drones

MDPI

Article

UAV Visual Object Tracking Based on Spatio-Temporal Context

Yongxiang He, Chuang Chao, Zhao Zhang, Hongwu Guo * and Jianjun Ma

College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410000, China; heyongxiang@nudt.edu.cn (Y.H.); chaochuang886@hotmail.com (C.C.); zhangzhao21@nudt.edu.cn (Z.Z.); abellp@buaa.edu.cn (J.M.)

* Correspondence: guohongwu@nudt.edu.cn

Abstract: To balance the real-time and robustness of UAV visual tracking on a single CPU, this paper proposes an object tracker based on spatio-temporal context (STCT). STCT integrates the correlation filter and Siamese network into a unified framework and introduces the target's motion model, enabling the tracker to adapt to target scale variations and effectively address challenges posed by rapid target motion, etc. Furthermore, a spatio-temporal regularization term based on the dynamic attention mechanism is proposed, and it is introduced into the correlation filter to suppress the aberrance of the response map. The filter solution is provided through the alternating direction method of multipliers (ADMM). In addition, to ensure efficiency, this paper proposes the average maximum response value-related energy (AMRE) for adaptive tracking state evaluation, which considers the time context of the tracking process in STCT. Experimental results show that the proposed STCT tracker can achieve a favorable balance between tracking robustness and real-time performance for UAV object tracking while running at ∼38 frames/s on a low-cost CPU.

Keywords: UAV object tracking; the correlation filter; Siamese networks; spatio-temporal context

1. Introduction

Unmanned aerial vehicles (UAVs) are favored in both military and civilian fields due to their small size, flexible movement, and low cost [1,2]. Visual object tracking plays a pivotal role in the collaborative reconnaissance process of unmanned aerial vehicles. The primary objective of visual object tracking is to accurately predict the spatial location in successive consecutive frames, leveraging an initial frame as a basis. Therefore, robust and real-time object tracking algorithms are crucial [3,4].

However, the challenges of UAV visual object tracking are more complex compared with general tracking scenarios. It is mainly manifested in the following aspects: (i) Rapid motion problem. UAVs are highly maneuverable. The rapid motion of the target, as well as the relative motion of the UAV and target, makes the tracking task more difficult. (ii) Scale variation and aspect ratio change [5]. During the course of movement, the morphology of the target may change. In addition, viewpoint variations lead to substantial alterations of the target scale. The majority of contemporary real-time object trackers that are dependent on a sole CPU have a fixed size or aspect ratio of the target tracking frame, which cannot cope well with the challenge posed by target scale variations and aspect ratio changes. (iii) Partial occlusion and target appearance change. In scenarios where the target undergoes partial occlusion or moves partially out of the field of view, its appearance may experience significant deformations. Failing to adaptively update the tracking model in real-time can result in the failure of the tracking process. (iv) Scarcity of computational resources. According to practical applications, object tracking algorithms need to be deployed on onboard processors or ground portable mobile platforms. However, due to limited battery capacity and constrained computational resources, higher requirements are placed on the real-time performance of target tracking algorithms [6]. Therefore, the pursuit of

Citation: He, Y.; Chao, C.; Zhang, Z.; Guo, H.; Ma, J. UAV Visual Object Tracking Based on Spatio-Temporal Context. *Drones* **2024**, *8*, 700. https://doi.org/10.3390/drones8120700

Academic Editor: Andrey V. Savkin

Received: 9 October 2024
Revised: 16 November 2024
Accepted: 21 November 2024
Published: 22 November 2024

stable, precise, and instantaneous object tracking amidst intricate surroundings stands as a formidable task.

Currently, there are two main classes of mainstream object tracking algorithms: discriminative correlation filter (CF) trackers and Siamese network-based trackers. Due to their remarkable computational efficiency, correlation filters have garnered immense prominence in the realm of UAV object tracking. Their extensive adoption within this domain stems from their capacity to handle real-time tracking scenarios adeptly while minimizing computational overhead. It adopts an 'online learning' mechanism, which leverages Fourier transformation to perform complex convolution operations in the frequency domain rather than the spatial domain. This innovative technique significantly boosts the computational speed. Although the traditional CF can achieve relatively strong real-time performance, its limited search regions and issues such as boundary effects result in relatively lower success rates and precision in complex and dynamic environments [7].

Recently, emerging deep learning techniques have attained remarkable triumphs in the domains of image classification, target detection, and image segmentation [8]. The deep learning theory provides new ideas to solve the object tracking problem. Among them, Siamese network-based trackers are getting more and more attention. It adopts an 'offline learning' mechanism. The target deep features are extracted by convolutional neural networks (CNNs), which have stronger adaptability.

Despite the significant improvement in tracking success and precision of Siamese networks, the ensuing problem is that the large network structure takes up a large amount of computational resources. However, the high performance of Siamese networks is usually realized on GPUs, and such algorithms consume relatively large amounts of power. The scarcity of computational resources in processors based on a single CPU limits the application of deep learning-based object tracking algorithms. Therefore, the pursuit of robust and real-time object tracking in intricate scenarios continues to present an arduous challenge that demands meticulous attention and resolution.

Aiming at the limitations observed in current methodologies, this paper proposes a real-time object tracking method for UAVs. It fuses spatial and temporal contexts by considering the spatial contextual information of the tracking process, as well as the motion information and global response changes in the time series. Robust tracking based on a single CPU has been achieved while ensuring real-time tracking performance.

The main contributions are summarized as follows:

(i) STCT integrates correlation filtering and Siamese networks into a unified vision architecture to strike a balance between robustness and real-time performance in a single CPU platform.

(ii) A spatio-temporal regularity term based on a dynamic attention mechanism is proposed to suppress aberrations and improve tracking robustness.

(iii) The average maximum response value related energy (AMRE) is proposed for adaptive tracking state assessment, making the tracking algorithm more efficient.

The subsequent segments of this paper are organized as follows: Section 2 delves into an exploration of pertinent literature in the field. Section 3 elucidates the proposed STCT tracker, providing comprehensive details regarding its underlying methodology. Section 4 showcases the experiments and corresponding results. Section 5 is a discussion about the proposed methods. Finally, Section 6 encapsulates the paper with a conclusive summary.

2. Related Works

For completeness, a concise overview of the most relevant works is presented as follows.

2.1. Correlation Filters for Online Object Tracking

In 2014, MOSSE [9] applied the correlation filtering method to object tracking for the first time. It achieved online template updates by minimizing the sum of squares error. On this basis, Henriques et al. proposed KCF [10], which adds a regularization term to

the optimization equation to prevent overfitting. It constructs the correlation filter by ridge regression. For feature extraction, KCF introduces a multi-channel histogram of oriented gradient (HOG) feature to describe the target shape information, which replaces the single-channel grayscale feature and extends the feature to a multi-channel nonlinear feature space. In addition, KCF utilizes cyclic shift sampling to construct dense samples, overcoming the problem of insufficient training samples in MOSSE. DSST [11] pays more attention to the target scale estimation. It considers the object tracking issue as a pair of distinct challenges: the translation of the target's center position and the adjustment of its scale variation. The displacement correlation filter is trained using the HOG features, and then another scale filter is trained for predicting the scale change, which improves the precision and success rate. However, the speed is slightly slower compared with KCF. To improve the discriminative ability, the SRDCF [12] algorithm introduces a spatial domain regularization component and filter coefficients that penalize the boundary region based on KCF optimization objectives. The Gauss–Seidel iteration is used for filter solving. However, the real-time performance of SRDCF is poor. Subsequently, Danelljan et al. proposed ECO-HC [13], which provides a new optimization method with respect to the dimensions of the model, the size of the training set, and the update strategy of the model, respectively. ECO-HC achieved better results in UAV object tracking tasks. In 2017, Gloogahi et al. proposed the background awareness correlation filter (BACF) [14], which abandons the negative samples generated through cyclic shifting. It employs negative samples generated by real shifting instead, which contain a larger search region and can better suppress the boundary effect. Since then, some methods [15–17] improved on BACF to enhance the adaptability of target tracking. CSR-DCF [18] is different from BACF in that it adds a mask to the filter from the spatial domain to suppress the boundary effect. In this case, the mask is constructed from color histograms of the foreground and background. CSR-DCF constructs the weighting coefficients of different channels using the response map information of different channels. STRCF [19] and RISTrack [20] improve on SRDCF by introducing regular term constraints. In addition, STRCF employs ADMM iteration to solve the filter, thereby enhancing its real-time capabilities.

To summarize, the above classical correlation filters possess their individual strengths and weaknesses in terms of performance on object tracking datasets in different domains. So far, on the dataset of UAV object tracking, ECO-HC and STRCF showcase impressive results on CPU [21].

2.2. Siamese Network-Based Trackers for Online Object Tracking [22,23]

Except for some correlation filters that introduce the deep-CNN in feature extraction, such as ECO [13], DeepSTRCF [19], etc., the most widely researched deep learning object tracking algorithms are based on the structure of Siamese networks [24]. In 2016, SiamFC [25] first applied the twinned convolutional neural network structure to object tracking. The Siamese network uses two identical CNN branches, one of which extracts the target features and the other extracts the search image features. Finally, the response map of the target position is generated through feature matching using the cross-correlation operation. However, SiamFC encounters challenges in effectively handling issues related to target scale variations. For this reason, SiamRPN [26] borrows the region proposal network (RPN) for predicting target scales in a new image. Instead of directly performing the mutual correlation operation after its backbone network extracts template features and search image features, both feature maps are fed into respective branches of the RPN, wherein they perform the discernment of target probability and the refinement of the target bounding box through meticulous prediction and regression. In addition, advanced deep trackers like SiamRPN++ [27] and SiamMask [28] are built on the foundation of Siamese network architecture. Currently, the feature extraction module of Siamese networks mainly uses a pre-trained backbone such as AlexNet [29], VGG [30], and ResNet [31]. The feature extraction network serves as the very essence of the Siamese network, playing a pivotal role in the extraction of target templates and the search for image features. The pre-trained

feature extraction network only needs to traverse the computation in a feed-forward manner to obtain the similarity score for target localization. However, the feature extraction network has a large structure and parameter size. Its superior performance is usually realized on GPUs with high performance. It cannot meet real-time requirements when running on processors with only a single CPU. This limitation hinders the application of Siamese network trackers within the realm of UAV visual object tracking.

2.3. Discussion of Related Works

In the realm of visual object tracking, prior to 2020, the majority of research centered around correlation filters, which can be implemented to achieve real-time performance on CPUs. Although the correlation filter has made some achievements in the field of object tracking, it still cannot well solve the problems of restricted search area and boundary effects. Efforts should be devoted towards refining the tracking precision and success rate for further improvements.

However, since 2020, Siamese neural networks have garnered widespread attention for their remarkable robustness and have progressively surpassed correlation filters. In fact, most Siamese network trackers have demonstrated superior performance in both tracking success rates and accuracy compared with correlation filters so far.

However, the Siamese network tracker has a large neural network structure and parameters. It is usually deployed on GPU-based platforms. The tracking speed is very slow on CPU-based platforms. In some areas of object tracking tasks, platforms with larger computational resources can be provided for Siamese network trackers. However, the research context of this paper is UAV visual object tracking, where the computational resources of an onboard processor carrying only a CPU are limited. Therefore, in the field of UAV visual object tracking, despite the excellent performance of the Siamese network tracker, its weakness in achieving real-time on CPU-based processors is a key limiting factor for its application.

To address the above issues, the algorithm presented in this paper is a real-time tracker proposed for CPU platforms. The importance of CPU deployment is reflected in the following points:

(i) Computational resource limitations: In the field of visual object tracking for UAVs, onboard processors or portable ground processors are often limited by their computational capabilities and may only be equipped with CPUs, lacking GPUs. In such cases, the effectiveness and computational efficiency of the algorithms must be validated in an environment that relies solely on CPU resources.

(ii) The payload limitations of small UAVs: Processing boards equipped with GPUs are heavier, which poses a challenge to the payload capacity of small UAVs. It is essential to consider their effective payload when designing a UAV. Therefore, in certain cases, opting for CPU deployment can better balance performance and weight, enabling the UAV to undertake a wider variety of tasks.

(iii) Economic considerations: The visual object tracking examined in this paper provides a foundation for future intelligent autonomous UAV swarms for cooperative reconnaissance. Utilizing CPUs for calculations can significantly reduce costs. In large-scale deployments, selecting cost-effective and low-power computing solutions will greatly enhance the overall feasibility and universality of the system.

This paper proposes a real-time target tracking algorithm that incorporates spatio-temporal context. The subsequent sections will provide a detailed overview of the proposed methodology, along with the experimental results, discussions, and conclusions.

3. Object Tracking Based on Spatio-Temporal Context (STCT)

3.1. The Algorithm Framework

The correlation filter exhibits excellent real-time performance but typically searches in a relatively constrained region. In situations where the target exhibits swift motion, the center of search in CF falls behind the target, consequently leading to subpar tracking performance.

In contrast, Siamese neural networks have a larger search area. Utilizing convolutional neural networks, it harnesses the power of deep feature extraction for capturing intricate details of the target. The deep features are highly robust when the target undergoes local occlusion or background interference. In addition, the correlation filter performs scale estimation by scale pooling or scale filtering. However, the aspect ratio of the target size is usually kept constant during scale estimation by CF. When the viewpoint changes significantly, the original scale cannot adapt well to the new target appearance. Siamese networks demonstrate greater scale adaptability. Therefore, this paper proposes an object tracking method incorporating spatio-temporal context (STCT), which invokes the Siamese network to expand the search space when necessary. Simultaneously, it performs online calibration of the scale parameters for the correlation filter, enabling the algorithm to balance robustness and real-time performance in visual object tracking.

Figure 1 shows the framework of the STCT algorithm proposed in this paper. The algorithm contains the following four components:

(i) The correlation filter.

If the tracking state is deemed satisfactory, the correlation filter undergoes online updates to accurately predict both the target's position and scale. The response $\boldsymbol{G}_k$ of the k-th frame is computed by Equation (1):

$$\boldsymbol{G}_k = \hat{\boldsymbol{x}}_k \odot \hat{\boldsymbol{f}}_k \tag{1}$$

Among them, $\odot$ represents element-wise multiplication. $\hat{\boldsymbol{x}}_k$ is the target feature after Fourier transformation in frame kth, and $\hat{\boldsymbol{f}}_k$ is the CF in the frequency domain. The CF algorithm computes the anticipated target position in the present frame by identifying the location associated with the utmost response value.

(ii) The Siamese network.

When the tracking state evaluation metric reaches the preset condition, SiamRPN is invoked to perform the target search in a larger spatial range. Its output $\boldsymbol{Y}_k$ is directly used as the result for the current frame (Equation (2)).

$$\boldsymbol{Y}_k = f(\boldsymbol{x}, \boldsymbol{z}_k) \tag{2}$$

Among them, $\boldsymbol{Y}_k$ is the similarity score, $\boldsymbol{x}$ is the target map, and $\boldsymbol{z}_k$ is the search map. In SiamRPN, the target position predicted in the current frame is determined by identifying the position exhibiting the utmost resemblance score.

Simultaneously, the search area and the fundamental target dimensions are adjusted based on the size $[w^{siam}, h^{siam}]$ of the tracking box (Equation (3)), and the initialization parameters of the CF are corrected. Then, the correlation filter is trained.

$$sz_{base} = \left[\frac{w^{siam}}{sc}, \frac{h^{siam}}{sc} \right] \tag{3}$$

where sc is the scale factor.

(iii) The target motion model.

To address the challenge of a lagging search focal point during swift target motion, this paper extracts the target positions of adjacent two frames in the time series to establish a motion model, enabling the prediction of its potential location in the upcoming frame. Since the time interval of consecutive frames is very short, it can be assumed that the target exhibits uniform linear motion in a short time.

Take the system state variable as $\boldsymbol{X} = [u_k^R, \dot{u}_k^R, v_k^R, \dot{v}_k^R]^T$.

Let

$$\boldsymbol{\phi}(k+1, k) = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

where T represents the sampling interval of the discrete system, and $\boldsymbol{\phi}$ is the state transition matrix.

Establish a target motion model through observation data:

$$\boldsymbol{X}(k+1) = \boldsymbol{\phi}(k+1,k)\boldsymbol{X}(k) \tag{5}$$

(iv) Tracking status evaluation.

Following the tracking outcomes, a comprehensive assessment of the object tracking state is performed to determine whether it is necessary to invoke SiamRPN to correct CF in the next frame. The state evaluation method combined with temporal context proposed in this paper is detailed in Section 3.3.

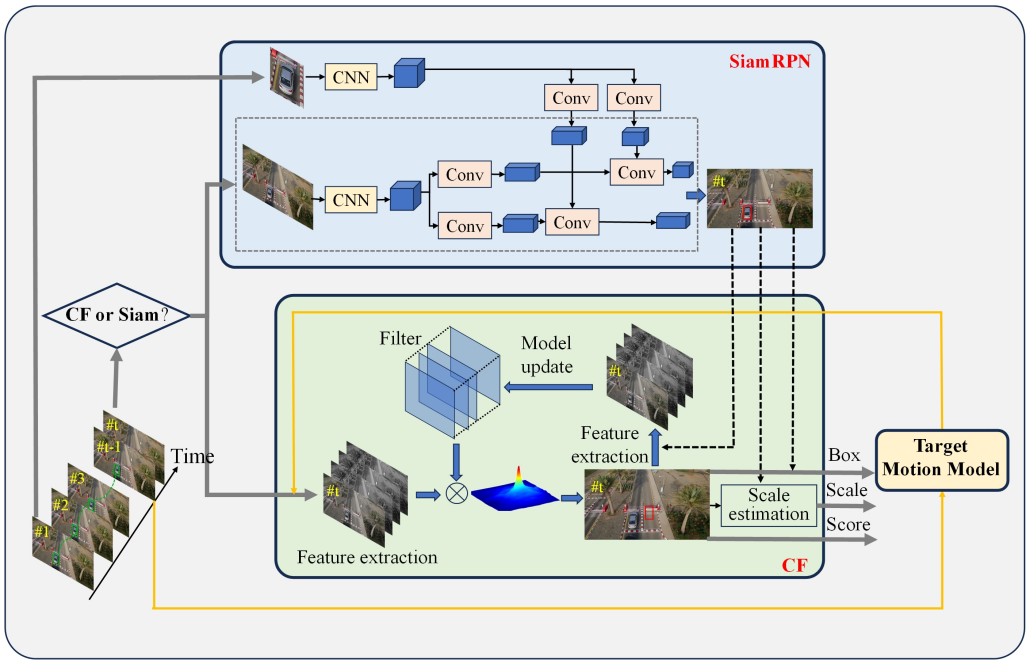


Figure 1. The algorithmic framework of STCT.

3.2. Aberration Suppression Correlation Filtering Based on Dynamic Attention Mechanism

The STRCF algorithm serves as the foundational framework in this paper. The objective in frame k is to minimize the following function in order to acquire the optimal correlation filter:

$$\arg\min_{f} \frac{1}{2}\left\|\sum_{d=1}^{D} \boldsymbol{x}_k^d * \boldsymbol{f}^d - \boldsymbol{y}\right\|^2 + \frac{1}{2}\sum_{d=1}^{D}\left\|\boldsymbol{w}\cdot\boldsymbol{f}^d\right\|^2 + \frac{\mu}{2}\left\|\boldsymbol{f}-\boldsymbol{f}_{k-1}\right\|^2 \tag{6}$$

Among them, $\boldsymbol{x}_k^d$ represents the extracted target feature from frame k, $d = 1,2,\ldots,D$ denotes the quantity of channels, $\boldsymbol{f}^d$ represents the filter trained on the d-th channel of frame k, $\boldsymbol{y}$ represents the expected Gaussian response, and $\boldsymbol{f}_{k-1}$ is the filter trained in frame $k-1$. $*$ represents a convolution operator. $\cdot$ represents the Hadamard product.

Although STRCF has demonstrated promising performance in UAV visual object tracking, it does have two inherent limitations:

(i) The target scale aspect ratio setting is fixed during the scale estimation process. It is determined by the labeled box in the first frame. This problem is addressed in Section 3.1 of this paper.

(ii) The spatial regularity term of the STRCF is fixed, making it difficult to achieve adaptability in scenarios where the target appearance undergoes unforeseen transformations during the tracking process.

Therefore, to tackle the aforementioned limitations, this paper introduces a spatio-temporal regularization term based on STRCF to suppress the aberrance in the response map. A dynamic attention module is devised to fine-tune the hyperparameters associated with the spatio-temporal regularization terms, rendering them adaptable and automatic. To accelerate the computational speed of the algorithm, the color features extracted in the baseline algorithm are discarded. In lieu of that, it amalgamates the HOG features, grayscale attributes, and color histogram features to extract intricate features.

First of all, the Euclidean paradigm is used to define the level of difference between the two response maps $\boldsymbol{R}_{k-1}$ and $\boldsymbol{R}_k$ as follows:

$$\begin{aligned}\left\|\boldsymbol{A}_{k-1,k}^{R}\right\|^2 &= \left\|\boldsymbol{R}_{k-1}[\psi_{q_1,q_2}] - \boldsymbol{R}_k\right\|^2 \\ &= \left\|\sum_{d=1}^{D}(\boldsymbol{x}_{k-1}^{d} * \boldsymbol{f}_{k-1}^{d})[\psi_{q_1,q_2}] - \sum_{d=1}^{D}\boldsymbol{x}_{k}^{d} * \boldsymbol{f}_{k}^{d}\right\|^2\end{aligned} \tag{7}$$

where $\boldsymbol{R}_{k-1}$ and $\boldsymbol{R}_k$ denote the response maps at frames k-1 and k, respectively; q_1 and q_2 denote the differences of the peak coordinates in the response maps $\boldsymbol{R}_{k-1}$ and $\boldsymbol{R}_k$, respectively. To facilitate the computation of differences between the response in the continuous time series, the peaks of $\boldsymbol{R}_{k-1}$ and $\boldsymbol{R}_k$ are overlapped by cyclically shifting ψ_{q_1,q_2}. When the target is abnormal, $\boldsymbol{R}_k$ suddenly changes, the similarity between $\boldsymbol{R}_k$ and $\boldsymbol{R}_{k-1}$ decreases, and the value of $\left\|\boldsymbol{A}_{k-1,k}^{R}\right\|^2$ increases. Anomalies can be recognized by judging $\left\|\boldsymbol{A}_{k-1,k}^{R}\right\|^2$.

To suppress the impact of response map aberrance on filter learning, the minimization objective function is designed as Equation (8) in this paper, where the third and fourth terms capture the temporal context and spatial context information for filter learning, respectively. When SiamRPN is called to correct CF, the temporal and spatial constraints of the correlation filter will be relaxed while the scale parameters are adjusted.

$$\arg\min_{f}\frac{1}{2}\left\|\sum_{d=1}^{D}\boldsymbol{x}_{k}^{d} * \boldsymbol{f}^{d} - \boldsymbol{y}\right\|^2 + \frac{1}{2}\sum_{d=1}^{D}\left\|\boldsymbol{w}\cdot\boldsymbol{f}^{d}\right\|^2 + \frac{\mu_0}{2}\left\|\boldsymbol{f}-\boldsymbol{f}_{k-1}\right\|^2 + \frac{\mu_1}{2}\left\|\boldsymbol{B}_k\boldsymbol{A}_{k-1,k}^{R}\right\|^2 \tag{8}$$

where $\boldsymbol{B}_k$ is the dynamic attention module designed in this paper, which dynamically adjusts the intensity of attention in different regions according to the spatial context information, making the filter more focused on the changes of the response map near the target and ignoring the secondary information, which in turn improves the performance.

Assume that the coordinates of the response map in frame k is $p_k = (u_k^R, v_k^R)$. The peak coordinates $\widehat{p}_k = (\tilde{u}_k^R, \tilde{v}_k^R)$ can be defined as follows:

$$[\tilde{u}_k^R, \tilde{v}_k^R] = \arg\max_{i,j}\boldsymbol{R}_k^{i,j} \tag{9}$$

The weight $\boldsymbol{B}_k$ should be assigned to each region is

$$\boldsymbol{B}_k(u_k^R, v_k^R) = \mu_2 + \mu_3 e^{-\left\|p_k-\widehat{p}_k\right\|/\left\|\widehat{p}_k\right\|} \tag{10}$$

Subsequently, the optimization objective pursued in this manuscript can be succinctly articulated as follows:

$$\begin{aligned}\varepsilon(\boldsymbol{f}) = &\frac{1}{2}\left\|\sum_{d=1}^{D}\boldsymbol{x}_{k}^{d} * \boldsymbol{f}^{d} - \boldsymbol{y}\right\|^2 + \frac{1}{2}\sum_{d=1}^{D}\left\|\boldsymbol{w}\cdot\boldsymbol{f}^{d}\right\|^2 + \frac{\mu_0}{2}\left\|\boldsymbol{f}-\boldsymbol{f}_{k-1}\right\|^2 \\ &+ \frac{\mu_1}{2}\left\|\boldsymbol{B}_k(\boldsymbol{R}_{k-1}[\psi_{q_1,q_2}] - \boldsymbol{x}_k * \boldsymbol{f}_k)\right\|^2\end{aligned} \tag{11}$$

3.3. Optimization Through ADMM

The optimal solution of Equation (11) is obtained by the alternating direction method of multipliers (ADMM). First of all, an auxiliary variable is introduced:

$$g = f \tag{12}$$

Further, the augmented Lagrangian formulation of Equation (11) can be written as follows:

$$\begin{aligned}\mathcal{L}(f,g,v) = &\frac{1}{2}\left\|\sum_{d=1}^{D} x_k^d * f^d - y\right\|^2 + \frac{1}{2}\sum_{d=1}^{D}\left\|w \cdot g^d\right\|^2 + \frac{\mu_0}{2}\|f - f_{k-1}\|^2 \\ &+\frac{\mu_1}{2}\|B_k(R_{k-1}[\psi_{q_1,q_2}] - x_k g_k)\|^2 + \sum_{d=1}^{D}(f^d - g^d)v^d + \frac{\rho}{2}\sum_{d=1}^{D}\left\|f^d - g^d\right\|^2\end{aligned} \tag{13}$$

Introduce

$$u = \frac{1}{\rho}v \tag{14}$$

Then, Equation (13) can be written as

$$\begin{aligned}\mathcal{L}(f,g,v) = &\frac{1}{2}\left\|\sum_{d=1}^{D} x_k^d * f^d - y\right\|^2 + \frac{1}{2}\sum_{d=1}^{D}\left\|w \cdot g^d\right\|^2 + \frac{\mu_0}{2}\|f - f_{k-1}\|^2 \\ &+ \frac{\mu_1}{2}\|B_k(R_{k-1}[\psi_{q_1,q_2}] - x_k g_k)\|^2 + \frac{\rho}{2}\sum_{d=1}^{D}\left\|f^d - g^d + u^d\right\|^2\end{aligned} \tag{15}$$

The above equation can be decomposed into three sub-problems by ADMM iteration:

$$f_{k+1} := \arg\min_f \left(\left\|\sum_{d=1}^{D} x_k^d * f^d - y\right\|^2 + \mu_0\|f - f_{k-1}\|^2 + \rho\|f - g + u\|^2\right) \tag{16}$$

$$g_{k+1} := \arg\min_g \left(\sum_{d=1}^{D}\left\|w \cdot g^d\right\|^2 + \mu_1\|B_k(R_{k-1}[\psi_{q_1,q_2}] - x_k g_k)\|^2 + \rho\|f - g + u\|^2\right) \tag{17}$$

$$u_{k+1} := u_k + f_{k+1} - g_{k+1} \tag{18}$$

(i) Solution of subproblem f

According to Parseval's theorem, the subproblem f (Equation (16)) can be elegantly reformulated in the Fourier domain as follows:

$$\hat{f}_{k+1} := \arg\min_{\hat{f}} \left(\left\|\sum_{d=1}^{D} \hat{x}_k^d \cdot \hat{f}_k^d - \hat{y}\right\|^2 + \mu_0\left\|\hat{f} - \hat{f}_{k-1}\right\|^2 + \rho\left\|\hat{f} - \hat{g} + \hat{u}\right\|^2\right) \tag{19}$$

where $\hat{f}$ represents the discrete Fourier transform (DFT) of the filter f. It is evident that the j-th component of the label $\hat{y}$ solely relies on the j-th component of the filter $\hat{f}$ and the sample $\hat{x}_k$.

Define $V_j(f) \in \mathbb{R}^D$ to be a vector that is composed of the j-th element of the filter f across all channels. Then, Equation (19) can be further partitioned into subproblems, with each individual subproblem designated as follows:

$$\arg\min_{\mathcal{V}_j(\hat{f})}\left(\left\|\mathcal{V}_j(\hat{x}_k)^T\mathcal{V}_j(\hat{f}) - \hat{y}_j\right\|^2 + \mu_0\left\|\mathcal{V}_j(\hat{f}) - \mathcal{V}_j(\hat{f}_{k-1})\right\|^2 + \rho\left\|\left(\mathcal{V}_j(\hat{f}) - \mathcal{V}_j(\hat{g}) + \mathcal{V}_j(\hat{u})\right)\right\|^2\right) \tag{20}$$

Let

$$\boldsymbol{v} = \hat{y}_j V_j(\hat{\boldsymbol{x}}_k) + \mu V_j(\hat{\boldsymbol{f}}_{k-1}) + \rho V_j(\hat{\boldsymbol{g}}) - \rho V_j(\hat{\boldsymbol{u}}) \tag{21}$$

Then, the closed-form solution of $V_j(\hat{\boldsymbol{f}})$ can be derived:

$$\begin{aligned} \mathcal{V}_j(\hat{\boldsymbol{f}}) &= \left(\mathcal{V}_j(\hat{\boldsymbol{x}}_k)\mathcal{V}_j(\hat{\boldsymbol{x}}_k)^T + (\mu_0 + \rho)I\right)^{-1}\boldsymbol{v} \\ &= (\mu_0 + \rho)^{-1}\left[I - \frac{\mathcal{V}_j(\hat{\boldsymbol{x}}_k)\mathcal{V}_j(\hat{\boldsymbol{x}}_k)^T}{\mu_0 + \rho + \mathcal{V}_j(\hat{\boldsymbol{x}}_k)^T\mathcal{V}_j(\hat{\boldsymbol{x}}_k)}\right]\boldsymbol{v} \end{aligned} \tag{22}$$

(ii) Solution of subproblem *g*

Taking the derivative of Equation (17) and equating it to zero yields a closed-form solution.

$$\boldsymbol{g} = (\boldsymbol{W}^T\boldsymbol{W} + \mu_1 \boldsymbol{B}_k{}^2 \boldsymbol{x}_k{}^2 + \rho \boldsymbol{I})^{-1}[\mu_1 \boldsymbol{B}_k{}^2 \boldsymbol{x}_k \boldsymbol{R}_{k-1}[\psi_{q_1,q_2}] + \rho(\boldsymbol{f} + \boldsymbol{u})] \tag{23}$$

where $\boldsymbol{W}$ denotes the diagonal matrix concatenated with D diagonal, matrices $Diag(\boldsymbol{w})$.

3.4. Adaptive Object Tracking Status Evaluation

In terms of tracking state assessment, the information of the response map is crucial for the state assessment of object tracking. Currently widely adopted tracking state evaluation metrics, such as maximum response $R_{\max}$, average peak-to-correlation energy (APCE [32]), peak-to-sidelobe ratio (PSR), and so on, solely consider the response result of the current frame, ignoring the temporal context information. In this regard, this paper proposes the average maximum response value related energy (AMRE), an adaptive object tracking state assessment function considering time series. It learns the tracking state change characteristics to assess the dependability of the tracking outcomes, which is served as a basis for model correction in STCT.

First, normalize the response map $\boldsymbol{R}_f$ in current frame. The normalized response map $\tilde{\boldsymbol{R}}_f$ is

$$\tilde{\boldsymbol{R}}_f = \frac{\boldsymbol{R}_f - min\left(\boldsymbol{R}_f\right)}{max\left(\boldsymbol{R}_f\right) - min\left(\boldsymbol{R}_f\right)} \tag{24}$$

Select the top M largest response peaks $\max_M(\boldsymbol{R}^i_{\max})$ from the current response map; $\boldsymbol{R}^i_{\max}$ is the response peak of ranked i. Then, the expectation μ of the response peaks can be calculated according to Equation (25):

$$\mu = \sum_{i=1}^{M} \tilde{\boldsymbol{R}}^i_{\max} \cdot P(\tilde{\boldsymbol{R}}^i_{\max}) \tag{25}$$

where $P(\tilde{\boldsymbol{R}}^i_{\max})$ is the probability that the response peak $\tilde{\boldsymbol{R}}^i_{\max}$ occurs.

Define

$$MRE = \frac{\left|\tilde{\boldsymbol{R}}^1_{\max} - \tilde{\boldsymbol{R}}^2_{\max}\right|^2}{\sqrt{\sum\limits_{i=1}^{M}\left(\tilde{\boldsymbol{R}}^i_{\max} - \mu\right)^2}} \tag{26}$$

where $\tilde{\boldsymbol{R}}^1_{\max}$ and $\tilde{\boldsymbol{R}}^2_{\max}$ are the values of the primary and secondary peaks in the response map $\tilde{\boldsymbol{R}}_f$, respectively.

Based on the maximum N MRE values from all past frames and the average of the MRE values from the last three frames, define AMRE as Equation (27):

$$\begin{cases} MRE' = mean(\max_N(MRE[0:k-1]) + MRE[k-4:k-1]) \\ AMRE = \frac{MRE' - MRE_k}{MRE'} \end{cases} \tag{27}$$

When the tracking performance is excellent, the value of MRE_k approaches MRE' in the frame $[0 : k-1]$, so that AMRE presents a relatively small value. On the contrary, if the tracking status deteriorates, MRE_k will suddenly decrease, so that AMRE presents a relatively large value. Then, it is considered that the object in the current frame has undergone an abnormality such as a variation in scale, partial occlusion, or a shift in the background.

In this paper, the weighted average value is used as the judgment threshold. Smooth the AMRE curve according to Equation (28):

$$\delta_t = \begin{cases} AMRE \; if \; t < 1 \\ \lambda \delta_{t-1} + (1-\lambda) AMRE \end{cases} \tag{28}$$

Calculate the adaptive threshold T as

$$T = \eta \cdot \delta_t + \eta_0 \tag{29}$$

where λ is the interpolation coefficient, η is the amplification factor, and η_0 is the intercept.

The timing for STCT invoking the Siamese network and correcting correlation filtering is:

$$if AMRE > T \tag{30}$$

Taking the Car6 sequence in the UAV123 dataset as an example, the target is partially occluded at frame 1140 $\sim$ frame 1386 due to being out of the visual range. It reappears completely after frames 1386. In addition, the size of the car varies a lot due to the changes in perspective. At this point, the filter template is prone to missing information if the CF is continuously called. When the target reappears, the CF will continue to learn the previous template. Instead, in the algorithm proposed in this paper, when AMRE exceeds a given threshold, the STCT calls SiamRPN to correct the object tracking and achieve scale adaptation. The change curves of AMRE value and the adaptive threshold, object tracking results, and response map changes during this process are shown in Figure 2.

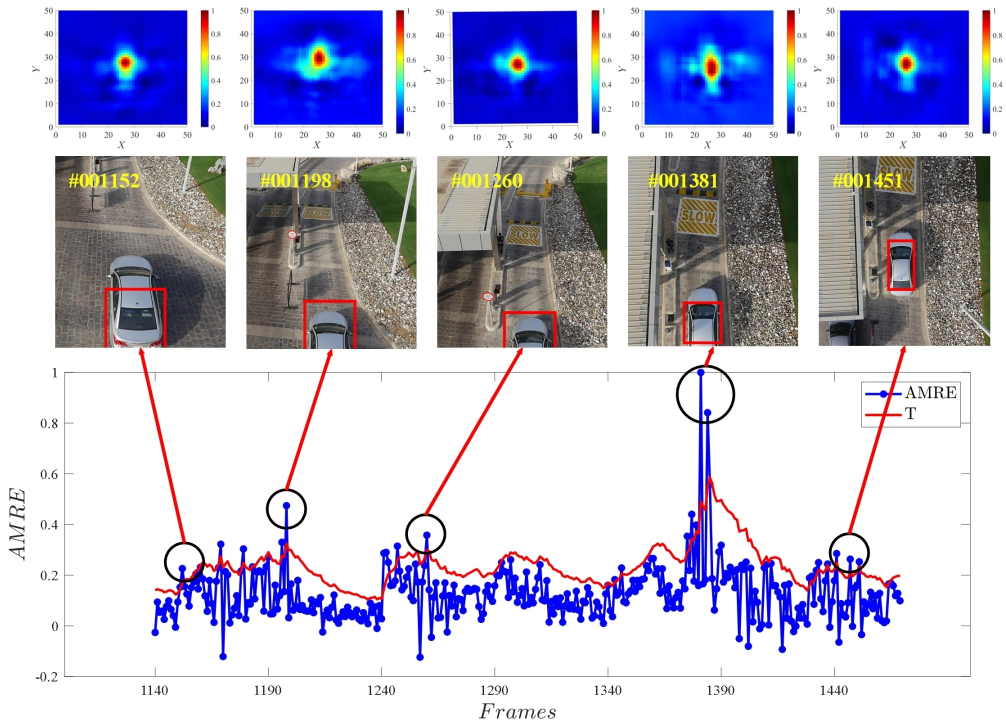


Figure 2. The dynamic adjustment process of AMRE.

4. Experiments and Results

In order to evaluate the algorithm performance, the proposed algorithm STCT is subjected to comparative experiments with other target trackers. The comparison algorithms mainly include EMCF, RISTrack, MRCF, MSCF, STRCF, BACF, KCF-CN, KCF-HOG, DSST, CSRDCF, ECO-HC, and SiamRPN.

4.1. Experimental Details

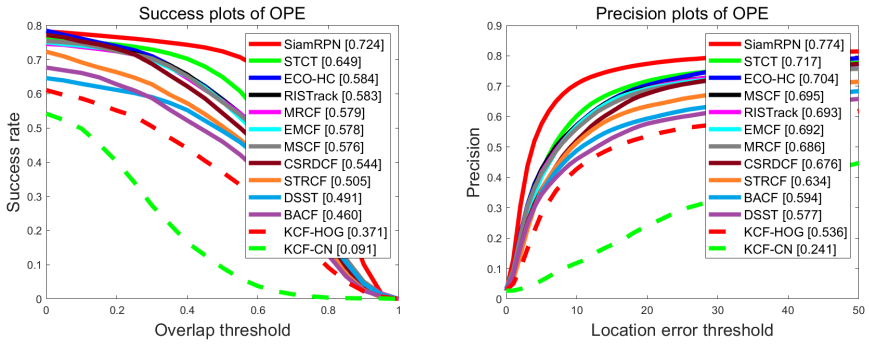
Datasets: This paper compares the STCT algorithm with other algorithms on the UAV123 and UAV20L datasets. The UAV123 dataset collected through rotor drones is specifically designed for testing object tracking algorithms in the UAV domain. It contains 123 low-altitude video sequences totaling about 110,000 frames. UAV123 covers a variety of environments and scenarios, including cities, countryside, oceans, forests, harbors, etc. Furthermore, UAV123 covers 12 attributes, including illumination change (IV), scale variation (SV), partial occlusion (POC), full occlusion (FOC), out-of-view (OV), fast motion (FM), camera motion (CM), background clutter (BC), similar object (SOB), aspect ratio change (ARC), viewpoint change (VC), and low resolution (LR). Due to the rapid relative motion between the target and UAV, combined with the significant variations in viewpoint, there is a substantial alteration in both the size and aspect ratio of the target bounding box compared with the initial frame, which fully simulates the complexity of the real application. The UAV20L dataset is a long-term object tracking dataset for UAVs. It contains 20 video sequences with an average sequence length of 2934 frames. Compared with the UAV123 dataset, the UAV20L dataset has a longer duration, more specific tasks, and greater challenges. Therefore, we chose these two datasets for the algorithms' evaluation in this paper.

Implementation details: Due to MATLAB's convenience and efficiency in numerical computation, most existing correlation filters are implemented in MATLAB. However, in practical applications, these algorithms are typically deployed using C++ or Python. Therefore, the correlation filtering algorithms developed in MATLAB require further translation into the corresponding languages for engineering deployment. Although Python-based correlation filtering object tracking algorithms tend to run more slowly, they are closer to engineering deployment and have better portability and cross-platform capabilities. To eliminate any potential impacts arising from the mixed use of different programming languages in engineering applications, all operations of the proposed algorithm, including the correlation filtering, are implemented using the Python programming language. Unfortunately, most of the existing open-source correlation filters are implemented only in MATLAB, limiting their flexibility and accessibility in practical applications. Based on this, the PyCFTrackers project has optimized and adjusted the official MATLAB code to implement Python versions of various related filtering tracking algorithms. Currently, the following correlation filters written in Python are implemented in PyCFTrackers: STRCF, BACF, KCF-CN, KCF-HOG, DSST, CSRDCF, and ECO-HC. SiamRPN is implemented in the open-source object tracking research platform PySOT. These algorithms will participate in the speed testing experiments in this paper. However, the EMCF, RISTrack, MRCF, and MSCF algorithms do not have a Python version available, lacking code suitable for engineering deployment. Therefore, this paper will still utilize the code provided in the original work, which is written in MATLAB, for implementation.

All comparison experiments have been conducted on a standardized platform with Lenovo R9000K hardware configuration and AMD Ryzen 9 5900HX CPU. In addition, to evaluate the real-time capabilities of different algorithms under constrained computational resources, this paper only conducted algorithm speed testing on the CPU. The parameters of STCT are set as follows: $\lambda = 0.9, \eta = 1.6, \eta_0 = 0.03$, $\mu_0 = 16, \mu_1 = 0.04$, $\mu_2 = 0.5, \mu_3 = 2$. When SiamRPN is called to correct CF, the temporal and spatial constraints of the correlation filter are relaxed such that $\mu_0 = 1, \mu_1 = 0.006$.

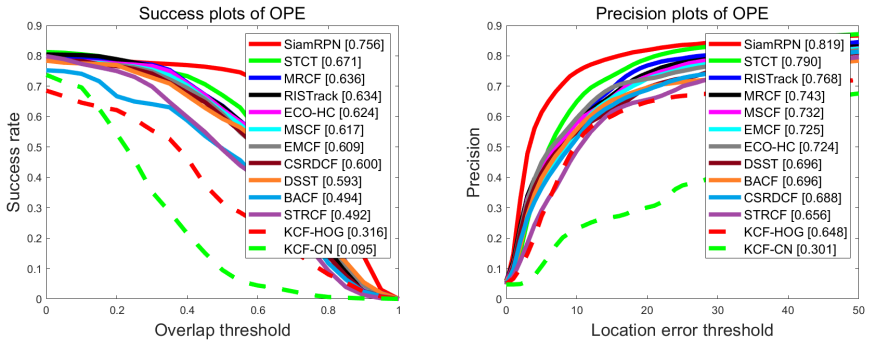
4.2. Comprehensive Evaluation

The evaluation of target trackers employs the one-pass evaluation (OPE) [33]. In the UAV123 and UAV20 datasets, the overall performance evaluation results of all trackers on the CPU are shown in Figures 3 and 4 and Tables 1 and 2. Among them, the most representative Siamese network, SiamRPN, performs optimally in terms of both tracking success rate and precision. The STCT proposed in this paper has a success rate of 0.649 and a precision of 0.717 on the UAV123 dataset, and a success rate of 0.671 and a precision of 0.790 on the UAV20L dataset, ranking second to SiamRPN.



(**a**) Success plots of OPE (**b**) Precision plots of OPE

Figure 3. Overall performance of CPU-based trackers on UAV123.



(**a**) Success plots of OPE (**b**) Precision plots of OPE

Figure 4. Overall performance of CPU-based trackers on UAV20L.

Table 1. Overall performance evaluation on UAV123.

Algorithms	AUC	Precision	Algorithms	AUC	Precision
SiamRPN	0.724	0.774	CSRDCF	0.544	0.676
STCT	0.649	0.717	STRCF	0.505	0.634
ECO-HC	0.584	0.704	DSST	0.491	0.577
RISTrack	0.583	0.693	BACF	0.460	0.594
MRCF	0.579	0.686	KCF-HOG	0.371	0.536
EMCF	0.578	0.692	KCF-CN	0.091	0.241
MSCF	0.576	0.695			

Table 2. Overall performance evaluation on UAV20L.

Algorithms	AUC	Precision	Algorithms	AUC	Precision
SiamRPN	0.756	0.819	CSRDCF	0.600	0.688
STCT	0.671	0.790	STRCF	0.492	0.656
ECO-HC	0.624	0.724	DSST	0.593	0.696
RISTrack	0.634	0.768	BACF	0.494	0.696
MRCF	0.636	0.743	KCF-HOG	0.316	0.648
EMCF	0.609	0.725	KCF-CN	0.095	0.301
MSCF	0.617	0.732			

More specifically, in the UAV123 datasets, STCT (0.649, 0.717) improves 28.5% in terms of success rate and 13.1% in terms of tracking precision compared with the benchmark algorithm STRCF (0.505, 0.634). Compared with the third-ranked ECO-HC (0.584, 0.704), STCT shows advantages of 11.1% in tracking success rate and 1.8% in tracking accuracy. In the UAV20L dataset, STCT (0.671) has advantages of 5.2% and 5.8% in tracking success rate over the third-ranked MRCF (0.636) and the fourth-ranked RISTrack (0.634), respectively. In terms of tracking precision, STCT (0.790) outperforms the third-ranked RISTrack (0.768) and the fourth-ranked MRCF (0.743) by 2.9% and 6.3%, respectively.

Runtime is an important metric to measure real-time performance. In the field of visual object tracking, the main concern is the inference time of the algorithm for online tracking. Since speed testing experiments need to be conducted on the same platform for fairness, this paper compares the running speed of the trackers written in Python on a single CPU. The results are shown in Table 3, where T_{ot} denotes the online training time of the algorithm and T_{tr} denotes the testing time, i.e., the online object tracking time. The correlation filter of STCT proposed in this paper is trained online, while the Siamese network tracker is trained offline. Only forward computation is needed in the actual tracking process. Therefore, the online training time of STCT consists solely of the training time for the correlation filter. The test time is the online tracking time, i.e., the average inference time for tracking each frame of the video sequence, which contains the online training phase. The smaller the value of the object tracking test time, the more real-time the algorithm is. Therefore, the real-time performance of the algorithm in the field of object tracking is mainly evaluated by the test time and FPS. It can be seen from Table 3 that the online training time and test time of STCT, ECO-HC, BACF, STRCF, and KCF-HOG are relatively short. The inference speeds of these algorithms all exceed 25FPS, meeting the real-time demands of UAV object tracking. However, the Siamese network tracker SiamRPN, which performs better in terms of performance, runs only 7.9FPS on the CPU. The running speed of the STCT proposed in this paper runs at a speed of 38.0FPS, which is 22.2% faster than the benchmark algorithm.

Table 3. The running time and average tracking speed (FPS) on the CPU of different trackers.

Algorithms	T_{ot} (CPU)	T_{tr} (CPU)	FPS (CPU)	Algorithms	T_{ot} (CPU)	T_{tr} (CPU)	FPS (CPU)
SiamRPN	-	126.6 ms	7.9	DSST	27.8 ms	94.3 ms	10.6
STCT	9.5 ms	26.3 ms	38.0	BACF	8.5 ms	25.4 ms	39.4
ECO-HC	12.8 ms	28.5 ms	35.1	KCF-HOG	5.3 ms	18.0 ms	55.6
CSRDCF	35.2 ms	107.5 ms	9.3	KCF-CN	32.5 ms	73.5 ms	13.6
STRCF	9.3 ms	32.2 ms	31.1				

Additionally, the performance and speed of the tracker are evaluated comprehensively in this paper. The results are illustrated in Figure 5, wherein the red dashed line delineates the real-time demarcation. Partial tracking results are presented in Figure 6.

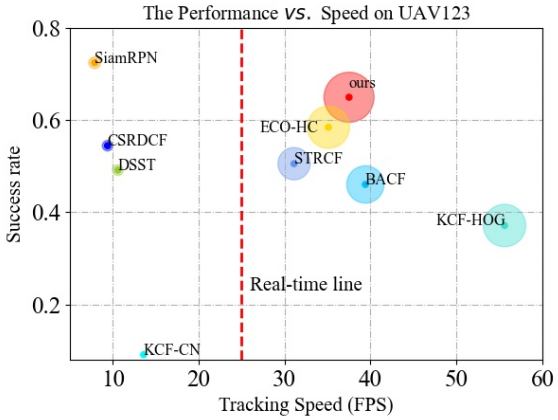


Figure 5. Comprehensive evaluation of the performance and speed.

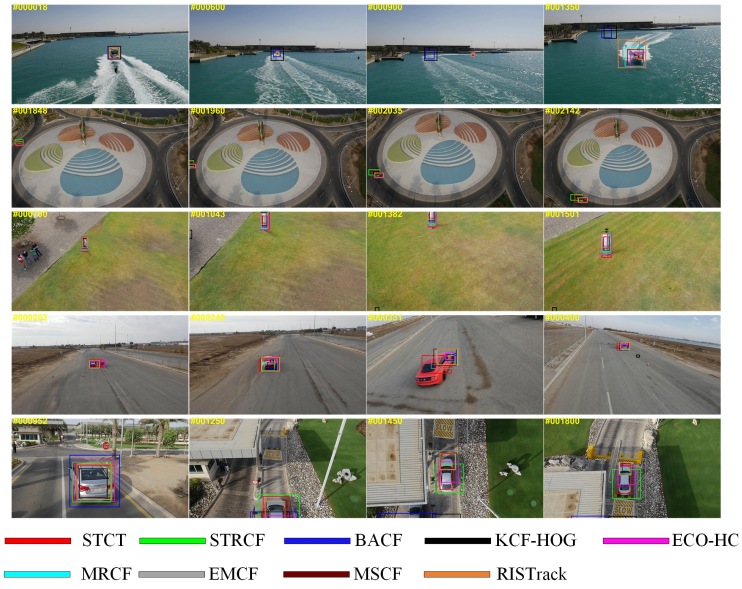


Figure 6. Visualization results of the real-time trackers.

4.3. The Attribute-Based Comparison

In this section, the tracking effectiveness of different algorithms for each attribute is quantitatively analyzed. The success rates and tracking precision for each attribute are shown in Figures 7–10. The experimental findings reveal that the proposed STCT algorithm exhibits better performance compared with other correlation filter trackers that rely solely on CPU processing in most attribute aspects, thus underscoring its superiority in performance evidently.

From the attribute evaluation of ARC, SV, and VC in the UAV123 dataset, the tracking success and precision of STCT surpass those of other correlation filters. Specifically, in the ARC scenario, the tracking success and precision of STCT (0.581, 0.659) are improved by 45.3% and 15.4%, respectively, compared with the benchmark algorithm (0.400, 0.571). In SV attribute assessment, STCT (0.621, 0.691) has shown an improvement of 32.4% in tracking success rate and 13.8% in precision compared with STRCF (0.469, 0.607). Compared with RISTrack, STCT has a 14.4% advantage. In VC attribute evaluation, STCT (0.628, 0.679) has improved the success rate and precision of the baseline (0.402, 0.538) by 56.2% and 26.2%, respectively. This indicates that STCT can effectively cope with complex situations such as target scale change and aspect ratio changes by combining spatio-temporal context and adaptively calling CF and Siamese network trackers.

Furthermore, benefiting from the target motion modeling combined with timing information in STCT, the attribute evaluation of both CM and FM outperforms that of other correlation filter trackers. Specifically in the attribute evaluation of CM, STCT achieves a tracking success rate of 0.656 and a precision of 0.707, which outperforms other correlation filters. Compared with STRCF (0.498, 0.600), the success rate and precision improved by 31.7% and 17.8%, respectively. In FM attribute evaluation, the tracking success and precision of STCT are 0.532 and 0.621, respectively. Compared with STRCF (0.382, 0.529), the success rate and precision are improved by 39.3% and 1.4%, respectively. Compared with the third-ranked ECO-HC, STCT has a 20.6% advantage. It can be seen that STCT demonstrates notable robustness when addressing camera movements and rapid object motions.

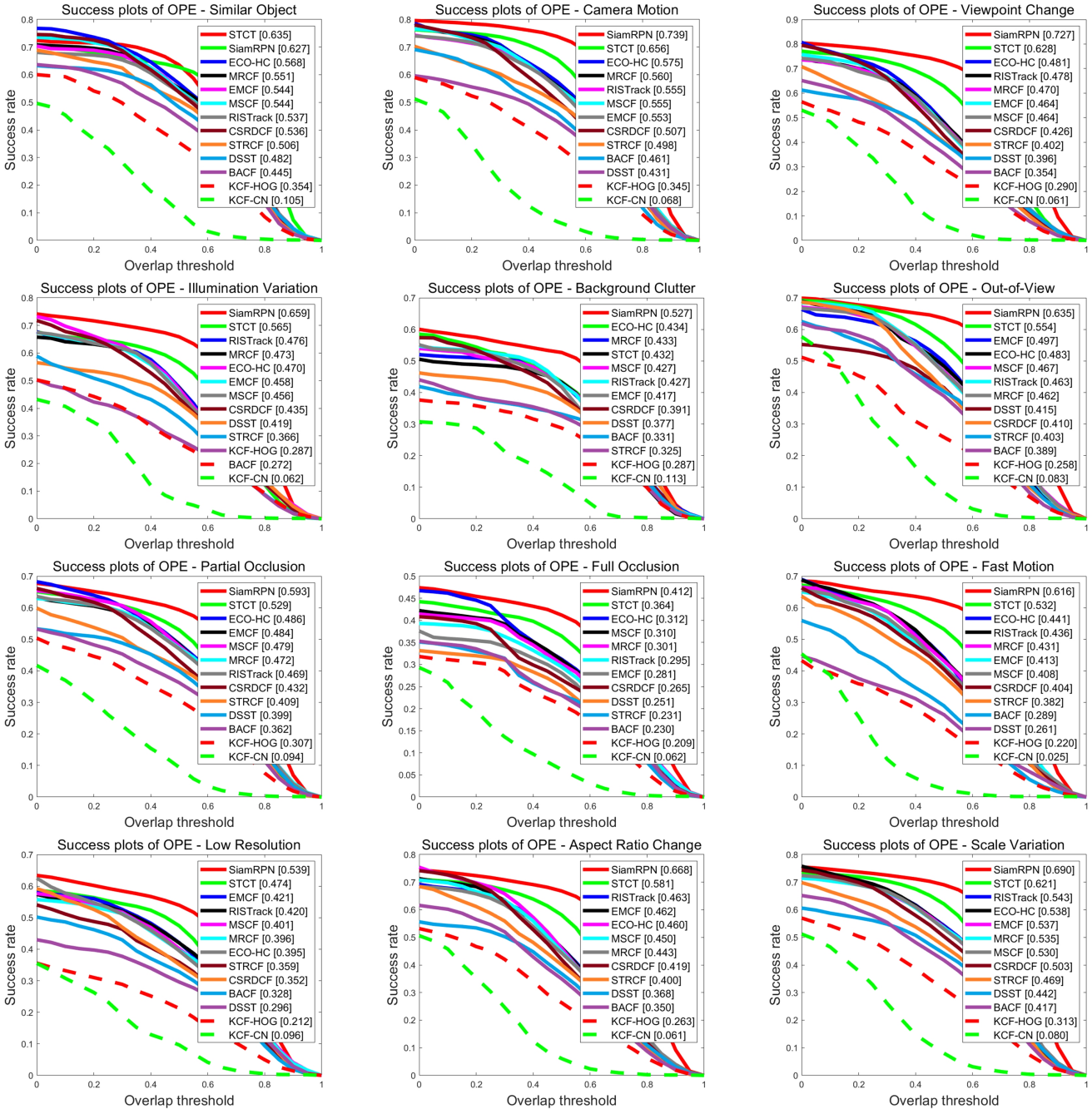


Figure 7. The testing success rate of each attribute on UAV123.

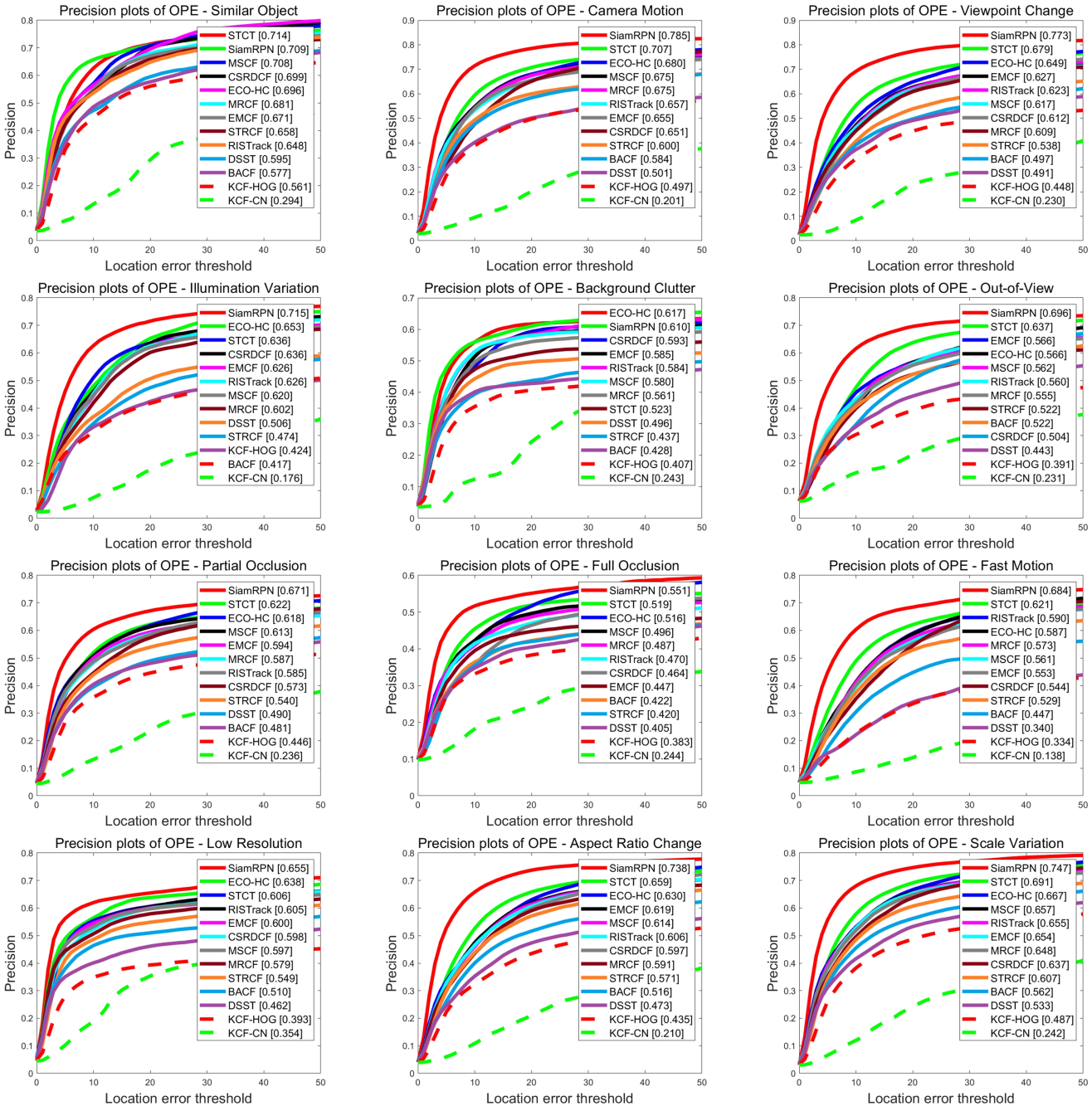


Figure 8. The testing precision of each attribute on UAV123.

Moreover, in OV, POC, and FOC attribute evaluation, STCT ranked second in both tracking success rate and precision. This shows that the spatio-temporal regularization term based on the dynamic attention mechanism proposed in this paper is robust in resolving target appearance changes due to partial occlusion or out-of-view scenarios.

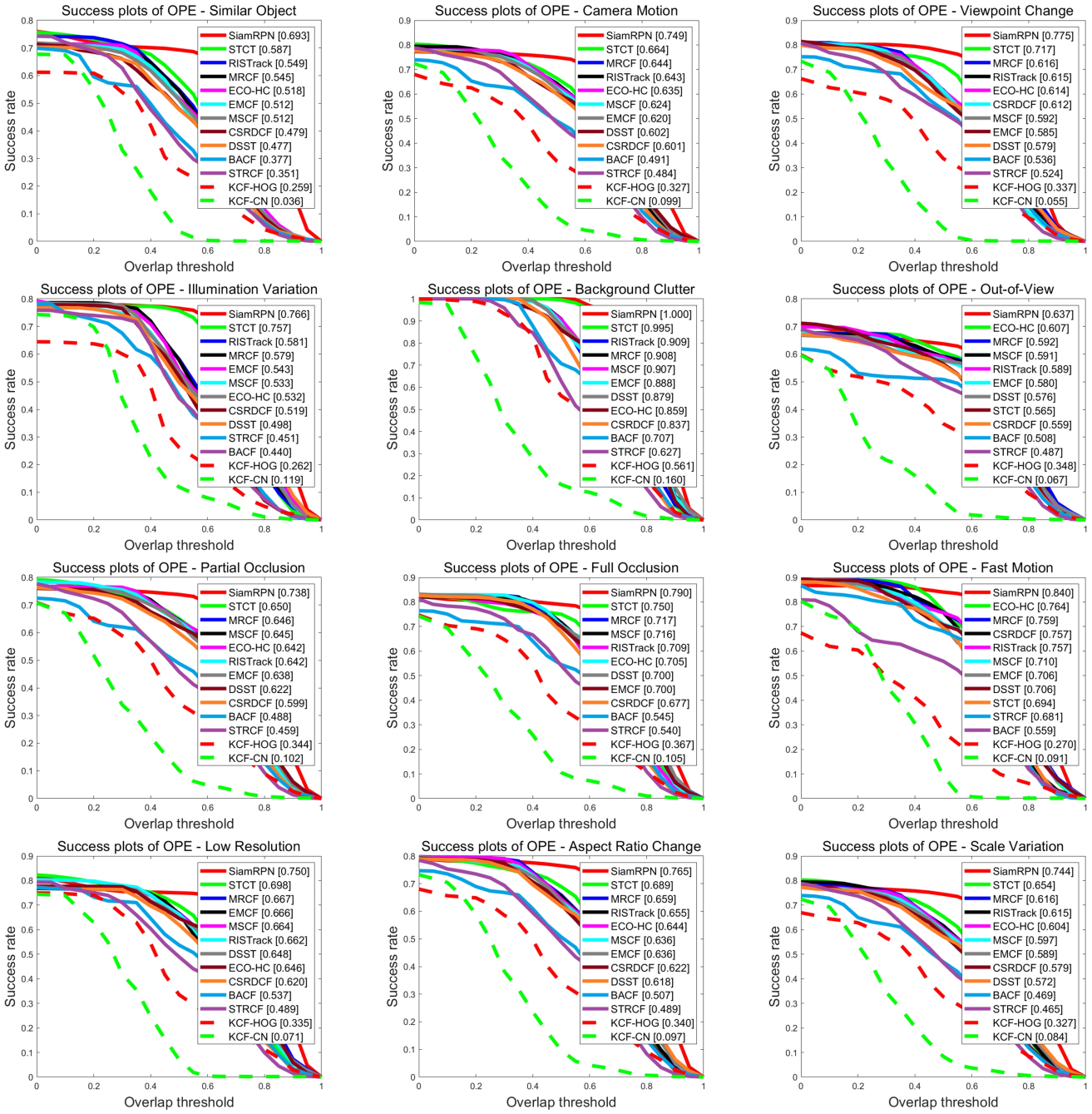


Figure 9. The testing success rate of each attribute on UAV20L.

In long-term object tracking, as shown in Figures 9 and 10, the STCT algorithm demonstrates better success rates than other related filters in the 10 attributes of SOB, CM, VC, IV, BC, POC, FOC, LR, ARC, and SV. The tracking precision in the 11 attributes, excluding OV, surpasses that of other related filters. Notably, in the evaluation of the full occlusion attribute, STCT (0.750, 0.828) achieves the best tracking accuracy. Compared with the third-ranked MRCF (0.717, 0.805), STCT shows advantages of 4.6% and 2.8% in success rate and accuracy, respectively. This indicates that the algorithm proposed in this paper has certain advantages in long-term object tracking.

In addition, this paper compares five trackers with better real-time performance based on a single CPU. Radar charts are used to show their capabilities under different attributes. The results are shown in Figure 11.

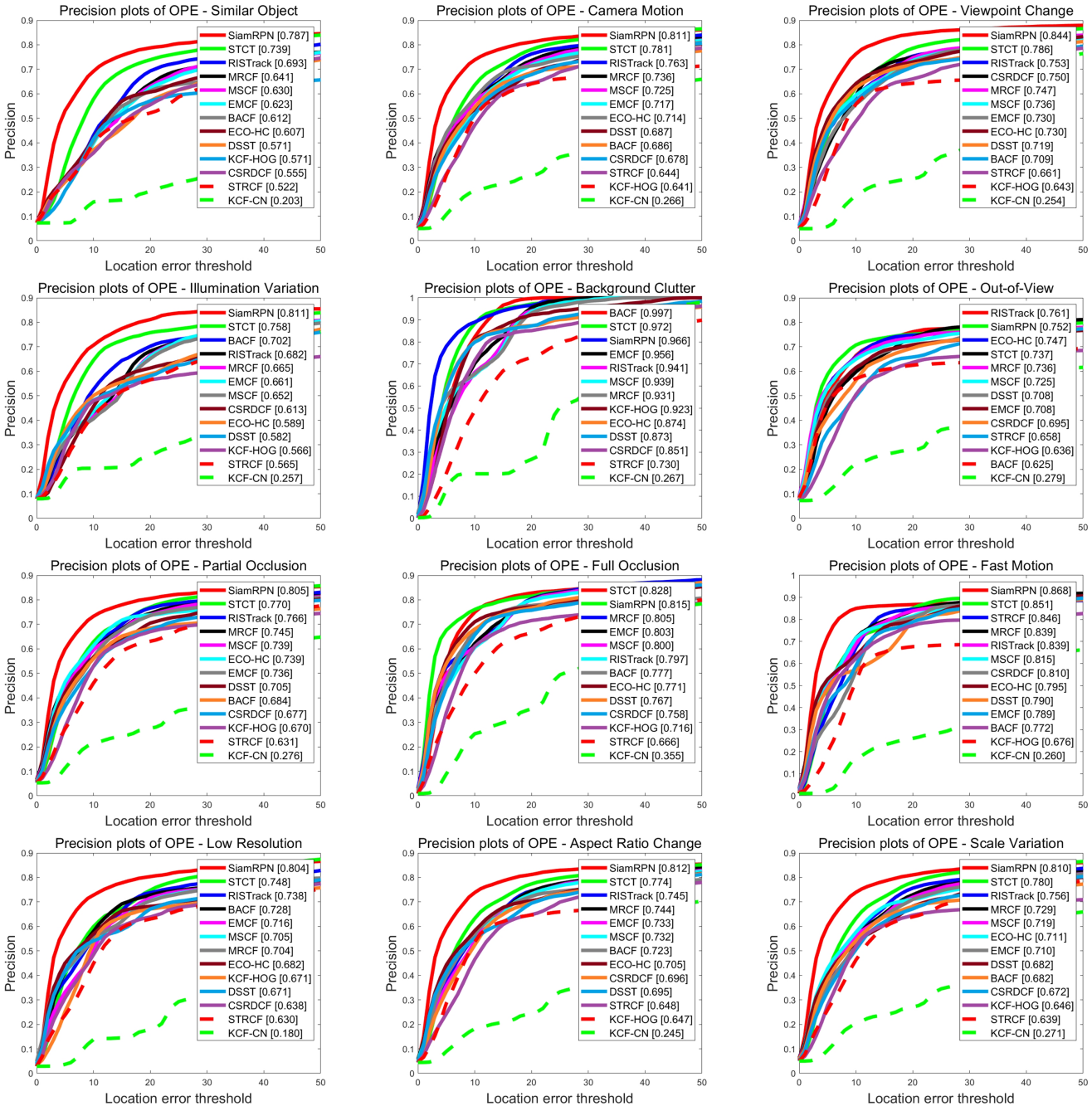
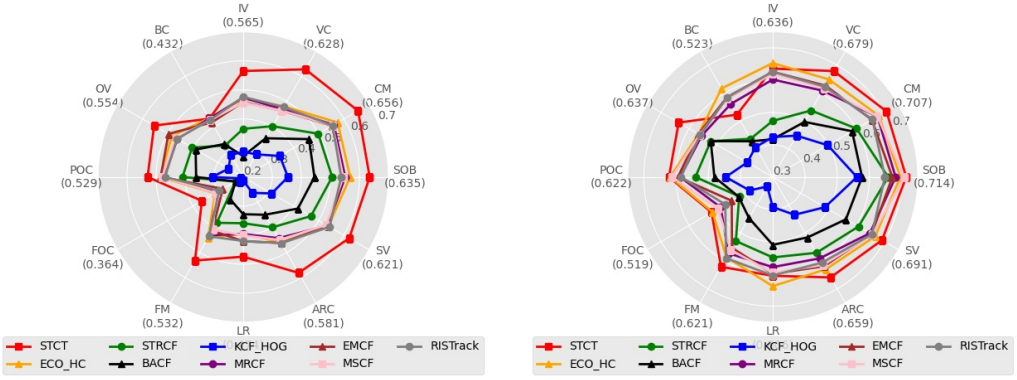


Figure 10. The testing precision of each attribute on UAV20L.



(**a**) Success rate (**b**) Precision

Figure 11. Attribute evaluation of real-time trackers.

5. Discussion

In this study, the proposed STCT algorithm excels in several aspects. In terms of overall performance evaluation, like other CPU-based correlation filters, STCT's robustness is slightly inferior to that of Siamese neural network trackers. However, STCT achieved the best performance with the benchmark algorithm (STRCF) and all other CPU-based real-time trackers in the comparison. It can be attributed to the 'corrective' functionality of the Siamese network on the correlation filter in the STCT framework. It significantly improves the robustness and accuracy of the algorithm. Although the supremacy of SiamRPN in terms of tracking success rate is clearly evident. However, the tracking speed of SiamRPN falls short of meeting the real-time demands on a single-CPU platform. Among all the trackers that fulfill the real-time requirements, the STCT algorithm proposed in this paper achieves the highest tracking success rate and stands out as the foremost real-time tracker based on a single CPU. In terms of attribute evaluation, it can be seen that STCT exhibits optimal performance in nine attributes: SOB, CM, FM, OV, POC, FOC, ARC, SV, and VC. This shows that the STCT algorithm is highly adaptable in dealing with diverse object tracking scenarios.

To summarize, STCT demonstrates a good balance between tracking robustness and real-time performance by effectively handling camera motion, illumination variation, low resolution, target out-of-view, occlusion, viewpoint change, scale variation, and aspect ratio variation while ensuring tracking efficiency. However, the current study also has some limitations. The effectiveness of the object tracking algorithm in specific environments needs further enhancement. Examples include applications in extreme lighting conditions and the effectiveness of tracking after a target has been occluded or lost for an extended period. Future research can be centered on strategies to enhance the performance of the algorithms in extreme conditions and enhance the adaptability of the algorithms in different scenarios.

6. Conclusions

In this paper, a real-time object tracking algorithm (STCT) that incorporates spatio-temporal context is proposed. The method integrates the Siamese network, the correlation filter, and the target motion model into a unified framework, aiming at real-time and robust tracking on a single CPU platform. Further, in STCT, a spatio-temporal regularization term based on the dynamic attention mechanism is introduced into CF to overcome the effects brought by response map aberrance. In addition, this paper proposes a temporal context-based state evaluation function AMRE, which makes the STCT algorithm more adaptable. The experimental results show that STCT has a higher tracking success rate and precision compared with the other real-time trackers. On the UAV123 and UAV20L datasets, the success rates are 0.649 and 0.671, respectively, while the precision rates are

0.717 and 0.790, respectively. Compared with the benchmark algorithm, the success rate and precision are improved by 28.5% and 13.1%, respectively. The STCT shows the best performance in multiple attribute evaluations. In addition, on the platform with a single CPU, STCT achieves a tracking speed of ∼38FPS, which meets the real-time requirements for UAV object tracking.

This study particularly emphasizes the importance of CPU deployment, as it accommodates the computational limitations of onboard and portable ground processors while also satisfying the lightweight requirements of small UAVs. Furthermore, it contributes to reducing economic costs in the future proliferation of UAV swarms. Future work could explore further enhancements to the STCT framework by improving correlation filtering and deep learning techniques. These techniques could utilize more powerful correlation filters to improve the robustness of target tracking in general scenarios or utilize larger datasets to train Siamese networks to cope with more complex scenarios.

Author Contributions: Conceptualization, Y.H. and H.G.; methodology, Y.H.; software, C.C.; Validation, Y.H. and Z.Z.; Formal analysis, J.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National University of Defense Technology Research Grant in 2023.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

Nomenclature

The following abbreviations are used in this manuscript:

Abbreviations	
STCT	The object tracker based on spatio-temporal context.
ADMM	Alternating direction method of multipliers.
AMRE	The average maximum response value related energy.
CF	The correlation filter.
OPE	One-pass evaluation.
Symbols	
$\hat{\boldsymbol{x}}_k$	The target feature after Fourier transformation in frame k-th.
$\hat{f}_k$	The correlation filter in the frequency domain.
z_k	The search map.
T_{ot}	Online training time.

Symbols	
sc	The scale factor.
$\boldsymbol{y}$	The expected Gaussian response.
$\boldsymbol{\phi}$	The state transition matrix.
$\boldsymbol{R}_k$	The response maps at frames k.
$\boldsymbol{B}_k$	The dynamic attention weight.
T	The adaptive threshold.
λ	The interpolation coefficient.
η	The amplification factor.
η_0	The intercept.
T_{tr}	Online object tracking time.

References

1. Fu, C.; Li, T.; Ye, J.; Zheng, G.; Li, S.; Lu, P. Scale-Aware Domain Adaptation for Robust UAV Tracking. *IEEE Robot. Autom. Lett.* **2023**, *8*, 3764–3771.
2. 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.
3. Zhang, Z.; He, Y.; Guo, H.; He, J.; Yan, L.; Li, X. Towards Robust Visual Tracking for Unmanned Aerial Vehicle with Spatial Attention Aberration Repressed Correlation Filters. *Drones* **2023**, *7*, 401.
4. Sun, L.; Li, X.; Yang, Z.; Gao, D. Visual Object Tracking Based on the Motion Prediction and Block Search in UAV Videos. *Drones* **2024**, *8*, 252.
5. Chen, F.; Wang, X.; Zhao, Y.; Lv, S.; Niu, X. Visual object tracking: A survey. *Comput. Vis. Image Underst.* **2022**, *222*, 103508.
6. El-Shafie, A.H.A.; Habib, S.E. Survey on hardware implementations of visual object trackers. *IET Image Process.* **2019**, *13*, 863–876.
7. Li, D.; Porikli, F.; Wen, G.; Kuai, Y. When Correlation Filters Meet Siamese Networks for Real-time Complementary Tracking. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 509–519.
8. He, Y.; Wu, J.; Xie, G.; Hong, X.; Zhang, Y. Data-driven relative position detection technology for high-speed maglev train. *Measurement* **2021**, *180*, 109468.
9. Bolme, D.S.; Beveridge, J.R.; Draper, B.A.; Lui, Y.M. 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.

10. 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.
11. Danelljan, M.; Häger, 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.
12. 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.
13. Danelljan, M.; Bhat, G.; Shahbaz Khan, F.; Felsberg, M. Eco: Efficient convolution operators for tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6638–6646.
14. Kiani Galoogahi, H.; Fagg, A.; Lucey, S. Learning background-aware correlation filters for visual tracking. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1135–1143.
15. Ye, J.; Fu, C.; Lin, F.; Ding, F.; Lu, G. Multi-Regularized Correlation Filter for UAV Tracking and Self-Localization. *IEEE Trans. Ind. Electron.* **2021**, *69*, 6004–6014.
16. Zheng, G.; Fu, C.; Ye, J.; Lin, F.; Ding, F. Mutation Sensitive Correlation Filter for Real-Time UAV Tracking with Adaptive Hybrid Label. *arXiv* **2021**, arXiv:2106.08073.
17. Zhang, F.; Ma, S.; Zhang, Y.; Qiu, Z. Perceiving Temporal Environment for Correlation Filters in Real-Time UAV Tracking. *IEEE Signal Process. Lett.* **2022**, *29*, 6–10.
18. Lukezic, A.; Vojir, T.; ˇCehovin 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.
19. 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.
20. Li, Y.; Zhang, H.; Yang, Y.; Liu, H.; Yuan, D. RISTrack: Learning response interference suppression correlation filters for UAV tracking. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 1-5.
21. Fu, C.; Li, B.; Ding, F.; Lin, F.; Lu, G. Correlation filters for unmanned aerial vehicle-based aerial tracking: a review and experimental evaluation. *arXiv* **2010**, arXiv:2010.06255.
22. Marvasti-Zadeh, S.M.; Cheng, L.; Ghanei-Yakhdan, H.; Kasaei, S. Deep Learning for Visual Tracking: A Comprehensive Survey. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 3943–3968.
23. Jiao, L.; Wang, D.; Bai, Y.; Chen, P.; Liu, F. Deep learning in visual tracking: A review. *IEEE Trans. Neural Netw. Learn. Syst. 2021*, **34**, 5497–5516.
24. Ondrasovic, M.; Tarabek, P. Siamese Visual Object Tracking: A Survey. *IEEE Access* **2021**, *9*, 110149–110172.
25. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H. Fully-convolutional siamese networks for object tracking. In *Proceedings of the Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, 8–10 and 15–16 October 2016*; Proceedings, Part II 14; Springer International Publishing: Cham, Switzerland, 2016; pp. 850–865.
26. 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.
27. Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; Yan, J. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4282–4291.
28. Wang, Q.; Zhang, L.; Bertinetto, L.; Hu, W.; Torr, P.H. Fast online object tracking and segmentation: A unifying approach. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 1328–1338.
29. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*.
30. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
31. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
32. Wang, M.; Liu, Y.; Huang, Z. Large Margin Object Tracking with Circulant Feature Maps. *IEEE Comput. Soc.* **2017**.
33. Wu, Y.; Lim, J.; Yang, M.-H. Object Tracking Benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848.