mathematics

MDPI

Article

Correlation Filter of Multiple Candidates Match for Anti-Obscure Tracking in Unmanned Aerial Vehicle Scenario

Zhen Chen [1], Hongyuan Zheng [1,2,*], Xiangping (Bryce) Zhai [1,2,*], Kangliang Zhang [1] and Hua Xia [1]

[1] College of Computer Science and Technology/College of Artificial Intelligence, Nanjing University of Aeronautics and Astronautics, Nanjing 211006, China
[2] Key Laboratory of Safety-Critical Software, Ministry of Industry and Information Technology, Nanjing 211106, China
* Correspondence: zhenghongyuan@nuaa.edu.cn (H.Z.); blueicezhaixp@nuaa.edu.cn (X.Z.)

Abstract: Due to the complexity of Unmanned Aerial Vehicle (UAV) target tracking scenarios, tracking drift caused by target occlusion is common and has no suitable solution. In this paper, an occlusion-resistant target tracking algorithm based on the correlated filter tracking model is proposed. First, instead of the traditional target tracking model that uses single template matching to locate the target, we locate the target by finding the optimal match based on multiple candidates templates matching. Then, in order to increase the accuracy of matching, we use the self-attentive mechanism for feature enhancement. We experiment our proposed algorithm on datasets OTB100 and UAV123, respectively, and the results show that the tracking accuracy of our algorithm outperforms the traditional correlated filtered target tracking model. In addition, we have also tested the anti-occlusion performance of our proposed algorithm on some video sequences in which the target is occluded. The results show that our proposed algorithm has a certain resistance to occlusion, especially in the UAV tracking scenario.

Keywords: target tracking; occlusion-resistant; multiple candidates; optimal match

MSC: 68T45

Citation: Chen, Z.; Zheng, H.; Zhai, X.; Zhang, K.; Xia, H. Correlation Filter of Multiple Candidates Match for Anti-Obscure Tracking in Unmanned Aerial Vehicle Scenario. *Mathematics* **2023**, *11*, 163. https://doi.org/10.3390/math11010163

Academic Editor: António Lopes

Received: 27 November 2022
Revised: 20 December 2022
Accepted: 23 December 2022
Published: 28 December 2022

1. Introduction

UAV target tracking enjoys a wide popularity recently and has been applied in many applications, such as aerial photography, reconnaissance, rescue, and so on. Unlike target tracking in fixed scenes where the camera is stationary, in UAV scenes, the camera moves together with the target. In this case, the tracking context will become very complex and tracking will face many challenges, such as target deformation, in-plane and out-plane rotation, light source change, background clutter, similar interference, occlusion, and so on. Among these challenges, occlusion is a difficult problem in UAV target tracking and there is still no suitable solution for it.

The two currently dominating tracking paradigms are correlation filter-based modules [1–10] and Siamese networks [11,12]. Correlation filter-based tracking modules correlate the pre-trained filter template with the search area to obtain a response score map, and determine the target position based on the response score map. Bolme firstly introduces correlation filtering in target tracking and proposes the least square error output and correlation filtering algorithm (MOOSE) [1]. Then, Henriques proposes the Kernelized Correlation Filters (KCF) on the basis of MOOSE [2], which greatly improves the accuracy and speed of the tracking algorithm by using Fourier variation and kernel functions. Later, Danelljan adds color features as learning features to KCF and proposes exploiting the Circulant Structure of Tracking-by-detection with Kernels (CSK) [3] which achieve effective tracking for deformation targets. Siamese network-based target tracking modules use two identical convolutional branching networks to locate target position through end-to-end network learning. Bertinetto proposes the Fully Convolutional Siamese Networks for

Object Tracking (SiamFC) [10], which applies the full convolutional network to tracking and greatly improves the tracking accuracy. Li applies the Region Proposal Network (RPN) network to tracking on the basis of SiamFC and proposes the High Performance Visual Tracking with Siamese Region Proposal Network (SiamRPN) [11], which divides the target tracking into two parts, target detection and regression of candidate frames. The Siamese network-based target tracking algorithm has high accuracy but slower tracking speed, while the correlation filter-based target tracking algorithm has lower tracking accuracy but faster tracking speed. With the continuous development of target tracking technology, there are more and more studies to apply it to mobile terminals to solve practical problems. Considering that target tracking in the UAV scenario has certain requirements for real-time performance of tracking, the correlation filtering-based target tracking model is more suitable for target tracking in UAV scenario compared to Siamese networks.

We can classify the current research directions for improving model resistance to occlusion into the following categories: (1) Improving model resistance to occlusion by adjusting the model update strategy. This is because adjusting the update strategy of the model can reduce the cumulative error and thus improve the tracking accuracy. The algorithm proposed in [12] adjusts the update strategy according to the degree of target occlusion and designs an event triggering mechanism for different situations. The algorithm proposed in [9] adjusts the model update strategy based on the oscillation parameters of the response matrix. They all achieve some effectiveness to some extent. (2) Improving model resistance to occlusion by adjusting model training strategies. Adjusting the training strategy of the model allows the model to learn obstruction-resistant features and thus improves the robustness of the model. Algorithms proposed in [13–17] reduce the impact of interference information from sample frames with occlusion on model performance through temporally consistent and spatially adaptive model training. Algorithms proposed in [18–21] avoid model drift during tracking by designing an effective learning strategy that allows the model to learn features with robustness and discriminability. (3) Improving the model's resistance to occlusion by increasing the training samples. For example, by introducing high-quality training samples [22–24], easily mis-detected negative samples [25–27], and generating class-obscuring hard-to-score positive samples [7,28] during training to allow the model to learn features that are less sensitive to occlusion.

Since the current tracking models are all appearance-based tracking models, most of the current algorithms improve the model's resistance to occlusion from the aspect of improving the discriminability of the model. However, when there is severe occlusion or similar target occlusion in the scene, the discriminable features of the target are reduced and the interference noise is increased, it will be difficult to identify the target using the above appearance-based tracking model. However, when there occurs severe occlusion or similar interference in the scene, the performance of the above appearance-based tracking models will be decreased due to the reduction in discriminative target features. Therefore, solving the occlusion problem only by improving the discriminative ability of the model has some limitations. To solve this problem, we propose an anti-obscuration model based on the appearance-based tracking model by introducing other discriminative cues to reduce the interference effects. We can summarize the major contributions of our work as follows:

- We propose a multi-template matching strategy instead of the traditional single-template matching strategy to locate targets.
- We introduce the self-attention mechanism to enhance the extracted candidate feature descriptions and improve the matching accuracy.

Experimentally, our algorithm proves to be robust to scenes with occlusion or similar interference.

2. Related Works

2.1. Discriminative Target Tracking Model

Traditional discriminative tracking models view the tracking problem as a classification or regression problem, which uses a discriminant function to separate the target from

the background. Such tracking models usually first train a target template using the target features extracted from the artificially given target region in the first frame, then calculate the similarity between the target template and the image features in the search area in the following tracking frames to obtain a match score map, and finally locate the target according to the peak position of the match score map. Such tracking model focuses only on the features of the target, and the performance of the model is largely limited by the discriminability of the model [1–10]. Improving the uniqueness of learned target features can enhance the discriminative ability of the model to some extent. However, when the target is occluded or similar targets appear, the discriminable features of the target will reduce, and the tracking performance of the model will also decrease. Therefore, we propose a novel anti-occlusion target tracking algorithm based on a discriminative target tracking model. We focus not only on the target but also on the interference targets appearing in the scene, and avoid the interference by short-time tracking of the interference targets.

2.2. Self-Attention Mechanism

An attention mechanism is a special structure embedded in machine learning models that can find correlations between data and highlight some important features [29]. The attention mechanism enjoys a wide popularity recently in computer vision and has been applied in many applications, such as image recognition, image vision, 3D vision, and so on [30,31]. A self-attention mechanism is a variation of attention mechanism, which is less dependent on external information and better at capturing correlations within data. Suppose the input data are denoted as a query and the data in context are denoted in the form of a key–value pair (key, value), attention mechanism can be represented as finding a mapping function onto the query to the key–value pair (key, value). In self attention mechanism, query, key, and value are equal, therefore, self attention mechanism can effectively capture the correlation within the dataset itself [29]. We apply the self-attention mechanism to generate the feature descriptions of candidate points, and enhance the feature description of the candidate points with the correlation of appearance and location features between the extracted candidate point sets. In this way, we can improve the matching accuracy.

2.3. Multi-Target Tracking

Multi-target tracking technology is the study of simultaneous tracking of multiple targets in a video sequence. Most of the multi-target tracking algorithms are detection-based tracking, which consists of three key steps: data segmentation, data association, and data filtering. During the data segmentation step, it segments sensor data using clustering or pattern recognition techniques [32]. During the data association step, data segments are associated with targets using data association algorithms. Finally, for each target, the position is estimated by taking the geometric mean of the data assigned to the target, and Kalman usually updates the position estimation filtering or particle filtering [33,34]. In this paper, we apply the idea of multi-target tracking to single-target tracking. Unlike traditional single-target tracking, our algorithm focuses not only on the given target but also on the interfering targets that appear during the tracking process. In addition, different from multi-target tracking, our algorithm does not need to predict the position of the interfering target, but only uses the interfering targets as references for predicting the position of the given target. We divide localization into two steps: candidate points generation and data association. In the candidate point generation step, all image points that have the potential to be the target are detected, and in the data association step, the target point is selected from the set of candidate points using data association techniques and thus the target position can be located.

3. Materials and Methods

3.1. Algorithmic Architecture

We show the algorithm architecture of this paper in Figure 1. The architecture consists of two modules: (1) the base tracking module and (2) the target location module. In the base tracking module, we input the image of the current frame and the target template into the correlation filter to calculate the response score map. In the target localization module, we introduce a multi-matching strategy to localize the target according to the response score map. We can describe the detailed process as follows: (1) Extract candidate targets according to the response score map. (2) Perform feature enhancement on candidate targets to generate feature descriptors. (3) Calculate the matching scores between the feature descriptors of the candidate targets in the current frame and those in the candidate set created in previous frames. (4) Maximize the total matching score to find the optimal match, and we take the position of the candidate matched with the target in the optimal match as the target position.

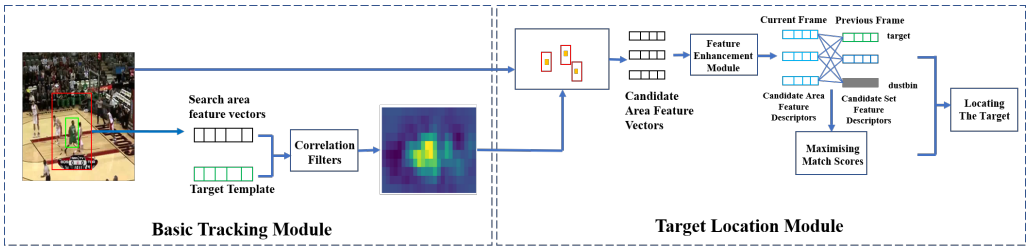


Figure 1. Algorithm framework diagram.

Base Tracking Module

We add an update strategy to KCF [2] as our base tracking model. First, we extract the HOG features of the image at the given target position in the first frame to train the correlation filtering template α. The calculation process is shown in (1), where y denotes the Gaussian label centered at the target location, x_1 denotes the Histogram of oriented gradients (HOG) feature map of the target region of the first frame, and $k(x_1, x_1)$ denotes the HOG feature map of the first frame with its own kernel correlation operation, w_1 denotes the correlation filter template of the target in the first input image frame, and λ denotes the regularization parameter. Then, in the following tracking frames, we extract the HOG feature of the image at the target location predicted in the previous frame and correlate it with the target template to obtain the corresponding score map $f(x_t)$. We show the calculation process in (2), where W_{t-1} is the target template of the previous frame and is equal to α_1 in the first frame.

$$\alpha_1 = \frac{y}{k(x_1, x_1) + \lambda}. \tag{1}$$

$$f(x_t) = k(x_t, W_{t-1}) \cdot \alpha_{t-1}, \tag{2}$$

The base tracking model can be update using (3) and (4), where ρ denotes the correlation filter template update parameter, α_t denotes the correlation filter template of the t-th frame, α_{t-1} denotes the correlation filter template of the previous frame, and x_t denotes the HOG feature map of the search region of the t-th frame.

$$\alpha_t = (1 - \rho)\frac{y}{k(x_t, x_t) + \lambda} + \rho\alpha_{t-1}. \tag{3}$$

$$W_t = (1 - \rho)W_{t-1} + \rho x_t. \tag{4}$$

KCF updates the correlation filtering template on each frame. However, when the target is occluded, the extracted target feature are less reliable, which may lead to error accumulation and thus reduce the robustness of the template. Therefore, we introduce

a strategic template update method into the model. We found that the fluctuation range of the response score map can reflect, to some extent, whether the target is obscured or not, as shown in Figure 2. Thus, we introduce the side flap ratio (PSR) [9] to evaluate the fluctuation of the response score map. The equation of PSR is shown in (5), where g_{max} is the peak response of the response score map, and μ and σ are the mean and variance of the response confidence map after excluding the peak, respectively.

$$PSR = \frac{g_{max} - \mu}{\sigma}. \tag{5}$$

The smaller the value of PSR is, the greater the fluctuation of the response score plot will be, and the more likely the target is obscured. Therefore, the base tracking model will not be updated when the PSR value of the response map is less than a certain threshold value σ_{psr}. When the PSR value of the response score map is greater than the threshold value, the model will be updated.

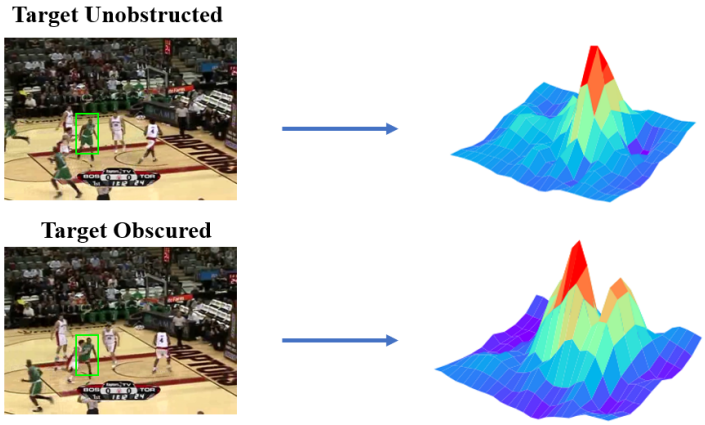


Figure 2. Response map with and without target occlusion.

3.2. Generation of Candidate Targets and Feature Descriptors

We take both the local peak and the peak location in the response map as candidate targets. We can describe the detailed steps as follows: (1) Use a slide window of size 5×5 to globally search the response map in the step of 1. (2) Extract the local maxima within the window and use their positions as candidate positions.

Due to the limitation of the search area, the positions of extracted candidate targets are close to each other and the appearance features are similar, so using only appearance feature cannot express the specificity of the candidates at this time; however, the insufficient candidate feature description will lead to an increase in the matching failure probability. As a result, enhancing the candidate feature description is very important. Considering that candidates are not isolated, and each candidate has certain appearance or location associations with other candidate, we introduce the self-attention mechanism to enhance the feature description of candidates. By using the self-attentive mechanism, we enhance the specificity of each candidate by exploiting the correlation with other candidates. We show the feature learning model in Figure 3.

First, the HOG features $\{a_1, a_2, \ldots, a_n\}$ and location features $\{d_1, d_2, \ldots, d_n\}$ of each candidate point in the candidate point set are extracted separately from the original image. Then, we encode the HOG features and location features of the candidate points to obtain the feature vector $a' = \{a'_1, a'_2, \ldots, a'_n\}$. The encoding process is $a'_i = x_i + conv(d_i)$, where conv(•) is used to perform a 1×1 convolution on d_i, which is used to up-dimension d_i to ensure that x_i has the same dimension as d_i. Finally, the encoded features $a' = \{a'_1, a'_2, \ldots, a'_n\}$ are used as the input to the self-attentive module. When computing the reinforcing feature b_i of candidate point a_i, a_i is token as query and other candidate points are token as keys. As shown in Figure 3, the attention network first calculates the correlation

between the query and each key, then uses the correlation as the weight of each value, and finally the product of each weight and value is summed to obtain the output. The weight parameters W_q, W_k, and W_v can be obtained by training.

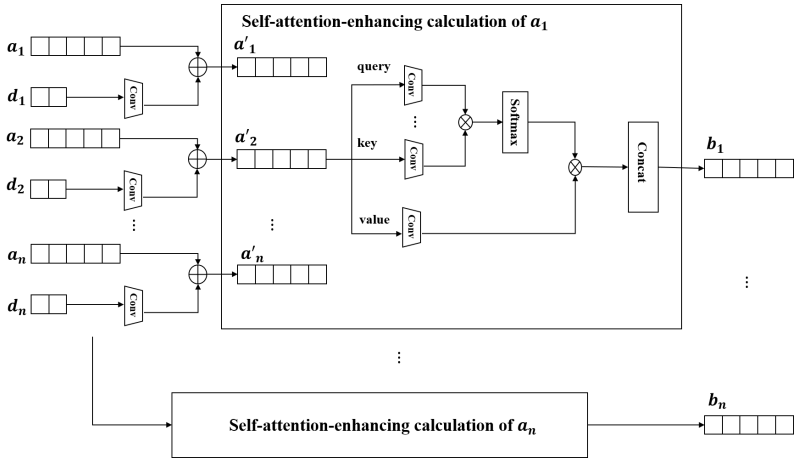


Figure 3. Feature learning model.

We use the self-supervised learning strategy to train the self-attentive module on LaSOT dataset [35]. Before training, we process the data. We can describe the detailed process of data processing as follows: (1) Input all the video frames in the LaSOT dataset into the base tracking model to calculate the response map of each frame. (2) Remove the image frames with only one candidate detected, since we only focus on the image frames with multiple candidates. (3) Divide the training dataset into two parts: the training set and the validation set to train the model.

We then train the model with the processed data. For each frame in the training set, we first perform affine transformation (zoom in or out, rotation, translation) on it, and then extract the candidates' HOG and location features in the original frame and the transformed image according to the corresponding response map. Finally, we input the candidate point features before and after the transformation into the model to calculate the feature descriptors $f = \{f_0, f_1, \ldots, f_n\}$ and $f' = \{f'_0, f'_1, \ldots, f'_n\}$. We calculate the similarity scores between feature descriptors by (7) to obtain match matrix M. The loss function can be calculated as shown in (6). We set the ground truth of the similarity score between f and f' as $M = \{(i,j)\}_{i,j=1,\ldots,n}$, whose value is 1 if i equals j otherwise it is 0. In addition, to simulate the occlusion we remove some candidate points randomly in some of the training data set.

$$loss = \sum_{i,j} -log(M_{i,j}). \tag{6}$$

3.3. Target Position Determination

Let the set of candidate feature descriptors for frame t be $P_t = \{p_0^t, p_1^t, p_2^t, \ldots, p_n^t\}$, where p_t^j is the feature descriptors of the i-th target in the frame t, then the set of candidate feature description subvectors in the frame $t-1$ is $P_{t-1} = \{p_0^{t-1}, p_1^{t-1}, \ldots, p_n^{t-1}\}$. For each $p_i^t \in P_t$, we compute the Euclidean distance between p_i^t and all points in the set of P_{t-1} as the similarity score. The calculation formula is shown in (7), where $p_{i,k}^t$ is the value at the position k of the p_i^t vector and $p_{j,k}^t$ is the value at the kth position of the p_j^t vector. The larger similarity score is, the more similar p_i^t is to p_j^t.

$$s_{i,j} = \frac{1}{\sqrt{\sum_{k=1}^{c}(p_{i,k}^t - p_{j,k}^{t-1})^2}}. \tag{7}$$

For each set of perfect matches M^k from P_t to P_{t-1}, we calculate the total match score as S^k_{total}, which is calculated as (8)–(10), where $m^k_{i,j}$ is the value of the row i and column j in the matrix M^k. The maximum value of k is equal to the number of perfect matches. We find the optimal match by maximizing the total match score, as shown in (11).

$$S^k_{total} = \sum_{i=1}^{i=n} \sum_{j=1}^{j=m} s'_{i,j}, \tag{8}$$

$$s'_{i,j} = \begin{cases} s_{i,j} & m^k_{i,j} = 1, \\ 0 & m^k_{i,j} = 0, \end{cases} \tag{9}$$

$$m^k_{i,j} = \begin{cases} 1 & \text{if } p^t_i \text{ matches } p^{t-1}_j, \\ 0 & \text{else}, \end{cases} \tag{10}$$

$$M_{best} = \underset{k}{argmax}\, S^k_{total}. \tag{11}$$

The candidate point in the optimal match that matches the previously detected-target is set as the target of the current frame. Considering that the candidate points detected on the previous and current frame are not necessarily a complete one-to-one match, we add a dustbin bit and set the matching score of all candidate points in P_t to this dustbin bit as a certain threshold σ_d, which represents the lowest limit of the match score. If a candidate point matches the dustbin bit in the optimal matching, the candidate point is a newly emerging candidate point and we need to add it to the candidate set. We show the flow of target localization in Figure 4. We store the 0th position in the candidate set as the target position. We use the KM matching algorithm to find the optimal match from P_t to P_{t-1} [36].

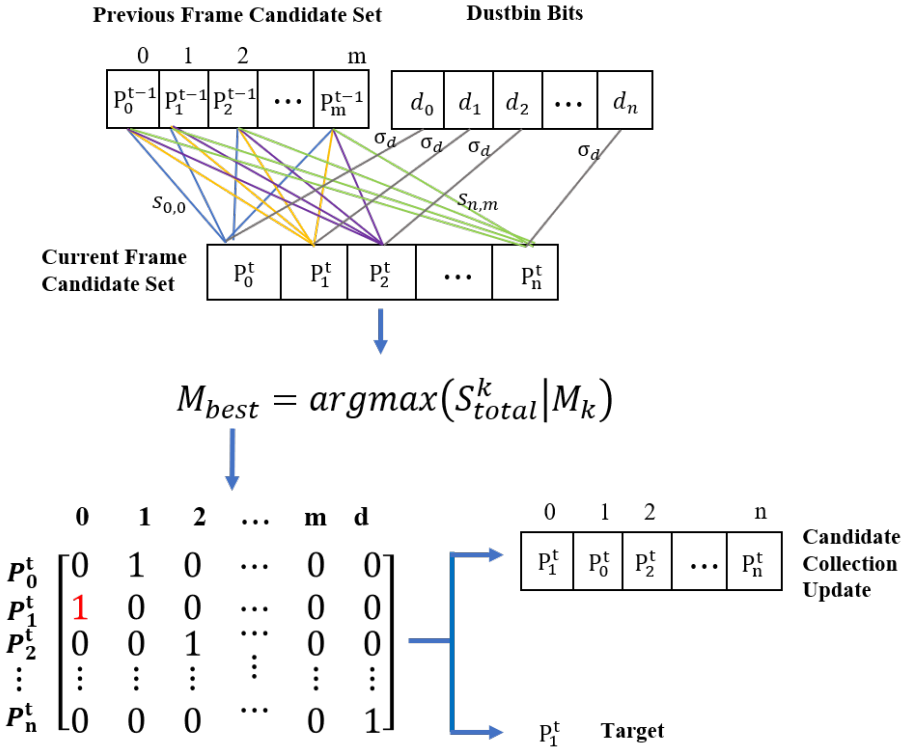


Figure 4. Target localization process.

An overview flow chart of the proposed method is summarized in Figure 5.

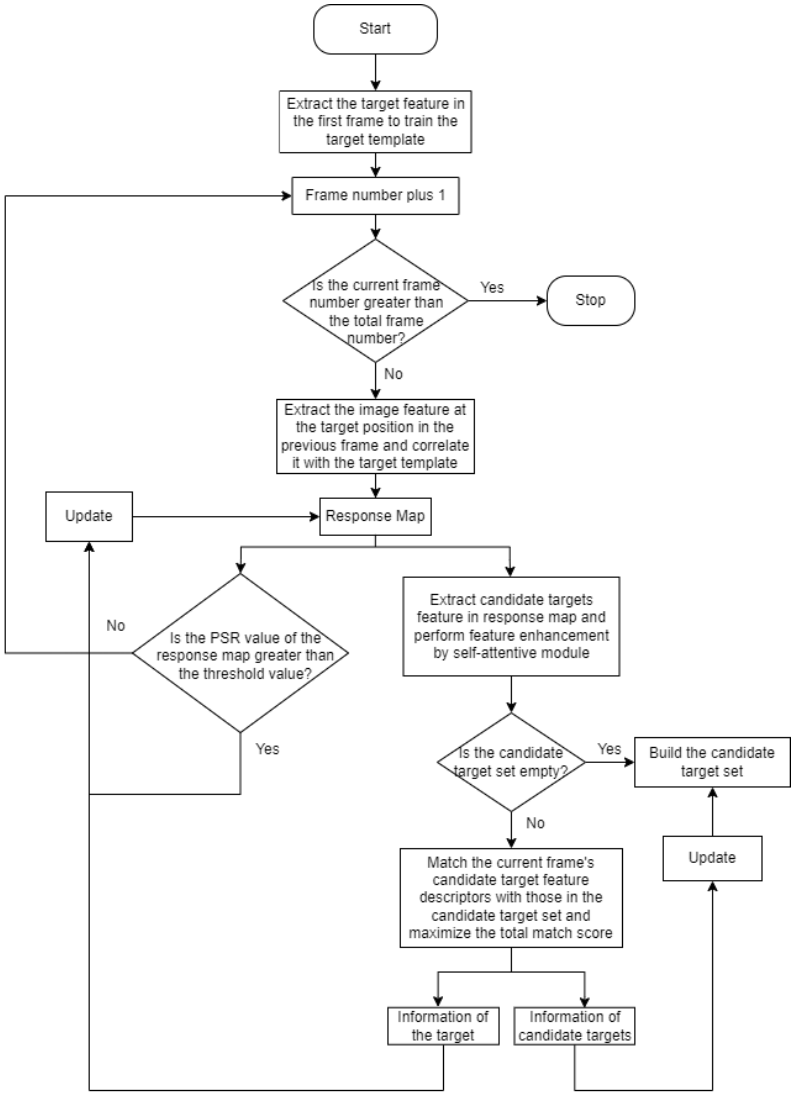


Figure 5. Algorithm flow chart.

3.4. Experimental Dataset and Evaluation Index

We evaluate our algorithm on the OTB2015 dataset containing 100 tracked video sequences [37] and the UAV123 dataset containing 91 tracked video sequences [38]. In order to evaluate the anti-occlusion performance of the algorithm, we also select some of the video sequences in which the target is occluded to evaluate our algorithm.

We use precision and success as the evaluation indexes of the experiment. The success is the ratio of the number of images whose distance difference between the target position detected by the tracking model and the true value is less than a certain threshold to all images. The precision refers to the ratio of the number of images whose overlap area between the target area output by the model and the actual target area is less than a certain threshold. We use the FPS (Frames Per Second) index as the speed test index of the algorithm—it indicates the number of frames per second that the algorithm can track, and the higher the FPS value, the faster the algorithm is.

4. Results

4.1. Experimental Environment and Parameters

The algorithm runs on a hardware platform configured with an Intel I5-10210U 1.60 GHz CPU and a NVIDIA GeForceMX250 GPU. The software platform is PyCharm2020. The model update threshold σ_{psr} in the base tracking module is set as 0.7. In the target

location module, the matching threshold σ_d between the candidate target feature descriptors and the dustbin bit is set to 0.02. Mini batch and gradient descent algorithm were used to optimize network parameters for the self-attention module training of feature strengthening. The batch size is 500, and the learning rate is 0.02. The ratio of training data to test data in the training set is 8:2. The number of training iterations is 15.

4.2. OTB Data Set Evaluation Results and Analysis

The OTB2015 dataset contains 100 video sequences, which is an extension of the OTB2013 dataset and is the mainstream tracking dataset [37]. The dataset includes 11 kinds of tracking challenges encountered during tracking, such as illumination change, target deformation, target occlusion, fast movement, in-plane rotation, out-of-plane rotation, out of view, background similar target interference, low resolution, scale transformation, motion blur, and so on. In this paper, we select some advanced correlation filter trackers with anti-occlusion ability, such as TLD (Tracking-Learning-Detection) [6], CSK (Exploiting the Circulant Structure of Tracking-by-detection with Kernels) [3], FDSST (Fast Discriminative Scale Space Tracking) [7], Staple (Complementary Learners for Real-Time Tracking) [8], SRDCF (Learning Spatially Regularized Correlation Filters for Visual Tracking) [5], LMCF (Large Margin Object Tracking with Circulant Feature Map) [9], ASRCF (Visual Tracking via Adaptive Spatially Regularized Correlation Filters) [39], and ARCF-H (Learning Aberrance Repressed Correlation Filters) [40] to compare with our algorithm on OTB2015 benchmark dataset. We abbreviate our algorithm as MCMCF. We show the comparison results in Tables 1 and 2:

Table 1. Comparison of the success and tracking speed of different tracking algorithms based on the OTB2015 dataset.

	All Sequences	Illumination Change	Occlusion	Motion Blur	Fast Moving	Out of View	Low Resolution	Tracking Speed
LMCF	0.800	0.737	0.810	0.660	0.691	0.702	**0.545**	7FPS
SRDCF	0.781	0.697	0.790	**0.763**	**0.706**	0.703	0.524	8FPS
ARCF-H	0.751	0.738	0.761	0.743	0.733	0.673	0.536	8FPS
Staple	0.738	0.692	0.732	0.628	0.605	0.586	0.499	8FPS
ASRCF	0.692	0.701	0.659	0.624	0.681	0.514	0.503	7FPS
FDSST	0.673	0.679	0.646	0.528	0.501	0.513	0.497	8FPS
TLD	0.521	0.460	0.468	0.482	0.473	0.516	0.327	5FPS
CSK	0.443	0.388	0.404	0.336	0.380	0.410	0.397	9FPS
MCMCF	**0.813**	**0.744**	**0.832**	0.647	0.700	**0.711**	0.513	5FPS

Bolded font in the table indicates the highest score in each column.

The success test results are shown in Table 1 and the precision test results are shown in Table 2. We can see that our algorithm outperforms other algorithms in terms of tracking precision and success in not only all sequences but also in data sequences with the occlusion problem. Therefore, the anti-occlusion algorithm proposed in this paper is effective. However, it can also be seen from the test results that the tracking algorithm of our algorithm is not as good as other algorithms, which is where our algorithm needs to be improved.

To further test the anti-occlusion performance of our algorithm, we selected three video sequences from the OTB100 dataset with occlusion problems to compare our algorithm with LMCF and SRDCF, which have similar or even better success and precision than our algorithm in quantitative tests, and the visualization results are shown in Figure 6. In the results shown in Figure 6 our algorithm is able to track the target stably when it is occluded; however, the other two algorithms drift to some extent. Therefore, our algorithm outperforms the other algorithms in terms of resistance to occlusion.

Table 2. Comparison of the precision and tracking speed of different tracking algorithms on the dataset with occlusion on OTB2015.

	All Sequences	Illumination Change	Occlusion	Motion Blur	Fast Moving	Out of View	Low Resolution	Tracking Speed
LMCF	0.842	**0.783**	0.844	0.714	0.730	**0.695**	0.555	7FPS
SRDCF	0.783	0.761	0.845	**0.790**	0.741	0.683	0.520	8FPS
ARCF-H	0.763	0.731	0.802	0.783	0.698	0.681	0.547	8FPS
Staple	0.742	0.728	0.776	0.671	0.642	0.670	0.505	8FPS
ASRCF	0.664	0.742	0.761	0.605	0.634	0.660	0.552	7FPS
FDSST	0.608	0.731	0.709	0.542	0.512	0.510	0.494	8FPS
TLD	0.545	0.537	0.563	0.518	0.551	0.576	0.349	5FPS
CSK	0.481	0.481	0.500	0.342	0.381	0.379	0.411	9FPS
MCMCF	**0.844**	0.780	**0.846**	0.699	**0.733**	0.687	**0.539**	5FPS

Bolded font in the table indicates the highest score in each column.

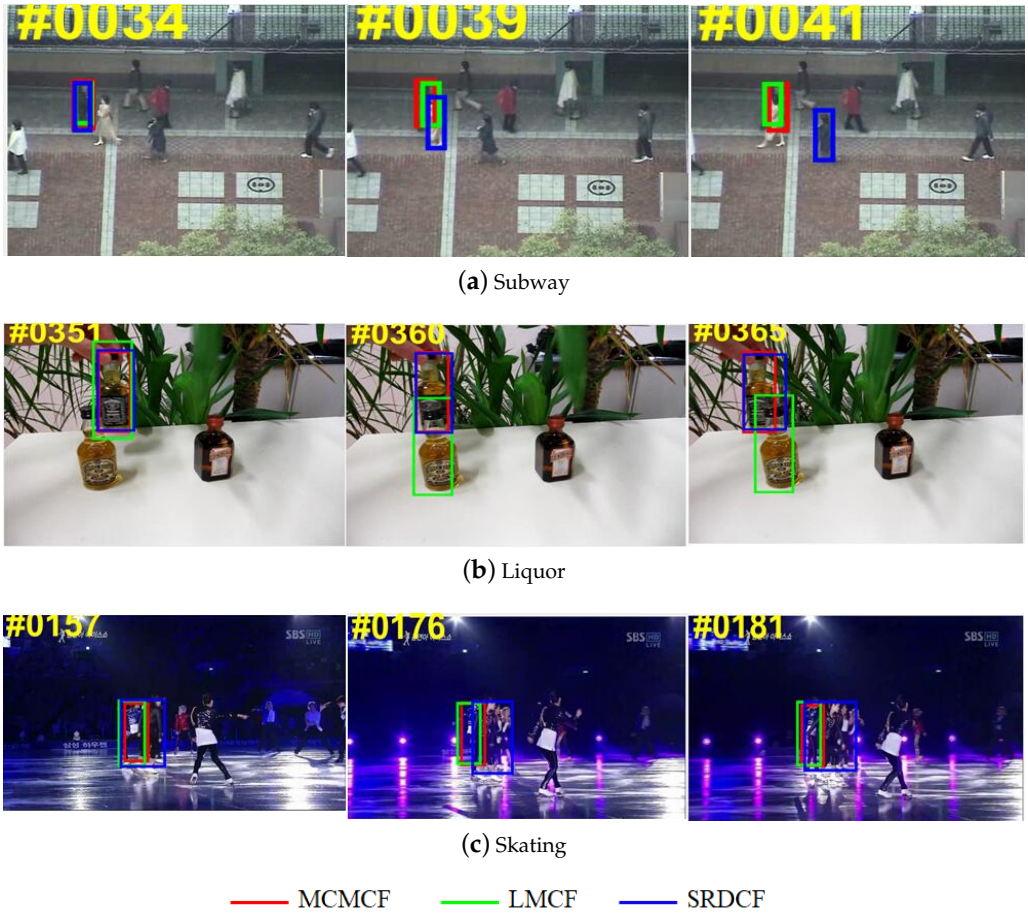


Figure 6. Results of different algorithms for some video sequences on the OTB100 dataset.

4.3. UAV123 Dataset Evaluation Results and Analysis

In order to evaluate the tracking performance of the proposed tracking model in UAV tracking scenarios, we also evaluate our tracking model on the UAV123 dataset. The UAV123 dataset is a dataset consisting of videos captured by low-altitude UAVs. It contains 91 video sequences, including 20 long video sequences [38]. In this paper, five tracking models are selected to compare with this paper's algorithm on UAV123, namely CSK [3], DCF (Discriminative Correlation Filter) [2], SRDCF [5], MUSTER (Multi-Store Tracker: A Cognitive Psychology Inspired Approach to Object Tracking) [4], and DSST (Discriminative Scale Space Tracking) [7]. The results are shown in Figure 7. Experimentally, our model outperforms other models not only in terms of success rate but also in terms of precision—

this shows that the proposed algorithm has good applicability for target tracking in UAV scenarios.

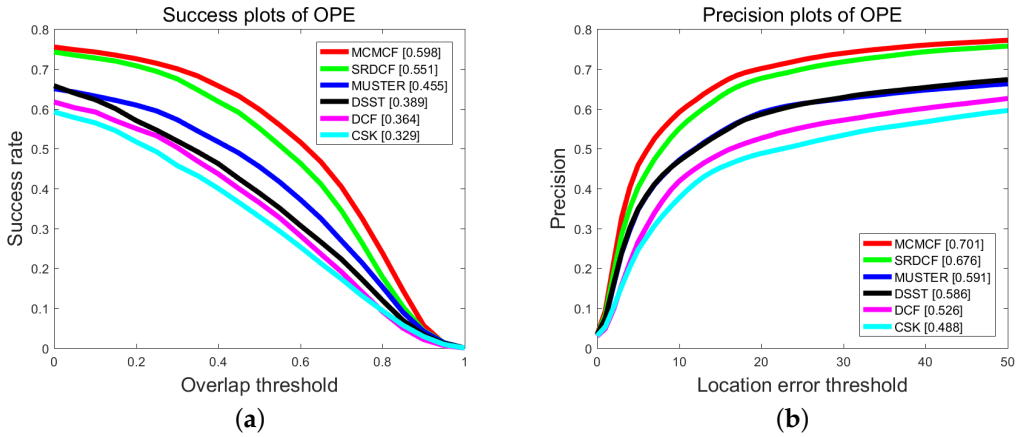


Figure 7. Comparison of success rate and accuracy with similar trackers on the UAV123 based dataset: (**a**) success result for models; (**b**) precision result for models.

In order to evaluate the anti-occlusion performance of the model in the UAV scene, this paper evaluates the performance of our model on some video sequences in the UAV123 dataset where occlusion and similar interference occur. The evaluation results are shown in Figures 8 and 9. The results show that the proposed algorithm still has good tracking performance when the target is occluded or similar interference occurs in the scene, so the anti-occlusion algorithm proposed in this paper is effective.

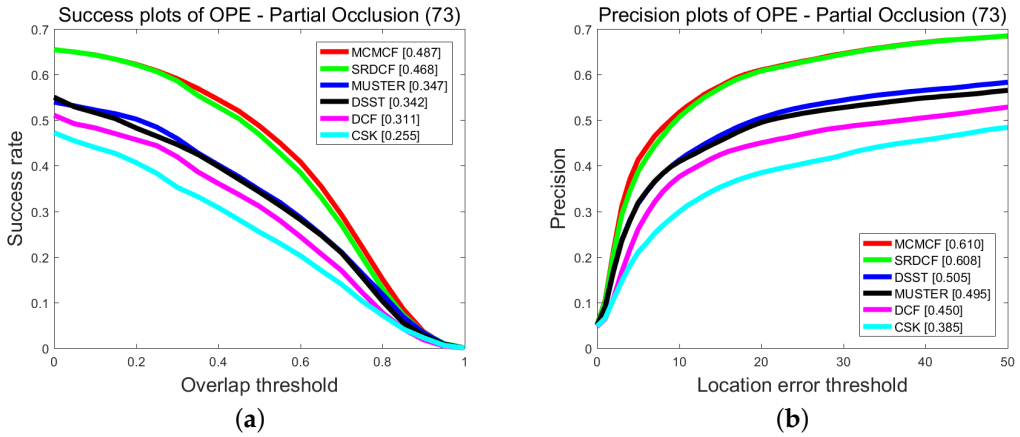
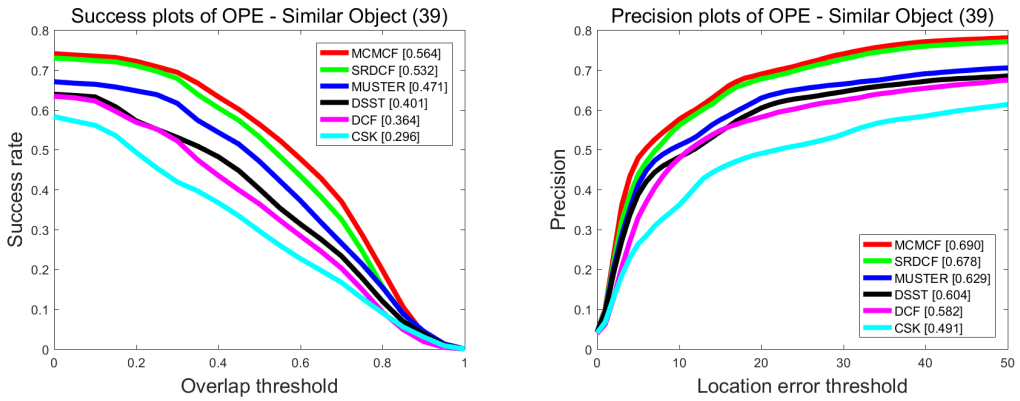


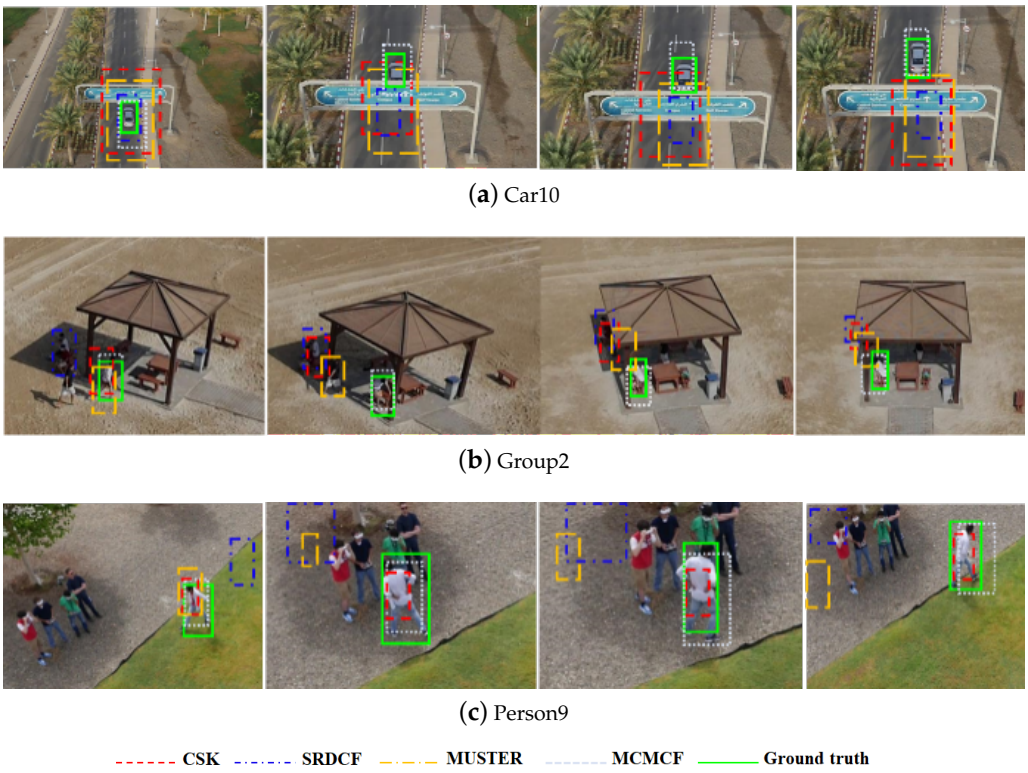
Figure 8. Comparison of success rate and accuracy with the same type of tracker on a dataset based on partial occlusion in UAV123: (**a**) success result for models; (**b**) precision result for models.

Figure 10 shows the results of our algorithm with some of the comparison algorithms on some of the UAV video sequences. In Figure 10a, partial occlusion appears in the scene. In Figure 10b, partial occlusion and similar interference appear in the scene. In Figure 10c, similar background interference appears in the scene. The results show that our target tracking model is still able to track the target stably in the above scenarios, while all other algorithms occur tracking drift to some extent.



(**a**) (**b**)

Figure 9. Comparison of success rate and accuracy with the same type of tracker on a dataset based on the presence of similar target interference in part of UAV123: (**a**) success result for models; (**b**) precision result for models.



(**a**) Car10

(**b**) Group2

(**c**) Person9

Figure 10. Results of different algorithms for some video sequences on the UAV123 dataset.

5. Conclusions

To address the tracking drift caused by target occlusion in UAV target tracking, we propose a novel anti-occlusion algorithm. To avoid the influence of interfering targets, we take into account the interfering target information when tracking. We first extract multiple candidate points in each frame based on the predicted response map. Then, we match the candidate points of the current frame with those of the previous frame and maximize the matching score to find an optimal match to locate the target position. In addition, to improve the matching accuracy, we introduce the self-attention mechanism to enhance the feature description of candidate points. The experimental results show that our algorithm outperforms the best-performing algorithm among the selected comparison algorithms by 1.3% and 0.2% on the OTB100 dataset and by 4.7% and 2.4% on the UAV123 dataset in

success and precision, respectively. In addition, the tracking success and precision on the dataset with occlusion attributes improved by 2.1% and 0.2%. It proves that our algorithm can effectively avoid the occurrence of target loss and has a certain anti-occlusion performance. Although our algorithm improves the tracking accuracy, it sacrifices the tracking speed to some extent. As a result, our future work will focus on finding a balance between improving target tracking accuracy and meeting UAV tracking real-time requirements.

Author Contributions: Conceptualization, H.Z. and X.Z.; methodology, Z.C.; software, Z.C.; validation, H.Z., X.Z. and Z.C.; formal analysis, Z.C.; investigation, Z.C.; resources, H.Z.; data curation Z.C.; writing—original draft preparation Z.C.; writing—review and editing, H.Z., X.Z., Z.C., K.Z. and H.X.; visualization, Z.C., K.Z. and H.X.; supervision, H.Z. and X.Z.; project administration, H.Z. and X.Z.; funding acquisition, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Natural Science Foundation of Jiangsu Province of China (Grant No. BK20222012), and in part by the Fund of Prospective Layout of Scientific Research for NUAA, and in part by the National Science Foundation of China (NSFC) with grant No. 61701231.

Data Availability Statement: Not applicable.

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

References

1. Bolme, D.S.; Beveridge, J.R.; Draper, B.A.; Lui, Y.M. Visual object tracking using adaptive correlation filters. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2544–2550.
2. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *37*, 583–596. [CrossRef] [PubMed]
3. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. *Exploiting the Circulant Structure of Tracking-by-Detection with Kernels*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 702–715.
4. Hong, Z.; Chen, Z.; Wang, C.; Mei, X.; Prokhorov, D.; Tao, D. Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 749–758.
5. Danelljan, M.; Hager, G.; Khan, F.S.; Felsberg, M. Learning Spatially Regularized Correlation Filters for Visual Tracking. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 4310–4318.
6. Kalal, Z.; Mikolajczyk, K.; Matas, J. Face-TLD: Tracking-Learning-Detection applied to faces. In Proceedings of the 2010 IEEE International Conference on Image Processing, Hong Kong, China, 26–29 September 2010; pp. 3789–3792.
7. Danelljan, M.; Häger, G.; Khan, F.S.; Felsberg, M. Discriminative Scale Space Tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1561–1575. [CrossRef] [PubMed]
8. Bertinetto, L.; Valmadre, J.; Golodetz, S.; Miksik, O.; Philip, H.S.T. Staple: Complementary learners for real-time tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1401–1409.
9. Wang, M.; Liu, Y.; Huang, Z. Large margin object tracking with circulant feature maps. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4021–4029.
10. Valmadre, J.; Bertinetto, L.; Henriques, J.; Vedaldi, A.; Torr, P.H. End-to-end representation learning for correlation filter based tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2805–2813.
11. 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, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8971–8980.
12. Guan, M.; Wen, C.; Shan, M.; Ng, C.L.; Zou, Y. Real-time event-triggered object tracking in the presence of model drift and occlusion. *IEEE Trans. Ind. Electron.* **2018**, *66*, 2054–2065. [CrossRef]
13. Li, F.; Tian, C.; Zuo, W.; Zhang, L.; Yang, M.H. Learning spatial-temporal regularized correlation filters for visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4904–4913.
14. Yuan, D.; Shu, X.; He, Z. TRBACF: Learning temporal regularized correlation filters for high performance online bvisual object tracking. *J. Vis. Commun. Image Represent.* **2020**, *72*, 102882. [CrossRef]
15. Fan, J.; Song, H.; Zhang, K.; Liu, Q.; Lian, W. Complementary tracking via dual color clustering and spatio-temporal regularized correlation learning. *IEEE Access.* **2018**, *6*, 56526–56538. [CrossRef]
16. Xu, T.; Feng, Z.H.; Wu, X.J.; 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. [CrossRef] [PubMed]

17. Zhang, Y.; Liu, G.; Zhang, H. Robust visual tracker combining temporal consistent constraint and adaptive spatial regularization. *Neural Comput. Appl.* **2021**, *33*, 1–20. [CrossRef]
18. Nam, H.; Baek, M.; Han, B. Modeling and propagating cnns in a tree structure for visual tracking. *arXiv* **2016**, arXiv:1608.07242.
19. Wang, N.; Zhou, W.; Tian, Q.; Hong, R.; Wang, M.; Li, H. Multi-cue correlation filters for robust visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4844–4853.
20. Kiani Galoogahi, H.; Fagg, A.; Lucey, S. Learning background-aware correlation filters for visual traking. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1135–1143.
21. Kiani Galoogahi, H.; Sim, T.; Lucey, S. Correlation filters with limited boundaries. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 4630–4638.
22. Danelljan, M.; Hager, G.; Shahbaz Khan, F.; Felsberg, M. 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, Las Vegas, NV, USA, 27–30 June 2016; pp. 1430–1438.
23. Zhu, Z.; Wang, Q.; Li, B.; Wu, W.; Yan, J.; Hu, W. Distractor-aware siamese networks for visual object tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 101–117.
24. 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, Las Vegas, NV, USA, 27–30 June 2016; pp. 4293–4302.
25. Fan, H.; Ling, H. Sanet: Structure-aware network for visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 42–49.
26. Choi, J.; Chang, H.J.; Fischer, T.; Yun, S.; Lee, K.; Jeong, J.; Demiris, Y.; Choi, J.Y. Context-aware deep feature compression for high-speed visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 479–488.
27. Zhang, X.; Wang, M.; Wei, J. Robust Visual Tracking based on Adversarial Fusion Networks. In Proceedings of the 2018 37th Chinese Control Conference (CCC), Wuhan, China, 25–27 July 2018; pp. 9142–9147.
28. Wang, X.; Li, C.; Luo, B.; Tang, J. Sint++: Robust visual tracking via adversarial positive instance generation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4864–4873.
29. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.
30. Wickens, C. Attention: Theory, Principles, Models and Applications. *Int. J. Human–Computer Interact.* **2021**, *37*, 403–417. [CrossRef]
31. Anwar, S.; Barnes, N. Real Image Denoising With Feature Attention. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3155–3164.
32. Luo, W.; Xing, J.; Milan, A.; Zhang, X.; Liu, W.; Kim, T. Multiple object tracking: A literature review. *Artif. Intell.* **2021**, *293*, 103448. [CrossRef]
33. Lu, W.-L.; Ting, J.-A.; Little, J.; Murphy, K. Learning to track and identify players from broadcast sports videos. *TPAMI* **2013**, *35*, 1704–1716.
34. Xing, J.; Ai, H.; Liu, L.; Lao, S. Multiple player tracking in sports video: A dual mode two-way bayesian inference approach with progressive observation modeling. *IEEE Trans. Image Process.* **2011**, *20*, 1652–1667. [CrossRef] [PubMed]
35. Fan, H.; Lin, L.; Yang, F.; Chu, P.; Deng, G.; Yu, S.; Bai, H.; Xu, Y.; Liao, C.; Ling, H. LaSOT: A High-Quality Benchmark for Large-Scale Single Object Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
36. Manne, A.S. A target-assignment problem. *Oper. Res.***1958**, *3*, 346–375. [CrossRef]
37. Wu, Y.; Lim, J.; Yang, M.H. Object tracking benchmark. *TPAMI* **2015**, *37*, 1834–1848. [CrossRef] [PubMed]
38. Mueller, M.; Smith, N.; Ghanem, B. A Benchmark and Simulator for UAV Tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016.
39. Dai, K.; Wang, D.; Lu, H.; Sun, C.; Li, J. Visual Tracking via Adaptive Spatially-Regularized Correlation Filters. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
40. Huang, Z.; Fu, C.; Li, Y.; Lin, F.; Lu, P. Learning aberrance repressed correlation filters for real-time uav tracking. *arXiv* **2019**, arXiv:1908.02231.