remote sensing

MDPI

Article

Learning Future-Aware Correlation Filters for Efficient UAV Tracking

Fei Zhang [1], Shiping Ma [1], Lixin Yu [2,*], Yule Zhang [3], Zhuling Qiu [1] and Zhenyu Li [4]

[1] Aeronautics Engineering College, Air Force Engineering University, Xi'an 710038, China; kgfzhang@163.com (F.Z.); mashiping@126.com (S.M.); kgdqzl@163.com (Z.Q.)
[2] Air Traffic Control and Navigation College, Air Force Engineering University, Xi'an 710051, China
[3] Air and Missile Defense College, Air Force Engineering University, Xi'an 710051, China; yule_zhang0921@163.com
[4] Harbin Institute of Technology, Harbing 150080, China; lzy011338@126.com
* Correspondence: ylxchinahai@sina.com

Abstract: In recent years, discriminative correlation filter (DCF)-based trackers have made considerable progress and drawn widespread attention in the unmanned aerial vehicle (UAV) tracking community. Most existing trackers collect historical information, e.g., training samples, previous filters, and response maps, to promote their discrimination and robustness. Under UAV-specific tracking challenges, e.g., fast motion and view change, variations of both the target and its environment in the new frame are unpredictable. Interfered by future unknown environments, trackers that trained with historical information may be confused by the new context, resulting in tracking failure. In this paper, we propose a novel future-aware correlation filter tracker, i.e., FACF. The proposed method aims at effectively utilizing context information in the new frame for better discriminative and robust abilities, which consists of two stages: future state awareness and future context awareness. In the former stage, an effective time series forecast method is employed to reason a coarse position of the target, which is the reference for obtaining a context patch in the new frame. In the latter stage, we firstly obtain the single context patch with an efficient target-aware method. Then, we train a filter with the future context information in order to perform robust tracking. Extensive experimental results obtained from three UAV benchmarks, i.e., UAV123_10fps, DTB70, and UAVTrack112, demonstrate the effectiveness and robustness of the proposed tracker. Our tracker has comparable performance with other state-of-the-art trackers while running at ∼49 FPS on a single CPU.

Keywords: visual tracking; unmanned aerial vehicle; discriminative correlation filter; future awareness; context learning; time series forecast

Citation: Zhang, F.; Ma, S.; Yu, L.; Zhang, Y.; Qiu, Z.; Li, Z. Learning Future-Aware Correlation Filters for Efficient UAV Tracking. *Remote Sens.* **2021**, *13*, 4111. https://doi.org/10.3390/rs13204111

Academic Editors: Peter Hofmann and Hossein M. Rizeei

Received: 15 September 2021
Accepted: 11 October 2021
Published: 14 October 2021

1. Introduction

Visual object tracking is a popular but challenging task in the domain of multimedia and computer vision. Given a video sequence, the task is to precisely estimate the position of the target of interest. With the popularity of unmanned aerial vehicles (UAVs), visual tracking applied for UAV platforms has attracted extensive attention, e.g., public security [1], disaster investigation [2], and remote sensor mounting [3]. Although this technique has acquired impressive progress, its performance is unsatisfactory when the target undergoes UAV-specific tracking challenges, such as viewpoint change, fast motion, and low resolution.

There are two main streams in visual tracking community: DCF-based trackers [4–7] and Siamese-based trackers [8]. Although Siamese-based trackers achieved impressive tracking performance using a GPU or GPUs, the complex calculation of the deep network inevitably brings large energy loss to the mobile platform such as UAVs. DCF-based trackers [9,10] is one of the most suitable choices for source-limited UAV platforms because of their balanced accuracy and speed as well as low cost. However, using synthesized training samples [11]

for training inevitably impedes the discriminative power of the filter, that is, boundary effects. In the literature, in order to solve this problem, many attempts have been conducted to enhance filter discrimination for the target and its environment, such as fixed or adaptive spatial regularization [12–15] and context learning [16–18]. For context learning, these methods [16–18] suppress the response of context patch in multiple directions to zero, thus achieving effective performance improvement. Nevertheless, multiple context patches may introduce irrelevant background noise, resulting in a suboptimal filter. Moreover, the feature extraction of these patches, especially when using deep features [18], hinders the real-time ability of trackers with context learning.

Moreover, traditional DCF-based trackers model the filter by virtue of historical information, e.g., accumulated training samples [4,11–15], previously generated filters [5,19,20], or response maps [21–23]. Although DCF-based trackers benefit from prevenient clues, this paradigm may fail to deal well with complex and changeable UAV tracking challenges, such as fast motion and viewpoint change. In these cases, both the volatile environment and the target appearance changes bring about severe uncertainties. It has been proved that the information in the future frame [20] has played a vital role in improving adaptability of the tracker. In [20], the object observation in the next frame is predicted by exploring spatial-temporal similarities of the target change in consecutive frames. Then, it is integrated with historical samples to form a more robust object model. However, this similarity assumption is not always valid for UAV object tracking because of the complex changeable nature of UAV tracking scenarios.

With respect to the above concerns, we propose a two-stage correlation filter tracker that can efficiently exploit the contextual information of the upcoming frame. The achievement of this purpose depends on two irreversible future-aware stages, i.e., future state awareness and future context awareness. The former stage is for predicting the spatial location change of the target in the upcoming frame, and the latter is for suppressing distractions caused by future complex background while enhancing filter discriminative power. In the first stage, when a new frame is coming, the simple yet effective single exponential smoothing forecast method [24] is used to predict a coarse target position. In the latter stage, we employ an efficient mask generation method to segment a single context patch based on the coarse position. Then, the segmented contextual information is incorporated into the training phase for discrimination improvement. Lastly, the more powerful filter rectifies the prediction error of the first stage. We perform comprehensive experiments on three challenging UAV benchmarks, i.e., DTB70 [25], UAV123_10fps [26], and UAVTrack112 [27]. The results confirm that the proposed tracker has superiority in terms of accuracy and speed compared with 29 other state-of-the-art trackers. Figure 1 shows the overall performance of all trackers on DTB70 [25] benchmark. Clearly, our tracker has comparable performance against other trackers while maintaining real-time speed on a single CPU, which demonstrates that the FACF tracker is suitable for real-time UAV applications.

The main contributions are summarized as follows:

- A coarse-to-fine DCF-based tracking framework is proposed to exploit the context information hidden in the frame that is to be detected;
- Single exponential smoothing forecast is used to provide a coarse position, which is the reference for acquiring a context patch;
- We obtain a single future-aware context patch through an efficient target-aware mask generation method without additional feature extraction;
- Experimental results on three UAV benchmarks verify the advancement of the proposed tracker. Our tracker can maintain real-time speed in real-world tracking scenarios.

The remainder of this paper is organized as follows: Section 2 generalizes the most relevant works; Sections 3 introduces the baseline tracker; Section 4 details the proposed method; Section 5 exhibits extensive and comprehensive experiments; and Section 6 provides a brief summary of this work.

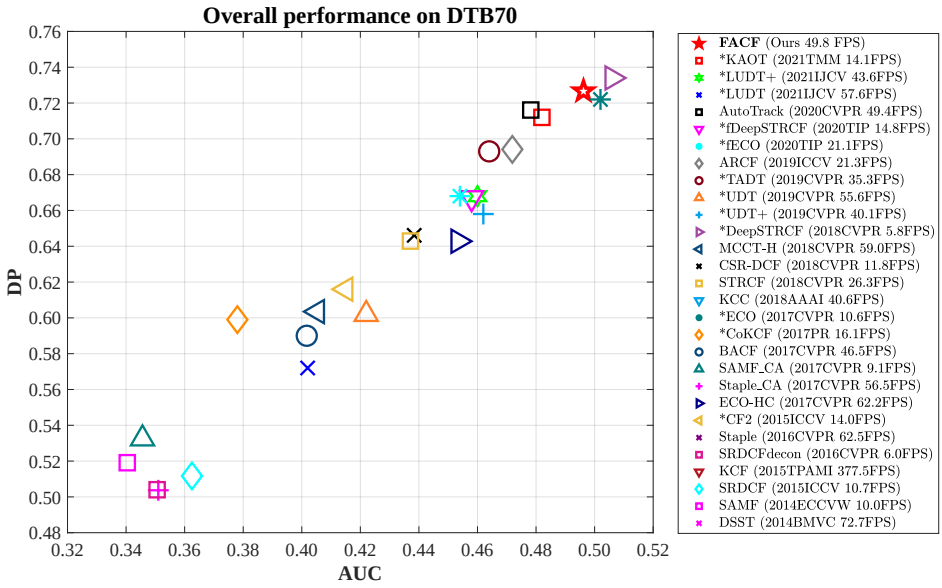

Figure 1. Overall performance based on area under curve (AUC) and distance precision (DP) between the proposed FACF tracker and 29 other state-of-the-art trackers on DTB70 [25] benchmark. AUC and DP are two metrics for evaluating tracking accuracy for which its detailed explanation is in Section 5. The legend provides detailed speed values of each tracker. The superscript $*$ represents the GPU-based tracker.

2. Related Works

In this section, we briefly discuss the most related trackers, including DCF-based trackers, trackers with context learning, trackers with future information, and trackers for UAVs.

2.1. DCF-Based Trackers

DCF-based trackers formulate the tracking task as a ridge regression problem, with the view of training a filter to distinguish the target from the background. The use of a cyclic matrix and calculation in the Fourier domain simplifies the filter optimization process. Recently, many methods were proposed for improving tracking accuracy from different aspects. These methods include kernel tricks [11], scale estimation [6,28,29], mitigation of boundary effects [12–15], solutions for temporal degradation [5,19], training-set management [7,30], more powerful feature representation [7,11,31–33], and consequent feature de-redundancy [19,34,35]. In general, these above methods collect historical known information to predict future unknown target states. Future information is not considered to be utilized for raising the robustness and adaptability of the tracker.

2.2. Trackers with Context Learning

In the literature, context learning is one of the most efficacious strategies for discriminating the enhancement of the filter. Mueller et al. [16] proposed a novel context-aware DCF-based tracker (CACF). They used multiple regularization terms to repress the response of context patches in four directions around the target. Later, different trackers [17,18,36] equipped with context learning all received significant performance improvement. In detail, Fu et al. [17] selected more reasonable surrounding samples according to the position and scale of the tracked object. Based on CACF [16], Yan et al. [36] cropped four context patches on the basis of the location of distractors response generated in the last frame. To repress context noise more adequately, Li et al. [18] considered the context samples located at the four corners. Moreover, to avoid frame-by-frame learning, they proposed a periodic key frame selection method for context learning and used temporal regularization to retain the discrimination for background interference. These methods mentioned above are all limited to heavy feature extraction of context samples, especially when using deep features.

In addition, it is easy for these methods to introduce background noise outside of the search region. Different from the methods mentioned above, our work produces a context-aware mask to segment a single pixel-level context sample, which can drastically increase speed and evade unrelated interference while boosting performance remarkably.

2.3. Trackers with Future Informarion

Most trackers [12–18,21–23,34,36,37] based on the DCF framework update the target appearance model with a fixed online learning rate through historical observations. Then, the filter trained with the appearance model is used for predicting the target state in the upcoming frame. By mining temporal-spatial similarities in consecutive video sequences, Zhang et al. [20] predicted a target observation of the next frame and incorporated it into the target model with a large learning rate. It can improve the adaptation of the model to target appearance variations, thus promoting tracking performance. This method implies that the change rate of the target appearance is constant. However, it is not always valid, as there often exists fast motion and viewpoint change in most UAV tracking cases. Different from the similarity assumption [20] for obtaining future information, we predicted a coarse position based on single exponential smoothing forecast. Based on this position, a true context patch sample in the next frame is efficiently segmented for context learning.

2.4. Trackers for UAVs

DCF-based trackers [10] are gradually becoming the most pervasive tracking paradigm in the UAV tracking community due to their high efficiency. In the real-world UAV object tracking process, low resolution, fast object motion, and viewpoint change pose extreme challenges. With respect to these issues, a large number of works are proposed for better tracking performance. They can mainly be divided into following strategies: spatial attention mechanism, adaptive regression label, and temporal consistency. Concretely, in contrast to the fixed spatial attention in [12], later works [23,38] proposed a dynamic spatial attention mechanism using target salient information or response variations. Different from traditional predefined label, References [39,40] generate an adaptive regression label to repress the distractors. On the other hand, References [21,41] keep temporal consistency at the responselevel, which largely improves positioning accuracy. With the exception of the above works, there also exists work [42] that focuses on adaptive templates and model updates by using cellular automata and high confidence assessment. Recently, some light-weight Siamese-based trackers [27,43,44] are designed for UAV object tracking, such as SiamAPN++ [27] and MultiRPN-DIDNet [44]. All the above-mentioned works ignore the threat posed by the rapid context changes in real-world UAV tracking. By incorporating the future contextual information into filter learning, the proposed tracker with handcrafted features is more discriminative to the scene changes of UAV object tracking.

3. Revisit BACF

In this section, we review the BACF [13] tracker, which is the baseline in this work. To mitigate the problem of boundary effects, background-aware correlation filter (BACF) [13] enlarges the search region and introduces a binary matrix $\mathbf{B} \in \mathbb{R}^{\mathrm{M}\times\mathrm{N}}(\mathrm{M}\ll\mathrm{N})$ to crop more complete negative samples, which largely improves tracking performance. In this work, we select the BACF tracker as the baseline. Given the vectorized desired response $\mathbf{y} \in \mathbb{R}^{\mathrm{N}\times 1}$ and the vectorized training samples of each channel $\mathbf{x}^d \in \mathbb{R}^{\mathrm{N}\times 1}$, the filter $\mathbf{w}$ in the current frame f can be obtained by minimizing the following objective:

$$\epsilon(\mathbf{w}_f) = \frac{1}{2}\left\| \sum_{d=1}^{\mathrm{D}} \mathbf{B}\mathbf{x}_f^d \circledast \mathbf{w}_f^d - \mathbf{y} \right\|_2^2 + \frac{\lambda}{2}\sum_{d=1}^{\mathrm{D}} \left\| \mathbf{w}_f^d \right\|_2^2, \tag{1}$$

where λ is the filter regularization parameter, and D is the total number of channels. $\circledast$ represents the correlation operator.

Unfortunately, a large number of real negative samples inevitably introduce background noise, which results in insufficient discrimination of the tracker, especially when distractors appear in the future frame.

4. Proposed Approach

In this section, we first carefully analyze the existing problems. Then, we introduce the proposed method, including two stages: future state awareness and future context awareness. Lastly, the complete tracking procedure is detailed.

4.1. Problem Formulation

In the tracking process of DCF-based trackers, the filter learned in the current frame is used to localize the object in the upcoming frame. Variations of the tracked object and its surroundings in the new frame are unforeseeable. The interference of surroundings, along with the boundary effects, may result in tracking drift. Many trackers [16–18,36] exploit current context patches surrounding the tracked object to enhance the discrimination power. However, this strategy cannot ensure robustness when facing unpredictable background variations in the new frame.

Usually, DCF-based trackers use a padding object patch, which contains certain context information, and its corresponding Gaussian label for filter training. It is expected that the response within the target region is a Gaussian shape while the response in the background region tends to zero. For trackers [16,18] with context learning, they obtain context information by cropping several context patches around the target and then exploits context regularization terms to restrain the response of context patches to zero. As shown in the top figure in Figure 2, context patches (blue dotted box) are the same size as the object patch (yellow dotted box). Therefore, these context patches may include the target of interest, which is contradictory to the regression term. Moreover, this strategy brings heavy calculation burden and redundancy. On the one hand, context patches need to be cropped and to extract features separately. On the other hand, context patches contain a large percentage of the overlap region, which is not efficient.

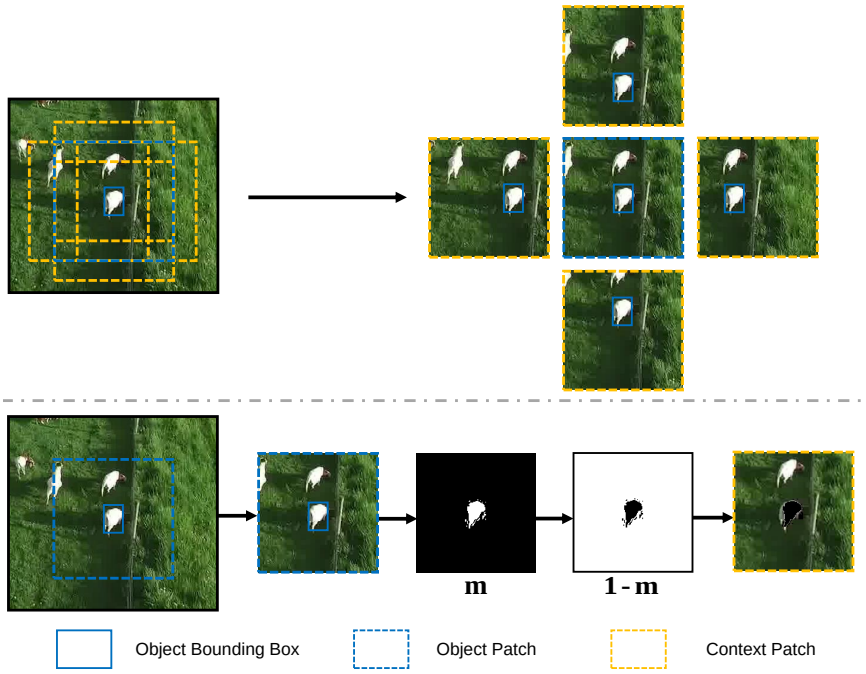

Figure 2. Different strategies of obtaining context patch. **Top**: traditional method. **Bottom**: our efficient method.

With respect to these concerns, we tried to utilize context information in the upcoming frame for filter training, which aims to cope with unpredictable background variations. In

addition, designing an efficient method to obtain the context patch is another goal of this work. Inspired from two-stage detectors [45,46], we propose a coarse-to-fine search strategy to improve localization precision. The pipeline of the proposed tracker is shown in Figure 3. Specifically, preliminary prediction of the object location is performed to precisely segment contextual pixels in the new frame. Then, the future-aware filter trained with future context information corrects the prediction bias in order to obtain the final prediction.

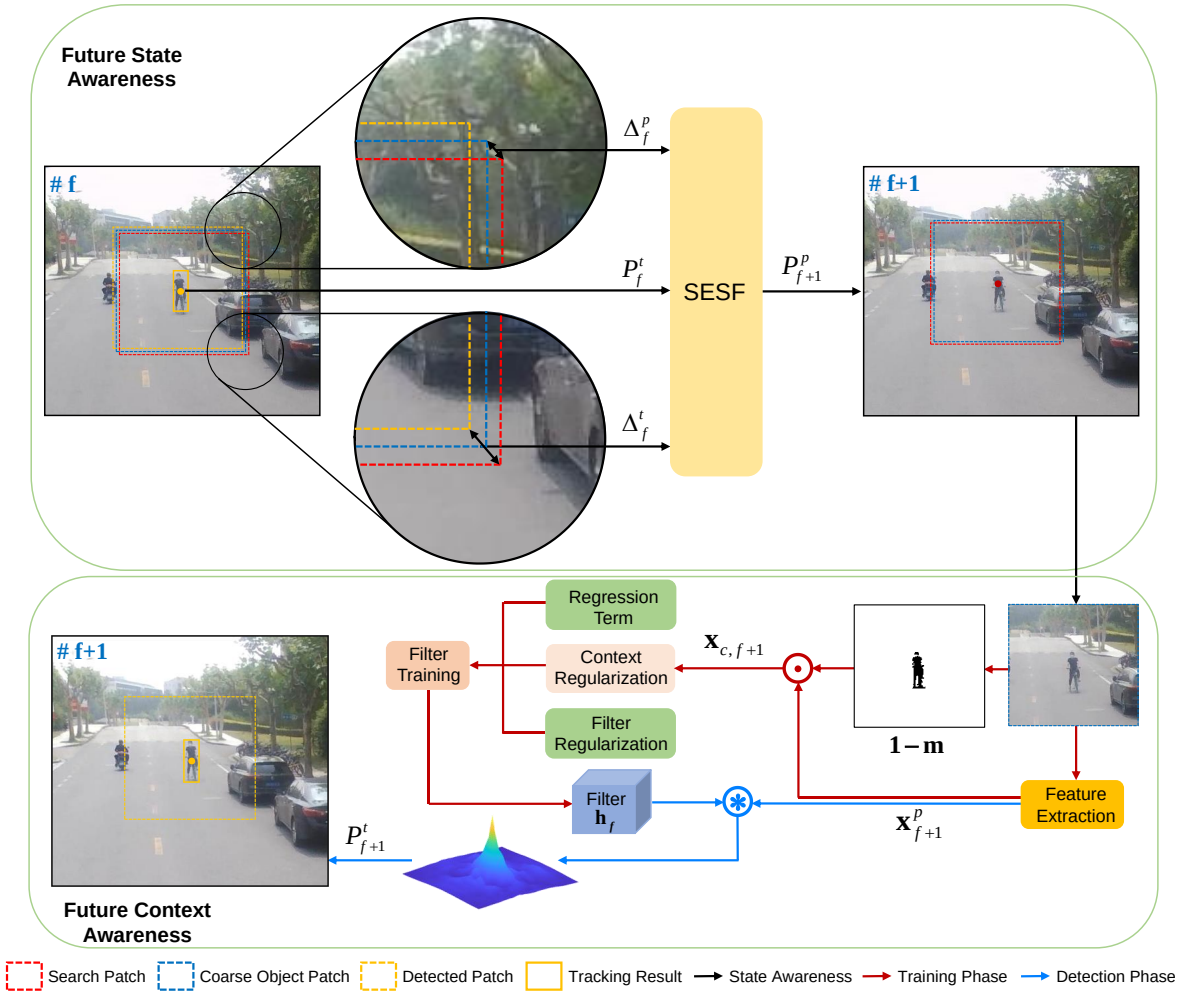

Figure 3. Tracking process of the proposed FACF tracker, which consists of two stages: future state awareness and future context awareness. Future state awareness: when a new frame is upcoming, we used the single exponential smoothing forecast to obtain a coarse target position. Future context awareness: we extract feature maps of the predicted region with the coarse position. Next, the feature maps are multiplied by the context mask to obtain the feature of the single context patch, which is then fed into filter training phase. Finally, the outputted filter performs target localization on the feature maps of the predicted region.

4.2. Stage One: Future State Awareness

From another perspective, the task of visual tracking is to predict target displacement in subsequent sequences. Normally, the motion state of the target within a certain time interval is approximately unchanged. Based on this assumption, we use a simple and effective time series forecasting method, i.e., single exponential smoothing forecast (SESF) [24], to roughly estimate the displacement of the target in the new frame. Let us assume that the true displacement vector (estimated by the filter) $\Delta_f^t = [\Delta x_f^t, \Delta y_f^t]$ of the f-th frame is given, the predicted displacement (estimated by SESF) Δ_{f+1}^p in the $(f+1)$-th frame can be obtained by following formula:

$$\Delta_{f+1}^{p} = \alpha\Delta_{f}^{t} + (1-\alpha)\Delta_{f}^{p}, \tag{2}$$

where Δ_{f}^{p} is the predicted displacement vector in the f-th frame. $[\Delta x, \Delta y]$ represents the displacement deviation in the x and y direction. α is the smoothing index. So far, we can obtain the initial prediction P_{f+1}^{p} of the target position in the next frame:

$$P_{f+1}^{p} = P_{f}^{t} + \Delta_{f+1}^{p}, \tag{3}$$

where P_{f}^{t} denotes the estimated position by the filter in the f-th frame. As shown in Figure 4, we provide some examples to verify the effectiveness of single exponential smoothing forecast for the initial prediction. We compare the center location error (CLE) of the initial and final prediction by the SESF module and our filter, respectively. In some cases, the initial prediction is more accurate than the final prediction.

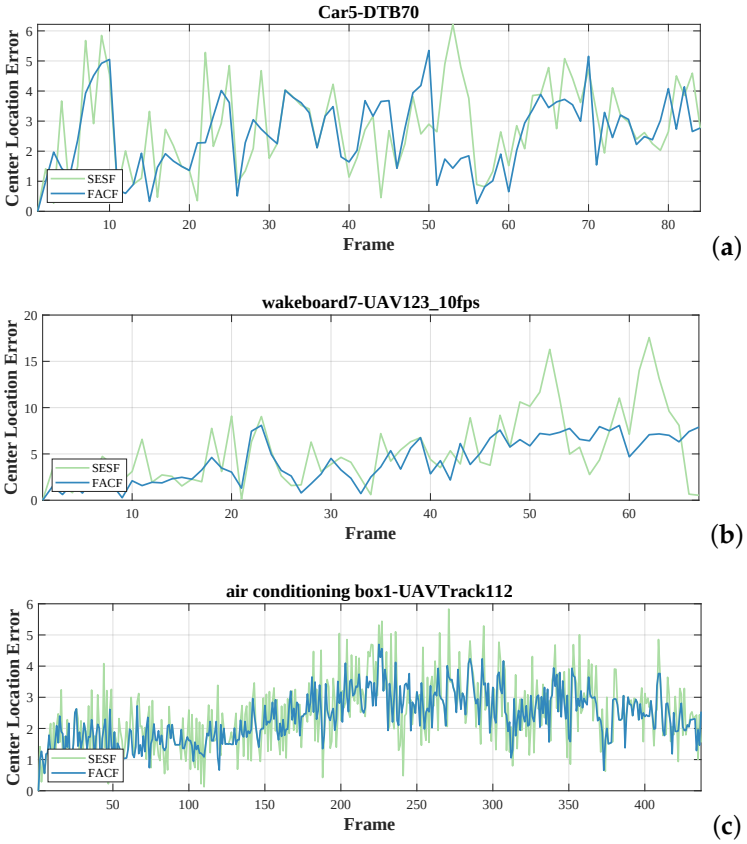

Figure 4. Center location error (CLE) comparison of the single exponential smoothing forecast method (SESF) and the proposed tracker (FASF) on three challenging seqeuences. From top to bottom are (**a**) Car5 from DTB70 [25], (**b**) wakeboard7 from UAV123_10fps [26], and (**c**) air conditioning box1 from UAVTrack112 [27].

Next, the methods for using future context information and obtaining the final position P_{f+1}^{t} on the basis of the result of single exponential smoothing forecasts will be discussed.

4.3. Stage Two: Future Context Awareness

4.3.1. Fast Context Acquisition

Usually, previous context learning methods [16–18] are limited to tedious feature extraction of context patches in multiple directions, which also increase the computation complexity of filter training. Moreover, context information outside of the search region

may introduce unnecessary information into the model. Furthermore, previous methods use the current context information for discrimination enhancement, which cannot deal with the unpredictable changes in the new frame, such as the appearance of similar targets or sudden viewpoint change.

While most trackers [14,47] strive to improve focus on the target through mask generation methods, we take an alternative approach. As shown in the bottom figure of Figures 2 and 3, when a new frame is coming, we first obtain the coarse object patch with initial prediction P_f^p. Then, we use an efficient and effective mask generation method [47] to acquire the mask $\mathbf{m}$ of the target. Finally, the context-aware mask $(1-\mathbf{m})$ is used to segment a single context patch based on the coarse object patch. In practice, we directly segment the features of the context patch after obtaining the features of the predicted patch for efficiency. The coarse object patch is regarded as the new search region to correlate with the filter. Then, we can acquire the final prediction.

4.3.2. Filter Training

Based on BACF [13], we incorporate future context information in the pixel-level into the training phase. The objective function of the proposed tracker is expressed as follows:

$$\begin{aligned}\epsilon(\mathbf{w}_f) = &\frac{1}{2}\left\|\sum_{d=1}^{\mathrm{D}}\mathbf{B}\mathbf{x}_f^d \circledast \mathbf{w}_f^d - \mathbf{y}\right\|_2^2 + \frac{\lambda}{2}\sum_{d=1}^{\mathrm{D}}\left\|\mathbf{w}_f^d\right\|_2^2 \\ &+ \frac{\gamma}{2}\left\|\sum_{d=1}^{\mathrm{D}}\mathbf{B}\mathbf{x}_{c,f+1}^d \circledast \mathbf{w}_f^d\right\|_2^2,\end{aligned} \tag{4}$$

where $\mathbf{x}_{c,f+1} = \mathbf{x}_{f+1}^p \odot (1-\mathbf{m})$ represents the surrounding context of the sought object in the upcoming frame $(f+1)$, and $\odot$ is the dot product operator. γ is the context regularization parameter.

Denoting auxiliary variable $\mathbf{h}^d = \mathbf{B}^{\mathsf{T}}\mathbf{w}^d \in \mathbb{R}^{\mathrm{N}\times 1}$, Equation (4) can be rewritten as follows.

$$\begin{aligned}\epsilon(\mathbf{w}_f, \mathbf{h}_f) = &\frac{1}{2}\left\|\sum_{d=1}^{\mathrm{D}}\mathbf{x}_f^d \circledast \mathbf{h}_f^d - \mathbf{y}\right\|_2^2 + \frac{\lambda}{2}\sum_{d=1}^{\mathrm{D}}\left\|\mathbf{w}_f^d\right\|_2^2 \\ &+ \frac{\gamma}{2}\left\|\sum_{d=1}^{\mathrm{D}}\mathbf{x}_{c,f+1}^d \circledast \mathbf{h}_f^d\right\|_2^2.\end{aligned} \tag{5}$$

After converting Equation (5) to the Fourier domain, the augmented Largrangian form of Equation (5) is expressed as follows:

$$\begin{aligned}\epsilon(\mathbf{w}_f, \hat{\mathbf{h}}_f, \hat{\boldsymbol{\zeta}}_f) = &\frac{1}{2}\left\|\sum_{d=1}^{\mathrm{D}}\hat{\mathbf{x}}_f^d \odot \hat{\mathbf{h}}_f^d - \hat{\mathbf{y}}\right\|_2^2 + \frac{\lambda}{2}\sum_{d=1}^{\mathrm{D}}\left\|\mathbf{w}_f^d\right\|_2^2 \\ &+ \frac{\gamma}{2}\left\|\sum_{c=1}^{\mathrm{D}}\hat{\mathbf{x}}_{c,f+1}^d \odot \hat{\mathbf{h}}_f^d\right\|_2^2 + \sum_{d=1}^{\mathrm{D}}(\hat{\mathbf{h}}_f^d - \sqrt{\mathrm{N}}\mathbf{F}\mathbf{B}^{\mathsf{T}}\mathbf{w}_f^d)^{\mathsf{T}}\hat{\boldsymbol{\zeta}}_f^d \\ &+ \frac{\mu}{2}\sum_{d=1}^{\mathrm{D}}\left\|\hat{\mathbf{h}}_f^d - \sqrt{\mathrm{N}}\mathbf{F}\mathbf{B}^{\mathsf{T}}\mathbf{w}_f^d\right\|_2^2,\end{aligned} \tag{6}$$

where $\hat{}$ is the Discrete Fourier Transformation (DFT). $\hat{\boldsymbol{\zeta}}_f = [\hat{\boldsymbol{\zeta}}_f^{1\mathsf{T}}, \hat{\boldsymbol{\zeta}}_f^{2\mathsf{T}}, \ldots, \hat{\boldsymbol{\zeta}}_f^{\mathrm{D}\mathsf{T}}] \in \mathbb{R}^{\mathrm{ND}\times 1}$ and μ are the Largrangian vector and a penalty factor, respectively.

Then, the ADMM [48] algorithm is adopted to optimize Equation (6) by alternately solving the following three subproblems. Each subproblem has its own closed-form solution.

Subproblem $\mathbf{w}$:

$$\begin{aligned}\mathbf{w}_f^* = &\frac{\lambda}{2}\sum_{d=1}^{\mathrm{D}}\left\|\mathbf{w}_f^d\right\|_2^2 + \sum_{d=1}^{\mathrm{D}}(\hat{\mathbf{h}}_f^d - \sqrt{\mathrm{N}}\mathbf{F}\mathbf{B}^{\mathsf{T}}\mathbf{w}_f^d)^{\mathsf{T}}\hat{\boldsymbol{\zeta}}_f^d \\ &+ \frac{\mu}{2}\sum_{d=1}^{\mathrm{D}}\left\|\hat{\mathbf{h}}_f^d - \sqrt{\mathrm{N}}\mathbf{F}\mathbf{B}^{\mathsf{T}}\mathbf{w}_f^d\right\|_2^2.\end{aligned} \tag{7}$$

The solution of Equation (7) can be solved in the following spatial domain:

$$\mathbf{w}_f^* = \frac{\boldsymbol{\zeta}_f + \mu\mathbf{h}_f}{\lambda/\mathrm{N} + \mu}, \tag{8}$$

where $\boldsymbol{\zeta}_f$ and $\mathbf{h}_f$ can be obtained, respectively, by the inverse Fourier Transform, i.e., $\boldsymbol{\zeta}_f = \frac{1}{\sqrt{\mathrm{N}}}\mathbf{B}\mathbf{F}^{\mathsf{T}}\hat{\boldsymbol{\zeta}}_f$ and $\mathbf{h}_f = \frac{1}{\sqrt{\mathrm{N}}}\mathbf{B}\mathbf{F}^{\mathsf{T}}\hat{\mathbf{h}}_f$.

Subproblem $\hat{\mathbf{h}}_f$:

$$\begin{aligned}\hat{\mathbf{h}}_f^* = &\frac{1}{2}\left\|\sum_{d=1}^{\mathrm{D}}\hat{\mathbf{x}}_f^d \odot \hat{\mathbf{h}}_f^d - \hat{\mathbf{y}}\right\|_2^2 + \frac{\gamma}{2}\left\|\sum_{c=1}^{\mathrm{D}}\hat{\mathbf{x}}_{c,f+1}^d \odot \hat{\mathbf{h}}_f^d\right\|_2^2 \\ &+ \sum_{d=1}^{\mathrm{D}}(\hat{\mathbf{h}}_f^d - \sqrt{\mathrm{N}}\mathbf{F}\mathbf{B}^{\mathsf{T}}\mathbf{w}_f^d)^{\mathsf{T}}\hat{\boldsymbol{\zeta}}_f^d + \frac{\mu}{2}\sum_{d=1}^{\mathrm{D}}\left\|\hat{\mathbf{h}}_f^d - \sqrt{\mathrm{N}}\mathbf{F}\mathbf{B}^{\mathsf{T}}\mathbf{w}_f^d\right\|_2^2.\end{aligned} \tag{9}$$

Since Equation (9) has element-wise dot product operation, we try to process the pixels on same location in order to decrease its nigh computational complexity. Equation (9) can be reformulated as follows.

$$\begin{aligned}\hat{\mathbf{h}}_f(n)^* = &\frac{1}{2}\left\|\hat{\mathbf{x}}_f(n)^{\mathsf{T}}\hat{\mathbf{h}}_f(n) - \hat{\mathbf{y}}(n)\right\|_2^2 + \frac{\gamma}{2}\left\|\hat{\mathbf{x}}_{c,f+1}(n)^{\mathsf{T}}\hat{\mathbf{h}}_f(n)\right\|_2^2 \\ &+ (\hat{\mathbf{h}}_f(n) - \mathbf{w}_f(n))^{\mathsf{T}}\hat{\boldsymbol{\zeta}}_f(n) + \frac{\mu}{2}\left\|\hat{\mathbf{h}}_f(n) - \mathbf{w}_f(n))\right\|_2^2.\end{aligned} \tag{10}$$

Take the derivative of Equation (10) with respect to $\hat{\mathbf{h}}_f(n)$ and set the result equal to zero, we can obtain the following.

$$\begin{aligned}\hat{\mathbf{h}}_f(n)^* = &\left(\hat{\mathbf{x}}_f(n)\hat{\mathbf{x}}_f(n)^{\mathsf{T}} + \gamma\hat{\mathbf{x}}_{c,f+1}(n)\hat{\mathbf{x}}_{c,f+1}(n)^{\mathsf{T}} + \mu\mathrm{N}\mathbf{I}_{\mathrm{D}}\right)^{-1} \\ &\left(\hat{\mathbf{x}}_f(n)\hat{\mathbf{y}}(n) - \mathrm{N}\hat{\boldsymbol{\zeta}}_f + \mu\mathrm{N}\hat{\mathbf{w}}_f(n)\right).\end{aligned} \tag{11}$$

In Equation (11), there exist matrix inversion, which is computationally heavy. With the assumption that $\hat{\mathbf{x}}_f(n)\hat{\mathbf{x}}_f(n)^{\mathsf{T}} + \gamma\hat{\mathbf{x}}_{c,f+1}(n)\hat{\mathbf{x}}_{c,f+1}(n)^{\mathsf{T}} = \sum_{a=0}^{1} S_a\hat{\mathbf{x}}_a(n)^{\mathsf{T}}\hat{\mathbf{x}}_a(n)$ (where $S_0 = 1$ and $S_1 = \gamma$, $\hat{\mathbf{x}}_0(n) = \hat{\mathbf{x}}_f(n)$ and $\hat{\mathbf{x}}_1(n) = \hat{\mathbf{x}}_{c,f+1}(n)$), the Sherman–Morrison [49] formula can be applied to accelerate computation. For convenience, we denote that $b = \mu\mathrm{N} + \sum_{a=0}^{1} S_a\hat{\mathbf{x}}_a(n)^{\mathsf{T}}\hat{\mathbf{x}}_a(n), \hat{s}_p(n) = \sum_{a=0}^{1} S_a\hat{\mathbf{x}}_a(n)^{\mathsf{T}}\hat{\mathbf{x}}_f(n)$. Then, Equation (11) can be converted into the following.

$$\begin{aligned}\hat{\mathbf{h}}_f(n)^* = &\frac{1}{\mu\mathrm{N}}\left(\hat{\mathbf{x}}_f(n)\hat{\mathbf{y}}(n) - \mathrm{N}\hat{\boldsymbol{\zeta}}_f + \mu\mathrm{N}\hat{\mathbf{w}}_f(n)\right) - \frac{\sum_{a=0}^{1} S_a\hat{\mathbf{x}}_a(n)}{\mu b} \\ &\left(\frac{1}{\mathrm{N}}\hat{s}_p(n)\hat{\mathbf{y}}(n) - \sum_{a=0}^{1}\hat{\mathbf{x}}_a(n)^{\mathsf{T}}\hat{\boldsymbol{\zeta}}(n) + \mu\sum_{a=0}^{1}\hat{\mathbf{x}}_a(n)^{\mathsf{T}}\hat{\mathbf{w}}_f(n)\right).\end{aligned} \tag{12}$$

The Lagrangian parameter $\hat{\boldsymbol{\zeta}}$ is described as follows:

$$\hat{\boldsymbol{\zeta}}^{(i+1)} = \hat{\boldsymbol{\zeta}}^{(i)} + \mu(\hat{\mathbf{h}}^{(i)} - \hat{\mathbf{w}}^{(i)}), \tag{13}$$

where (i) and $(i+1)$ represent the (i)-th and $(i+1)$-th iteration, respectively. The penalty fator μ is updated as $\mu^{(i+1)} = \min(\mu_{\max}, \delta\mu^{(i)})$.

4.3.3. Object Detection

The final object position can be estimated through the peak of the generated response map $\mathbf{R}$. Given the predicted patch $\hat{\mathbf{x}}^p_{f+1}$ and the trained filter $\hat{\mathbf{h}}_f$, the response map in frame $f+1$ can be obtained by the following:

$$\mathbf{R}_{f+1} = \mathcal{F}^{-1}\left(\sum_{d=1}^{\mathrm{D}} \hat{\mathbf{x}}^{p,d}_{f+1} \odot \hat{\mathbf{h}}^d_f\right), \tag{14}$$

where $\mathcal{F}^{-1}$ represents the inverse Fourier Transform (IFT). The biggest difference between our tracker and all previous trackers is that it contains future context information, resulting in more robustness relative to uncertain environment change.

4.3.4. Model Update

The object appearance model is updated by using linear weighted combination frame-by-frame:

$$\hat{\mathbf{x}}^M_f = (1-\beta)\hat{\mathbf{x}}^M_{f-1} + \beta\hat{\mathbf{x}}^o_f, \tag{15}$$

where $\hat{\mathbf{x}}^M$ represents the object models, and $\hat{\mathbf{x}}^o_f$ is the training sample of the current frame. β is the online learning rate.

4.4. Tracking Procedure

In this work, we train the scale filter [6] to estimate the scale variation. The complete tracking procedure of our tracker is shown in Algorithm 1.

Algorithm 1: FACF Tracker

Input: A video sequence with F frames.
The position P_1 and scale S_1 of the target in the first frame I_1.

Output: The position P_f and scale S_f of the target in subsequent frames $\mathrm{I}_f, f > 1$.

for *frame* $f = 1$ *to end* **do**
  if $f > 1$ **then**
    Stage I: Future state awareness
    Obtain the initial position P^p_f using Equations (2) and (3).
    Stage II: Future context awareness
    Training phase:
    Extract feature $\mathbf{x}^p_f$ of the search patch with position P^p_f and scale S_{f-1} and then obtain the context feature $\mathbf{x}_{c,f}$.
    Learn the filter $\hat{\mathbf{h}}_f$ using the context information of the upcoming frame with Equation (12).
    Detection phase:
    Generate the response map $\mathbf{R}_f$ with $\mathbf{x}^p_f$ and $\hat{\mathbf{h}}_f$ using Equation (14).
    Obtain the final object position P^t_f (P_f) via the response map $\mathbf{R}_f$ and estimate the scale S_f.
    Return P_f and S_f.
  end
  Update object appearance model $\mathbf{x}^M_f$ using Equation (15).
end

5. Experiments

In this section, we perform extensive and comprehensive experiments on three challenging UAV benchmarks. First, we introduce the detailed experimental settings, including parameters, benchmarks, metrics, and platform. Next, we compare our tracker with 29 other state-of-the-art trackers with handcrafted or deep features. Then, we verify the rationality of the parameters and the effectiveness of each component. Afterward, different context learning strategies are analyzed. The last subsection provides some failure cases of the proposed tracker.

5.1. Implementation Details

5.1.1. Parameters

Our tracker uses Hog, CN, and Grayscale features for object representation. We use two ADMM iterations to train the filter. The learning rate β of the model update is set to 0.019, and the context regularization parameter is chosen as $\gamma = 0.009$. The smoothing index α is set to 0.88. During the entire experiment, the parameters of the proposed tracker remain unchanged. The other trackers used for comparison retain their initial parameter setting. The code of our tracker is available at https://github.com/FreeZhang96/FACF, accessed on 10 September 2021.

5.1.2. Benchmarks

Experiments are conducted on three well-known UAV benchmarks involving DTB70 [25], UAV123_10fps [26], and the recent built UAVTrack112 [27]. These benchmarks have 305 video sequences in total, which are captured on UAV platforms.

5.1.3. Metrics

We use the one pass evaluation (OPE) norm to test all trackers. Evaluations of tracking accuracy are based on IoU and CLE. IoU (the Intersection Over Union) refers to the intersection of the bounding boxes between the prediction and groundtruth. CLE (Center Location Error) denotes the location error (pixels) between the predicted center location and the true location. When IoU or CLE exceeds a given threshold, the tracking results are deemed successful. If we set different thresholds (IoU $\in [0, 1]$ and CLE $\in [0, 50]$), we can obtain the success plots and precision plots. The area under the curve (success plots) is denoted as AUC. DP (Distance Precision) represents the score in precision plots when CLE = 20 pixels. FPS (Frame Per Second) is used for speed measurement of each tracker.

5.1.4. Platform

All experiments are carried out using Matlab2019b. The experimental platform is a PC with an Intel(R) Core(TM) i7-9750H CPU (2.60 GHz), 32 GB RAM, and a single RTX 2060 GPU.

5.2. Performance Comparison

5.2.1. Comparison With Handcrafted-Based Trackers

In this part, we comprehensively compare our tracker with other 16 state-of-the-art handcrafted trackers, i.e., STRCF [5], SAMF [28], KCF [11], DSST [6], ECO_HC [7], Staple [50], KCC [29], SAMF_CA [16], ARCF [21], AutoTrack [23], SRDCF [12], MCCT_H [37], CSRDCF [14], BACF [13], Staple_CA [16], and SRDCFdecon [30].

Overall Evaluation. Precision and success plots of our tracker and other trackers on all three benchmarks are presented in Figure 5.

DTB70 [25] benchmark contains 50 video sequences with 12 attributes. Our tracker has the best AUC and DP scores, namely 0.496 and 0.727, respectively. The AUC and DP scores surpass the second excellent tracker AutoTrack [23] 1.8% and 1.1%, respectively.

UAV123_10fps [26] is a large benchmark composed of 123 challenging UAV video sequences. We report the precision and success plots in Figure 5a. The proposed tracker FACF outperforms other trackers in terms of AUC and DP scores.

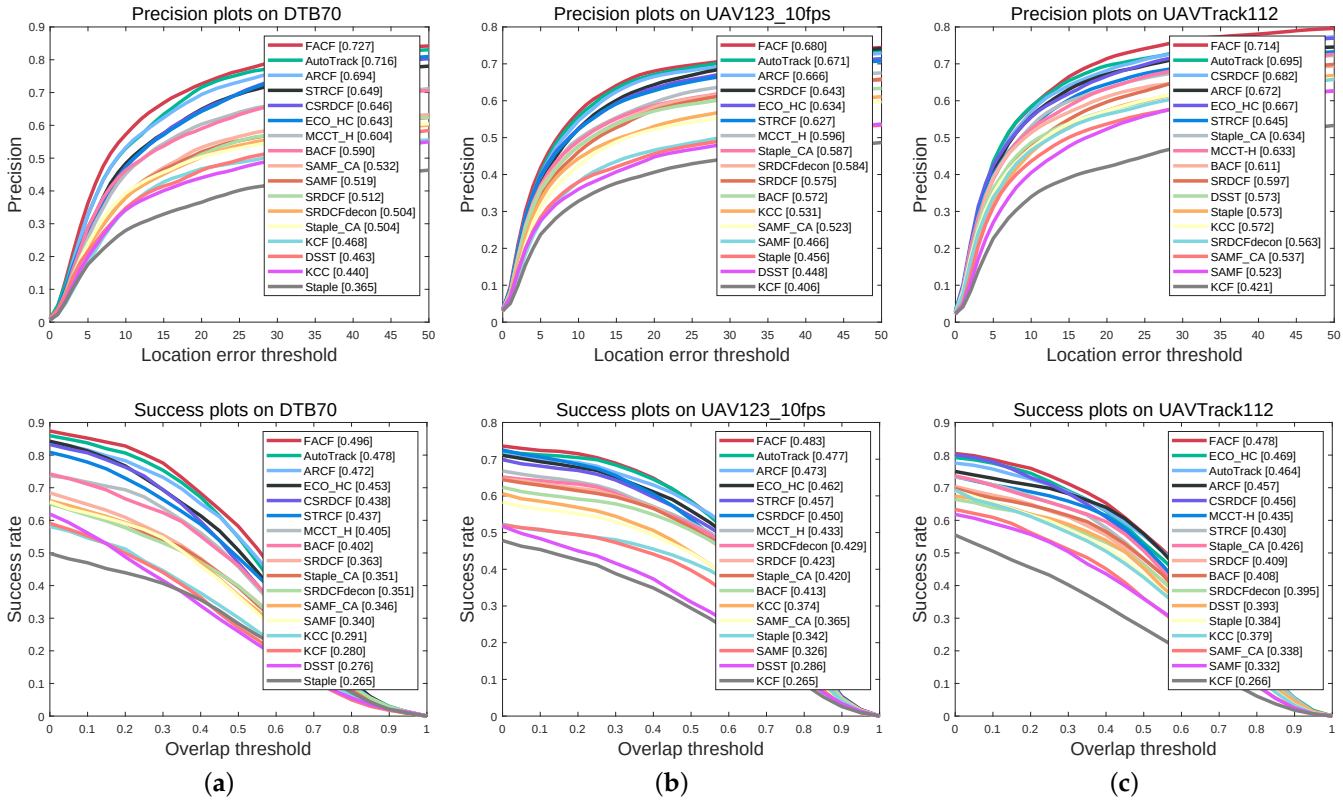

Figure 5. Overall performance comparison of the proposed FACF tracker and other 16 state-of-the-art handcrafted feature-based trackers on (**a**) DTB70 [25], (**b**) UAV123_10fps [26], and (**c**) UAVTrack112 [27]. First row: precision plots. Second row: success plots.

UAVTrack112 [27] is a recent newly built benchmark that is collected by DJI Mavic Air2. This benchmark contains 112 sequences with 13 attributes. From the plots in Figure 5c, our tracker performs the best with the AUC and DP scores of 0.478 and 0.709, respectively.

Table 1 presents the average AUC, DP, and speed comparison between our tracker and other handcrafted-based trackers on three benchmarks. In terms of AUC and DP, our tracker FACF performs best (0.486 and 0.707), followed by AutoTrack [23] (0.473 and 0.694) and ARCF [21] (0.467 and 0.677). The average speed of our FACF tracker can reach 48.816 FPS, which is sufficient for real-time applications.

Attribute-oriented Evaluation. To verify the performance of the proposed tracker in UAV-specific scenarios, this part provides extensive attribute-based analysis following attribute categorization in [10]. The new attributes for all benchmarks include VC (camera motion and viewpoint change), FM (fast motion), LR (low resolution), OCC (partial occlusion and full occlusion), and IV (illumination variation). Following the new attributes, we take the average AUC/DP score of all attributes (in all benchmarks) belonging to a new attribute as the final score. For example, as for the DP score of VC, five scores of all related attributes (camera motion and viewpoint change in both of UAV123_10fps [26] and UAVTrack112 [27] and fast camera motion in DTB70 [25]) are averaged in order to obtain the desired result. Table 2 exhibits the average performance of 17 different trackers under these specific attributes. Our tracker achieves the best AUC and DP scores under the VC, FM, and LR attributes.

Table 1. Average performance of all trackers on benchmarks DTB70 [25], UAV123@10fps [26], and UAVTrack112 [27]. **Red**, **green**, and **blue** represent the top three trackers in terms of DP, AUC, and FPS, respectively.

Tracker	FACF	AutoTrack	ARCF	STRCF	MCCT-H	KCC	CSRDCF	BACF	ECO-HC	Staple_CA	SAMF_CA	Staple	SRDCFdecon	KCF	SRDCF	SAMF	DSST
Venue	-	'20CVPR	'19ICCV	'18CVPR	'18CVPR	'18AAAI	'17CVPR	'17ICCV	'17CVPR	'17CVPR	'17CVPR	'16CVPR	'16CVPR	'15TPAMI	'15ICCV	'14ECCV	'14BMVC
DP	**0.707**	**0.694**	**0.677**	0.579	0.611	0.514	0.657	0.591	0.648	0.579	0.531	0.465	0.550	0.432	0.561	0.503	0.495
AUC	**0.486**	**0.473**	**0.467**	0.441	0.425	0.348	0.448	0.408	0.461	0.399	0.349	0.331	0.391	0.270	0.398	0.0.333	0.318
FPS	48.816	50.263	24.690	25.057	51.743	36.620	11.207	47.710	**62.163**	44.807	9.220	57.207	6.290	**533.250**	12.007	10.260	**87.777**

Table 2. AUC and DP scores of all trackers under UAV special attributes, including VC, FM, LR, OCC, and IV. **Red**, **green**, and **blue** denote the top three results.

Metric / Tracker	DP					AUC				
	VC	FM	LR	OCC	IV	VC	FM	LR	OCC	IV
FACF	**0.707**	**0.583**	**0.643**	**0.580**	**0.585**	**0.464**	**0.382**	**0.374**	**0.388**	**0.390**
AutoTrack [23]	**0.681**	**0.549**	**0.594**	**0.581**	**0.573**	**0.451**	**0.366**	**0.349**	**0.386**	**0.374**
ARCF [21]	**0.662**	**0.555**	**0.617**	0.600	**0.593**	**0.439**	**0.372**	**0.366**	**0.398**	**0.409**
STRCF [5]	0.612	0.480	0.524	0.546	0.470	0.401	0.315	0.300	0.355	0.314
MCCT-H [37]	0.540	0.400	0.472	0.529	0.406	0.355	0.262	0.252	0.332	0.267
KCC [29]	0.494	0.417	0.466	0.496	0.459	0.322	0.271	0.253	0.311	0.295
DSST [6]	0.503	0.396	0.475	0.439	0.429	0.310	0.247	0.249	0.273	0.262
CSRDCF [14]	0.633	0.510	0.607	**0.586**	0.531	0.411	0.345	0.324	0.371	0.331
BACF [13]	0.625	0.507	0.562	0.527	0.513	0.415	0.336	0.317	0.345	0.348
ECO-HC [7]	0.615	0.505	0.527	0.547	0.559	0.416	0.346	0.294	0.358	0.354
Staple_CA [16]	0.503	0.377	0.425	0.512	0.434	0.332	0.243	0.228	0.327	0.269
SAMF_CA [16]	0.542	0.458	0.505	0.516	0.442	0.360	0.307	0.283	0.331	0.304
Staple [50]	0.447	0.373	0.426	0.430	0.421	0.310	0.263	0.243	0.288	0.283
SRDCFdecon [30]	0.578	0.469	0.542	0.530	0.502	0.384	0.313	0.303	0.342	0.316
KCF [11]	0.376	0.310	0.380	0.363	0.353	0.251	0.227	0.262	0.242	0.240
SRDCF [12]	0.480	0.397	0.385	0.451	0.361	0.322	0.264	0.199	0.286	0.241
SAMF [28]	0.538	0.456	0.499	0.528	0.458	0.340	0.303	0.263	0.334	0.287

Figure 6 provides some detailed success plots of representative attribute-based analysis on different benchmarks. In terms of camera motion, fast motion, and low resolution, our tracker is in a leading position, surpassing the second place by a large margin. As shown in Figure 7, we compared the tracking results of the proposed tracker with five other state-of-the-art on 10 challenging video sequences. These compared trackers are STRCF [5], BACF [13], ECO_HC [7], AutoTrack [23], and ARCF [21]. In these UAV-specific scenarios (including VC, LR, and FM), the proposed tracker can achieve robust tracking while other trackers fail.

5.2.2. Comparison with Deep-based Trackers

Thirteen state-of-the-art deep-based trackers, i.e., LUDT [51], LUDT+ [51], fECO [52], fDeepSTRCF [52], TADT [53], CoKCF [54], UDT [55], CF2 [56], UDT+ [55], ECO [7], DeepSTRCF [5], and KAOT [18], are used for comparison. The overall performance of all trackers on DTB70 benchmarks is shown in Table 3. Our tracker has comparable performance with respect to other deep-based trackers. In particular, the AUC and DP scores (0.496 and 0.727) of the proposed tracker rank third and second, respectively. Meanwhile, our tracker FACF can achieve real-time speed, depending on a single CPU, while other deep-based trackers use GPU for acceleration.

5.3. Parameter Analysis and Ablation Study

5.3.1. The Impact of Key Parameter

To investigate the impact of key parameters for performance, we perform extensive experiments on DTB70 [25], UAV123_10fps [26], and UAVTrack112 [27] benchmarks. As shown in Figures 8 and 9, we only provide the results of the most important parameters, i.e., the smoothing index α and context regularization parameter γ.

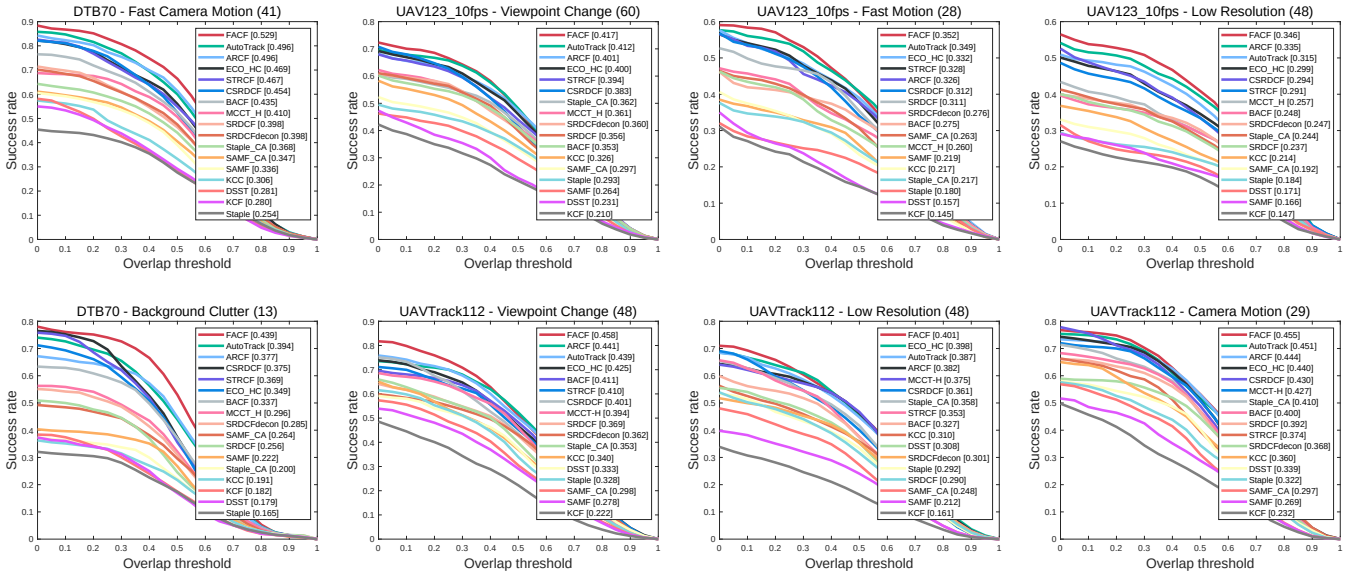

Figure 6. Attribute-based analysis of the proposed tracker and the other 16 state-of-the-art handcrafted feature-based trackers on DTB70 [25], UAV123_10fps [26] , and UAVTrack112 [27].

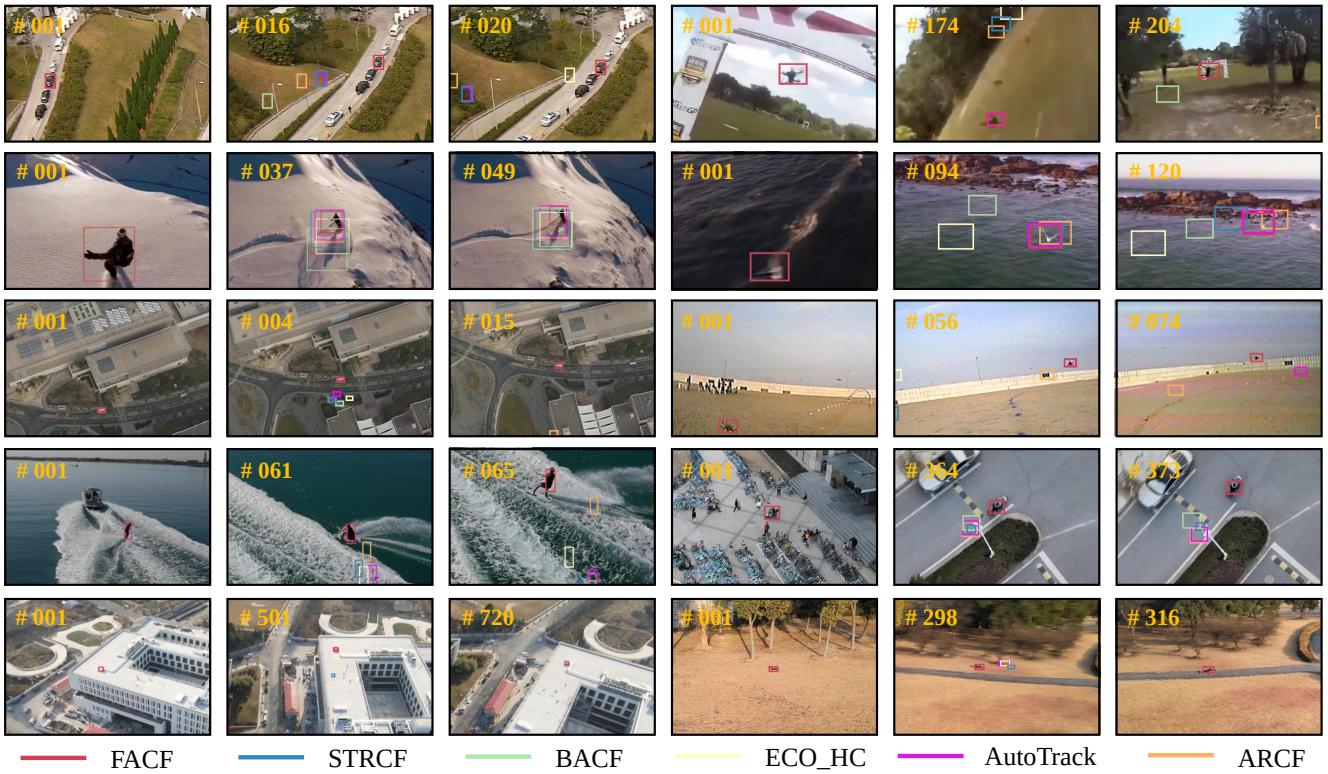

Figure 7. Visualization of the tracking results between the proposed tracker and 5 other state-of-the-art trackers on 10 challenging sequences. From left to right and from top to bottom are Car2, ChasingDrones, Gull1, and Snowboarding from DTB70 [25]; car13, uav3, and wakeboard from UAV123_10fps [26]; and courier1, electric box,and uav1 from UAVTrack112 [27].

Table 3. Performance comparision of the proposed tracker and 13 other deep-based trackers on DTB70 [25] benchmark. **Red**, **green**, and **blue** represent the top three trackers in terms of DP, AUC and FPS, respectively. The superscript $*$ means GPU spped.

Tracker	Venue	Type	DP	AUC	FPS
FACF	-	Hog+CN+Grayscale	**0.727**	**0.496**	**51.412**
KAOT [18]	'21TMM	Deep+Hog+CN	0.712	0.482	*14.045
LUDT+ [51]	'21IJCV	End-to-end	0.668	0.460	***43.592**
LUDT [51]	'21IJCV	End-to-end	0.572	0.402	***57.638**
fDeepSTRCF [52]	'20TIP	Deep+Hog+CN	0.667	0.458	*14.800
fECO [52]	'20TIP	Deep+Hog+CN	0.668	0.454	*21.085
TADT [53]	'19CVPR	End-to-end	0.693	0.464	*35.314
UDT+ [55]	'19CVPR	End-to-end	0.658	0.462	*40.135
UDT [55]	'19CVPR	End-to-end	0.602	0.422	*55.621
DeepSTRCF [5]	'18CVPR	Deep+Hog+CN	**0.734**	**0.506**	*5.816
MCCT [37]	'18CVPR	Deep+Hog+CN	**0.725**	0.484	*8.622
ECO [7]	'17CVPR	Deep+Hog	0.722	**0.502**	*10.589
CoKCF [54]	'17PR	Deep	0.599	0.378	*16.132
CF2 [56]	'15ICCV	End-to-end	0.616	0.415	*13.962

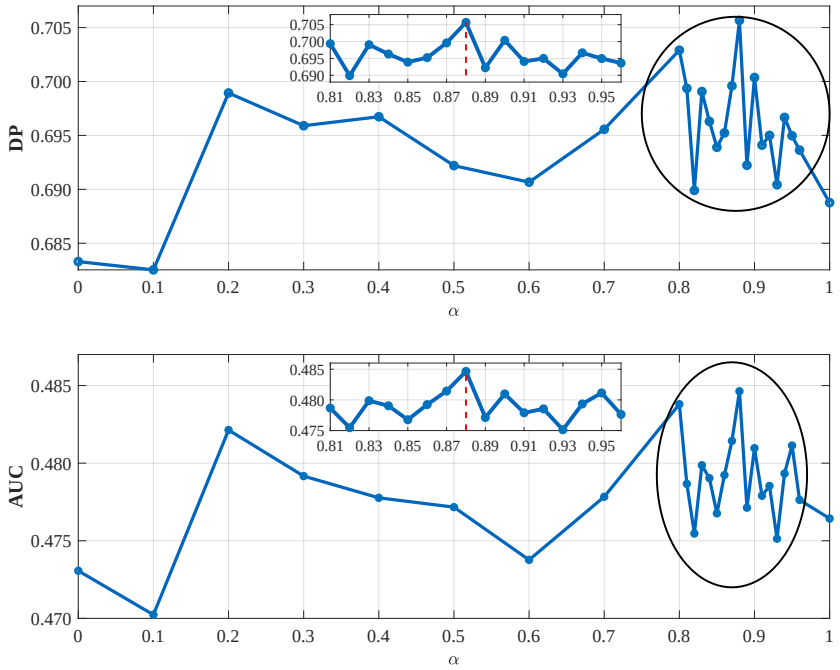

Figure 8. Average performance on three benchmarks when the smooth index α varies from 0 to 1. **Top**: the curve of AUC score. **Bottom**: the curve of DP score.

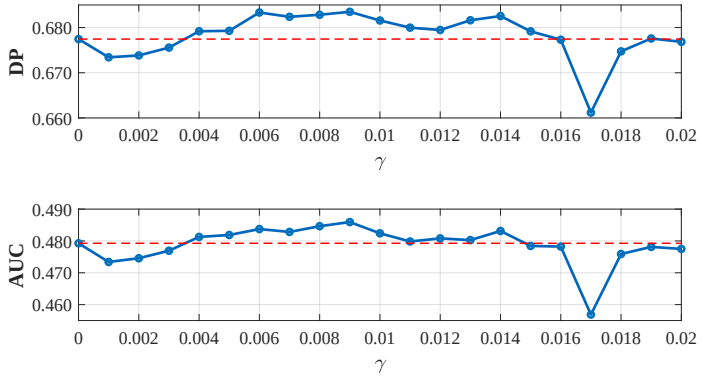

Figure 9. Average performance on three benchmarks when the context regularization parameter γ varies from 0 to 0.02. **Top**: the curve of AUC score. **Bottom**: the curve of DP score.

Smoothing index α. The smoothing index $\alpha \in [0, 1]$ controls the importance of the true displacement in the current frame for prediction. As for the tracking task, the displacement of the tracked object in a short time is approximate. Therefore, the value of the smoothing index is absolutely close to one. In Figure 8, we provide the average AUC and DP scores when α varies from 0 to 1. When the smoothing index reaches $\alpha = 0.88$ (red dotted line in Figure 8), our tracker achieves the best performance in terms of AUC and DP. The result confirms our analysis, and we select $\alpha = 0.88$.

Context regularization parameter γ. Figure 9 shows the average AUC and DP scores on all three benchmarks when the value of context regularization parameter varies from 0 to 0.02 with a step of 0.001. The red dotted line denotes the performance when $\gamma = 0$. As γ increases, AUC and DP reach the maximum value when $\gamma = 0.009$. Therefore, this work selects $\gamma = 0.009$.

5.3.2. The Vality of Component

To verify the effectiveness of each component, we develop four trackers equipped with different components. The average performance of different trackers on three benchmarks is shown in Figure 10. FACF-FCA denotes the FACF tracker disabled with future context awareness (FCA). FACF-FSA stands for the FACF tracker without future state awareness (FSA). FACF-(FSA+FCA) represents the BACF tracker with Hog, CN, and Grayscale features (baseline tracker). Clearly, both FSA and FCA modules can improve the tracking performance. The FSA module is beneficial for obtaining the future context patch as well as for improving tracking accuracy. Only when based on FSA can FCA contribute to more accurate tracking.

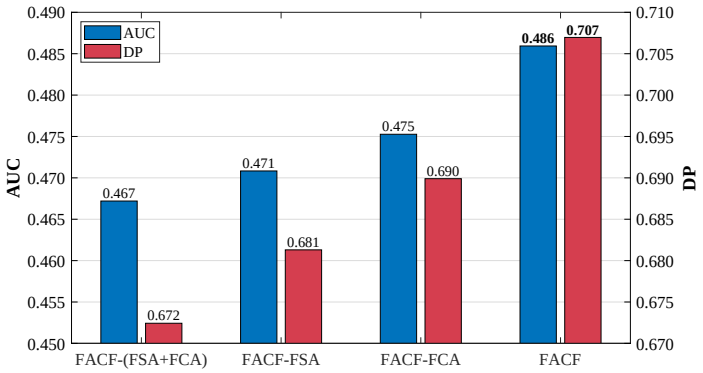

Figure 10. Ablation analysis of the proposed FACF tracker on three benchmarks.

5.4. The Strategy for Context Learning

In this part, different context learning methods based on the current or upcoming frame are compared on DTB70 [25] benchmark. We denote the context learning methods of

FACF and CACF as FCA and CA, respectively. FACF+CA means the FACF-FCA tracker with the context learning method in CACF [16]. BACF+CA and BACF+FCA are the baseline trackers (with Hog, CN, and Grayscales) that are using the context learning strategy in CACF [16] and FCA in our tracker, respectively. For these trackers, we finetuned the context regularization parameter to obtain the best performance on DTB70 [25] benchmarks (0.01 for FACF+CA, 0.008 for BACF+CA, and 0.001 for BACF+FCA). The results displayed in Table 4 indicate that the context learning strategy proposed in this paper is superior to that in CACF [16]. The fast context segmentation can not only avoid the context patch from containing the target but also effectively reduce computational complexity.

Table 4. Performance with different context learning strategies on DTB70 [25] benchmark.

Tracker	DP	AUC	FPS
FACF	0.727	0.496	51.412
FACF + CA	0.687	0.481	21.578
BACF + FCA	0.701	0.484	46.007
BACF + CA	0.679	0.477	19.427

5.5. Failure Cases

Figure 11 visualizes three representative and challenging sequences from three benchmarks that the proposed method fails to track. In the sequence Animal3, there are similar targets around the alpaca to be tracked. Although our tracker uses context learning to repress interference, their similar appearance still confuses the proposed tracker. In the sequence uav1_3, the UAV moves so fast and irregularly that the SESF module cannot work well, resulting in tracking failure. In the sequence sand truck, the sand truck is under a low illumination condition. When the target enters a dark environment, the proposed tracker cannot localize it. Table 2 also confirms that the performance of our tracker is not state of the art under the attribute of illumination variation.

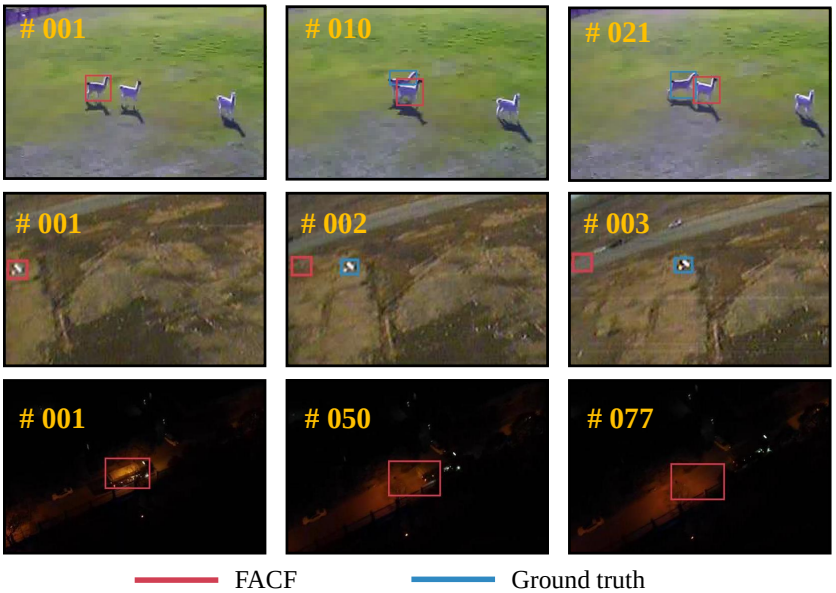

Figure 11. Some representative tracking failure cases of the proposed FACF tracker. From top to bottom: Animal3, uav1_3, and sand truck from DTB70 [25], UAV123_10fps [26], and UAVTrack112 [27], respectively.

6. Conclusions

In this work, in order to enhance filter discriminative power in future unknown environments, we proposed a novel future-aware correlation filter tracker, namely FACF. By virtue of an effective time series forecast method, we obtained the predicted patch with

the coarse target position, which is beneficial for localizing the target more precisely and for obtaining the predicted patch. Then, a context-aware mask is produced through an efficient target-aware method. Afterward, we obtain a single context patch at the pixel-level by the element-wise dot product between the context-aware mask and the feature maps of the predicted patch. Finally, feature maps of the context patch are utilized for improving discriminative ability. Extensive experiments on three UAV benchmarks verify the superiority of the FACF tracker against other state-of-the-art handcrafted-based and deep-based trackers.

The proposed future-aware strategy aims at dealing with unpredicted surrounding changes by learning the future context rather than the current context. The fast context acquisition avoids additional feature extraction as well as unrelated background noise. In general, our method guarantees the robustness, accuracy, and efficiency, which is promising for UAV real-time application. We think the proposed context learning method can be extended to other trackers for more robust UAV tracking. In future work, we will explore more accurate and efficient strategies to exploit future information in order to boost tracking performance without sacrificing speed drastically.

Author Contributions: Methodology, S.M.; software, F.Z.; validation, Z.Q., L.Y. and Y.Z.; formal analysis, L.Y.; investigation, S.M.; resources, Z.Q.; data curation, Y.Z.; writing—original draft preparation, F.Z.; writing—review and editing, F.Z.; visualization, Z.Q.; supervision, S.M.; project administration, Z.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

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

References

1. Miao, Y.; Li, J.; Bao, Y.; Liu, F.; Hu, C. Efficient Multipath Clutter Cancellation for UAV Monitoring Using DAB Satellite-Based PBR. *Remote Sens.* **2021**, *13*, 3429. [CrossRef]
2. Zhang, F.; Yang, T.; Liu, L.; Liang, B.; Bai, Y.; Li, J. Image-Only Real-Time Incremental UAV Image Mosaic for Multi-Strip Flight. *IEEE Trans. Multim.* **2021**, *23*, 1410–1425. [CrossRef]
3. Mcarthur, D.R.; Chowdhury, A.B.; Cappelleri, D.J. Autonomous Control of the Interacting-BoomCopter UAV for Remote Sensor Mounting. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 5219–5224.
4. Bolme, D.S.; Beveridge, J.R.; Draper, B.A.; Lui, Y.M. Visual object tracking using adaptive correlation filters. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010; pp. 2544–2550.
5. 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 (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 4904–4913.
6. Danelljan, M.; Häger, G.; Khan, F.; Felsberg, M. Accurate scale estimation for robust visual tracking. In Proceedings of the British Machine Vision Conference (BMVC), Nottingham, UK, 1–5 September 2014.
7. Danelljan, M.; Bhat, G.; Shahbaz Khan, F.; Felsberg, M. Eco: Efficient convolution operators for tracking. In Proceedings of the IEEE Conference on Computer Cision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6638–6646.
8. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H. Fully-convolutional siamese networks for object tracking. In Proceedings of the European Conference on Computer Cision (ECCVW), Amsterdam, The Netherlands, 8–16 October 2016; pp. 850–865.
9. Fu, C.; Lin, F.; Li, Y.; Chen, G. Correlation Filter-Based Visual Tracking for UAV with Online Multi-Feature Learning. *Remote Sens.* **2019**, *11*, 549. [CrossRef]
10. Fu, C.; Li, B.; Ding, F.; Lin, F.; Lu, G. Correlation Filter for UAV-Based Aerial Tracking: A Review and Experimental Evaluation. *arXiv* **2020**, arXiv:2010.06255.
11. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *37*, 583–596. [CrossRef]

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 (ICCV), Santiago, Chile, 7–13 December 2015; pp. 4310–4318.
13. 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 (ICCV), Venice, Italy, 22–29 October 2017; pp. 1135–1143.
14. Lukezic, A.; Vojir, T.; Čehovin 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 (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6309–6318.
15. Dai, K.; Wang, D.; Lu, H.; Sun, C.; Li, J. Visual Tracking via Adaptive Spatially-Regularized Correlation Filters. In Proceedings of the IEEE Conference on Computer Cision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 4665–4674.
16. Mueller, M.; Smith, N.; Ghanem, B. Context-aware correlation filter tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1396–1404.
17. Fu, C.; Jiang, W.; Lin, F.; Yue, Y. Surrounding-Aware Correlation Filter for UAV Tracking with Selective Spatial Regularization. *Signal Process.* **2020**, *167*, 1–17. [CrossRef]
18. Li, Y.; Fu, C.; Huang, Z.; Zhang, Y.; Pan, J. Intermittent Contextual Learning for Keyfilter-Aware UAV Object Tracking Using Deep Convolutional Feature. *IEEE Trans. Multimed.* **2021**, *23*, 810–822. doi: 10.1109/TMM.2020.2990064. [CrossRef]
19. Xu, T.; Feng, Z.H.; Wu, X.J.; Kittler, J. Joint group feature selection and discriminative filter learning for robust visual object tracking. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 7950–7960.
20. Zhang, Y.; Gao, X.; Chen, Z.; Zhong, H.; Xie, H.; Yan, C. Mining Spatial-Temporal Similarity for Visual Tracking. *IEEE Trans. Image Process.* **2020**, *29*, 8107–8119. [CrossRef] [PubMed]
21. Huang, Z.; Fu, C.; Li, Y.; Lin, F.; Lu, P. Learning Aberrance Repressed Correlation Filters for Real-Time UAV Tracking. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 2891–2900.
22. Fu, C.; Ye, J.; Xu, J.; He, Y.; Lin, F. Disruptor-Aware Interval-Based Response Inconsistency for Correlation Filters in Real-Time Aerial Tracking. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 6301–6313. [CrossRef]
23. Li, Y.; Fu, C.; Ding, F.; Huang, Z.; Lu, G. AutoTrack: Towards High-Performance Visual Tracking for UAV with Automatic Spatio-Temporal Regularization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11923–11932.
24. Nazim, A.; Afthanorhan, A. A comparison between single exponential smoothing (SES), double exponential smoothing (DES), holt's (brown) and adaptive response rate exponential smoothing (ARRES) techniques in forecasting Malaysia population. *Glob. J. Math. Anal.* **2014**, *2*, 276–280.
25. Li, S.; Yeung, D.Y. Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 4140–4146.
26. Mueller, M.; Smith, N.; Ghanem, B. A benchmark and simulator for uav tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; pp. 445–461.
27. Fu, C.; Cao, Z.; Li, Y.; Ye, J.; Feng, C. Onboard Real-Time Aerial Tracking with Efficient Siamese Anchor Proposal Network. *IEEE Trans. Geosci. Remote Sens.* **2021**, 1–13. [CrossRef]
28. Li, Y.; Zhu, J. A scale adaptive kernel correlation filter tracker with feature integration. In Proceedings of the European Conference on Computer Vision Workshops (ECCV), Zurich, Switzerland, 6–12 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 254–265.
29. Wang, C.; Zhang, L.; Xie, L.; Yuan, J. Kernel cross-correlator. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2017; pp. 4179–4186.
30. Danelljan, M.; Hager, G.; Shahbaz Khan, F.; Felsberg, M. Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1430–1438.
31. Danelljan, M.; Khan, F.S.; Felsberg, M.; Van De Weijer, J. Adaptive Color Attributes for Real-Time Visual Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 1090–1097.
32. Qi, Y.; Zhang, S.; Qin, L.; Yao, H.; Huang, Q.; Lim, J.; Yang, M.H. Hedged Deep Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 4303–4311.
33. Danelljan, M.; Robinson, A.; Khan, F.S.; Felsberg, M. Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 472–488.
34. Xu, T.; Feng, Z.; Wu, X.; Kittler, J. Learning Adaptive Discriminative Correlation Filters via Temporal Consistency Preserving Spatial Feature Selection for Robust Visual Object Tracking. *IEEE Trans. Image Process.* **2019**, *28*, 5596–5609. [CrossRef]
35. Zhu, X.F.; Wu, X.J.; Xu, T.; Feng, Z.; Kittler, J. Robust Visual Object Tracking via Adaptive Attribute-Aware Discriminative Correlation Filters. *IEEE Trans. Multim.* **2021**, 1. doi: 10.1109/TMM.2021.3050073. [CrossRef]

36. Yan, Y.; Guo, X.; Tang, J.; Li, C.; Wang, X. Learning spatio-temporal correlation filter for visual tracking. *Neurocomputing* **2021**, *436*, 273–282. [CrossRef]
37. Wang, N.; Zhou, W.; Tian, Q.; Hong, R.; Wang, M.; Li, H. Multi-cue correlation filters for robust visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 4844–4853.
38. Fu, C.; Xu, J.; Lin, F.; Guo, F.; Liu, T.; Zhang, Z. Object Saliency-Aware Dual Regularized Correlation Filter for Real-Time Aerial Tracking. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 8940–8951. [CrossRef]
39. Fu, C.; Ding, F.; Li, Y.; Jin, J.; Feng, C. Learning dynamic regression with automatic distractor repression for real-time UAV tracking. *Eng. Appl. Artif. Intell.* **2021**, *98*, 104116. [CrossRef]
40. Zheng, G.; Fu, C.; Ye, J.; Lin, F.; Ding, F. Mutation Sensitive Correlation Filter for Real-Time UAV Tracking with Adaptive Hybrid Label. In Proceedings of the International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 1–8.
41. Lin, F.; Fu, C.; He, Y.; Guo, F.; Tang, Q. Learning Temporary Block-Based Bidirectional Incongruity-Aware Correlation Filters for Efficient UAV Object Tracking. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 2160–2174. [CrossRef]
42. Xue, X.; Li, Y.; Shen, Q. Unmanned aerial vehicle object tracking by correlation filter with adaptive appearance model. *Sensors* **2018**, *18*, 2751. [CrossRef]
43. Zha, Y.; Wu, M.; Qiu, Z.; Sun, J.; Zhang, P.; Huang, W. Online Semantic Subspace Learning with Siamese Network for UAV Tracking. *Remote Sens.* **2020**, *12*, 325. [CrossRef]
44. Zhuo, L.; Liu, B.; Zhang, H.; Zhang, S.; Li, J. MultiRPN-DIDNet: Multiple RPNs and Distance-IoU Discriminative Network for Real-Time UAV Target Tracking. *Remote Sens.* **2021**, *13*, 2772. [CrossRef]
45. Ren, S.; He, K.; Girshick, R.B.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Proceedings of the 29th International Conference on Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
46. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R.B. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
47. Li, B.; Fu, C.; Ding, F.; Ye, J.; Lin, F. All-Day Object Tracking for Unmanned Aerial Vehicle. *arXiv* **2021**, arXiv:2101.08446.
48. Boyd, S.; Parikh, N.; Chu, E. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* **2011**, *3*, 1–122. [CrossRef]
49. Sherman, J.; Morrison, W.J. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *Ann. Math. Stat.* **1950**, *21*, 124–127. [CrossRef]
50. 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 (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1401–1409.
51. Wang, N.; Zhou, W.; Song, Y.; Ma, C.; Liu, W.; Li, H. Unsupervised Deep Representation Learning for Real-Time Tracking. *Int. J. Comput. Vis.* **2021**, *129*, 400–418. [CrossRef]
52. Wang, N.; Zhou, W.; Song, Y.; Ma, C.; Li, H. Real-Time Correlation Tracking Via Joint Model Compression and Transfer. *IEEE Trans. Image Process.* **2020**, *29*, 6123–6135. [CrossRef] [PubMed]
53. Li, X.; Ma, C.; Wu, B.; He, Z.; Yang, M.H. Target-aware deep tracking. In Proceedings of the IEEE Conference on Computer Cision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 1369–1378.
54. Zhang, L.; Suganthan, P. Robust Visual Tracking via Co-trained Kernelized Correlation Filters. *Pattern Recognit.* **2017**, *69*, 82–93. doi: 10.1016/j.patcog.2017.04.004. [CrossRef]
55. Wang, N.; Song, Y.; Ma, C.; Zhou, W.; Liu, W.; Li, H. Unsupervised deep tracking. In Proceedings of the IEEE Conference on Computer Cision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 1308–1317.
56. Valmadre, J.; Bertinetto, L.; Henriques, J.F.; Vedaldi, A.; Torr, P.H.S. End-to-End Representation Learning for Correlation Filter Based Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5000–5008.