machines

MDPI

Article

UAV Real-Time Target Detection and Tracking Algorithm Based on Improved KCF and YOLOv5s_MSES

Shihai Cao [1], Ting Wang [2], Tao Li [1,*] and Shumin Fei [3]

[1] School of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China; shihaicao@nuaa.edu.cn
[2] School of Information Science and Technology, Nanjing Forestry University, Nanjing 210037, China; wangting10402@njfu.edu.cn
[3] School of Automation, Southeast University, Nanjing 210096, China; smfei@seu.edu.cn
* Correspondence: autolitao@nuaa.edu.cn

Abstract: In past decade, even though correlation filter (CF) has achieved rapid developments in the field of unmanned aerial vehicle (UAV) tracking, the discrimination ability between target and background still needs further investigation due to boundary effects. Moreover, when the target is occluded or leaves the view field, it may result in tracking loss of the target. To address these limitations, this work proposes an improved CF tracking algorithm based on some existent ones. Firstly, as for the scale changing of tracking target, an adaptive scale box is proposed to adjustably change the scale of the target box. Secondly, to address boundary effects caused by fast maneuvering, a spatio-temporal search strategy is presented, utilizing spatial context from the target region in the current frame and temporal information from preceding frames. Thirdly, aiming at the problem of tracking loss due to occlusion or out-of-view situations, this work proposes a fusion strategy based on the YOLOv5s_MSES target detection algorithm. Finally, the experimental results show that, compared to the baseline algorithm on the UAV123 dataset, our DP and AUC increased by 14.07% and 14.39%, respectively, and the frames per second (FPS) amounts to 37.5. Additionally, on the OTB100 dataset, the proposed algorithm demonstrates significant improvements in distance precision (DP) metrics across four challenging attributes compared to the baseline algorithm, showing a 12.85% increase for scale variation (SV), 16.45% for fast motion (FM), 18.66% for occlusion (OCC), and 17.09% for out-of-view (OV) scenarios. To sum up, the proposed algorithm not only achieves the ideal tracking effect, but also meets the real-time requirement with higher precision, which means that the comprehensive performance is superior to some existing methods.

Keywords: kernelized correlation filter (KCF); average peak correlation energy (APCE); target tracking; YOLOv5s

check for updates

Academic Editor: Dan Zhang

Received: 12 March 2025
Revised: 14 April 2025
Accepted: 27 April 2025
Published: 28 April 2025

Citation: Cao, S.; Wang, T.; Li, T.; Fei, S. UAV Real-Time Target Detection and Tracking Algorithm Based on Improved KCF and YOLOv5s_MSES. *Machines* **2025**, *13*, 364. https://doi.org/10.3390/machines13050364

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Currently, as one of the crucial tasks in computer vision, target tracking plays an important role in various related fields such as radar tracking [1,2], video surveillance [3–5], and UAV detection [6–9]. The attainment of precise, stable, and efficient target tracking remains a formidable challenge owing to the intricate nature of UAV application scenarios and the constrained computational resources onboard the platform [10]. The challenges faced by UAV target tracking mainly include the following aspects. (1) There are a large number of ground interference targets in the wide field of view in the air, especially in bad weather conditions (such as fog, rain, strong light) and complex terrain environments (such

as urban tall buildings, mountain terrain), resulting in reduced target feature prominence and increased confusion between target and background. (2) Limited shooting altitude and oblique viewing angles typically result in small pixel proportions of tracked targets in ground imaging, making them vulnerable to background noise interference. (3) The UAV is easily affected by wind disturbance during flight, resulting in blurred imaging and target displacement, which seriously affects the tracking continuity. (4) Due to limited computing resources of a UAV, it is challenging to maintain high accuracy while using target detection and a tracking algorithm with low complexity. Despite great advancements in UAV target detection and tracking algorithms, there still exists significant potential for improvement in terms of both high precision and real-time performance due to the aforementioned challenges [11,12].

Normally, CF tracking algorithms determine target positions by generating response graphs via online learning CF, thereby transforming time domain computation into frequency domain computation and significantly enhancing tracking speed. The classical CF trackers encompass MOSSE [13], KCF [14], spatial and temporal regularization [15], adaptive scale estimation [16] and the fusion of complementary learners [17]. In 2010, the MOSSE algorithm [13] was proposed, the core idea of which is to realize real-time target positioning based on filtering and convolution operation. In 2014, the target tracking algorithm based on correlation filtering made a breakthrough. Henriques et al. [14] presented the KCF algorithm, which introduced a kernel technique to optimize the correlation filter in the frequency domain, thus greatly improving the speed and accuracy of target tracking. In the same year, Danelljan et al. [18] first considered the role of color features in single-target tracking, comprehensively evaluated the effects of features extracted from various color spaces in target tracking, and finally proposed a single-target tracking method based on multi-channel color features. In 2017, the CSRDCF algorithm proposed by Lukezic et al. [19] used the color model to construct a spatial-restricted mask, which enhanced the robustness and accuracy of tracking and suppressed the boundary effect. Building upon these algorithms, researchers have made many great improvements [20–22]. However, the majority of existing studies primarily focused on how to enhance the detection accuracy of tracking algorithms, while the real-time requirements always received less attention. For instance, the FPS of the algorithm [20] was only 6.1, which makes it difficult to deploy for real-time UAV target tracking tasks. Among these algorithms, the KCF algorithm, which represents a target tracking algorithm based on CF, not only exhibits high tracking accuracy and robustness, but also possesses fast tracking speed to meet the real-time requirements. However, the KCF algorithm may lose target features in case of serious occlusion, target deformation, and background confusion, which leads to a reduction in tracking accuracy and even the failure of tracking. Additionally, the fixed tracking target box in the KCF algorithm does not account for changes in scale of the target tracking, which can result in a reduction in tracking accuracy.

Motivated by the above discussions, this work proposes a real-time target detection and tracking algorithm for the UAV. Firstly, an adaptive scale box is introduced to enable the target box for dynamically adjusting its scale in response to the changes in the target size. Secondly, due to the boundary effects, the tracker is prone to drifting when the target is occluded or rotated. Then, a spatio-temporal search strategy is introduced to track the target by utilizing the spatial information surrounding the target in the current frame, along with the temporal information from the previous frame. This approach aims to mitigate the effects of boundary influences. Finally, the average peak correlation energy (APCE) [23] is used to determine whether the target is occluded or lost, and the YOLOv5s_MSES [24] target detection algorithm is employed to redetect and retrack the lost target. In summary, the innovations of this work are listed as follows:

- Based on some existent search strategies, this work initially proposes a new spatio-temporal search strategy (STS), which can comprehensively integrate the information of time and space to dynamically capture the target changes by incorporating historical data while retaining spatial information. Then, different from the traditional ones, the proposed STS can more effectively preserve the valuable feature information of the target, mitigate the target drift issues induced by the boundary effect, and efficiently enhance the search accuracy.
- This work innovatively puts forward an anti-loss strategy for target retracking (TR) based on the YOLOv5s_MSES algorithm. Such a strategy firstly utilizes the APCE to decide whether the tracking target will be obscured or out of the view field, and if so, the YOLOv5s_MSES is exploited to redetect all similar targets. Then, the APCE is further used to determine the real tracking target and track it again by resorting to the KCF algorithm. Thus, our TR strategy not only solves the problem that the current CF algorithm cannot retrack the lost target, but also facilitates the enhancement of tracking accuracy.
- In order to solve the issue of the fixed scale induced by the KCF algorithm, this work introduces an adaptive scale box method (ASB), enabling the dynamic adjustment of the scale of the target tracking box, which can improve the accuracy and stability for our derived algorithm, particularly in the case of large size variation.

The remainder of this work is organized as follows. The related works are reviewed in Section 2. The details of the improvement of the KCF algorithm are presented in Section 3. Section 4 shows the experimental results. Finally, we summarize the work in Section 5.

Notation 1. *The notations of this work are standard.* $\odot$ *represents the element-wise dot multiplication, and* $\hat{x}$ *denotes the discrete Fourier transform of the term* x*, and the kernel function* $k(z, x)$ *is defined as the dot product of the vectors* $\phi(z)$ *and* $\phi(x)$*.*

2. Related Work

2.1. Boundary Effects

Even though it is possible to prevent the tracker from overfitting by increasing training samples through the cyclic shift operation, it inevitably leads to the boundary effects of the CF tracker. Recently, many methods have been proposed to solve these negative effects [25–28]. Liu et al. [26] designed a context pyramid to effectively employ the target and its contextual information in the DCF learning process. Moorthy et al. [28] added wraparound patches to the tracker for suppressing the target drift caused by the boundary effects. However, these above studies only considered the spatial information of the current frame of the tracking target. Based on Ref. [28], this work proposes an improved spatio-temporal search strategy, which fully considers the position information of previous frames tracking the target and the contextual information around the current frame tracking the target, so that it can capture the dynamic changes of the target by combining historical data while retaining spatial information. In particular, the negative impacts of the boundary effects can be effectively eliminated.

2.2. Tracking Loss

Normally, there exist two primary categories for the tracking target loss issue: (1) the tracking target is obscured by other objects or interfered with the background; (2) the tracking target exceeds the field of view of the tracker. However, most studies only considered the issues of the background interference and object occlusion in trackers [29–32].

In recent years, together with the rapid developments of deep learning, researchers have combined target detection algorithms with the CF ones, and achieved some elegant results [33–35]. Cai et al. [33] first detected all moving targets using the YOLO target detection algorithm, and then combined the KCF algorithm with the Kalman filter to track multiple targets. Chen et al. [34] employed an SSD-FT target detection algorithm to redetect the tracking target to eliminate the error after each fixed frame of tracking. Although these above algorithms combine target detection algorithms with the tracking ones, they cannot be effectively utilized and struggle to meet the real-time requirements on tracking. Moreover, whether the background is occluded or the tracking target is out of the field of view, the final result means the loss of the target. Therefore, this work introduces the small target detection algorithm YOLOv5s_MSES on the basis of the KCF algorithm to solve the problem of target loss. In contrast to other detection algorithms, YOLOv5s_MSES has smaller parameter scale and higher tracking accuracy. It not only effectively detects the target for tracking, but also preserve the real-time performance of the tracking algorithm. Ref. [34] employed the SSD-FT target detection algorithm at each fixed frame. However, since the object loss may not occur during the whole tracking procedure, excessive use of the object detection algorithm will reduce the FPS, affect the real-time performance, and waste the limited resources of the computer. Moreover, in comparison with the SSD-FT target detection algorithm [34], the FPS of YOLOv5s_MSES is as high as 51, which more than meets the real-time requirement, while SSD-FT has less than 1 FPS. Therefore, inspired by Ref. [34], this work first utilizes the APCE to determine whether the tracked target appears to be obscured or out of the tracker's field of view, and if so, the YOLOv5s_MSES will be employed to redetect all similar targets. Then, the APCE is further utilized to select the tracking target and track it again via the proposed algorithm.

3. Methods

3.1. YOLOv5s_MSES Target Detection Algorithm

In order to improve the detection accuracy of the UAV target tracking algorithm, this work introduces an improved YOLOv5s_MSES algorithm, whose overall architecture is shown in Figure 1, including the following four key components. Firstly, in the data preprocessing stage, the Mosaic data enhancement method is used to enhance the diversity of the training data, and the input image is uniformly adjusted to the preset size with adaptive image scaling technology. Secondly, the backbone network is composed of several innovative modules. The Focus module uses feature slicing and replication operations to achieve efficient feature extraction; the newly introduced MSES attention module significantly enhances the ability of the network to capture features of small-sized targets; the SPP space pyramid module extends the receptive field of the network effectively through the combination of multi-scale pooling operation and jump connection. Thirdly, in the feature fusion part, the algorithm innovatively integrates the bidirectional feature transfer mechanism of FPN and PAN. Finally, the detection head adopts a multi-scale detection strategy, including three different scale detection layers of 40×40, 80×80 and 160×160. What is more, experimental results show that compared with the existing advanced algorithms [36,37], YOLOv5s_MSES exhibits superior performance on the VisDrone2019 dataset and is well suited for small object detection tasks.

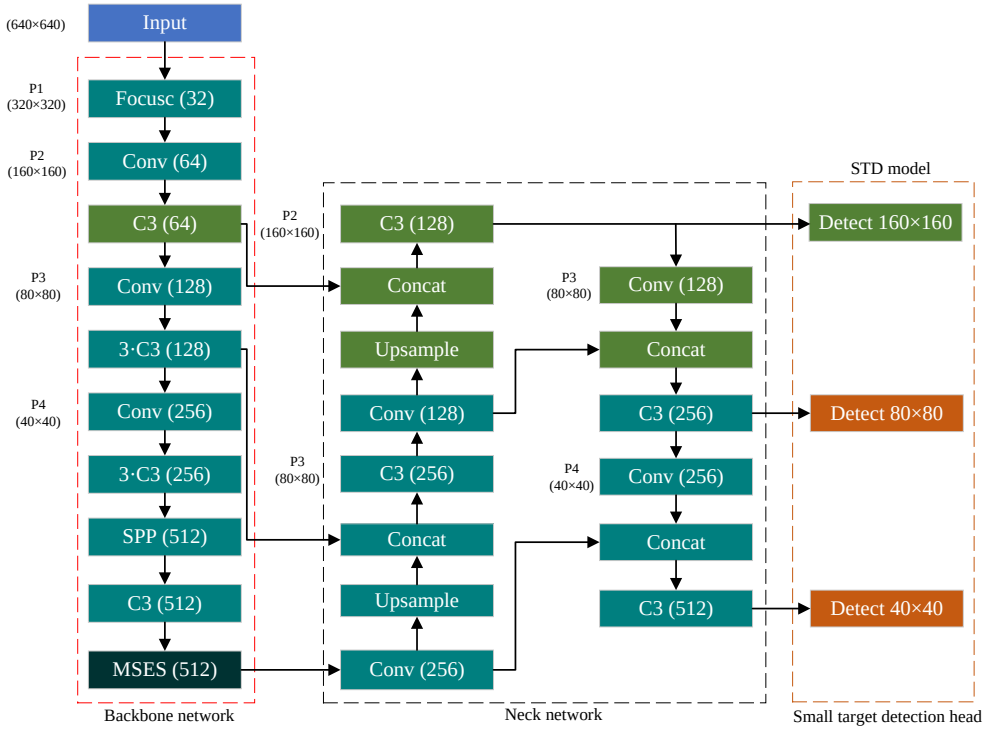


Figure 1. The YOLOv5s_MSES model structure.

3.2. KCF Target Tracking Algorithm

The KCF algorithm acquires a filled box by filling the target box given in the initial frame of the tracking. In the next frame of the image, the fill box is cyclically shifted to obtain a cyclic matrix to collect a large number of positive samples and negative ones. The region containing the target is designated as the positive samples, while the remainder is considered as the negative ones. These collected samples are used to train the target detector. Subsequently, the trained target detector is employed to ascertain whether the predicted location is the location where the target appears, and the filled box of the previous frame is utilized to continue the cyclic shift. These images in the obtained sample box are divided into positive samples and negative ones again, and the sample with the largest response is selected as the filling box of the target in the current frame. Finally, these samples obtained from the current frame are used to update the target detection classifier. During this process, these samples can be computed by utilizing the diagonalization property of the circulant matrix in Fourier space and fast Fourier transform to improve the calculation speed. The process is represented by the following formulas. (1) Let a set of training samples be $(x_i,\ y_i)\ i \in \{1, 2, \cdots, n\}$, then x_i denotes input sample i, and y_i represents the response of the i th sample. (2) The regression function is $f(x_i) = w^T x_i$, where w means the weight coefficient for the column vector, which can be solved by using the least square method. Equation (1) is given as follows:

$$\min_{w} \sum_{i=1}^{n} [f(x_i) - y_i]^2 + \lambda \|w\|^2 \tag{1}$$

where λ is the regularization coefficient to prevent the tracker from overfitting, and $\|w\|^2$ denotes the L2-norm of w.

By introducing the nonlinear mapping function $\phi(x_i)$, the low-dimensional nonlinear inseparable problem is mapped to the high-dimensional space, which can be transformed into a linear solution, whose weighted linear combination is expressed as follows:

$$w = \sum_{i=1}^{n} \alpha_i \phi(x_i) \tag{2}$$

$$f(z) = w^T z = \sum_{i=1}^{n} \alpha_i k(z, x_i) \tag{3}$$

where z represents the single test sample.

In Equations (2) and (3), the optimization problem of w is transformed into the optimization one for α. The final result is:

$$\hat{\alpha} = \frac{\hat{y}}{\hat{k}^{xx} + \lambda} \tag{4}$$

where k^{xx} means the first row element of the kernel function matrix. Here, according to Notation 1, $\hat{\alpha}$, $\hat{y}$, $\hat{k}^{xx}$ denote the discrete Fourier transforms of α, y, k^{xx}, respectively.

In the detection stage, firstly, the template obtained by training the target in the previous frame is used, and the kernelized correlation filter circularly shifts to the original target position region in the current frame for obtaining the candidate region. Then, z is correlated with the filter template to calculate its response value. The specific formula is given as follows:

$$f(z) = (K^z)^T \alpha \tag{5}$$

where the asymmetric matrix K^z denotes the kernel matrix between the training sample and the candidate one. Each element of K^z is defined as $K^z_{ij} = K(P^{i-1}z, P^{j-1}x)$, where P is the permutation matrix used to realize displacement. Here, α is a vector composed of each coefficient α^i. Furthermore, in order to compute Equation (5), this work diagonalizes it and further obtains it as:

$$\hat{f}(z) = \hat{K}^{xz} \odot \hat{\alpha} \tag{6}$$

The value of $f(z)$ is the response value of all the shifted candidate regions. The candidate sample corresponding to the maximum value means the candidate sample that is most likely to be the target after this detection.

3.3. Spatio-Temporal Search Strategy

Due to the boundary effects, the tracker easily tends to drift when the target undergoes occlusion and rotation. Inspired by Ref. [28], this work utilizes the appearance features and contextual information of the target for joint modeling, to better distinguish between the target and the background. Modify Equation (6) to the following formula:

$$\hat{f}(z) = \sum_{i=1}^{N} \hat{K}^{(x+\sigma_i)z} \odot \hat{\alpha} \tag{7}$$

where σ_i denotes the range size of the surrounding search with a size of 2.5. Moreover, N represents the number of surrounding search boxes, and since the algorithm searches in four directions, up, down, left, and right of the target, the value of N is 4 to make full use of the spatial context information around the target.

However, the tracking target may have different motion patterns in different time periods. For faster tracking targets, there may also be a problem of leaving the search

field in the next frame. Only relying on spatial information cannot capture this change and accurately predict the future position of the target. Therefore, in order to improve the accuracy and robustness of the tracker, except for considering spatial information, this work further involves temporal information. By modeling the historical trajectory of the target, the future position of the target can be predicted. In this way, even if the target experiences occlusion or rotation, its possible position can be inferred based on its motion pattern, thereby reducing the error caused by the boundary effects. Thus, we modify Equation (7) to the following formulas:

$$D_t = (X_t + X_{t-1} - X_{t-2}, Y_t + Y_{t-1} - Y_{t-2}) \tag{8}$$

$$\hat{f}_1(z) = \hat{K}^{x_{D_t}z} \odot \hat{\alpha} \tag{9}$$

$$\hat{f}_2(z) = \sum_{i=1}^{N} \hat{K}^{(x+\sigma_i)z} \odot \hat{\alpha} \tag{10}$$

where D_t represents the predicted position of the target, and x_{D_t} means an input sample whose center position coordinate corresponds to D_t. What is more, t implies the current frame, X denotes the horizontal coordinate of the target position center, and Y means the vertical coordinate of the target position center.

According to Equations (9) and (10), the spatial–temporal search strategy first predicts the position of the current frame of the target according to the historical information of the target tracking, and then carries out spatial search near the predicted position via the time information to obtain the feature information of the target for determining the final predicted position and tracking the target effectively.

3.4. Retracking Strategy for Target Loss

The KCF algorithm can always track the specified target under normal tracking conditions. However, when the target is obscured by other objects or out of the field of tracker's view, it is easy to lose the target. Furthermore, although YOLOv5s_MSES can detect UAV small targets well, it cannot complete the tracking task of a single target due to the limitations of the target detection algorithm. Therefore, combining two above algorithms is a good solution to the problem of tracking target loss.

The retracking strategy for target loss first utilizes the APCE to determine whether the target is lost. When the cumulative number of APCE values exceeding the threshold reaches 3, the target is considered to be lost, where the threshold is set to 0.15 times the initial APCE value. Subsequently, YOLOv5s_MSES is employed to detect the target. The YOLOv5s_MSES small target detection algorithm based on UAV can extract more small target feature information by improving the network structure of the YOLOv5s algorithm, and then use an attention mechanism to highlight small target feature information to suppress background information, which greatly improves the detection accuracy of small targets. Finally, APCE is used to compare the correlation filter response graphs of all candidate targets detected by YOLOv5s_MSES with the response graph of the initial tracking target, and the candidate target with the highest response pattern matching degree is selected as the follow-up tracking target. As shown in Figures 2 and 3, when the target is lost, there is a significant difference between the maximum value and fluctuation of the response graph. Therefore, the APCE can be utilized to determine whether the target is lost. The calculation formula for the APCE is shown as follows:

$$APCE = \frac{(R_{\max} - R_{\min})^2}{\text{mean}\left(\sum_{ij}(R_{ij} - R_{\min})^2\right)} \tag{11}$$

where R denotes the response matrix, $R_{\min}$ means the minimum value in the response matrix, $R_{\max}$ is the maximum value in the response matrix, and R_{ij} represents the response value at (i, j) in the response matrix.

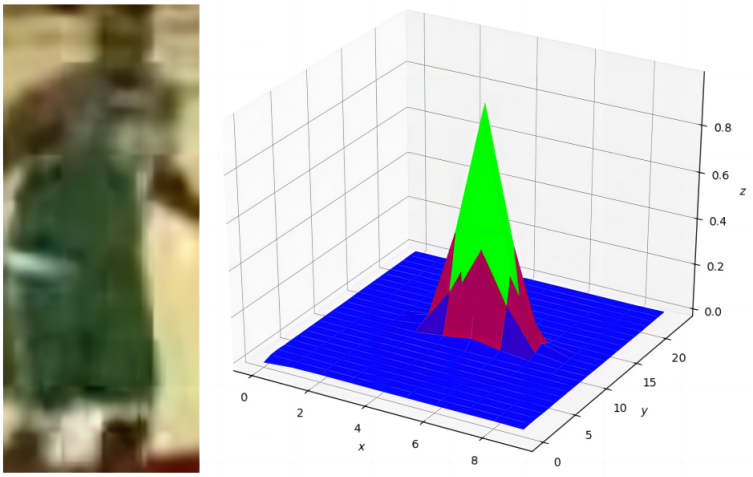


Figure 2. The response plot of the target image.

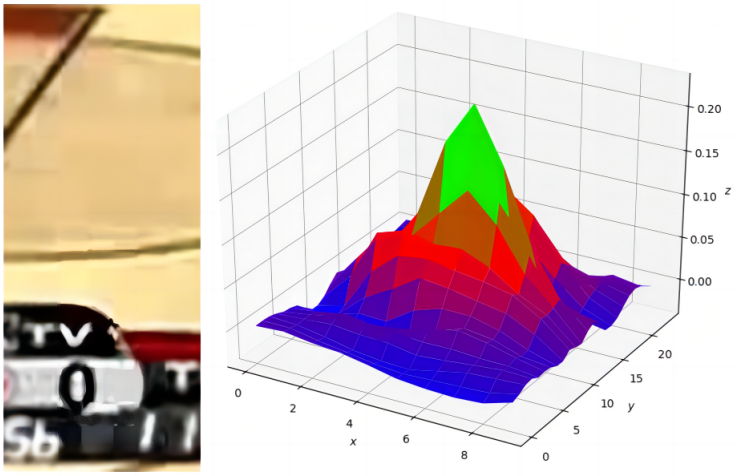


Figure 3. The response plot of the interference image.

3.5. Adaptive Scale Box

The KCF algorithm utilizes a fixed scale box for the target. When the size of the tracking target changes, the target box cannot adapt to the size of the target, thereby affecting tracking accuracy. In Ref. [38], the adaptive multi-scale pyramid (AMP) only used multiple fixed scale factors to resize the scale box, which may not yield the optimal result for the current target. In order to solve this problem, this work proposes an effective method to dynamically adjust the scale factors according to the differences between the maximum response value of different-sized scale boxes and the maximum response value of the current scale box, through which the scale box can be more suitable for the size of the current target. Specifically, the bidirectional scale sampling of the current target region is carried out first, and the peak response values R_S and R_L are extracted by calculating the reduced and enlarged correlation filter response graphs of the current target region. Then, the peak R_S or R_L is used to determine whether the next frame should be scaled down or enlarged. Finally, the scale factor and the difference between the current frame response

peak and the corresponding peak after reduction or amplification are used to determine the actual scaled frame scale size for the next frame. The specific formula is given as follows:

$$B = \begin{cases} \left(e^{(R-R_S)\beta_S} w,\ e^{(R-R_S)\beta_S} h \right),\ R_S > R_L,\ R_S > R \\ \left(e^{(R_L-R)\beta_L} w,\ e^{(R_L-R)\beta_L} h \right),\ R_L > R_S,\ R_L > R \\ (\ w\ ,\ h\ ),\ R > R_L \text{ and } R > R_S \end{cases} \tag{12}$$

where B represents the scale box, and w, h denote the width and height, respectively. Then, R means the maximum response value of the current scale box, R_S denotes the maximum response value of the small scale box, and R_L represents the maximum response value of the large scale box. Moreover, β_S and β_L are the scale factors for the size of the scale box where the size of β_S is chosen as 0.45 and the size of β_L is selected as 0.40.

4. Experiment

4.1. Experimental Data and Parameter Setting

(1) Implementation details: The experimental running environment for this experiment is the Windows 11 operating system, a computer with Nvidia RTX3060 GPU hardware configuration and Intel i7-12700 2.70 GHz CPU, and the compiling software for program code is PyCharm with version 2021.3.3. The algorithm is written in the Python language with version 3.8.5, and the parameter settings are consistent with the open-source KCF code. In sum, this work uses the above experimental environment for the tracking experiment.

(2) OTB100 Dataset: The experimental data environment consists of 98 videos and 100 test scenarios in the OTB100 [39] dataset. Among them, the video sequences Skating2 and Jogging have two test scenes for each case. The dataset covers 11 complex scene attributes, namely lighting changes (IV), SV, OCC, deformation (DEF), motion blur (MB), FM, in-plane rotation (IPR), out-of-plane rotation (OPR), OV, background clutter (BC), and low resolution (LR).

(3) UAV123 Dataset: UAV123 is about 14G in size and contains 123 video sequences [40]. These video sequences cover a variety of complex scenes, including indoor, outdoor, sky, urban and rural, etc. The moving targets in the dataset include people, cars, animals, etc., and they move at different speeds and in different directions in the different environments. Having a high dynamic range and high definition while containing a challenging variety of tracking scenarios makes the UAV123 data ensemble one of the ideal datasets for the evaluation of target tracking algorithms.

(4) Evaluation indicators: In order to quantitatively evaluate the performance of the target tracking algorithms in the following experiments, the tracking results of different trackers are estimated by using precision and success rates [39]. This work employs two evaluation metrics in one-pass evaluation (OPE). One is the average distance precision (DP), which is expressed as the percentage of frames within a given ground truth threshold distance, and the threshold distance is 20 pixels. The other metric is the area under curve (AUC), and it represents the proportion of frames where the bounding box overlap exceeds a specified threshold, determined as 0.50 [41]. Moreover, the FPS indicates the number of frames at per second.

4.2. The Ablation Experiments on OTB100 Dataset

In order to verify each component in our proposed algorithm, the effectiveness of each module is tested on the OTB100 dataset. The ablation results are shown in Table 1.

The ablation results show that the addition of different modules into the KCF algorithm significantly improves the DP and AUC. According to these data, it can be observed that the improved STS module has enhanced the DP by 1.73% and AUC by 1.64% in contrast to the

basic module SI [28], which indicates that incorporating STS effectively mitigates boundary effects, occlusion drift, and other impacts of the algorithm. Since the YOLOv5s_MSES target detection algorithm is trained on the VisDrone2019 dataset, it can only detect a portion of the OTB100 video sequence, as shown in Table 2. After adding the TR module, although only a portion of the video sequence is detected, there exists an improvement of DP by 4.88% and AUC by 3.65%. It demonstrates that combining the target detection algorithm with the CF target tracking one is highly feasible. In comparison with the baseline algorithm, our proposed algorithm has increased the DP by 13.25% and the AUC by 11.98%, indicating its effectiveness and advantage in improving performance. In summary, the proposed algorithm of this work can effectively guide the tracking task of the UAV image target.

Table 1. The ablation experiments on the OTB100 dataset.

Methods	AMP [38]	ASB	SI [28]	STS	TR	DP	AUC
1	-	-	-	-	-	64.54	44.82
2	√	-	-	-	-	68.38	51.01
3	-	√	-	-	-	70.27	51.41
4	-	√	√	-	-	71.18	51.51
5	-	√	-	√	-	72.91	53.15
6	-	√	-	√	√	77.79	56.80

The symbol "√" indicates that the strategy is included in the experiment. The same applies below.

Table 2. The video sequences detected by TR on the OTB100 dataset.

Names					
BlurOwl	Bolt2	CarScale	Couple	Crowds	Dog
DragonBaby	Human3	Human6	Human8	Ironman	Jogging-2
Jump	Jumping	KiteSurf	MountainBike	Panda	Shaking
Skating1	Soccer	Toy	Vase	-	-

4.3. Contrast Experiments on the OTB100 Dataset

Firstly, this work utilizes 100 video sequences from the OTB100 test set to analyze DP and AUC of the ten algorithms. The experimental results are shown in Figures 4 and 5. From these results, the DP by our algorithm amounts to 77.79%, which can be improved by 13.25% when compared with the original KCF one. Then, the AUC of our proposed algorithm reaches 56.80%, which means 11.98% higher than the original KCF one. In contrast with the second ranked SAMF method, our proposed algorithm improves the DP by 3.33% and the AUC by 1.88%. In addition, our algorithm can achieve 53.9 FPS, indicating that the proposed method can achieve the ideal real-time tracking.

Secondly, by comprehensively analyzing the tracking performance of our algorithm in various scenarios, this work assesses the DP and AUC of the algorithm across 11 distinct video attributes. As illustrated in Tables 3 and 4, the scores of various algorithms under the different video attributes are presented. Based on the above experiments, it can be discerned that our proposed algorithm ranks the first among the eight attributes in the DP metric. Under these attributes of IPR, LR and MB, although our algorithm ranks the second, it is not obviously different from the top-ranked algorithm. In the AUC metric, the proposed algorithm ranks the first in ten attributes. Moreover, under the attributes of MB and LR, it is slightly insufficient when being compared with the top-ranked algorithm. Then, based on these comparisons with these original KCF algorithms, our proposed method can achieve an elegant improvement over the 11 attributes. In addition, in the four attributes of OCC, SV, FM and OV, the values of DP and AUC are much higher than those of the second-ranked algorithm. Thus, it is concluded that the proposed modules of this work can effectively help to achieve more accurate target tracking.

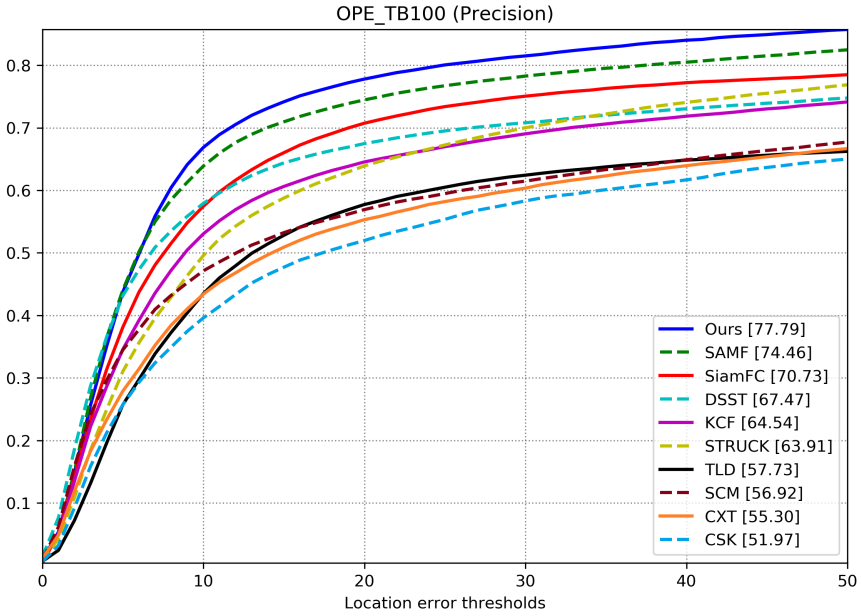


Figure 4. The DP performance on the OTB100 dataset.

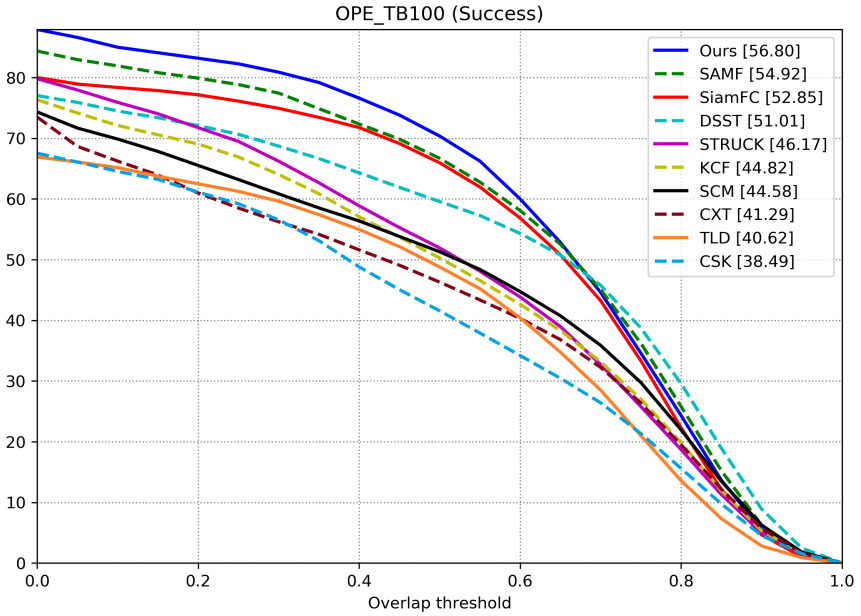


Figure 5. The AUC performance on the OTB100 dataset.

Table 3. The DP values under 11 attributes on the OTB100 dataset.

Methods	Ours	SAMF	SiamFC	DSST	KCF	STRUCK	TLD	SCM	CXT	CSK
IV	70.74	66.11	63.85	66.11	59.53	50.65	50.26	56.21	47.68	45.07
SV	73.31	70.97	69.39	64.34	61.74	60.06	55.17	55.47	53.33	46.33
OCC	75.16	71.25	63.19	58.80	59.32	52.45	49.94	54.24	42.89	43.08
DEF	70.37	66.29	65.41	52.02	56.73	51.33	45.18	52.46	39.15	43.62
MB	69.77	65.54	70.09	57.96	57.20	59.49	52.73	31.94	56.54	38.85
FM	73.27	64.94	68.10	56.09	57.25	62.06	53.53	35.75	54.91	42.01
IPR	70.22	70.83	67.59	67.62	62.67	64.05	58.87	53.43	60.98	53.14
OPR	75.20	73.84	68.89	64.10	63.17	60.60	55.65	57.44	52.76	49.94
OV	65.70	62.75	54.63	48.06	52.82	50.31	47.55	43.55	41.64	31.52
BC	75.19	67.66	57.93	69.09	58.61	55.01	43.24	55.00	43.78	57.42
LR	66.77	70.99	61.85	54.96	55.37	62.82	53.66	55.75	49.24	36.72

Table 4. The ACU values under 11 attributes on the OTB100 dataset.

Methods	Ours	SAMF	SiamFC	DSST	KCF	STRUCK	TLD	SCM	CXT	CSK
IV	52.07	49.37	47.13	49.37	37.85	39.56	36.07	45.43	35.53	34.20
SV	52.63	50.15	50.97	47.46	38.50	40.75	37.21	43.28	38.76	32.84
OCC	55.37	53.60	48.18	45.56	40.81	38.56	34.48	42.08	33.02	33.36
DEF	51.75	49.50	47.79	40.62	38.94	37.66	31.90	39.09	29.71	33.06
MB	52.60	52.10	54.09	47.93	40.26	46.09	40.34	31.41	40.78	32.71
FM	55.25	50.04	52.31	45.48	41.39	46.16	39.64	33.08	40.53	33.83
IPR	51.73	51.64	49.80	50.21	43.12	45.74	40.56	41.00	45.41	39.28
OPR	54.62	54.05	50.13	47.28	42.79	43.42	37.39	43.96	39.59	36.18
OV	49.19	48.03	40.16	38.59	36.85	38.42	34.23	34.81	34.38	28.31
BC	53.77	51.03	42.06	50.03	40.76	42.29	31.62	43.78	34.01	41.05
LR	43.83	45.94	42.69	37.89	28.62	34.71	34.54	38.10	36.64	25.08

Finally, Tables 5 and 6 clearly show the improvement of the addition of several modules on scaling, target loss, and other issues. It is noteworthy that the inclusion of the target detection algorithm has significantly enhanced both DP and AUC metrics in terms of the OCC and OV attributes, which indicates that the combination of KCF and YOLOv5s_MSES can effectively solve the problem of losing track of the target.

Table 5. The comparisons of DP indicators based on four attributes of OTB100 dataset.

Methods	ASB	SI [28]	STS	TR	SV	FM	OCC	OV
1	-	-	-	-	60.46	56.82	56.50	48.61
2	√	-	-	-	65.30	62.08	65.12	54.47
3	√	√	-	-	66.43	63.82	65.24	56.52
3	√	-	√	-	69.03	67.10	68.83	60.46
4	√	-	√	√	73.31	73.27	75.16	65.70

Table 6. The comparisons of AUC indicators based on four attributes of OTB100 dataset.

Methods	ASB	SI [28]	STS	TR	SV	FM	OCC	OV
1	-	-	-	-	38.50	41.39	40.81	36.85
2	√	-	-	-	46.54	47.37	47.76	41.33
3	√	√	-	-	46.78	48.12	47.43	42.03
3	√	-	√	-	49.04	51.20	50.75	45.85
4	√	-	√	√	52.63	55.25	55.37	49.19

4.4. Contrast Experiments on the β_S and β_L Parameters

In the experiments of scale factor selection, the benchmark is the KCF algorithm. In order to facilitate selection, the mean values of AUC and DP are used as evaluation criteria. As can be seen from Tables 7 and 8, when the β_S value is 0.45 and β_L value is 0.40, the comprehensive performance of the ASB method is the best.

Table 7. The comparison experiments of β_S scale factor in ASB method based on OTB100 dataset.

Methods	β_S	β_L	DP	AUC	Average Values
1	0.20	0.40	68.25	48.95	58.60
2	0.30	0.40	70.41	51.02	60.72
3	0.40	0.40	70.32	51.06	60.69
4	0.42	0.40	70.18	51.30	60.74
5	0.45	0.40	70.27	51.41	60.84

Table 7. *Cont.*

Methods	β_S	β_L	DP	AUC	Average Values
6	0.47	0.40	69.75	50.59	60.17
7	0.50	0.40	69.65	51.13	60.39

Table 8. The comparison experiments of β_L scale factor in ASB method based on OTB100 dataset.

Methods	β_S	β_L	DP	AUC	Average Values
1	0.45	0.20	67.77	49.15	58.46
2	0.45	0.30	70.15	51.08	60.62
3	0.45	0.35	68.95	50.32	59.64
4	0.45	0.37	69.01	50.63	59.82
5	0.45	0.40	70.27	51.41	60.84
6	0.45	0.42	70.07	50.80	60.44
7	0.45	0.45	67.83	50.14	58.99
8	0.45	0.50	69.96	51.15	60.56

4.5. Contrast Experiments on the σ_i Parameter

In the comparative experiment of the selection of the size of the surrounding search range σ_i in the STS method, the benchmark is that the KCF algorithm has been combined with the ASB method. In order to facilitate the selection, the average value of AUC and DP is used as the evaluation standard. As can be seen from Table 9, when the value of σ_i is 0.25, the comprehensive performance of the STS method is the best.

Table 9. The comparison experiments of σ_i in STS method based on OTB100 dataset.

Methods	σ_i	DP	AUC	Average Values
1	0.20	72.07	52.52	62.30
2	0.22	69.27	51.82	60.55
3	0.25	72.91	53.15	63.03
4	0.27	70.96	51.93	61.45
5	0.30	70.70	51.05	60.88

4.6. Experiments on UAV123 Dataset

Since the UAV123 dataset is taken from the perspective of the UAV, in order to verify the effectiveness of our proposed algorithm, the ablation experiments are carried out on UAV123. The ablation results are shown in Table 10, which shows that when different modules are added into the KCF algorithm, the DP and AUC of our proposed algorithm are significantly increased. Then, in comparison with the baseline algorithm, the DP of our proposed algorithm can be improved by 14.07%, and the AUC is improved by 14.39%, indicating that our algorithm effectively promotes the performance of the basic KCF algorithm.

In order to reflect the effectiveness of the combination of YOLOv5s_MSES and KCF, Table 11 shows the comparison results of DP indicators after partially adding the target detection algorithm. It follows from the data that our proposed method can improve the performance of the target tracking algorithm very effectively.

Table 10. The ablation experiments on the UAV123 dataset.

Methods	ASB	STS	TR	DP	AUC
1	-	-	-	53.54	33.14
2	√	-	-	55.78	39.05
3	√	√	-	58.62	40.91
4	√	√	√	67.61	47.53

Table 11. The comparisons of DP indicators on the UAV123 dataset (The wakeboard is abbreviated as wb.).

Videos	car7	car9	group2_1	person10	person16	truck4_2	wb5	wb8
KCF + ASB + STS	15.02	36.00	8.20	17.65	10.70	4.65	3.80	25.15
Ours	54.71	34.58	68.31	51.79	42.92	55.63	34.19	40.33

4.7. Comprehensive Performance Comparison

As shown in Tables 12 and 13, DP, AUC and FPS are selected as evaluation criteria in the experiment. In the OTB100 dataset and UAV123 dataset, although the DP and AUC of these algorithms (such as ASTSAELT, STCDFF, MixFormer-1k, NeighborTrack, and EFSCF) are higher than ours, the FPS is too low to meet the real-time requirements of tracking. In the UAV target tracking task, if the algorithm cannot meet the real-time requirement, it is difficult for the UAV to complete the task requirements in time. Thus, our proposed algorithm more meets the real-time requirements in the comparisons with the two datasets. Moreover, in contrast to [20,42], our algorithm achieves higher DP, AUC and FPS. In summary, it is concluded that our algorithm can achieve overall better performance.

Table 12. The comprehensive performance comparison experiments on OTB100 dataset.

Methods	STAPLE$_+$SS 2021 [20]	ASTSAELT 2023 [28]	STCDFF 2022 [43]	Ours -
DP	70.5	89.8	91.8	77.8
AUC	54.7	67.5	69.5	56.8
FPS	42.6	2.7	3.4	53.9

Table 13. The comprehensive performance comparison experiments on UAV123 dataset.

Methods	EFSCF 2023 [21]	LADCF 2023 [21]	SPMCF 2023 [42]	MixFormer_1k 2022 [44]	NeighborTrack 2022 [45]	FRD_BACF 2022 [46]	Ours -
DP	70.4	65.8	63.0	89.5	93.4	67.4	67.6
AUC	49.6	49.8	41.7	68.7	72.6	47.9	47.5
FPS	18.0	7.0	13.0	2.0	1.3	26.7	37.5

4.8. Experimental Effect and Analysis

Four kinds of key frames are selected for the tracking experiments by using some existing algorithms. Firstly, Figure 6 shows the tracking results based on the OTB100 dataset, where the green boxes represent the results for our algorithm. The red boxes indicate the results for the KCF algorithm and the blue boxes denote the results for the SiamFC algorithm. In Figure 6a–c, when the tracking target changes in terms of light and shadow, the SiamFC algorithm cannot be adapted and thus the tracking target is lost. Even though KCF can continue to track, the scale box cannot be changed adaptively, while our proposed algorithm can adapt effectively. In Figure 6d–f, when there exists occlusion of similar targets, the tracking error of the SiamFC algorithm is large, while our algorithm can

track the target in a normal way. In addition, it can be seen from Figure 6 that the proposed algorithm also has a good effect on scale change.

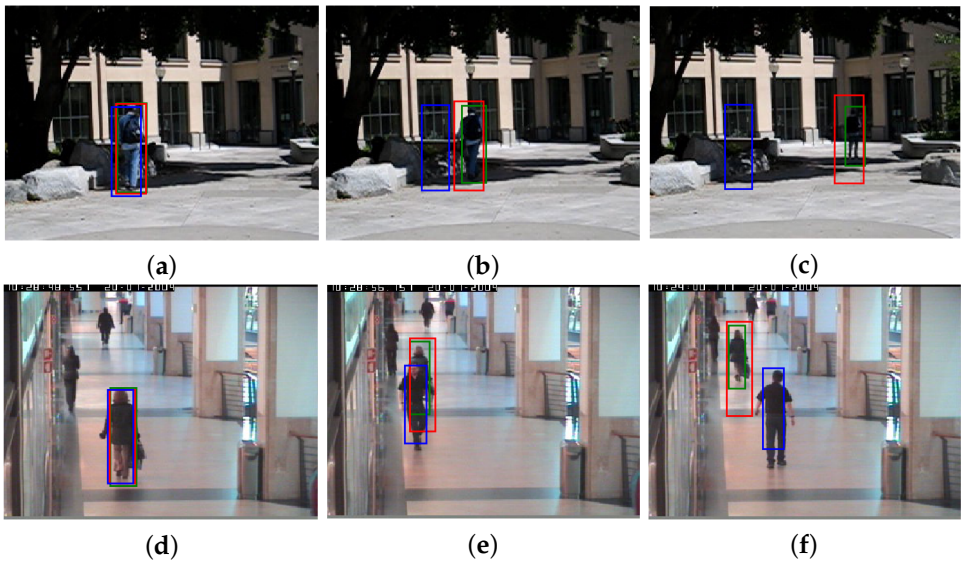


Figure 6. The OTB100 video sequences. (**a**) Human8: Frame 6; (**b**) Human8: Frame 24; (**c**) Human8: Frame 94; (**d**) Walking2: Frame 9; (**e**) Walking2: Frame 200; (**f**) Walking2: Frame 314.

Figure 7 shows the tracking results on the UAV123 dataset, where the green boxes denote the results for our algorithm. The red boxes indicate the results for the KCF + ASB + STS algorithm. As can be seen from Figure 7a–c, the UAV first tracks the rear of the car. Thus, when the tail of the car is out of the view range of the UAV, KCF + ASB + STS cannot perform tracking, while our algorithm can exploit YOLOv5s_MSES to redetect the target and continue to track. As can be seen from Figure 7d–f, the target tracked by the UAV aims for the person in green. Then, when the group is occluded by trees, both KCF + ASB + STS and our algorithms cannot continue the tracking. Yet, when the target reappears in view, our algorithm can still utilize YOLOv5s_MSES to accurately redetect the target and continue tracking. The experimental results show that our algorithms are more feasible and effective for solving the problem of lost target retracking.

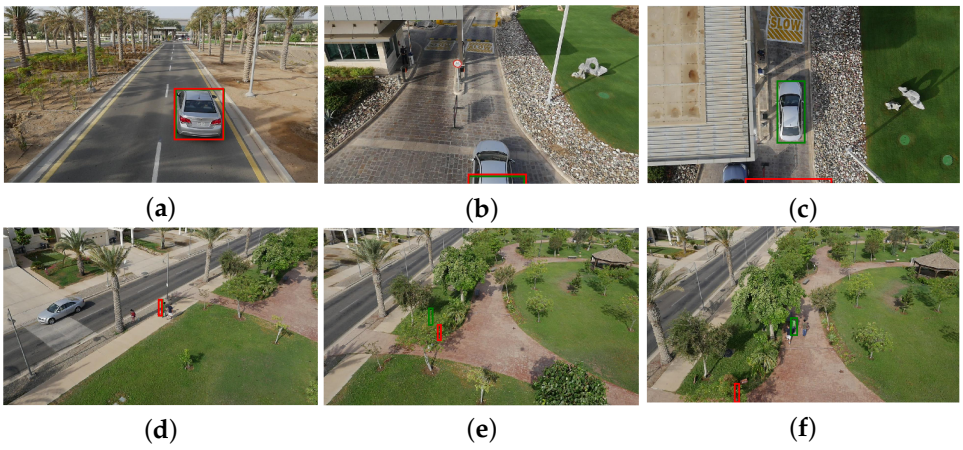


Figure 7. The UAV123 video sequences. (**a**) Car6_2: Frame 29; (**b**) Car6_2: Frame 703; (**c**) Car6_2: Frame 958; (**d**) Group2_3: Frame 29; (**e**) Group2_3: Frame 340; (**f**) Group2_3: Frame 622.

Finally, the dynamic display of the complete video sequence tracking trajectory is presented, especially focusing on stable tracking after successful retracking. The

link to the video is https://github.com/csh265/Complete_video_sequence (accessed on 26 April 2025).

5. Conclusions

This work has proposed an improved KCF tracking algorithm based on the target detection algorithm to address the problem of the difficulty involving retracking the target when it is lost and tracking drift caused by boundary effects. First of all, an adaptive scale box was proposed to change the scale of the target box according to the changing scale of the target. Then, based on the spatial information around the current frame of the target and the temporal information of the previous frames, a spatio-temporal search strategy was introduced to suppress the influence of boundary effects. Additionally, the KCF algorithm integrated the YOLOv5s_MSES target detection one and the APCE to enable the UAV to retrack the target when it is lost. Finally, The experimental results of the UAV123 dataset showed that although the YOLOv5s_MSES algorithm could only detect part of the videos, the DP and the AUC were, respectively, increased by 8.99% and 6.62% on the basis of the KCF + ASB + STS detection results. It is verified that the combination of the CF target tracking algorithm and target detection one could achieve great feasibility. Therefore, our proposed algorithm can better solve the problem induced by target loss, scale change, and boundary effects, meet the real-time requirements with high precision, and demonstrate superior comprehensive performance over some existing algorithms.

Author Contributions: Conceptualization, S.C. and T.W.; methodology, S.C.; software, S.C.; validation, S.C., T.W. and T.L.; formal analysis, T.W.; investigation, T.W. and S.F.; writing—original draft preparation, S.C; writing—review and editing, T.L. and S.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundations of China (Nos. 62073164, 61922042), Aeronautical Science Foundation of China (No. 2023Z032052001).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

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

References

1. Yan, B.; Paolini, E.; Xu, L.; Lu, H. A target detection and tracking method for multiple radar systems. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–21. [CrossRef]
2. Fang, H.; Liao, G.; Liu, Y.; Zeng, C. Shadow-assisted moving target tracking based on multi-discriminant correlation filters network in video SAR. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 1–5.
3. Lee, M.; Lin, C. Object tracking for an autonomous unmanned surface vehicle. *Machines* **2022**, *10*, 378. [CrossRef]
4. Hayat, M.; Ali, A.; Khan, B.; Mehmood, K.; Ullah, K.; Amir, M. An improved spatial–temporal regularization method for visual object tracking. *Signal Image Video Process.* **2024**, *18*, 2065–2077. [CrossRef]
5. Song, P.; Li, P.; Dai, L.; Wang, T.; Chen, Z. Boosting R-CNN: Reweighting R-CNN samples by RPN's error for underwater object detection. *Neurocomputing* **2023**, *530*, 150–164. [CrossRef]
6. Zhang, R.; Shao, Z.; Huang, X.; Wang, J.; Wang, Y.; Li, D. Adaptive dense pyramid network for object detection in UAV imagery. *Neurocomputing* **2022**, *489*, 377–389. [CrossRef]
7. Wang, J.; Meng, C.; Deng, C.; Wang, Y. Learning convolutional self-attention module for unmanned aerial vehicle tracking. *Signal Image Video Process.* **2023**, *17*, 2323–2331. [CrossRef]
8. Wang, H.; Qi, L.; Qu, H.; Ma, W.; Yuan, W.; Hao, W. End-to-end wavelet block feature purification network for efficient and effective UAV object tracking. *J. Vis. Commun. Image Represent.* **2023**, *97*, 103950. [CrossRef]
9. Daramouskas, I.; Meimetis, D.; Patrinopoulou, N.; Lappas, V.; Kostopoulos, V. Camera-based local and global target detection, tracking, and localization techniques for UAVs. *Machines* **2023**, *11*, 315. [CrossRef]

10. Fu, C.; Li, B.; Ding, F.; Lin, F.; Lu, G. Correlation filters for unmanned aerial vehicle-based aerial tracking: A review and experimental evaluation. *IEEE Geosci. Remote Sens. Mag.* **2021**, *10*, 125–160. [CrossRef]
11. Zhang, F.; Ma, S.; Qiu, Z.; Qi, T. Learning target-aware background-suppressed correlation filters with dual regression for real-time UAV tracking. *Signal Process.* **2022**, *191*, 108352. [CrossRef]
12. Li, S.; Liu, Y.; Zhao, Q.; Feng, Z. Learning residue-aware correlation filters and refining scale for real-time UAV tracking. *Pattern Recognit.* **2022**, *127*, 108614. [CrossRef]
13. Bolme, D.; Beveridge, J.; Draper, B.; Lui, Y. Visual object tracking using adaptive correlation filters. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2544–2550.
14. Henriques, J.; 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]
15. Danelljan, M.; Hager, G.; Shahbaz Khan, F.; Felsberg, M. Learning spatially regularized correlation filters for visual tracking. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4310–4318.
16. Danelljan, M.; Hager, G.; Khan, F.; Felsberg, M. Accurate scale estimation for robust visual tracking. In Proceedings of the British Machine Vision Conference, Nottingham, UK, 1–5 September 2014.
17. Bertinetto, L.; Valmadre, J.; Golodetz, S.; Miksik, O.; Torr, P.H. 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.
18. Danelljan, M.; Shahbaz Khan, F.; 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, Columbus, OH, USA, 23–28 June 2014; pp. 1090–1097.
19. Lukezic, A.; Vojir, T.; Cehovin Zajc, L.; Matas, J.; Kristan, M. Discriminative correlation filter with channel and spatial reliability. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6309–6318.
20. Tai, Y.; Tan, Y.; Xiong, S.; Tian, J. Subspace reconstruction based correlation filter for object tracking. *Comput. Vis. Image Underst.* **2021**, *212*, 103272. [CrossRef]
21. Wen, J.; Chu, H.; Lai, Z.; Xu, T.; Shen, L. Enhanced robust spatial feature selection and correlation filter learning for UAV tracking. *Neural Netw.* **2023**, *161*, 39–54. [CrossRef]
22. Moorthy, S.; Joo, Y. Learning dynamic spatial-temporal regularized correlation filter tracking with response deviation suppression via multi-feature fusion. *Neural Netw.* **2023**, *167*, 360–379. [CrossRef]
23. Ma, H.; Acton, S.; Lin, Z. Situp: Scale invariant tracking using average peak-to-correlation energy. *IEEE Trans. Image Process.* **2020**, *29*, 3546–3557. [CrossRef]
24. Cao, S.; Wang, T.; Li, T.; Mao, Z. UAV small target detection algorithm based on an improved YOLOv5s model. *J. Vis. Commun. Image Represent.* **2023**, *97*, 103936. [CrossRef]
25. Wang, H.; Wang, Z.; Fang, B.; Bu, Y. Temporal–spatial consistency of self-adaptive target response for long-term correlation filter tracking. *Signal Image Video Process.* **2020**, *14*, 639–644. [CrossRef]
26. Liu, P.; Liu, C.; Zhao, W.; Tang, X. Multi-level context-adaptive correlation tracking. *Pattern Recognit.* **2019**, *87*, 216–225. [CrossRef]
27. Elayaperumal, D.; Joo, Y. Robust visual object tracking using context-based spatial variation via multi-feature fusion. *Inf. Sci.* **2021**, *577*, 467–482. [CrossRef]
28. Moorthy, S.; Joo, Y. Adaptive spatial-temporal surrounding-aware correlation filter tracking via ensemble learning. *Pattern Recognit.* **2023**, *139*, 109457. [CrossRef]
29. Li, S.; Wu, O.; Zhu, C.; Chang, H. Visual object tracking using spatial context information and global tracking skills. *Comput. Vis. Image Underst.* **2014**, *125*, 1–15. [CrossRef]
30. Yan, P.; Yao, S.; Zhu, Q.; Zhang, T.; Cui, W. Real-time detection and tracking of infrared small targets based on grid fast density peaks searching and improved KCF. *Infrared Phys. Technol.* **2022**, *123*, 104181. [CrossRef]
31. Yang, X.; Li, S.; Yu, J.; Zhang, K.; Yang, J.; Yan, J. GF-KCF: Aerial infrared target tracking algorithm based on kernel correlation filters under complex interference environment. *Infrared Phys. Technol.* **2021**, *119*, 103958. [CrossRef]
32. Jia, Y.; Zhang, Y.; Zhou, C.; Yang, Y. Helop: Multi-target tracking based on heuristic empirical learning algorithm and occlusion processing. *Displays* **2023**, *79*, 102488. [CrossRef]
33. Cai, D.; Guan, Z.; Bamisile, O.; Zhang, W.; Li, J.; Zhang, Z.; Wu, J.; Chang, Z.; Huang, Q. Multi-objective tracking for smart substation onsite surveillance based on YOLO approach and AKCF. *Energy Rep.* **2023**, *9*, 1429–1438. [CrossRef]
34. Chen, R.; Wu, J.; Peng, Y.; Li, Z.; Shang, H. Detection and tracking of floating objects based on spatial-temporal information fusion. *Expert Syst. Appl.* **2023**, *225*, 120185. [CrossRef]
35. Kinasih, F.; Machbub, C.; Yulianti, L.; Rohman, A. Two-stage multiple object detection using CNN and correlative filter for accuracy improvement. *Heliyon* **2023**, *9*, e12716. [CrossRef]

36. Wang, M.; Yang, W.; Wang, L.; Chen, D.; Wei, F.; KeZiErBieKe, H.; Liao, Y. FE-YOLOv5: Feature enhancement network based on YOLOv5 for small object detection. *J. Vis. Commun. Image Represent.* **2023**, *90*, 103752. [CrossRef]
37. Liu, C.; Yang, D.; Tang, L.; Zhou, X.; Deng, Y. A lightweight object detector based on spatial coordinate self-attention for UAV aerial images. *Remote Sens.* **2022**, *15*, 83. [CrossRef]
38. Zheng, K.; Zhang, Z.; Qiu, C. A fast adaptive multi-scale kernel correlation filter tracker for rigid object. *Sensors* **2022**, *22*, 7812. [CrossRef] [PubMed]
39. Wu, Y.; Lim, J.; Yang, M. Online object tracking: A benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2411–2418.
40. Mueller, M.; Smith, N.; Ghanem, B. A benchmark and simulator for UAV tracking. In *Computer Vision–ECCV 2016, Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016*; Proceedings, Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 445–461.
41. Babenko, B.; Yang, M.; Belongie, S. Robust object tracking with online multiple instance learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *33*, 1619–1632. [CrossRef]
42. Zhang, Y.; Zheng, Y. Object tracking in UAV videos by multi-feature correlation filters with saliency proposals. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2023**, *16*, 5538–5548. [CrossRef]
43. Nai, K.; Li, Z.; Wang, H. Dynamic feature fusion with spatial-temporal context for robust object tracking. *Pattern Recognit.* **2022**, *130*, 108775. [CrossRef]
44. Cui, Y.; Jiang, C.; Wang, L.; Wu, G. Mixformer: End-to-end tracking with iterative mixed attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 13608–13618.
45. Chen, Y.; Wang, C.; Yang, C.; Chang, H.; Lin, Y.; Chuang, Y.; Liao, H. Neighbortrack: Improving single object tracking by bipartite matching with neighbor tracklets. *arXiv* **2022**, arXiv:2211.06663.
46. Lin, B.; Bai, Y.; Bai, B.; Li, Y. Robust correlation tracking for UAV with feature integration and response map enhancement. *Remote Sens.* **2022**, *14*, 4073. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.