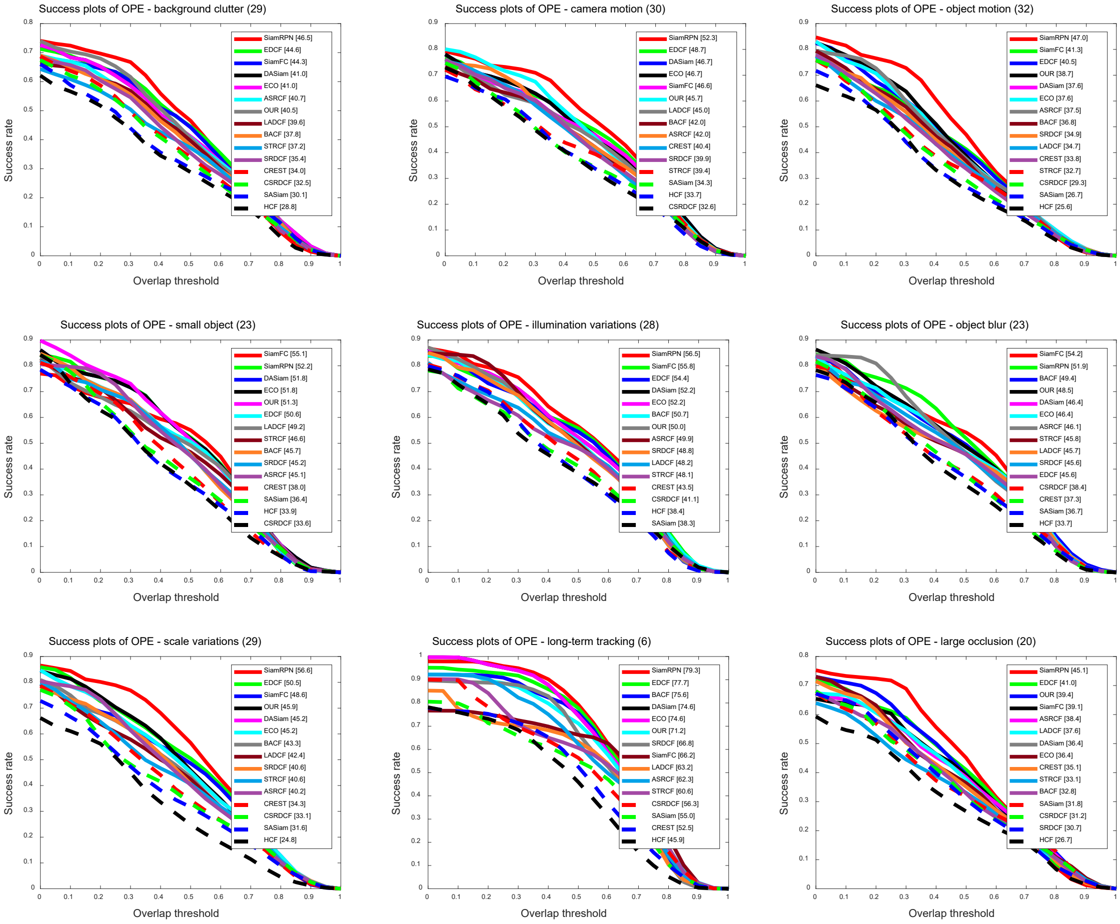


Figure 8. Precision plots based on tracking results on the UAVDT dataset in nine sequence attributes, including: background clutter (29), camera motion (30), object motion (32), small object (23), illumination variations (28), object blur (23), scale variations (29), long-term tracking (6), and large occlusion (20). The legend of distance precision is the threshold scores at 20 pixels. The proposed algorithm performs well against state-of-the-art results.

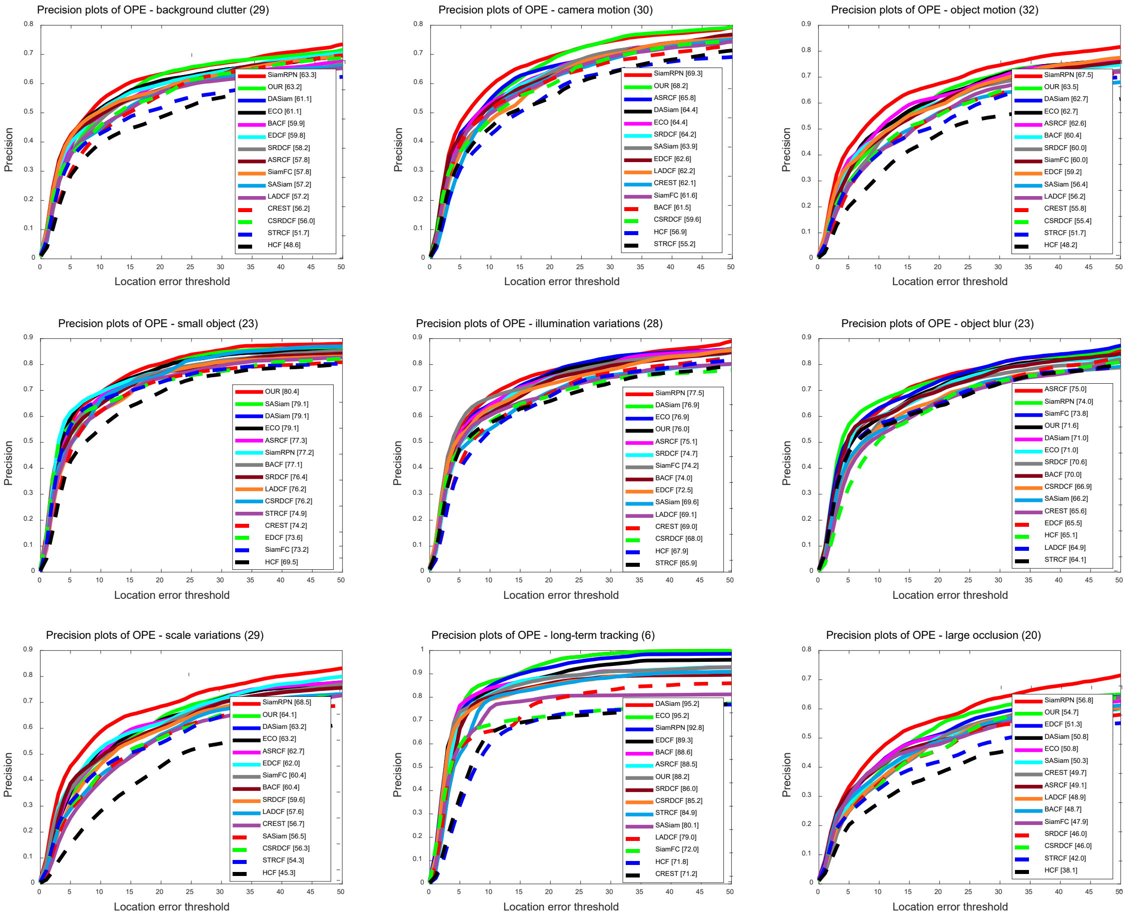


Figure 9. Success plots based on tracking results on the UAVDT [7] dataset in nine sequence attributes, including: background clutter (29), camera motion (30), object motion (32), small object (23), illumination variations (28), object blur (23), scale variations (29), long-term tracking (6), and large occlusion (20). The legend of overlap success is the area-under-the-curve score. The proposed algorithm performs well against state-of-the-art results.

Table 2. Performance comparison of different trackers under large occlusion and background clutter conditions. The distance precision (DP) is threshold scores at 20 pixels. The AUC is the area under the success rate curve score. The red, blue, and green colors represent the first, second, and third scores.

Trackers	Large Occlusion		Background Clutter	
	AUC(%)	DP(%)	AUC(%)	DP(%)
MDNet [38]	38.1	54.7	39.7	63.6
CSRDCF [42]	32.9	46.0	33.0	56.0
EDCF [19]	41.0	51.3	44.6	59.8
SiamRPN [16]	45.1	56.8	46.5	63.3
DASiam [22]	36.4	50.8	41.0	61.1
SASiam [21]	31.8	50.3	30.1	57.2
ECO [46]	36.0	50.8	41.0	61.1
ASRCF [41]	36.0	49.1	40.7	57.8
SiamFC [20]	35.9	47.9	44.3	57.8
LADCF [43]	35.6	48.9	39.6	57.2
CREST [39]	35.1	49.7	34.0	56.2
BACF [44]	33.5	48.7	37.8	59.9
SRDCF [40]	32.7	46.0	35.4	58.2
Staple [48]	32.5	49.6	32.9	59.2
KCF [13]	22.8	34.3	23.5	45.8
OUR	39.4	54.7	40.5	63.2

4.3.4. Qualitative Evaluation

Qualitative evaluation of the proposed algorithm with the other algorithms (including SiamFC [20], EDCF [19], CCOT [47], and HCF [37] in the UAVDT [7] dataset and the results are shown in Figure 10. Next, we will analyze the effects of different trackers in typical videos which contain challenges, such as light change, scale change, background clutter, and rotation.

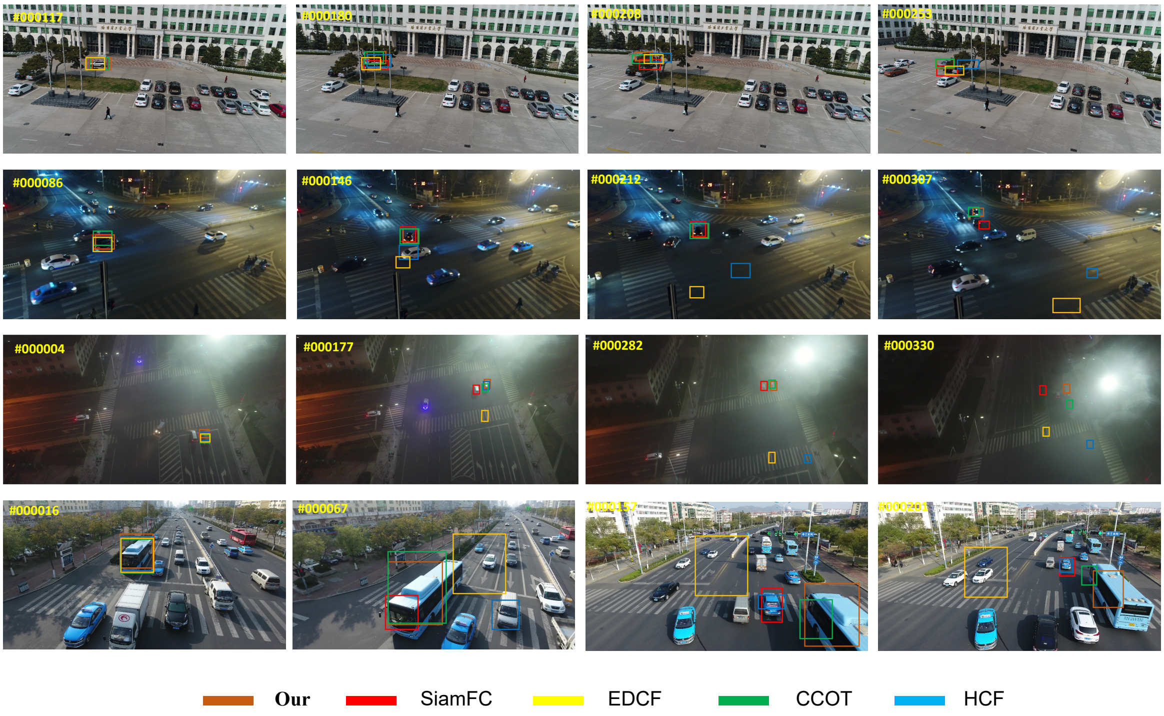


Figure 10. Experimental results of different trackers on challenging sequences (from top–down are *S0201*, *S0301*, *S1306* and *S0602* from the UAVDT [7] dataset). The tracking results of SiamFC [20], EDCF [19], CCOT [47], and HCF [37], as well as the proposed algorithm are denoted by different colored bounding boxes.

S0201: This video sequence is a car passing through a tree, including the target being occluded. We took the 117th, 180th, 219th, and 253rd frames to analyze the tracker's performance. At the 117th frame, each algorithm could achieve accurate target tracking. In the 180th frame, the target was occluded. At this time, the tracking performance of the other algorithms is reduced, except for our algorithm and CCOT [47]. In the 253rd frame, the target is partially occluded and similar interference terms appear around it. At this time, only the proposed algorithm in this paper and CCOT [47] can still achieve tracking. Additionally, the unoccluded part of the target was bounded tightly.

S0301: The sequence is a process in which a car turns and gradually moves away, including the rotation and the change of the target scale, and our algorithm performs well in this video sequence. We intercepted the 86th, 146th, 212nd, and 307th frames to analyze the performance of the tracker. In the 86th frame, the target did not change much, and all trackers could locate the target effectively. From the 146th frame, the target had a large deformation, and only our algorithm, SiamFC [20], and CCOT [47] could follow the target. In the 307th frame, one distractor appeared around the target, then SiamFC [20] could not identify the target from this interference. The experimental results show that our algorithm can effectively deal with deformation and distractors.

S1306: This video sequence is the process of a car passing through street lights, including changes in lighting and complex background clutter. We took the 4th, 177th, 282nd, and 330th frames to analyze the tracker's performance. Because the background was complex and the lighting changed greatly, the EDCF [19] and HCF [37] trackers lost their targets quickly. In the 177th frame, distractors appeared around the target, and the SiamFC [20] algorithm located the wrong target. Finally, only our algorithm and CCOT [47] could track the target.

S0602: The video sequence is a car across the intersection road. It contains the rapid rotation of the camera, the change in the scale of the target, and the similarity interference. We took the 68th, 154th, 196th, and 220th frames to analyze the performance. At the 67th, 157th, and 201st frames, the target had a large deformation. Only our tracker could track the target, although the tracking accuracy level was not high.

4.3.5. Speed Performance

Efficient algorithms are critical to real-time operation for fast-moving unmanned platforms. In our approach, the Alexnet network was used to extract the deep feature of the target. During the online training process, the parameters of the convolutional layer were not updated. When the filter could not achieve the expression of the target, the filter parameters were updated. Frames per second (FPS) of our approach and the related trackers are shown in Figure 11. It can be seen that trackers based on the Siamese network have fast advantages and operate in real-time. The SiamFC [20] has faster speed than ours, but their performances are worse than our tracker.

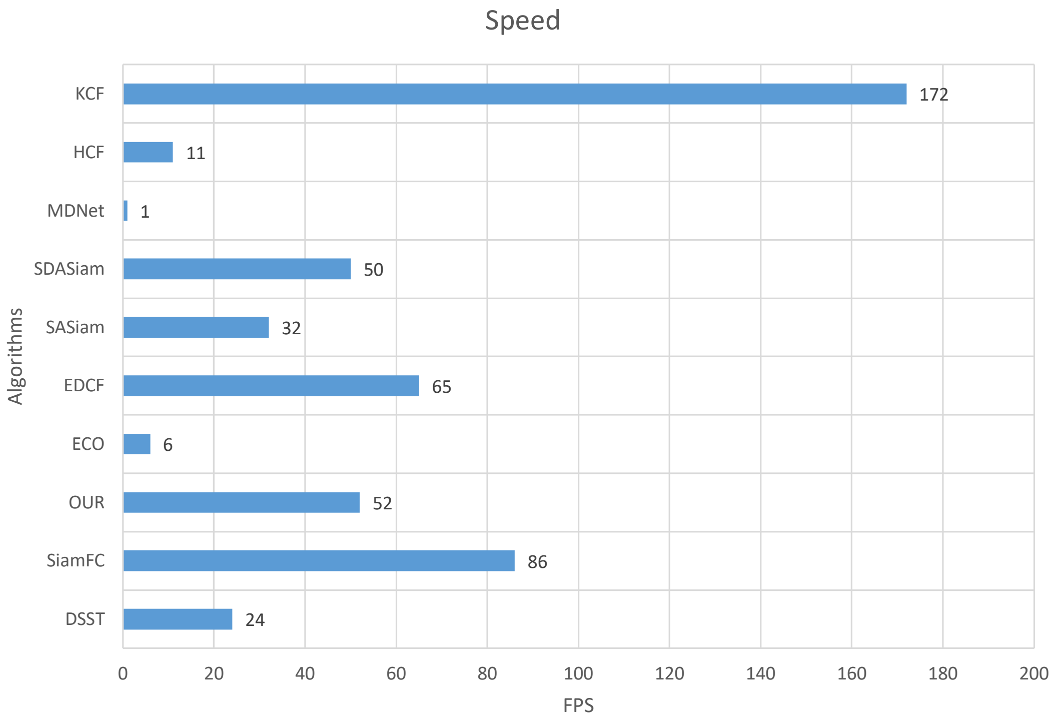


Figure 11. Comparison of speed performance of different trackers.

4.4. Results of UAV123 Dataset

UAV123 [6] is another popular public UAV dataset, containing 123 image sequences with 12 difficulty attributes. It was used to evaluate the overall performance of the tracker. Our tracker was compared to other state-of-the-art trackers, including ECO [46], SRDCF [40], MDNet [38], DASiam [22], SiamRPN [16], MEEM [49], ASRCF [41], BACF [44], SAMF [35], LDES [50], and TLD [51].

It can be seen from Figure 12 that our tracker achieved 70.3% and 53.4% in DP and AUC, respectively, and had a significant improvement in performance. Although MDNet [38] achieved the best score under the UAV123 [6] dataset, the complicated operation made its tracking speed not applicable in the real-time scenario.

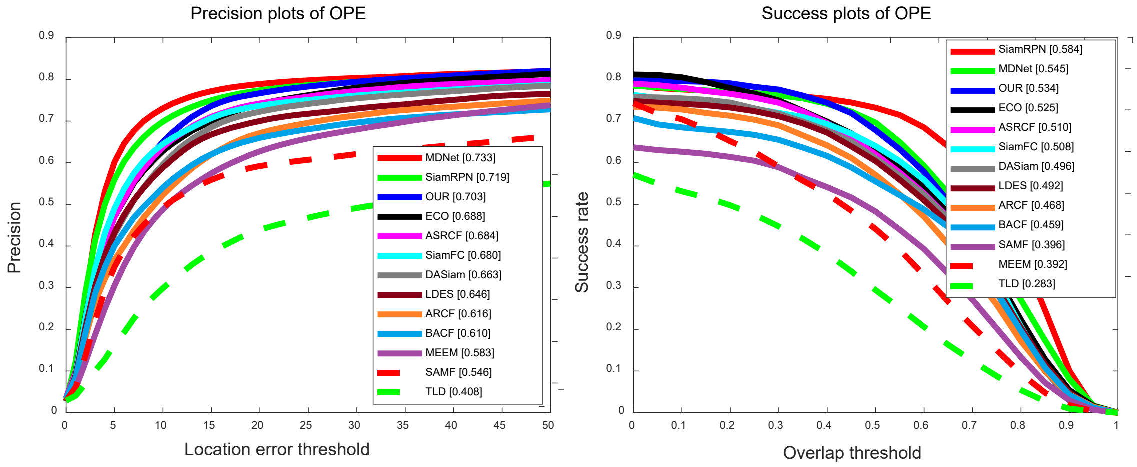


Figure 12. Distance precision and overlap success plots on the UAV123 [6] dataset. The legend of distance precision contains threshold scores at 20 pixels, while the legend of overlap success contains area-under-the-curve scores for each tracker. The proposed algorithm performs favorably against state-of-the-art trackers.

The UAV123 [6] dataset based on the drone has complex difficulty attributes, and contains 12 difficulty attributes, which are Scale Variation (SV), Aspect Ratio Change (ARC), Low Resolution (LR), Fast Motion (FM), Full Occlusion (FOC), Partial Occlusion (POC), Out-of-View (OV), Background Clutter (BC), Illumination Variation (IV), Viewpoint Chang e(VC), Camera Motion (CM), and Similar Object (SOB). More specifically, small targets make tracking more difficult in regard to judging a similar target. The low resolution and fast motion of drones also increase the difficulty of tracking. Figures 13 and 14 and Table 3 show the tracking performances, and our tracker achieved excellent performance under these attributes. At the same time, we also obtained the highest scores in background clutter and partial occlusions.

Table 3. Performances comparison of different trackers in cases of Similar Object (SOB), Fast Motion (FM) and Partial Occlusion (PO). The distance precision (DP) is threshold scores at 20 pixels, and the AUC is the area under the success rate curve score. The red, blue, and green colors represents the first, second, and third scores.

Trackers	**SOB**		**FM**		**PO**	
	AUC(%)	**DP(%)**	**AUC(%)**	**DP(%)**	**AUC(%)**	**DP(%)**
ECO [46]	51.8	69.4	41.5	59.7	45.6	62.9
SiamFC [20]	46.5	65.1	42.8	58.7	43.1	60.6
SiamRPN [16]	50.4	65.8	49.7	63.3	49.8	64.4
BACF [44]	44.9	63.2	33.9	48.3	37.2	53.0
LDES [50]	44.8	62.8	38.4	52.9	40.2	55.4
SAMF [35]	40.7	57.8	25.0	37.7	32.6	47.2
DASiam [22]	46.6	64.8	40.4	58.7	42.3	60.1
ARCF [52]	45.8	67.8	34.2	48.8	39.4	53.7
TLD [51]	30.0	46.7	19.4	29.6	22.8	35.3
MEEM [49]	39.7	59.7	26.0	41.4	33.1	50.2
OUR	49.0	66.2	46.0	66.1	46.4	64.1

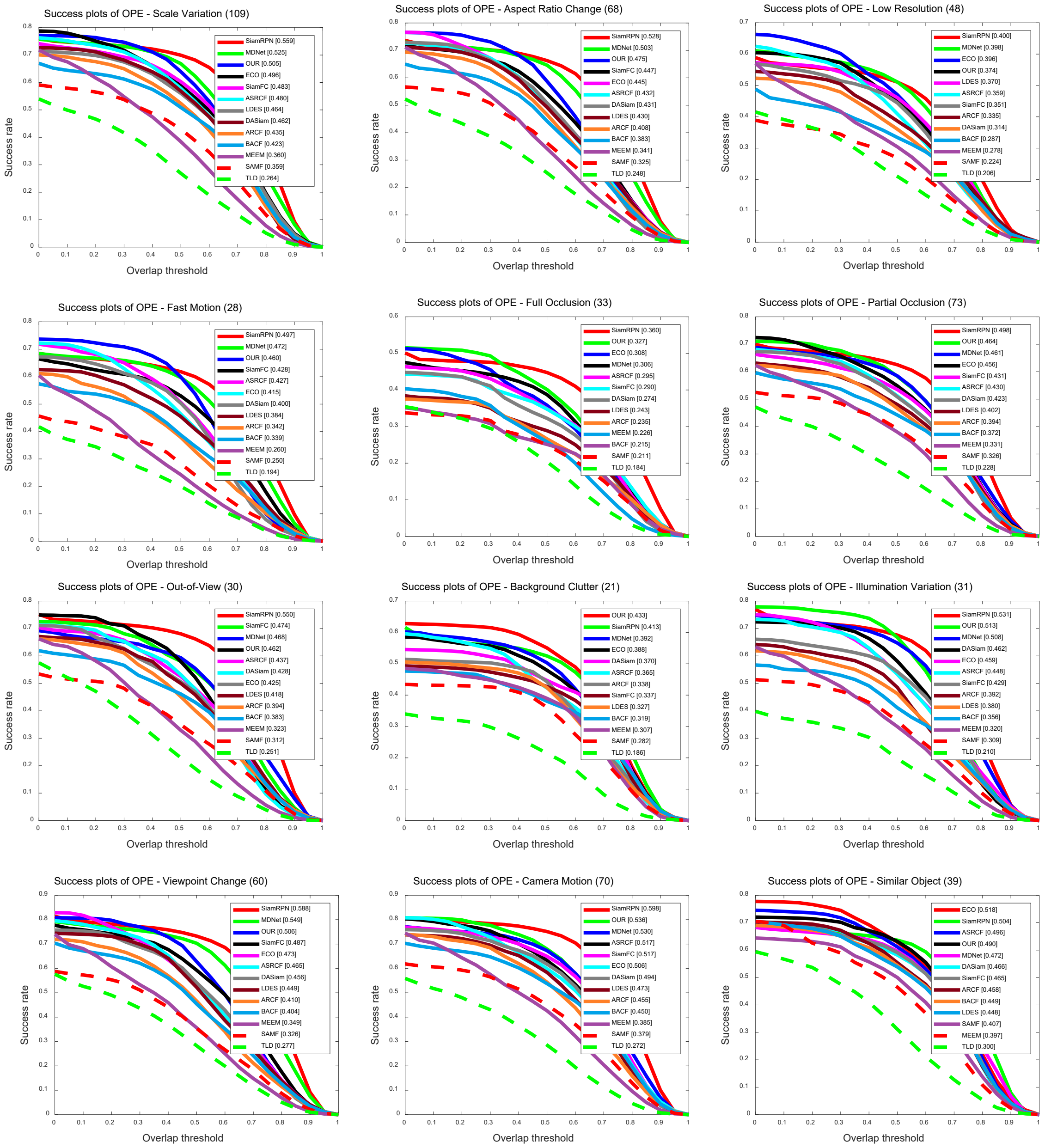


Figure 13. Precision plots based on tracking results on the UAV123 [6] dataset in 12 sequence attributes, including: Scale Variation (SV), Aspect Ratio Change (ARC), Low Resolution (LR), Fast Motion (FM), Full Occlusion (FOC), Partial Occlusion (POC), Out-of-View (OV), Background Clutter (BC), Illumination Variation (IV), Viewpoint Change (VC), Camera Motion (CM), and Similar Object (SOB). The legend of distance precision is threshold scores at 20 pixels. The proposed algorithm performs well against state-of-the-art results.

4.5. Limitations of Proposed Approach

Although the proposed method has achieved competitive results with other state-of-the-art trackers, it still has some limitations when the tracking is on-the-fly on the UAV platform.

Figure 15 shows two failed cases: *car1* and *s0101* are from the UAVDT [7] and UAV123 [6] dataset, respectively. The green box represents our results, and the red box represents ground truth.

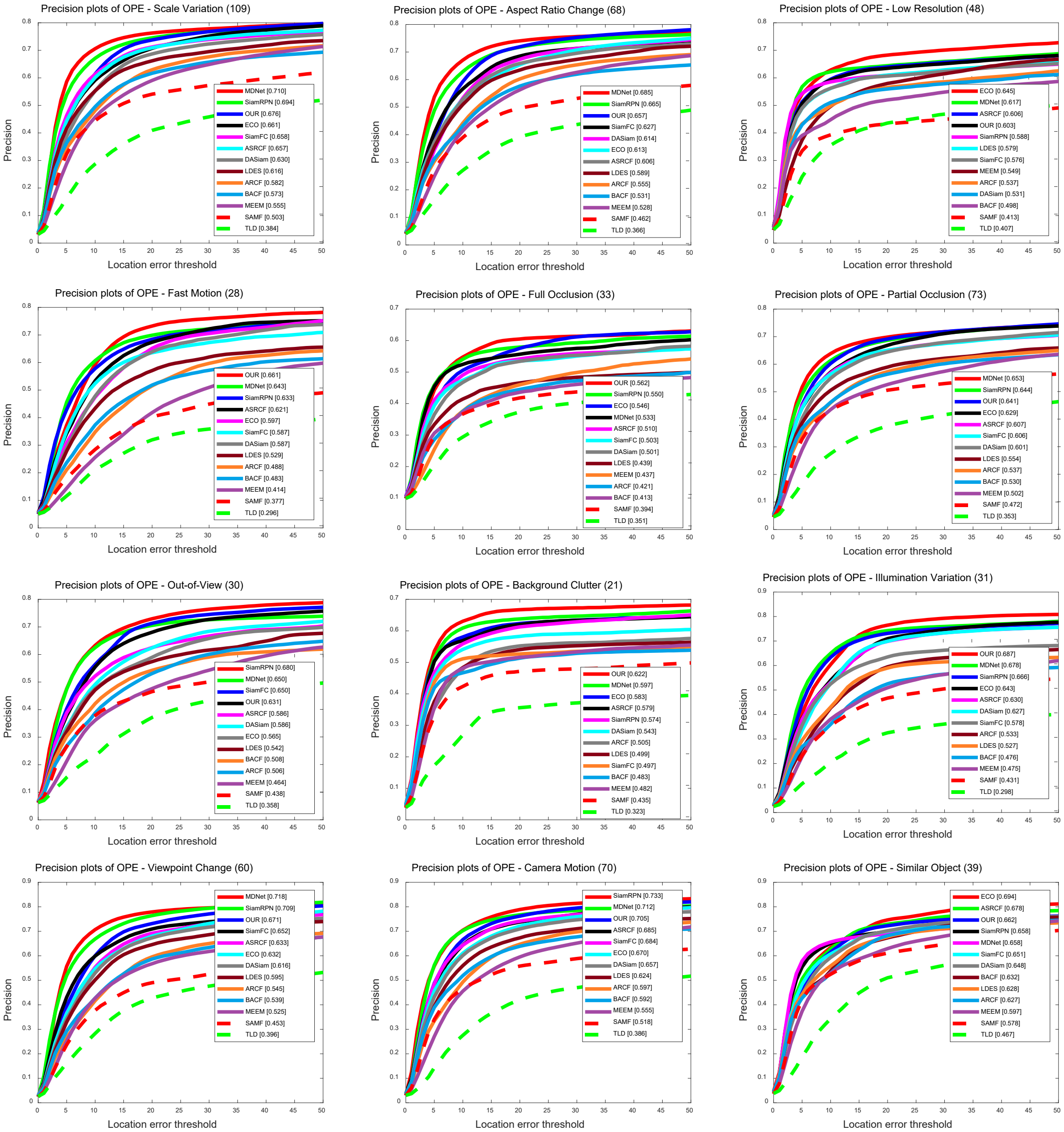
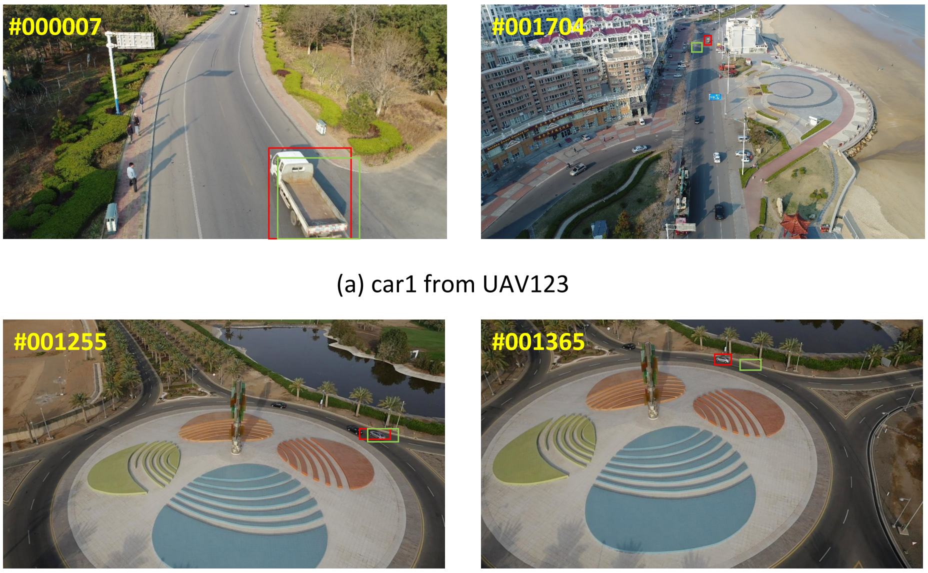


Figure 14. Success plots based on tracking results on the UAV123 [6] dataset in 12 sequence attributes, including: Scale Variation (SV), Aspect Ratio Change (ARC), Low Resolution (LR), Fast Motion (FM), Full Occlusion (FOC), Partial Occlusion (POC), Out-of-View (OV), Background Clutter (BC), Illumination Variation (IV), Viewpoint Change (VC), Camera Motion (CM), and Similar Object (SOB). The legend of overlap success is area-under-the-curve score. The proposed algorithm performs well against state-of-the-art results.

In the sequence *car1*, the truck undergoes large-scale variation from the 7th frame to the 1704th frame. The appearance of the target changed dramatically, so the tracker did not adapt to this variation. This is the reason why the tracker lost the target. The proposed method focuses on learning both similar and subspace semantic features, but they are invalid when the target suffers severely. We think the reason behind this is that the target model derived from the initial state changes gradually when the tracking is on-the-fly. However, the changed target largely makes the model confused with the current target and previous state, and the model does not represent the target discriminatively. Therefore, it is important to adapt to the scale variation of the target, especially for the UAV platform.



(a) car1 from UAV123

(b) s0101 from UAVDT

Figure 15. Two failures on sequence *s0101* from UAVDT [7] and *car1* from UAV123 [6]. The green box is the results of our tracker, and the ground truth is represented by the red box.

Additionally, the rotation and occlusion also degrades the tracking performance. In the sequence *s0101*, the car undergoes rotation and turns from side to side when it runs around the turntable. The proposed method cannot deal with this situation well and loses the target, because the representation is not robust enough to describe the target.

What is more, occlusion is another cause that leads the tracker to collapse. The tracking is forward-processing, and always thinks the target is on the scene. If the target has occluded or disappeared, the tracker still indicates a location as the result, even it is not the true target. These false results will be used as positive samples to train the tracker. Note that the false samples include a large background or distractors, so the discriminative ability of the tracker is not enough to identify the target from the background. One solution for this is to evaluate the results, which can be used as a criterion for an updated strategy.

5. Discussion

The analysis of the advantages and disadvantages of the proposed method can help choose the most suitable model for visual tracking. In Section 4, the proposed method achieved complete results with the state-of-the-art method on both UAV datasets [6,7]. In comparison with the Siamese network-based trackers, the results shown in Table 1 have demonstrated that the performance of our method is excellent according to the two criteria in Section 4.2, and also obtained fast tracking speed. More specifically, our method improved the success rate by 3.9% compared with the baseline tracker [20]. The reason for this is that our method can capture the semantic subspace details of online targets based on the traditional similarity measurement between the template and search region.

In comparison with CNN network-based trackers, some trackers have better performance than ours, but their speed is only about 1 FPS, and this limits them in the practical system. Overall, our method can obtain competitive results with state-of-the-art trackers, can run in real-time, and obtains similar performance, and this makes a good balance between performance and speed. Table 1 illustrates that our method improves the success rate by 7.7% and the precision by 6.2%, compared with the latest work, LADCF [43].

Considering each component of the proposed method, the ability of the single model is limited for tracking performance. However, the semantic subspace module can improve the performance in the Siamese tracking framework. We think the reason for this is that the semantic information is complementary to the similarity information obtained by the Siamese network.

Although the proposed method has achieved competitive results with other state-of-the-art trackers, it still has some limitations when the tracking is on-the-fly on the UAV platform. The tracker will lose the target it suffers from the dramatic appearance variation, especially for the scale variation in the UAV platform. Additionally, the rotation and occlusion also enable the tracker to collapse. In fact, the tracking looks forward in time, and always thinks that the target is on the scene. If the target has occluded or disappeared, the tracker still predicts results, whether it is the true target or not. After this false sample is fixed into the model update, the tracker will be blurred so as not to identify the target.

6. Conclusions

In this paper, we proposed an occlusion-aware online semantic subspace learning method with the Siamese network for UAV tracking. Instead of using linear dimension reduction, a new semantic subspace module was designed to encode the target's special information based on the shared Siamese network. Online learning enables the tracker to adapt to the variations of the target temporally. Additionally, the occlusion/disappearance detection avoids the polluted sample so as to update the model. Extensive experiments on the UAV benchmark tracking datasets verify the competitive performance of the proposed tracker with regard to performance and speed.

For future research, the network architecture is critical to improving the target representation, and the deeper and wider network needs to be explored for UAV tracking.

Author Contributions: All authors have devised the tracking approach and made significant contributions to this work. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported in part by National Natural Science Foundation of China (61773397, 61971352, 61862043, 61571362), grant 2018JM6015 approved as the Natural Science Basic Research Plan in Shaanxi Province of China, as well as supported by the Fundamental Research Funds for the Central Universities (3102019ZY1003, 3102019ZY1004).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fu, C.; Lin, F.; Li, Y.; Chen, G. Correlation Filter-Based Visual Tracking for UAV with Online Multi-Feature Learning. *Remote. Sens.* **2019**, *11*, 549.
2. Xia, G.S.; Datcu, M.; Yang, W.; Bai, X. information processing for unmanned aerial vehicles (UAVs) in surveying, mapping, and navigation. *Geo-Spat. Inf. Sci.* **2018**, *21*, 1.
3. Lu, Y.; Xue, Z.; Xia, G.; Zhang, L. A survey on vision-based UAV navigation. *Geo-Spat. Inf. Sci.* **2018**, *21*, 21–32.
4. Lyu, Y.; Vosselman, G.; Xia, G.; Yilmaz, A.; Yang, M.Y. The UAVid Dataset for Video Semantic Segmentation. *arXiv* **2018**, arXiv:1810.10438.
5. Xiang, T.; Xia, G.; Zhang, L. Mini-Unmanned Aerial Vehicle-Based Remote Sensing: Techniques, applications, and prospects. *IEEE Geosci. Remote. Sens. Mag.* **2019**, *7*, 29–63.
6. Mueller, M.; Smith, N.; Ghanem, B. A Benchmark and Simulator for UAV Tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; pp. 445–461.
7. Du, D.; Qi, Y.; Yu, H.; Yang, Y.; Duan, K.; Li, G.; Zhang, W.; Huang, Q.; Tian, Q. The Unmanned Aerial Vehicle Benchmark: Object Detection and Tracking. In Proceedings of the European Conference Computer Vision(ECCV), Munich, Germany, 8–14 September 2018; pp. 375–391.
8. Zhu, P.; Wen, L.; Bian, X.; Ling, H.; Hu, Q. Vision Meets Drones: A Challenge. *arXiv* **2018**, arXiv:1804.07437.
9. Deng, J.; Dong, W.; Socher, R.; Li, L.; Kai, L.; Li, F.-F. ImageNet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009; pp. 248–255.

10. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the Twenty-Sixth Annual Conference on Neural information Processing Systems (NIPS), Lake Tahoe, NV, USA, 3–8 December 2012; pp. 1106–1114.
11. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
12. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
13. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 583–596.
14. Visual object tracking by correlation filters and online learning. *ISPRS J. Photogramm. Remote. Sens.* **2018**, *140*, 77–89.
15. Danelljan, M.; Khan, F.S.; Felsberg, M.; van de Weijer, J. Adaptive Color Attributes for Real-Time Visual Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 1090–1097.
16. Li, B.; Yan, J.; Wu, W.; Zhu, Z.; Hu, X. High Performance Visual Tracking With Siamese Region Proposal Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 8971–8980.
17. Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; Yan, J. SiamRPN++: Evolution of Siamese Visual Tracking with Very Deep Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 4282–4291.
18. Guo, Q.; Feng, W.; Zhou, C.; Huang, R.; Wan, L.; Wang, S. Learning Dynamic Siamese Network for Visual Object Tracking. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1781–1789.
19. Wang, Q.; Zhang, M.; Xing, J.; Gao, J.; Hu, W.; Maybank, S. Do not Lose the Details: Reinformationrced Representation Learning for High Performance Visual Tracking. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Stockholm, Sweden, 13–19 July 2018; pp. 985–991.
20. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H.S. Fully-Convolutional Siamese Networks for Object Tracking. In Proceedings of the European Conference Computer Vision Workshops (ECCVW), Amsterdam, The Netherlands, 8–10 October 2016; pp. 850–865.
21. He, A.; Luo, C.; Tian, X.; Zeng, W. A Twofold Siamese Network for Real-Time Object Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 4834–4843.
22. Zha, Y.; Wu, M.; Qiu, Z.; Dong, S.; Yang, F.; Zhang, P. Distractor-Aware Visual Tracking by Online Siamese Network. *IEEE Access* **2019**, *7*, 89777–89788.
23. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI, Munich, Germany, 5–9 October 2015; pp. 234–241.
24. Li, P.; Wang, D.; Wang, L.; Lu, H. Deep visual tracking: Review and experimental comparison. *Pattern Recognit.* **2018**, *76*, 323–338.
25. Tao, R.; Gavves, E.; Smeulders, A.W.M. Siamese Instance Search for Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1420–1429.
26. Wang, Q.; Teng, Z.; Xing, J.; Gao, J.; Hu, W.; Maybank, S. Learning Attentions: Residual Attentional Siamese Network for High Performance Online Visual Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 4854–4863.
27. Zhang, Z.; Peng, H. Deeper and Wider Siamese Networks for Real-Time Visual Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 4591–4600.
28. Ross, D.A.; Lim, J.; Lin, R.S.; Yang, M.H. Incremental Learning for Robust Visual Tracking. *Int. J. Comput. Vis.* **2008**, *77*, 125–141.
29. Wright, J.; Yang, A.Y.; Ganesh, A.; Sastry, S.S.; Ma, Y. Robust Face Recognition via Sparse Representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *31*, 210–227.

30. Mei, X.; Ling, H.; Wu, Y.; Blasch, E.P.; Bai, L. Efficient Minimum Error Bounded Particle Resampling L1 Tracker With Occlusion Detection. *IEEE Trans. Image Process.* **2013**, *22*, 2661–2675.
31. Xiao, Z.; Lu, H.; Wang, D. L2-RLS-Based Object Tracking. *IEEE Trans. Circuits Syst. Video Technol.* **2014**, *24*, 1301–1309.
32. Roweis, S.T.; Saul, L.K. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science* **2000**, *290*, 2323–2326.
33. Hu, H.; Ma, B.; Shen, J.; Shao, L. Manifold Regularized Correlation Object Tracking. *IEEE Trans. Neural Netw. Learning Syst.* **2018**, *29*, 1786–1795.
34. Hu, H.; Ma, B.; Shen, J.; Sun, H.; Shao, L.; Porikli, F. Robust Object Tracking Using Manifold Regularized Convolutional Neural Networks. *IEEE Trans. Multimed.* **2019**, *21*, 510–521.
35. Li, Y.; Zhu, J. A Scale Adaptive Kernel Correlation Filter Tracker with Feature Integration. In Proceedings of the European Conference Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 254–265.
36. Wu, Y.; Lim, J.; Yang, M. Object Tracking Benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848.
37. Ma, C.; Huang, J.; Yang, X.; Yang, M. Hierarchical Convolutional Features for Visual Tracking. In Proceedings of the IEEE Conference on International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 3074–3082.
38. Nam, H.; Han, B. Learning Multi-domain Convolutional Neural Networks for Visual Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 4293–4302.
39. Song, Y.; Ma, C.; Gong, L.; Zhang, J.; Lau, R.W.H.; Yang, M. CREST: Convolutional Residual Learning for Visual Tracking. In Proceedings of the IEEE Conference on International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2574–2583.
40. Danelljan, M.; Häger, G.; Khan, F.S.; Felsberg, M. Learning Spatially Regularized Correlation Filters for Visual Tracking. In Proceedings of the IEEE Conference on International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 4310–4318.
41. Dai, K.; Wang, D.; Lu, H.; Sun, C.; Li, J. Visual Tracking via Adaptive Spatially-Regularized Correlation Filters. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 4670–4679.
42. Lukežic, A.; Vojír, T.; Zajc, L.C.; Matas, J.; Kristan, M. Discriminative Correlation Filter with Channel and Spatial Reliability. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4847–4856.
43. Xu, T.; Feng, Z.; Wu, X.; Kittler, J. Learning Adaptive Discriminative Correlation Filters via Temporal Consistency Preserving Spatial Feature Selection for Robust Visual Object Tracking. *IEEE Trans. Image Process.* **2019**, *28*, 5596–5609.
44. Galoogahi, H.K.; Fagg, A.; Lucey, S. Learning Background-Aware Correlation Filters for Visual Tracking. In Proceedings of the IEEE Conference on International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1144–1152.
45. Wu, Y.; Lim, J.; Yang, M. Online Object Tracking: A Benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013; pp. 2411–2418.
46. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. ECO: Efficient Convolution Operators for Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6931–6939.
47. Danelljan, M.; Robinson, A.; Shahbaz Khan, F.; Felsberg, M. Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking. In Proceedings of the European Conference Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; pp. 472–488.
48. Bertinetto, L.; Valmadre, J.; Golodetz, S.; Miksik, O.; Torr, P.H.S. Staple: Complementary Learners for Real-Time Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1401–1409.
49. Zhang, J.; Ma, S.; Sclaroff, S. MEEM: Robust Tracking via Multiple Experts Using Entropy Minimization. In Proceedings of the European Conference Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 188–203.

50. Li, Y.; Zhu, J.; Hoi, S.C.H.; Song, W.; Wang, Z.; Liu, H. Robust Estimation of Similarity Transformation for Visual Object Tracking. In Proceedings of the Thirty-Third Conference on Artificial Intelligence, AAAI, Honolulu, HI, USA, 27 January–1 February 2019; pp. 8666–8673.
51. Kalal, Z.; Mikolajczyk, K.; Matas, J. Tracking-Learning-Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1409–1422.
52. Danelljan, M.; Häger, G.; Khan, F.S.; Felsberg, M. Discriminative Scale Space Tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1561–1575.