remote sensing

MDPI

Article

Auto-Learning Correlation-Filter-Based Target State Estimation for Real-Time UAV Tracking

Ziyang Bian [1,2], Tingfa Xu [1,3], Junjie Chen [1], Liang Ma [2], Wenjing Cai [2] and Jianan Li [1,*]

[1] Key Laboratory of Photoelectronic Imaging Technology and System, Beijing Institute of Technology, Beijing 100081, China
[2] North China Research Institute of Electro-Optics, Beijing 100015, China
[3] Beijing Institute of Technology Chongqing Innovation Center, Chongqing 401120, China
* Correspondence: lijianan@bit.edu.cn

Abstract: Most existing tracking methods based on discriminative correlation filters (DCFs) update the tracker every frame with a fixed learning rate. However, constantly adjusting the tracker can hardly handle the fickle target appearance in UAV tracking (e.g., undergoing partial occlusion, illumination variation, or deformation). To mitigate this, we propose a novel auto-learning correlation filter for UAV tracking, which fully exploits valuable information behind response maps for adaptive feedback updating. Concretely, we first introduce a principled target state estimation (TSE) criterion to reveal the confidence level of the tracking results. We suggest an auto-learning strategy with the TSE metric to update the tracker with adaptive learning rates. Based on the target state estimation, we further developed an innovative lost-and-found strategy to recognize and handle temporal target missing. Finally, we incorporated the TSE regularization term into the DCF objective function, which by alternating optimization iterations can efficiently solve without much computational cost. Extensive experiments on four widely-used UAV benchmarks have demonstrated the superiority of the proposed method compared to both DCF and deep-based trackers. Notably, ALCF achieved state-of-the-art performance on several benchmarks while running over 50 FPS on a single CPU. Code will be released soon.

Keywords: single object tracking; UAV tracking; correlation filters; target state estimation; auto-learning

Citation: Bian, Z.; Xu, T.; Chen, J.; Ma, L.; Cai, W.; Li, J. Auto-Learning Correlation-Filter-Based Target State Estimation for Real-Time UAV Tracking. *Remote Sens.* **2022**, *14*, 5299. https://doi.org/10.3390/rs14215299

Academic Editors: Miltiadis D. Lytras and Andreea Claudia Serban

Received: 24 August 2022
Accepted: 17 October 2022
Published: 23 October 2022

1. Introduction

Given a target object specified by a bounding box in the first frame, visual object tracking aims to determine the target's exact location sequentially in a video, which serves as a fundamental task in the computer vision community. Recently, increasing attention has been paid to unmanned aerial vehicle (UAV) tracking [1–5] due to its promising development in the field of geomatics [6], agroforestry [7], transportation [8] and mid-air aircraft tracking [9].

Existing object tracking methods can mainly be divided into two types: discriminative correlation filter (DCF)-based [10–14] and deep-based [15–18]. Since UAVs only provide limited power capacity and computational resources, deep-based trackers usually suffer low inference speeds and high computational overheads. In contrast, DCF-based trackers achieve an ideal trade-off between accuracy and speed by efficiently computing the Fourier frequency domain, providing good solutions to real-time UAV tracking.

DCF-based trackers [3,11–13] detect the target by generating a response map and assigning the peak position as the target position. Then, the model will be updated with a fixed learning rate after every frame to adapt to the target's appearance change. However, due to the common challenges of fast motion and out-of-view in UAV tracking, target appearance could change drastically, rending the ineffectiveness of a stable update strategy. An appropriate update strategy is a goal to pursue. Recently, some trackers [2,19–21] have tried to mine information from response maps, such as fluctuation degree and peak

height, as updated guidance. Intuitively, the peak height can provide a rough indication of the similarity between the target and the tracker, and the fluctuation degree indicates background clutter to some extent. LMCF [19] proposed an criterion called average peak-to-correlation energy (APCE), beyond which a model update is required. We regard it as a good measure of the fluctuation degree, but this single criterion seems not robust enough to cope with the appearance change, as shown in Figure 1. It is not sensitive to slight target state changes (e.g., in frames 1300, 1400), and there is an unnecessarily sharp drop when the model state is credible (in frame 1268). In addition, although LMCF stops the model from updating in some frames, which can prevent model degradation to a certain extent, it still relies on the pre-set single learning rate to cope with the model updating.

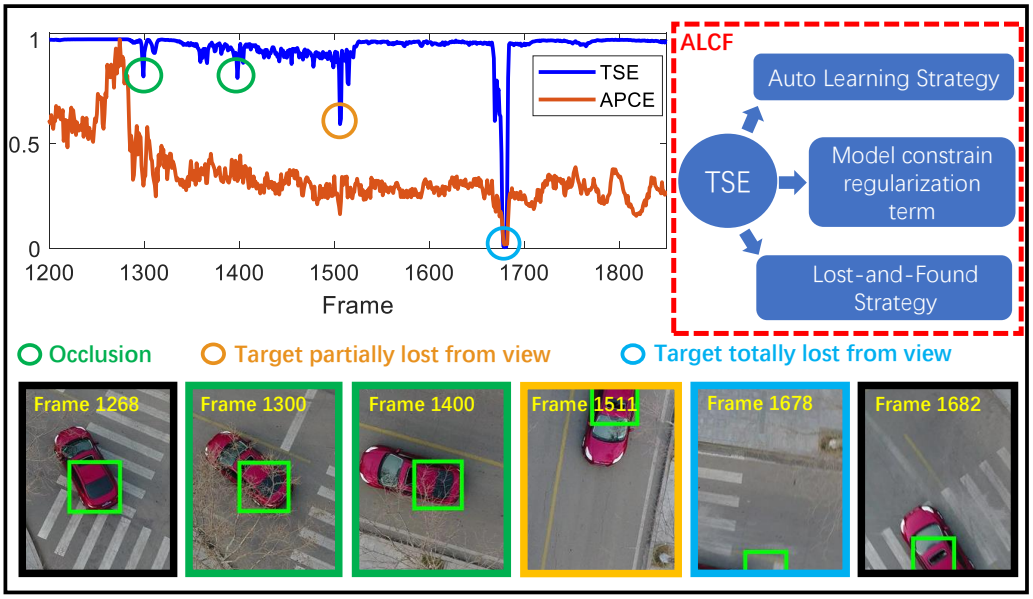

Figure 1. Comparison of the proposed TSE with APCE. The green box shows the tracking result.

1.1. Related Works

1.1.1. Discriminative Correlation Filter

Discriminative correlation filter (DCF)-based methods have been widely applied to visual tracking due to their high computational efficiency. Bolme et al. [22] proposed a minimum output sum of squared error (MOSSE) filter and were the first to apply a correlation filter to object tracking. Henriques et al. [23] introduced a circulant matrix to produce sufficient samples for training and detection while maintaining a fast tracking speed. However, the resulting periodic repetitions at boundary positions limit the discrimination of the tracker. To mitigate this issue, various regularization terms have been introduced [2,12,24]. Danelljan et al. [12] adopted a spatial regularization term that penalizes the filter coefficients for background regions to force the tracker to focus more on the target center. Huang et al. [3] used response maps as regularization terms for model training and repressed the aberrance of response maps to improve tracking performance. Chen et al. [4] designed a Gaussian-like response map as expected output to enhance the representation learning ability of the correlation filter. Xu et al. [25] proposed to gauge the relevance of multi-channel features for the purpose of channel selection to improve the performance of DCF in accuracy and provide a parsimonious model from the attribute perspective. Wang et al. [26] proposed a context and saliency-aware CF to strengthen the ability to extract targets of interest from complex background. Fu et al. [27] introduced an object-saliency-aware, dual-regularized correlation filter to suppress the boundary effect and consequently enhance the performance of the tracker. Li et al. [28] introduced a temporal regularization term, which uses the model trained in the previous frame as the reference model to constrain the model training in the current frame, thereby preventing

model degradation. However, it uses a fixed constant as the regularization coefficient and adopts the model in the previous frame as the reference model regardless of whether the target state in the previous frame is credible.

1.1.2. Prior Knowledge to Model Update

Recently, increasing attention has been paid to the model updating. Wang et al. [19] introduced a criterion called APCE to measure response map quality and guide model updates. Li et al. [2] used the variations of response maps between consecutive frames to estimate the credibility of the target state and restricted model update when needed. Huang et al. [29] updated the Siamese tracker by transferring the previous target knowledge with attention. Dai et al. [30] used an offline-trained meta-updater to determine whether or not to update the tracker with two fixed learning rates (0 or 1), which fails to adapt to fickle target appearances varying at different rates in UAV tracking scenarios. Though adopting adaptive learning rates, Yang et al. [31] highly relies on LSTM [32], causing a high computational cost. Chen et al. [33] constructed a Gaussian-like (GL) function label for correlation filter training and proposed an accurate incremental update to mitigate model degradation by combining target samples with adaptive aspect ratios. In this paper, we fully exploit the information of the response map and directly obtain the learning rate from it, which is computationally efficient and suitable for real-time tracking. To the best of our knowledge, no previous works have exploited the quality of response maps to evaluate the target state and adjust the model's learning rate adaptively.

1.1.3. Redetect the Lost Target

Due to the complex situations during tracking, targets may be missed in some frames. To address this problem, previous works [34–36] often designed an additional detector to search for the target on the entire screen after target loss. Some trackers [37,38] even introduce global detection into tracking and locate the target through spatio-temporal constraints. However, these methods are challenging to run in real-time on GPU, which limits their applications in UAV tracking. Huang et al. [9] exploited motion features of a moving target to redetect the target from clutter background after loss, but this rough approach cannot handle static targets or visible videos. Zhu et al. [21] expanded the search area to increase the possibility of recapturing the target. Recently, Alan Lukezic et al. [36] verified the possibility of correlation filters as a detector using multiple correlation filters updated at different time scales. However, the large number of filters used in [36] slows the tracking speed, which limits its application in the field of UAV. Therefore, exploring an efficient re-detection method for UAV tracking is very meaningful. Unlike previous works, we introduce an iterative search method with only one correlation filter, which tracks the target and searches for the target after it is lost, achieving tracking performance improvement at the expense of only a slight decrease in tracking speed.

1.2. Contributions

We present a novel auto-learning correlation filter (short for ALCF) tracking algorithm for efficient UAV tracking. We first introduce an innovative criterion, target state estimation (TSE), to comprehensively get aware of the target state. Considering the peak height and fluctuation degree, both informative features of response maps, simultaneously, the resulting TSE value can represent the uncertainty in the target state of each frame reliably, thereby having a keener sense of slight target state changes and being more resistant to noise. As shown in Figure 1, TSE remains steady when the target moves smoothly and fluctuates significantly when occlusion or target lost occurs (as denoted by the circles). Our TSE successfully indicates whether the target is currently in the presence of occlusions (green circle), partially lost (yellow circle), or out-of-view (blue circle). It is noted that the comparison of APCE and TSE in Figure 1 was performed on the same tracker (ALCF).

Based on the TSE, we propose a novel auto-learning strategy that automatically derives a rapid and optimal learning rate from TSE adaptively depending on the estimated target

states. Specifically, when the TSE remains large, indicating a transparent and credible target state, no update for the tracker is required, since it nicely fits the target. Moreover, when the TSE decreases slightly, indicating the target undergoes deformation, a significant learning rate is produced to adapt to rapid appearance changes. Furthermore, when the TSE drops sharply and the target state is uncertain, the target may suffer severe occlusion or be lost from view such that a small learning rate is generated to retain prior knowledge of the target and avoid model degradation. This way, ALCF can adjust its tracker properly under complex and changeable situations.

Moreover, the target in UAV tracking usually suffers large positional displacement between successive frames or even temporal disappearance from view, which quickly results in tracking failures. The challenges lie in recapturing the target when it appears in subsequent frames, as large positional displacement may exist. Thanks to the TSE criterion, which would decrease sharply when the target is lost (as exampled by the blue circle in Figure 1), we present a new lost-and-found strategy that perceives such a TSE decrease and enlarging search areas centered at the location, where the target last appears immediately. ALCF can quickly recapture the lost target with a new alternate search strategy with negligible extra computational cost.

The proposed TSE can also contribute to model optimization. We introduce a novel regularization term by using the trusted model trained with previous frames as a reference model to prevent model degradation. Instead of manually setting the coefficient for the regularization term, ALCF can automatically adjust the coefficient according to TSE. When the target state is credible, the coefficients are small to relax the model variation, and vice versa—the coefficients are significant in constraining the variation of the model in the current frame. We also incorporate two additional regularization terms to help suppress the aberration of computed responses for consecutive frames [3] and focus more on the central portion of targets [12].

We extensively evaluated our ALCF on several commonly used benchmark datasets, including UAVDT [39], UAV123@10fps [40], DTB70 [41], and VisDrone2018-test-dev [42], which contain large-scale, challenging aerial video sequences captured by UAVs. We provide two version trackers of ALCF, one based on a simple scale regression strategy (proposed by DSST [43]), which far exceeds real-time on a single CPU and performs favorably against other state-of-the-art trackers with hand-drafted features. Another version of ALCF exploits deep features for scale regression (proposed by Alpha-Refine [44]), which shows competitive performance against recent deep trackers.

To sum up, this work makes the following contributions:

- We present a novel TSE metric to the community which allows accurate estimation of the target state.
- We introduce a new ALCF tracker upon TSE, which consists of a novel auto-learning strategy, a fast lost-and-found strategy, and an effective regularization term for efficient and robust UAV tracking.
- We propose a deep version model which outperforms numerous recently popular deep framework-based trackers and established the new state-of-the-art on several UAV datasets.
- We provide the community with an optimal hand-drafted feature-correlation filter at over 50 FPS on a single CPU.

2. Method

In this section, we first describe the tracking mechanism of the correlation filter and the overall framework of the ALCF tracker, followed by a detailed description of the construction process of target state estimation (TSE), and finally describe the three strategies based on TSE.

2.1. Overview

The overall framework of our method is shown in Figure 2. The DCF-based tracker consists of three stages: the detection stage for target localization, the online model update stage for learning target appearance information, and the training stage for computing correlation filters. Based on the TSE value computed from the response map, we propose a lost-and-found strategy in the detection stage to cope with target loss. Then, an auto-learning strategy to learn new information from the target sample is presented, which learns adaptively according to the target state. Finally, a model that constrains regularization is introduced into model training to obtain a robust correlation filter model for tracking.

In the detection stage, the correlation filter f_{t-1} from the previous frame is used to search for the target in the current frame. The correlation response map is used to locate the target position, and the peak position is considered the target position of the current frame. Besides the peak position of the response map, we also exploit other information (i.e., peak value, fluctuation degree) to calculate TSE to evaluate the target state. When there is a sudden drop in the TSE value, we judge that the target has been lost and start using the lost-and-found strategy to re-detect the target, skipping the online update stage and the training stage if the target is lost.

For the online update stage, we propose an auto-learning strategy to learn new information from the current image. Unlike traditional DCF-based trackers that use a pre-set constant as the learning rate, ALCF uses TSE to compute an adaptively changing learning rate η, which allows the appearance model to learn more from the current image when the target state is credible and less when the target state is in doubt, thus maintaining robust tracking.

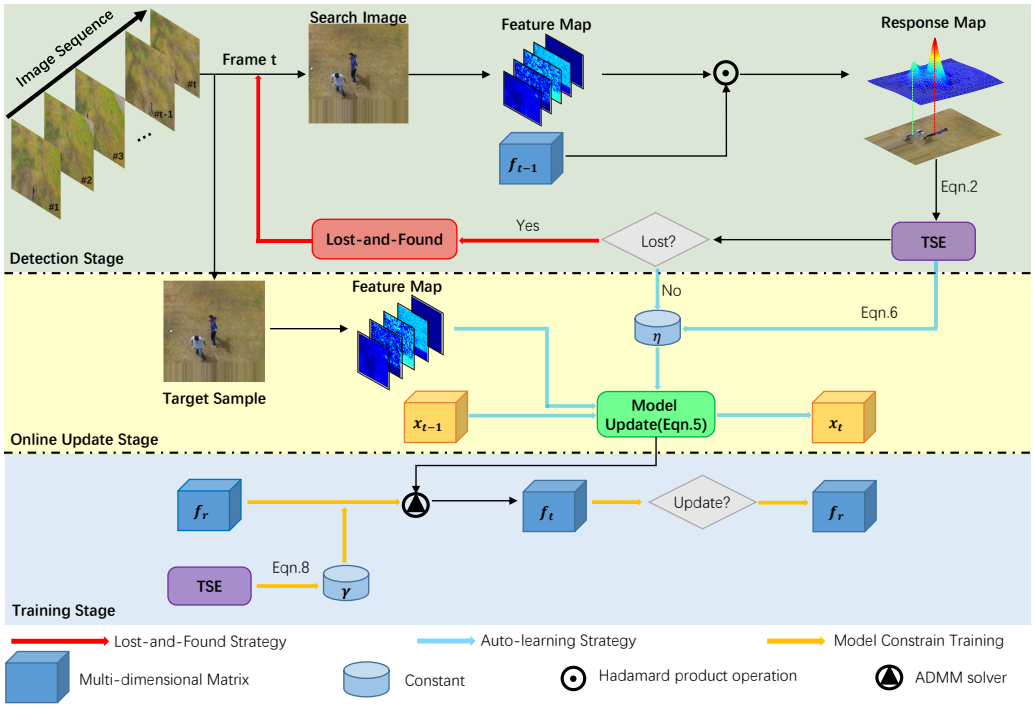

Figure 2. The overall framework of ALCF comprises three stages, i.e., detection, online updating, and training.

For the training phase, we present the model constrain regularization strategy to train the model. Specifically, we use the reference model f_r preserved from historical frames to prevent model degradation and use TSE to guide the update of the reference model and its weight coefficient γ during model training. When the target state is credible, we reduce the weight coefficient of the reference model and relax the restriction on model learning from the current frame, in the meanwhile updating the reference model to use the newly

computed correlation filter f_t as the reference model for model training in subsequent frames. Conversely, we increase the weight coefficient of the reference model and restrict the model changes to prevent model degradation. Then, the model can be solved efficiently via the alternating direction method of multipliers (ADMM) algorithm.

2.2. Target State Estimation (TSE)

We developed TSE by comprehensively evaluating the peak value F_{max} and fluctuation degree (short for FD) of response maps, which can indicate the uncertainty in the target state and thus reflect the target state.

Denote F as the response map of dimensions $W \times H$ (width×height in pixels). We first compute mean-square error (MSE) to measure the difference between F and the ideal Gaussian response:

$$MSE = \frac{1}{WH} \cdot \sum_{i=1}^{W} \sum_{j=1}^{H} (\boldsymbol{F}(i,j) - \boldsymbol{y}(i,j))^2, \tag{1}$$

where $F(i,j)$ represents the response value on pixel (i,j), y is the ideal response map (Gaussian distribution), and F is expected to be similar to y when the target state is credible, leading to a low MSE value.

Given MSE, the TSE is computed as:

$$TSE = [1 + e^{-(T-\mu)}]^{-1}, \tag{2}$$

$$T = \alpha_1 F_{max} + \alpha_2 FD, \tag{3}$$

$$FD = \frac{{F_{max}}^2}{MSE}, \tag{4}$$

where F_{max} is the peak value of F, which measures the similarity score for the tracker and the target; FD reflects the fluctuation degree of F and the degree of background disturbance. When there are no distractors (i.e., frames 7, 21, and 307 in Figure 3), the value of FD is about 20 times that of F_{max}. Thus, we use two constant coefficients, α_1 and α_2, to ensure F_{max} and FD are of the same magnitude. μ is an offset constant. When the target state is credible, F_{max} is large while MSE is small, resulting in a considerable TSE value (close to 1). In contrast, TSE would drop to 0 when the target is lost. When the target suffers deformation (in frame 21, the target suffers out-of-plane rotation), the peak value drops slightly. When there exist distractors (in frame 228), sub-peaks lead to a significant drop to FD, but F_{max} is still high, keeping the TSE value enormous. Thus, with the combined effect of peak and fluctuation degree, our TSE is more robust and better guides the target state changes. More details about the construction of TSE can be found in Section 3.3.5.

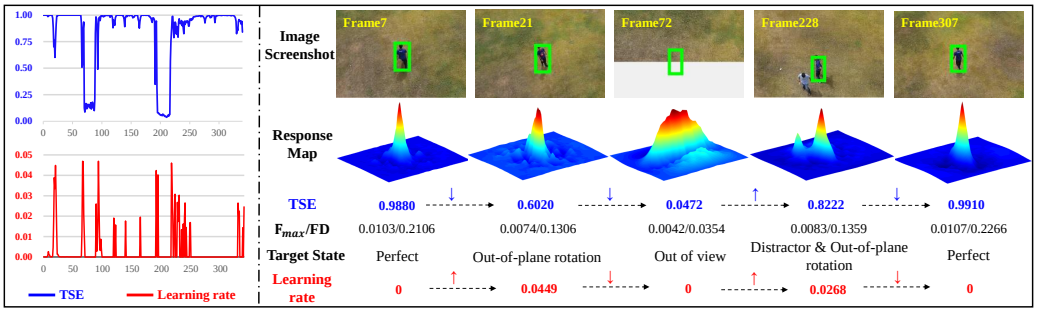

Figure 3. Illustration of the auto-learning strategy. The green box shows the tracking result.

2.3. Auto-Learning Correlation Filter

Based on TSE, we developed a new ALCF tracker comprised of an auto-learning strategy, a lost-and-found strategy, and an effective regularization term.

2.3.1. Auto-Learning Strategy

Existing methods update the tracking model with a single learning rate as follows:

$$\hat{\boldsymbol{x}}_k^{model} = (1-\eta)\hat{\boldsymbol{x}}_{k-1}^{model} + \eta\hat{\boldsymbol{x}}_k, \tag{5}$$

where $\hat{}$ denotes the discrete fourier transform (DFT) of a signal, $\hat{x}_k$ is the vectorized image of frame k, and $\hat{x}_{k-1}^{model}$ is the tracking model learned from frame $k-1$. η is a fixed learning rate that is hard to set manually for UAV tracking, as constant and uncertain appearance changes exist. For example, when the correlation filter model can describe the target well, the model does not need to learn new information from the target sample image. When the target suffers severe occlusion or is lost from the screen, the model should stop updating to prevent contamination by untrustworthy samples. On the other hand, when there exists a difference between the model and the current target appearance, the model should accelerate learning new information from the target appearance. When there exists a huge difference between the target appearance and the model, we argue that maybe the target is lost, and the learning rate should reduce to keep the prior knowledge of the target.

Given this, we developed an auto-learning strategy that updates the tracking model with varying learning rates depending on the target state indicated by TSE, as shown in Figure 3. We automatically adjust the learning rate depending on different target states revealed by TSE computed from the distribution of response maps. When the target moves smoothly (frame 12), TSE is close to one, and the learning rate drops to zero, since updating the tracker is not necessary. However, when the target suffers occlusion or deformation (frames 32, 48, and 156), TSE decreases while the learning rate increases to adapt to rapid changes in target appearance. Moreover, when the target is lost from view (frame 20), TSE drops sharply, and the learning rate becomes zero to stop the model updating and retain the prior knowledge of the target. Concretely, we use TSE to compute the learning rate of frame k from a Gaussian distribution:

$$\eta_k = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(TSE-\beta)^2}{2\sigma^2}} - \varphi, \tag{6}$$

where β is set as 0.5, since TSE is bounded between 0 and 1, the standard deviations σ is set as 1, and φ is a constant to make η_k fall in a reasonable value range. When the value of TSE is close to 1, the target state is trusted, and the learning rate is low; and as the value of TSE drops, the learning rate takes the lead and then decreases. When TSE decreases to 0, we think the target may have been lost and stopped updating.

2.3.2. Lost-and-Found Strategy

We designed a novel lost-and-found strategy based on TSE. When the target is lost from view, TSE will drop suddenly. For example, it decreases by over seventy percent within ten frames, as illustrated in Figure 4. ALCF keeps a close eye on the trend of TSE and initiates the search for the lost target around the location where it last appeared (green rectangle) upon a sharp drop in the TSE. TSE drops rapidly when the target suffers complete occlusion, as shown by the TSE curve. ALCF initiates four additional search windows (yellow dashed box) and iterates their space intervals over three increased distances frame by frame to enlarge search areas. Until any one of the search windows regains a large TSE (more extensive than the sum value of the rest windows, as shown in frame 49), the tracker successfully recaptures the lost target and terminates the re-found process.

Since large positional displacement may exist when the lost target reappears in subsequent frames, previous works have introduced additional detectors to search for the target in the entire frame, leading to heavy memory and computation cost. Instead, we increase the number of search windows to cover a large search area and thus avoid using a detector. However, simply increasing search windows, though increasing the chances of recapturing the target, would hurt tracking speed at the same time. Instead of augmenting massive search windows in every single frame, we explored an efficient alternating search scheme

that gradually enlarges the distance of four additional search windows in consecutive frames to cover large search areas sequentially, as shown by the yellow dashed rectangle in Figure 4. We set three distance intervals based on the width w and height h of the target and iterate over them every three consecutive frames until the target is re-found.

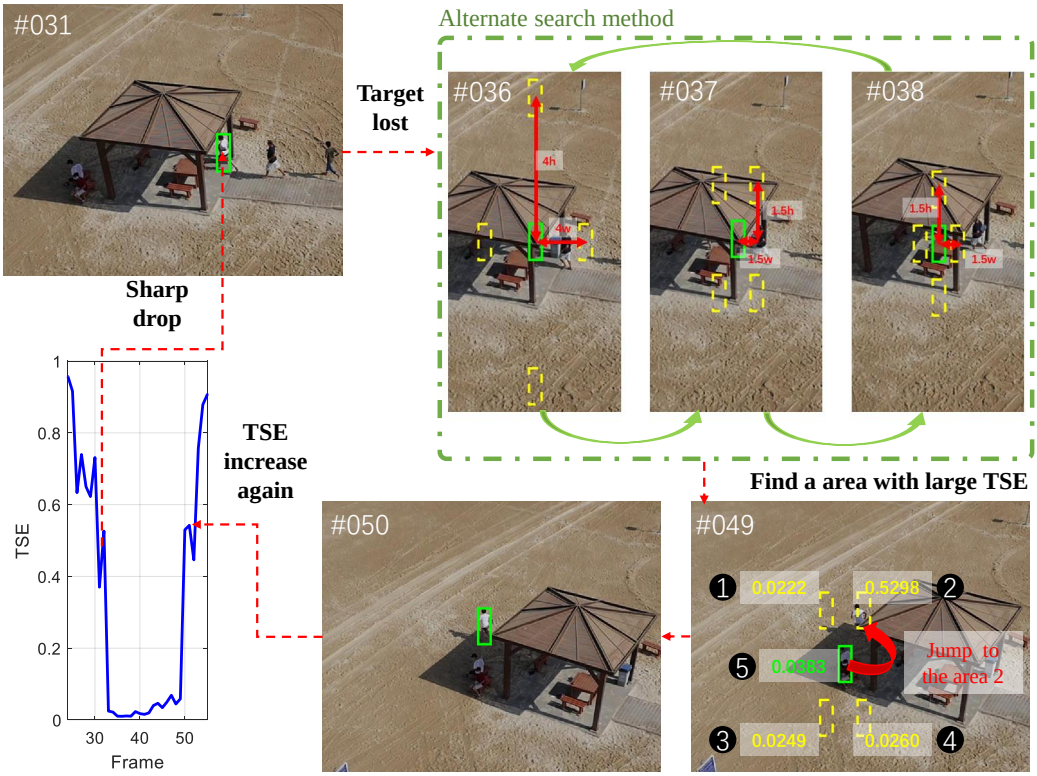

Figure 4. Diagram of lost-and-found strategy.

Our lost-and-found strategy exploits the correlation filter as the detector, which has two advantages: (i) The re-detection process is efficiently computed in the Fourier domain. The alternate search mechanism does not introduce a significant additional computational burden, which is friendly to the limited computational resources of UAVs. (ii) The correlation filter can be used not only as a tracking model but also as a detector when the target is lost, which brings no additional memory pressure and has low memory requirements for the UAVs.

2.3.3. Model Constrain Regularization

We further introduced a new regularization term by employing the prior knowledge learned from previous frames to constrain the filter learning of the current frame:

$$R_1 = \frac{\gamma_k}{2} \| \boldsymbol{f}_k - \boldsymbol{f}_r \|_2^2, \tag{7}$$

where $f_k \in \mathbb{R}^T$ denotes the filter of frame k, and $f_r \in \mathbb{R}^T$ represents the reference filter learned from previous frames, which encodes informative prior knowledge about the target. Instead of updating f_r after each frame [28], we only use the filter learned from the frame with favorable target states (e.g., no severe occlusion and deformation) as a reliable reference filter. To this end, we use TSE to estimate the target state of every frame and only use the filter from the frame whose computed TSE is larger than a pre-set threshold τ to refresh f_r.

In addition, we use an adaptive coefficient γ_k to adjust the weight of the regularization term depending on the current target state:

$$\gamma_k = \frac{1}{2}(1 - TSE^2), \tag{8}$$

when the current target state is trustworthy, i.e., TSE is large, γ_k is small such that f_k learns less from f_r to adapt to the latest appearance change of the target. Otherwise, f_k borrows more knowledge from f_r to prevent model degradation.

2.4. Model Optimization

The objective function can be formulated as:

$$E(f_k) = \frac{1}{2}\left\| \sum_{d=1}^{D} (\boldsymbol{x}_k^d * \boldsymbol{f}_k^d) - \boldsymbol{y} \right\|_2^2 + \sum_{i=[1,2,3]} R_i, \tag{9}$$

where $x_k^d \in \mathbb{R}^T$ denotes the *d-th* channel of the vectorized image of frame k, and D is the total channel number. $y \in R^T$ is the expected response (Gaussian distribution). $*$ indicates the convolution operator. $R_{i\in\{1,2,3\}}$ are three additional regularization terms to improve the discrimination of the model, which are detailed below.

R_1 is the regularization term defined in Equation (7). We further employ the spatial regularization term [12] to eliminate boundary effects caused by circulant shifted samples:

$$R_2 = \frac{1}{2}\sum_{d=1}^{D} \left\| \boldsymbol{w} \odot \boldsymbol{f}_k^d \right\|_2^2, \tag{10}$$

where w is a negative Gaussian-shaped spatial weight vector to make the tracking model emphasize more on the target center. $\odot$ indicates the Hadamard product operation.

In addition, a regularization term for response maps [3] is also used to exploit informative features of response distribution:

$$R_3 = \frac{\theta}{2}\left\| \sum_{d=1}^{D} (\boldsymbol{x}_k^d * \boldsymbol{f}_k^d) - \sum_{d=1}^{D} (\boldsymbol{x}_{k-1}^d * \boldsymbol{f}_{k-1}^d)[\Psi_{p,q}] \right\|_2^2, \tag{11}$$

where θ is a regularization coefficient; $x_{k-1}, \boldsymbol{f}_{k-1} \in \mathbb{R}^T$ denote the vectorized image and the filter of frame $k-1$, respectively; $\sum_{d=1}^{D} (\boldsymbol{x}_{k-1}^d * \boldsymbol{f}_{k-1}^d)$ is the computed response map. p and q denote the location differences of two peaks in both response maps in two-dimensional space, and $[\Psi_{p,q}]$ is used to align the peak position of two consecutive response maps.

2.5. Optimization Solutions

For the ease of optimization, we introduce an auxiliary variable $\hat{g}_k = \sqrt{N}\boldsymbol{F}\boldsymbol{f}_k$, where F is the orthonormal $N \times N$ matrix of complex basis vectors for mapping any N dimensional vectorized signal to Fourier frequency domain. Thus, we transformed the objective function into the frequency domain:

$$\begin{aligned} E(\boldsymbol{f}_k, \hat{\boldsymbol{g}}_k^d) = {} & \frac{1}{2}\left\| \sum_{d=1}^{D} (\hat{\boldsymbol{x}}_k^d * \hat{\boldsymbol{g}}_k^d) - \hat{\boldsymbol{y}} \right\|_2^2 + \frac{1}{2}\sum_{d=1}^{D} \left\| \boldsymbol{w} \odot \boldsymbol{f}_k^d \right\|_2^2 \\ & + \frac{\theta}{2}\left\| \sum_{d=1}^{D} (\hat{\boldsymbol{x}}_k^d * \hat{\boldsymbol{g}}_k^d) - \boldsymbol{R}_{pre} \right\|_2^2 + \frac{\gamma_k}{2}\sum_{d=1}^{D} \left\| \hat{\boldsymbol{g}}_k^d - \hat{\boldsymbol{g}}_r^d \right\|_2^2, \end{aligned} \tag{12}$$

where R_{pre} represents the discrete Fourier transformation of the previous frame's response map, which is a constant for the current frame. We used the alternating direction method

of multipliers (ADMM) algorithm to minimize Equation (12) to achieve a locally optimal solution. The augmented Lagrangian form of the equation can be formulated as:

$$\begin{aligned} L_k(\boldsymbol{f}_k, \hat{\boldsymbol{g}}_k^d, \hat{\boldsymbol{\xi}}^T) = E(\boldsymbol{f}_k, \hat{\boldsymbol{g}}_k^d) + \hat{\boldsymbol{\xi}}^T(\hat{\boldsymbol{g}}_k - \sqrt{N}\boldsymbol{F}\boldsymbol{f}_k) \\ + \frac{\rho}{2}\left\|\hat{\boldsymbol{g}}_k - \sqrt{N}\boldsymbol{F}\boldsymbol{f}_k\right\|_2^2, \end{aligned} \tag{13}$$

where $\hat{\xi}^T = [\hat{\boldsymbol{\xi}}_1^T, \hat{\boldsymbol{\xi}}_2^T, \ldots, \hat{\boldsymbol{\xi}}_D^T]$ is the $1 \times DN$ Lagrangian vector in the Fourier domain, ρ is the penalty factor, and superscript T indicates the conjugate transpose operation. Then, ADMM is applied by alternately solving the following two sub-problems:

2.5.1. The Solution to Sub-Problem $\hat{g}_{k+1}$

The sub-problem $\hat{g}_{k+1}$ can be formulated as:

$$\begin{aligned} \hat{\boldsymbol{g}}_{k+1} = argmin\frac{1}{2}\left\|\sum_{d=1}^{D}(\hat{\boldsymbol{x}}_k^d \odot \hat{\boldsymbol{g}}_k^d) - \hat{\boldsymbol{y}}\right\|_2^2 \\ + \frac{\gamma_k}{2}\sum_{d=1}^{D}\left\|\hat{\boldsymbol{g}}_k^d - \hat{\boldsymbol{g}}_r^d\right\|_2^2 + \frac{\theta}{2}\left\|\sum_{d=1}^{D}(\hat{\boldsymbol{x}}_k^d \odot \hat{\boldsymbol{g}}_k^d) - \boldsymbol{R}_{pre}\right\|_2^2 \\ + \hat{\boldsymbol{\xi}}^T(\hat{\boldsymbol{g}}_k - \sqrt{N}\boldsymbol{F}\boldsymbol{f}_k) + \frac{\rho}{2}\left\|\hat{\boldsymbol{g}}_k - \sqrt{N}\boldsymbol{F}\boldsymbol{f}_k\right\|_2^2, \end{aligned} \tag{14}$$

which is too complex to solve directly. Given the fact that x_k is sparse-banded, and thus each element of $\hat{y}(\hat{\boldsymbol{y}}(t), t = 1, 2, \ldots, N)$ is dependent only on each $\boldsymbol{x}_k(t) = [x_k^1(t), x_k^2(t), \ldots, x_k^D(t)]^T$ and $\hat{g}_k(t) = [conj(\hat{g}_k^1(t)), conj(\hat{g}_k^2(t)), \ldots, conj(\hat{g}_k^D)(t)]^T$ [45], $conj(.)$ denotes the complex conjugate operation. Thus, we divided the vectorized image $\boldsymbol{x}_k$ into N elements and solved the problem by N smaller and independent problems, over $t = [1, 2, \ldots, N]$:

$$\begin{aligned} \hat{\boldsymbol{g}}_{k+1}(t) = argmin\frac{1}{2}\left\|\hat{\boldsymbol{x}}_k^T(t) \odot \hat{\boldsymbol{g}}_k(t) - \hat{\boldsymbol{y}}(t)\right\|_2^2 \\ + \frac{\gamma_k}{2}\|\hat{\boldsymbol{g}}_k(t) - \hat{\boldsymbol{g}}_r\|_2^2 + \frac{\theta}{2}\left\|(\hat{\boldsymbol{x}}_k^T(t) \odot \hat{\boldsymbol{g}}_k(t)) - \boldsymbol{R}_{pre}\right\|_2^2 \\ + \hat{\boldsymbol{\xi}}^T(\hat{\boldsymbol{g}}_k(t) - \sqrt{N}\boldsymbol{F}\boldsymbol{f}_k(t)) + \frac{\rho}{2}\left\|\hat{\boldsymbol{g}}_k(t) - \sqrt{N}\boldsymbol{F}\boldsymbol{f}_k(t)\right\|_2^2, \end{aligned} \tag{15}$$

where $f_k(t) = [f_k^1(t), f_k^2(t), \ldots f_k^D(t)]$, $\hat{x}_k^T(t)$ is the discrete Fourier transformation of $x_k^T(t)$. Each smaller problem can be calculated efficiently to achieve the solution of Equation (15):

$$\hat{\boldsymbol{g}}_{k+1}(t) = ((1 + N\theta)\hat{\boldsymbol{x}}_k(t)\hat{\boldsymbol{x}}_k^T(t) + (\rho + \gamma_k)N)^{-1}\boldsymbol{H}, \tag{16}$$

$$\boldsymbol{H} = \hat{\boldsymbol{x}}_k^T(t)\hat{\boldsymbol{y}}(t) + N\theta\boldsymbol{R}_{pre}\hat{\boldsymbol{x}}_k^T(t) + N\hat{\boldsymbol{g}}_r - N\hat{\boldsymbol{\xi}}^T + \rho N\hat{\boldsymbol{f}}_k. \tag{17}$$

We use Sherman–Morrison formula [46] to accelerate the inverse operation:

$$(M + uv^T)^{-1} = M^{-1} - M^{-1}uv^TM^{-1}(1 + v^TM^{-1}u). \tag{18}$$

Hence, Equation (16) can be reformulated as:

$$\hat{\boldsymbol{g}}_{k+1}(t) = \frac{1}{(\rho + \gamma_k)N}(I - \frac{\hat{\boldsymbol{x}}_k(t)\hat{\boldsymbol{x}}_k^T(t)}{\frac{(\rho+\gamma_k)N}{1+\theta N} + \hat{\boldsymbol{x}}_k(t)\hat{\boldsymbol{x}}_k^T(t)})\boldsymbol{H}. \tag{19}$$

2.5.2. The Solution to Sub-Problem f_k

The subproblem f_k can be written as:

$$\begin{aligned} f_{k+1} = argmin \frac{1}{2} \sum_{d=1}^{D} \left\| \boldsymbol{w} \odot \boldsymbol{f}_k^d \right\|_2^2 \\ + \hat{\boldsymbol{\xi}}^T (\hat{\boldsymbol{g}}_k - \sqrt{N} \boldsymbol{F} \boldsymbol{f}_k) + \frac{\rho}{2} \left\| \hat{\boldsymbol{g}}_k - \sqrt{N} \boldsymbol{F} \boldsymbol{f}_k \right\|_2^2, \end{aligned} \tag{20}$$

which can be easily solved as follows:

$$\boldsymbol{f}_{k+1} = \frac{\boldsymbol{\xi}^T N + \rho N \boldsymbol{g}_{k+1}}{\boldsymbol{w} \odot \boldsymbol{w} + \rho N}. \tag{21}$$

2.5.3. Update of Lagrangian Parameter

We update Lagrangian multipliers as:

$$\boldsymbol{\xi}^{i+1} = \boldsymbol{\xi}^i + \rho(\boldsymbol{g}_{k+1}^{i+1} - \boldsymbol{f}_{k+1}^{i+1}), \tag{22}$$

where i denotes the iteration number. g_{k+1}^{i+1} and f_{k+1}^{i+1} indicate the solution of subproblem g_{k+1} and f_{k+1}, respectively. The coefficient ρ (initially equals 1) is computed as:

$$\rho^{i+1} = min(\rho_{max}, \beta \rho^i), \tag{23}$$

where β and ρ_{max} are set as 10 and 10,000, respectively.

3. Results and Discussion

In this section, we first compare our ALCF with state-of-the-art trackers with four challenging UAV datasets, then provide experimental analysis to validate our design choices. Finally, a deep version of ALCF is proposed to compare with recent deep trackers to validate the superior performance of ALCF.

3.1. Implementation Details

All experiments were performed using MATLAB R2019a on a computer with Intel i7-9700 CPU@3.0 GHz, 16 GB RAM, and NVIDIA GTX 1080. We set $\theta = 2.2, \tau = 0.9$, $\varphi = 0.35$, $\mu = 6$. Considering the numerical magnitude difference between F_{max} and FD, we set $\alpha_2 = 20 \times \alpha_1 = 500$. The iteration number of ADMM was 2. We extracted histogram of oriented gradients (HOG) [47] and color name (CN) [48] features for input images. A separate filter in [43] was used in our ALCF for scale estimation, and the scale refine module proposed in [44] was used in DeepALCF for scale estimation. Note that the parameters of ALCF remained fixed on all image sequences on all benchmarks. ALCF and DeepALCF had the same parameters except for the size estimation method.

Two widely-used metrics, the precision plot and success plot, describe the performance of all trackers through one-pass evaluation (OPE). Concretely, a precision plot computed the percentages of frames in which the estimated target location is within a given distance threshold from the ground truth. We used the score at the threshold of 20 pixels for ranking the trackers. The success plot measures the fractions of successful frames in which the intersection over union (IoU) between the predicted bounding box and the ground truth is more significant than a certain threshold varied from 0 to 1. We use the area under curve (AUC) of the success plots to rank all trackers.

3.2. Comparison with Hand-Crafted Trackers

We compared our ALCF with thirteen state-of-the-art hand-crafted trackers, including AutoTrack [2], STRCF [28], ECO [49], CSR-DCF [10], MCCT-H [50], ARCF [3], BACF [13], DSST [43], fDSST [11], SRDCF [12], Staple [51], SAMF [52], and KCF [53]. The detailed comparisons on different benchmarks are listed below. Note that the publicly available

codes and default parameters offered by the authors were utilized for impartial comparison. In Figures 5–8, the numbers in the legend indicate the average precision scores for the precision plot and average AUC scores for the success plot.

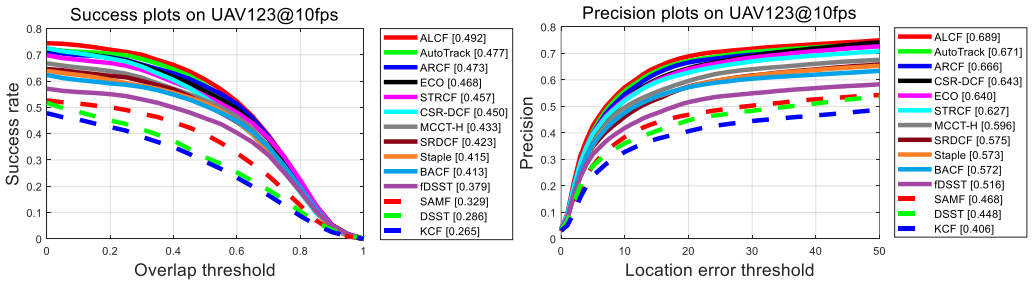

Figure 5. Overall performance of ALCF and other state-of-the-art trackers on UAV123@10fps.

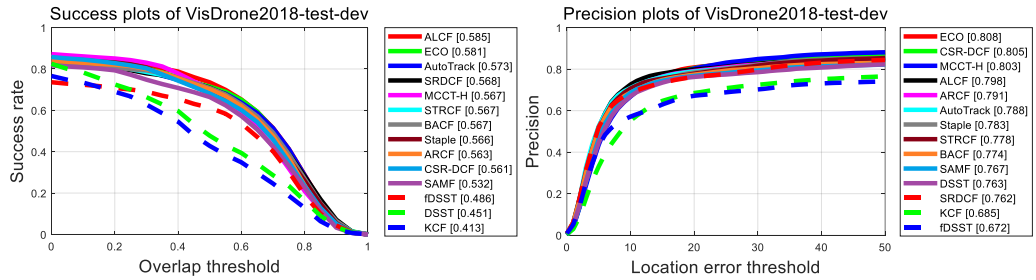

Figure 6. Overall performance of ALCF and other state-of-the-art trackers on VisDrone2018-test-dev.

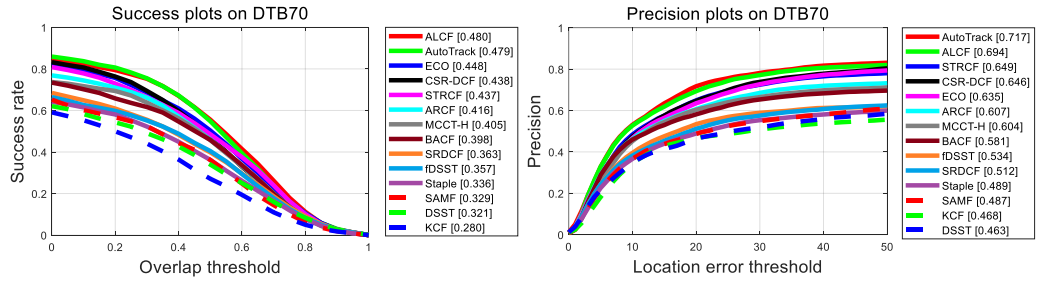

Figure 7. Overall performance of ALCF and other state-of-the-art trackers on DTB70.

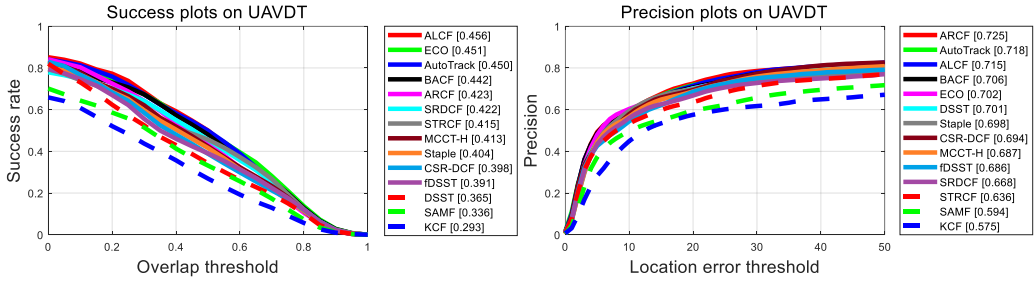

Figure 8. Overall performance of ALCF and other state-of-the-art trackers on UAVDT.

3.2.1. Results on UAV123@10fps

UAV123@10fps [40] consists of 123 challenging sequences with 12 different attributes. All sequences were temporally down-sampled to 10 FPS to simulate challenging large target displacements between consecutive frames. Figure 5 provides comparisons with other preeminent trackers. ALCF achieved the best performance in both precision and success. Notably, ALCF outperformed the previous state-of-the-art tracker, AutoTrack, by a large margin, i.e., 1.8% and 1.5%, in precision and success, respectively. This evidences that ALCF can cope well with sudden changes in target appearance and large position shifts between adjacent frames, which commonly appear in UAV tracking.

3.2.2. Results on VisDrone2018-Test-Dev

VisDrone2018-test-dev [42] contains 35 sequences captured by various UAV platforms in over 14 different cites in China, featuring diverse real-world scenarios. Figure 6 shows that our ALCF obtained the best success score of 0.585 and a competitive precision score of 0.798, which is on par with the best-performing tracker ECO, mainly due to the auto-learning and the lost-and-found strategy in ALCF. The auto-learning strategy can adjust the learning rate according to the target state to prevent model degradation. The lost-and-found strategy can help the tracker recapture target after the target is lost, enhancing the performance in coping with long-term tracking (the average length of VisDrone2018-test-dev is around 940 frames; the most extended sequence can reach a length of 2569 frames).

3.2.3. Results on DTB70

DTB70 [41] comprises 70 sequences covering different types of UAV movements, including rapid translation and rotation. Targets include humans, animals, and rigid objects. In Figure 7, the numbers in the legend indicate the average precision scores for the precision plot and average AUC scores for the success plot. Figure 7 depicts that our ALCF performed the best in terms of success rate and second-best in precision. We attribute this to the auto-learning update strategy in ALCF, since there are many viewpoint changes in DTB70, where auto-learning plays an important role.

3.2.4. Results on UAVDT

UAVDT [39] was selected from over 10 h of videos taken by a UAVs at different locations in urban areas. The targets are mainly vehicles with additional attributions, i.e., weather conditions, flying altitude, and camera view. As we can see in Figure 8, our tracker is the best in terms of success. It outperformed the previous best tracker, ECO [49], and second-best tracker, AutoTrack [2], by 0.5% and 0.6%, respectively.

3.2.5. Average Performance Results

Figure 9 reports the average precision and success rate of the 13 trackers and ALCF on the four benchmarks. ALCF yielded the best average success rate score and precision score. It is noted that ALCF was the only tracker with a success rate score (0.503) above 0.5, surpassing the second-best tracker AutoTracker (0.494) by 0.9%. As far as we know, ALCF is the best tracker with hand-drafted features on the four benchmarks.

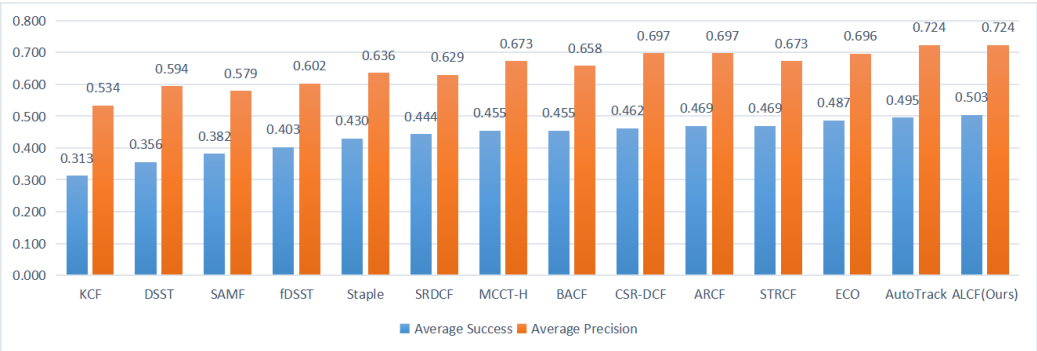

Figure 9. Overall evaluation of ALCF and other trackers regarding average precision and success rate.

3.2.6. Per-Attribute Evaluation

Figure 10 provides the precision comparison of twelve attributes defined on UAV123@10fps. The number in parentheses is the precision of ALCF. Detailed definitions of these attributions can be found on UAV123@10fps. Our method performed the best against state-of-the-art trackers (i.e., AutoTrack [2], ARCF [3], BACF [13], ECO [49], and STRCF [28]) in most attributes. It is worth noting that our ALCF performed extremely well not only in coping with rapid changes (e.g., viewpoint or illumination variation, and camera motion), but also in preventing model degradation (e.g., out-of-view, partial and full occlusion).

These challenges abound in real-world UAV object tracking, which is ample evidence that our strategies are quite effective for UAV tracking.

3.2.7. Visualization

Figure 11 illustrates some qualitative comparisons of ALCF with other top-performing trackers on some challenging sequences (from top down they are *ChasingDrones (DTB70), StreetBasketball3 (DTB70), bike2 (UAV123@10fps), person10 (UAV123@10fps),* and *S0602 (UAVDT)*). Our ALCF showed superiority over other trackers in handling challenging scenarios, including fast motion (the first sequence), heavy occlusion (the second sequence), low resolution (the third sequence), loss from view (the fourth sequence), and viewpoint change (the last sequence). Of course, our algorithm still has limitations. In the case of long-term occlusion and superimposition of target deformation, tracking loss will occur.

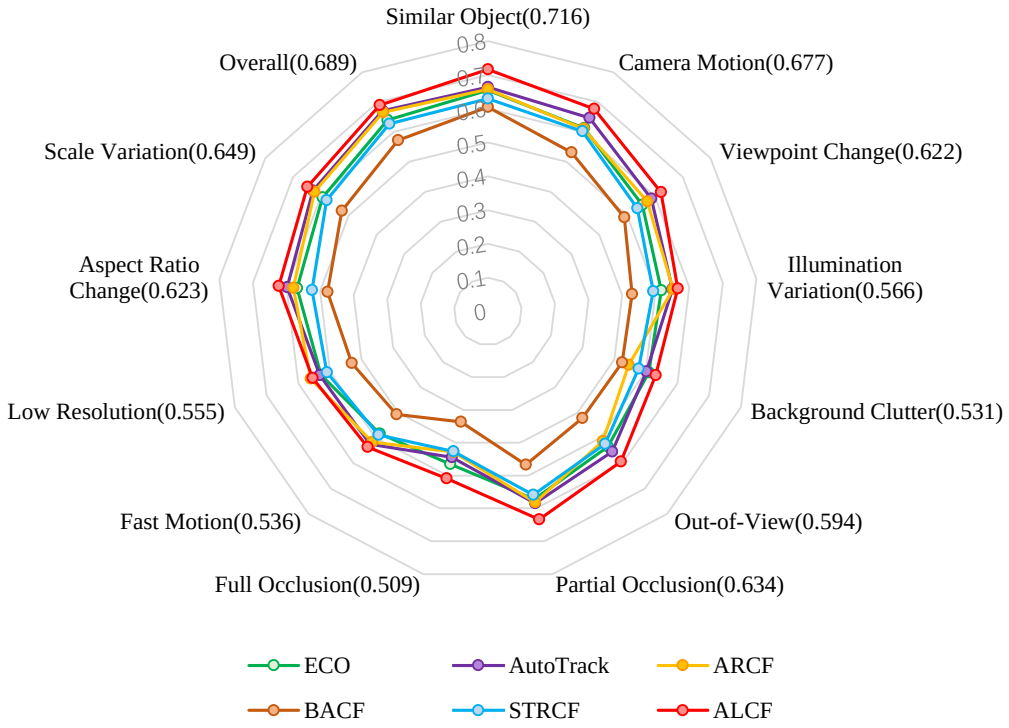

Figure 10. Attribute-based evaluation of ALCF and some other trackers in precision.

3.2.8. Speed

Table 1 compares the tracking speeds of ALCF and other DCF-based trackers (top-5 trackers with hand-drafted features in terms of success over four benchmarks). All the trackers can run on a single CPU. ALCF reached a real-time speed of 51.3 FPS, comparable to that of ECO and AutoTrack, with better predictive performance.

Table 1. Average speed (FPS) of top five trackers on [39–42], ranked by average success (top 3 in **bold**).

	ALCF	**AutoTrack**	**ECO-HC**	**STRCF-HC**	**ARCF**
FPS	**51.3**	**58.3**	**69.4**	23.2	23.3
Success	**0.503**	**0.494**	**0.487**	0.469	0.469
Precision	**0.724**	**0.723**	0.696	0.672	**0.697**

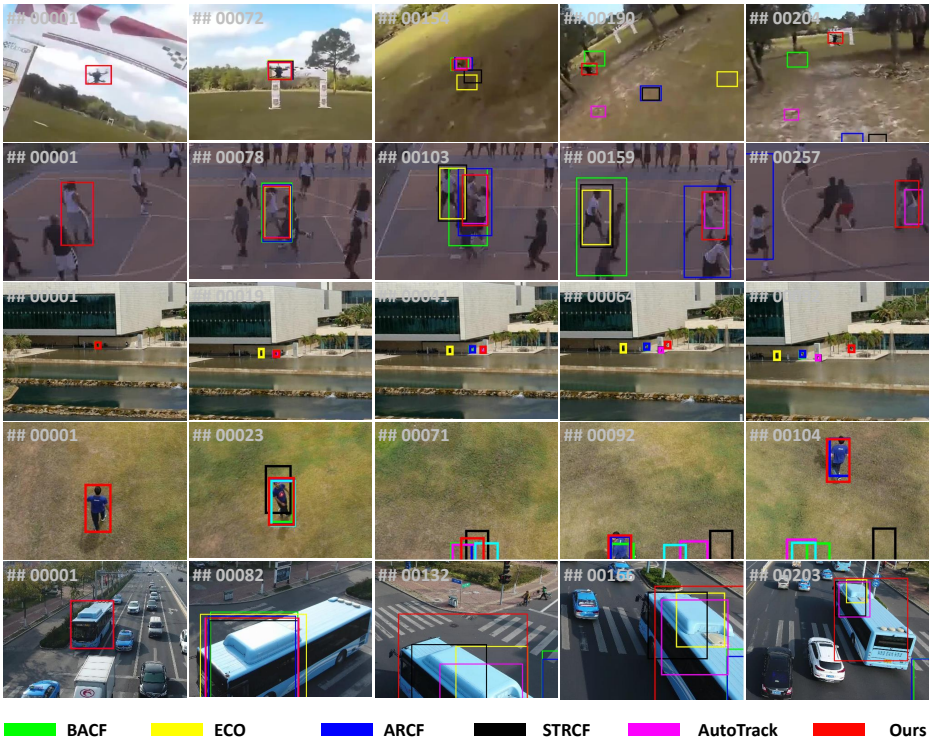

Figure 11. Qualitative comparison of ALCF with other top-performing trackers on some challenging sequences.

3.3. Ablation Studies

We investigated the effectiveness of different components of ALCF on UAV123@10fps. As presented in Table 2, we used the model variant, which employs the same objective function and fixes the learning rate of 0.025 as in SRDCF [12] as a baseline. For a fair comparison, the baseline model adopted the same features extraction and scale estimation scheme as ALCF (original SRDCF utilizes only HOG features, so the performance in Figure 5 is not as good as that in Table 2). AL, MC, and LF denote the auto-learning strategy, model constraint regularization term, and lost-and-found strategy, respectively. Different components are incrementally added to the baseline model individually or in combination to verify their effectiveness.

Table 2. Ablation analysis. R represents the response map regularization term. AUC represents the success rate (top 1 in **bold.**)

	AL	MC	LF	R	Precision	AUC	FPS
Baseline					0.600	0.432	36.94
	✓				0.637	0.460	45.64
		✓			0.649	0.463	34.03
			✓		0.648	0.465	32.38
		✓	✓		0.654	0.468	32.36
	✓	✓	✓		0.667	0.473	47.89
	✓			✓	0.654	0.469	42.44
		✓		✓	0.650	0.464	33.70
			✓	✓	0.652	0.474	36.52
		✓	✓	✓	0.662	0.475	29.91
	✓	✓	✓	✓	**0.689**	**0.492**	**41.31**
Baseline+STRCF					0.638	0.457	31.87
Baseline+MC+Fix					0.639	0.459	33.40

3.3.1. Effect of Auto-Learning Strategy

As shown in Table 2, Baseline+AL outperformed the baseline in both precision and success rate by 3.7% and 2.8%, respectively. In addition, incorporating an auto-learning strategy improves tracking speed by 23.6%, as it helps reduce meaningless or detrimental training of individual samples. The tracking speed in Table 2 is lower than the average speed in Table 1 because the targets in UAV123@10fps vary drastically between consecutive frames compared to those in the other three datasets, resulting in the frequent model updating in most frames. Nevertheless, our auto-learning strategy still improved the tracking speed significantly.

3.3.2. Effect of Regularization Term

Baseline+MC denotes the model variant that incorporated the proposed regularization term into the baseline model, Baseline+MC+Fix replaced the adaptive regularization coefficient γ calculated from Equation (8) with a fixed constant, and Baseline+STRCF updated the reference model in Equation (7) after each frame without considering target states [28]. As is shown in Table 2, using a credible model as a reference and an adaptive regularization coefficient contributes more to the performance boost.

3.3.3. Effect of the Lost-and-Found Strategy

We present a model variant (Baseline+LF) with a lost-and-found strategy added to the baseline model. Notably, employing the lost-and-found strategy boosted the performance significantly by 4.8% and 3.3% in precision and success rate, respectively, along with a negligible increase in tracking speed.

3.3.4. Effect of the Objective Function

We also found that all three components can be further improved to different degrees with the help of the additional response map regularization term used in [3], mainly attributed to its aberration suppression of the response maps between two adjacent frames, which helps the response maps to be more stable. The TSE proved to be more accurate (the performance of Baseline+AL+MC+LF+R (precision: 0.689, AUC: 0.492, FPS: 41.31) is higher than that of Baseline+AL+MC+LF (precision: 0.667, AUC: 0.473, FPS: 47.89) with a slight sacrifice in speed). We also verified the generality of our three innovations, which can improve the tracking performance for trackers with different objective functions.

3.3.5. Construction of Target State Estimation

In our experiment, we found that T (calculated from Equation (3)) has an extensive variation range, which exceeded 10 when the target state was trustworthy and dropped below 2 when the target was lost. We should focus on the variation when T falls between 2 to 10 to perceive the transition of target states. Hence, we introduce the sigmoid activation function (with μ as 6) upon T to magnify its variation in the interval [2,10] while suppressing noisy fluctuations when the target state is entirely credible ($T > 10$) or completely untrustworthy ($T < 2$).

We also experimented on UAV123@10fps to validate our parameter settings. Figure 12 shows the performance peaks at $\mu = 6$. When μ becomes small, the value of TSE becomes large (close to 1), which leads to a gradual reduction of the learning rate to 0. When $\mu = 0$, it is equivalent to not updating the model, so the model's performance was the worst. Moreover, when μ becomes large, the value of TSE will become smaller, which caused the model to use a large learning rate to update when it did not need to be updated and zero when it needed to be updated. However, the performance was slightly better compared to the case of not updating the model ($\mu = 0$), which verified the importance of the model update.

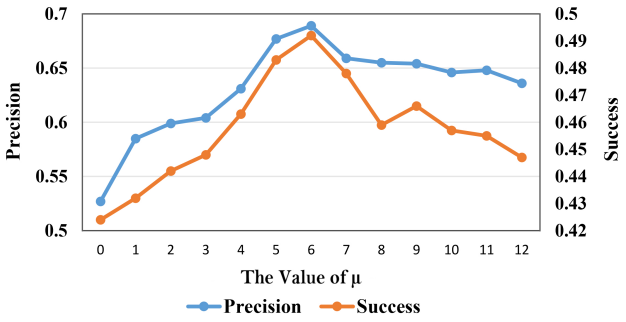

Figure 12. Effect of μ on the performance of ALCF.

3.3.6. Sensitivity Analysis of Model Constraint Regularization Term

Based on STRCF [28], we further designed an adaptive model constraint regularization term for UAV tracking, which can update the reference model adaptively and adjust the effect of the reference model on the correlation filter learning. Moreover, we validated our parameter settings on UAV123@10fps. We set a threshold τ to refresh f_r. When the value of TSE is larger than τ, the reference model f_r will be updated. To find the optimal value of τ, we refined the search within the range of values of τ, taking values at 0.1 intervals. The result is shown in Figure 13. We can see that when $\tau = 0$, the reference filter f_r is updated every frame with an adaptive parameter γ. When $\tau = 1$, the reference filter f_r is never updated during tracking. By comprehensively considering the precision and success rate, we set $\tau = 0.9$.

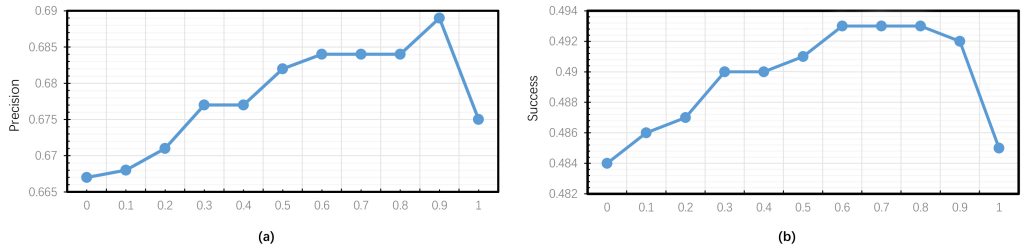

Figure 13. Effect of τ on the performance of ALCF. (**a**) shows the precision while (**b**) shows the success rate.

3.4. Extension to Deep Trackers

We further evaluated the tracking performance of ALCF in comparison with other state-of-the-art deep-based trackers on the four benchmarks. Inspired by [44], we proposed a deep version of ALCF called DeepALCF, which exploits Alpha-Refine as our scale estimation. Note that DeepALCF still exploits HOG and CN as feature presentations for object location, but it exploits ResNet-34 to extract deep features for scale estimation, which is more accurate. More detail about the scale estimation can be found in Alpha-Refine [44].

3.4.1. Results on VisDrone2018-Test-Dev

We selected 10 new state-of-the-art deep trackers for comparison. The results are shown in Table 3. DeepALCF yielded the best precision score (0.816), surpassing the second-best tracker KYS [54] and third-best tracker DiMP [55] by 0.6% and 1%. DeepALCF also obtained the best success score (0.619), outperforming the second-best tracker, Super_DiMP [56] (0.610), and the third-best tracker, DiMP [55] (0.605) by 0.9% and 1.4%. This proves that the auto-learning strategy can cope well with object tracking in natural scenes.

Table 3. Performance comparison of ALCF with other ten state-of-the-art deep trackers (top 3 in **bold**).

	MDNet [16]	UDT+ [15]	PrDiMP [57]	ATOM [20]	DiMP [55]	SiamRPN [17]	CFNet [58]	Super_DiMP [56]	KYS [54]	HiFT [59]	ALCF Ours	DeepALCF Ours
Precision	0.790	0.800	0.797	0.751	**0.806**	0.781	0.778	0.800	**0.810**	0.721	0.798	**0.816**
Success	0.579	0.584	0.600	0.563	**0.605**	0.573	0.568	**0.610**	0.600	0.527	0.585	**0.619**

3.4.2. Results on UAVDT

Table 4 shows the precision and success scores of ALCF and other 12 deep trackers on UAVDT. ALCF was very competitive in comparison with PrDiMP [57] in both precision and success and outperformed third-best tracker MDNet [16] by 2.3% and 7% in terms of precision and success, respectively.

3.4.3. Results on DTB70

We compared DeepALCF with 13 popular deep trackers. Table 5 shows the comparison results. DeepALCF outperformed the third-best tracker DaSiamRPN [21] (0.512) by 1.8% in terms of success, and DeepALCF (0.727) is very close to DaSiamRPN (0.735) in terms of precision. Compared with ARTracker [33], ALCF reached a real-time speed of 51.3 FPS, which is far more than ARTracker (FPS: 13.28).

3.4.4. Results on UAV123@10fps

We conducted a comprehensive experiment in Table 6, which compares DeepALCF and ALCF with numerous recent deep trackers. DeepALCF achieved the best precision (0.712) and success rate (0.516) scores. It can be found that ALCF with hand-drafted features obtained a third-best precision score, which outperformed a lot of deep trackers while running at over 50 FPS on a single CPU.

Table 4. Overall evaluation of ALCF, DeepALCF, and some other state-of-the-art trackers on UAVDT [39] (top 3 in **bold**).

Trackers	Venue	Precision	Success
MDNet [16]	CVPR2016	**0.725**	0.464
MCCT [50]	CVPR2018	0.691	0.448
ASRCF [24]	CVPR2019	0.720	0.449
TADT [60]	CVPR2019	0.700	0.441
UDT [15]	CVPR2019	0.674	0.442
UDT+ [15]	CVPR2020	0.696	0.415
PrDiMP [57]	CVPR2020	**0.757**	**0.559**
fECO [61]	TIP2020	0.699	0.415
fDSTRCF [61]	TIP2020	0.677	0.454
HiFT [59]	ICCV2021	0.652	**0.474**
LUDT [62]	IJCV2021	0.631	0.418
LUDT+ [62]	IJCV2021	0.701	0.406
ALCF	Ours	**0.725**	0.456
DeepALCF	Ours	**0.748**	**0.534**

Table 5. Overall evaluation of ALCF and some other state-of-the-art trackers on DTB70 [41] (top 3 in **bold**).

Trackers	Venue	Precision	Success
HCF [63]	ICCV2015	0.616	0.415
MDNet [16]	CVPR2016	0.703	0.466
CFNet [58]	CVPR2017	0.587	0.398
IBCCF [64]	ICCVW2017	0.669	0.460
CREST [65]	ICCV2017	0.650	0.452
ADNet [66]	CVPR2017	0.637	0.422
DaSiamRPN [21]	ECCV2018	**0.735**	**0.512**
UDT+ [15]	CVPR2019	0.650	0.457
MCCT [50]	CVPR2019	0.725	0.484
TADT [60]	CVPR2019	0.690	0.460
ASRCF [24]	CVPR2019	0.696	0.468
KAOT [67]	TMM2020	0.692	0.469
ARTracker [33]	GRSL2022	**0.752**	**0.588**
ALCF	Ours	0.694	0.480
DeepALCF	Ours	**0.727**	**0.530**

Table 6. Overall evaluation of ALCF and some other state-of-the-art trackers on UAV123@10fps [40] (top 3 in **bold**).

Trackers	Precision	Success	Trackers	Precision	Success
HCF	0.601	0.426	MDNet	0.664	0.477
CFNet	0.568	0.422	IBCCF	0.651	0.481
SiamFC [68]	0.678	0.472	CREST	0.600	0.445
C-COT [69]	**0.704**	**0.502**	DeepSTRCF	0.680	0.499
DaSiamRPN	**0.689**	0.481	UDT+	0.674	0.470
MCCT	**0.689**	0.496	ADNet	0.647	0.422
TADT	0.685	**0.507**	ASRCF	0.685	0.477
ALCF	**0.689**	0.492	DeepALCF	**0.712**	**0.516**

4. Conclusions

In this work, we proposed a novel tracking method called ALCF to deal with challenging situations (occlusion, deformation, etc.) in UAV tracking. Specifically, we introduced a novel criterion named target state estimation (TSE) to identify exact target states and adjust the learning rate for model update adaptively. In addition, a new model constraining regularization terms and an efficient lost-and-found strategy were further developed to enhance the robustness of the tracker to target variation and loss. Comprehensive experiments on four widely-used UAV datasets demonstrated that ALCF performs better than most state-of-the-art trackers in both precision and success scores while running over 50 FPS on a single CPU. We also provided a deep version tracker called DeepALCF, which is competitive against recently popularized deep trackers. We believe our work would contribute to the field of UAV tracking. We will focus on improving the real-time performance of the model in edge computing and running our algorithm on a hardware system on a UAV to evaluate its performance.

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

Funding: This research was funded by the National Key Laboratory Foundation of China, grant number TCGZ2020C004 and 202020429036.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

Acknowledgments: This work was supported by the Key Laboratory Foundation under grant TCGZ2020C004 and grant 202020429036.

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

References

1. Lin, B.; Bai, Y.; Bai, B.; Li, Y. Robust Correlation Tracking for UAV with Feature Integration and Response Map Enhancement. *Remote Sens.* **2022**, *14*, 4073. [CrossRef]
2. 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 Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11923–11932.
3. 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, Seoul, Korea, 27 October–2 November 2019; pp. 2891–2900.
4. Chen, J.; Xu, T.; Li, J.; Wang, L.; Wang, Y.; Li, X. Adaptive Gaussian-Like Response Correlation Filter for UAV Tracking. In Proceedings of the ICIG, Haikou, China, 6–8 August 2021; pp. 596–609.
5. Wang, L.; Li, J.; Huang, B.; Chen, J.; Li, X.; Wang, J.; Xu, T. Auto-Perceiving Correlation Filter for UAV Tracking. *IEEE Trans. Circuits Syst. Video Technol.* **2022,** *32*, 5748–5761. [CrossRef]
6. Nex, F.; Remondino, F. UAV for 3D mapping applications: A review. *Appl. Geomat.* **2014**, *6*, 1–15. [CrossRef]
7. Torresan, C.; Berton, A.; Carotenuto, F.; Gennaro, S.F.D.; Gioli, B.; Matese, A.; Miglietta, F.; Vagnoli, C.; Zaldei, A.; Wallace, L. Forestry applications of UAVs in Europe: A review. *Int. J. Remote Sens.* **2017**, *38*, 2427–2447. [CrossRef]
8. Karaduman, M.; Cınar, A.; Eren, H. UAV Traffic Patrolling via Road Detection and Tracking in Anonymous Aerial Video Frames. *J. Intell. Robot. Syst.* **2019**, *95*, 675–690. [CrossRef]
9. Huang, B.; Chen, J.; Xu, T.; Wang, Y.; Jiang, S.; Wang, Y.; Wang, L.; Li, J. SiamSTA: Spatio-Temporal Attention based Siamese Tracker for Tracking UAVs. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, 11–17 October 2021; pp. 1204–1212.
10. Lukežič, A.; Vojíř, 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, Honolulu, HI, USA, 21–26 July 2017; pp. 6309–6318.
11. Danelljan, M.; Häger, G.; Khan, F.S.; Felsberg, M. Discriminative Scale Space Tracker. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1561–1575. [CrossRef] [PubMed]
12. Danelljan, M.; Häger, G.; 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. Galoogahi, H.K.; 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 October2017; pp. 1144–1152.
14. Huang, B.; Xu, T.; Jiang, S.; Chen, Y.; Bai, Y. Robust visual tracking via constrained multi-kernel correlation filters. *IEEE Trans. Multimed.* **2020**, *22*, 2820–2832. [CrossRef]
15. Wang, N.; Song, Y.; Ma, C.; Zhou, W.; Liu, W.; Li, H. Unsupervised Deep Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 1308–1317.
16. Nam, H.; Han, B. Learning Multi-Domain Convolutional Neural Networks for Visual Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4293–4302.
17. 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, 18–23 June 2018; pp. 8971–8980.
18. Wang, Y.; Xu, T.; Jiang, S.; Chen, J.; Li, J. Pyramid Correlation based Deep Hough Voting for Visual Object Tracking. In Proceedings of the Asian Conference on Machine Learning, PMLR, Virtual Event, 17–19 November 2021; pp. 610–625.
19. Wang, M.; Liu, Y.; Huang, Z. Large margin object tracking with circulant feature maps. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4021–4029.
20. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. ATOM: Accurate Tracking by Overlap Maximization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4660–4669.
21. Zhu, Z.; Wang, Q.; Li, B.; Wu, W.; Yan, J.; Hu, W. Distractor-Aware Siamese Networks for Visual Object Tracking. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 101–117.
22. Bolme, D.; Beveridge, J.; Draper, B.; Lui, Y.M. Visual object tracking using adaptive correlation filters. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2544–2550.
23. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J.P. Exploiting the Circulant Structure of Tracking-by-Detection with Kernels. In Proceedings of the European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 702–715.

24. Dai, K.; Wang, D.; Lu, H.; Sun, C.; Li, J. Visual Tracking via Adaptive Spatially-Regularized Correlation Filters. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4670–4679.
25. Xu, T.; Feng, Z.; Wu, X.J.; Kittler, J. Adaptive Channel Selection for Robust Visual Object Tracking with Discriminative Correlation Filters. *Int. J. Comput. Vis.* **2021**, *129*, 1359–1375. [CrossRef]
26. Wang, F.; Yin, S.; Mbelwa, J.T.; Sun, F. Context and saliency aware correlation filter for visual tracking. *Multimed. Tools Appl.* **2022**, *81*, 27879–27893. [CrossRef]
27. 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]
28. 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.
29. Huang, B.; Xu, T.; Shen, Z.; Jiang, S.; Zhao, B.; Bian, Z. SiamATL: Online Update of Siamese Tracking Network via Attentional Transfer Learning. *IEEE Trans. Cybern.* **2021**, *52*, 7527–7540. [CrossRef] [PubMed]
30. Dai, K.; Zhang, Y.; Wang, D.; Li, J.; Lu, H.; Yang, X. High-performance long-term tracking with meta-updater. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6298–6307.
31. Yang, T.; Xu, P.; Hu, R.; Chai, H.; Chan, A.B. ROAM: Recurrently optimizing tracking model. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6718–6727.
32. Xingjian, S.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; Woo, W.c. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 802–810.
33. Chen, J.; Xu, T.; Huang, B.; Wang, Y.; Li, J. ARTracker: Compute a More Accurate and Robust Correlation Filter for UAV Tracking. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [CrossRef]
34. Kalal, Z.; Mikolajczyk, K.; Matas, J. Tracking-Learning-Detection. *IEEE Trans. Softw. Eng.* **2011**, *34*, 1409–1422. [CrossRef]
35. Fan, H.; Ling, H. Parallel Tracking and Verifying: A Framework for Real-Time and High Accuracy Visual Tracking. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5487–5495.
36. Lukezic, A.; Zajc, L.C.; Vojír, T.; Matas, J.; Kristan, M. FCLT-A Fully-Correlational Long-Term Tracker. *arXiv* **2018**, arXiv:1804.07056.
37. Voigtlaender, P.; Luiten, J.; Torr, P.H.; Leibe, B. Siam R-CNN: Visual Tracking by Re-Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6578–6588.
38. Huang, L.; Zhao, X.; Huang, K. GlobalTrack: A Simple and Strong Baseline for Long-Term Tracking. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 11037–11044.
39. Du, D.; Qi, Y.; Yu, H.; Yang, Y.; Duan, K.; Li, G.; Zhang, W.; Huang, Q.; Tian, Q. The Unmanned Aerial Vehicle Benchmark: Object Detection and Tracking. In Proceedings of the European conference on computer vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 370–386.
40. Müller, M.; Smith, N.; Ghanem, B. A Benchmark and Simulator for UAV Tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 445–461.
41. Li, S.; Yeung, D.Y. Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31.
42. Zhu, P.; Wen, L.; Bian, X.; Ling, H.; Hu, Q. Vision Meets Drones: A Challenge. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; Springer: Berlin/Heidelberg, Germany, 2018.
43. 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; Bmva Press: Newcastle, UK, 2014.
44. Yan, B.; Zhang, X.; Wang, D.; Lu, H.; Yang, X. Alpha-refine: Boosting tracking performance by precise bounding box estimation. In Proceedings of the CVPR, Virtual, 19–25 June 2021; pp. 5289–5298.
45. Galoogahi, H.K.; Sim, T.; Lucey, S. Multi-channel Correlation Filters. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 3072–3079.
46. 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]
47. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–26 June 2005; Volume 1, pp. 886–893.
48. Danelljan, M.; Khan, F.; Felsberg, M.; van de Weijer, J. Adaptive Color Attributes for Real-Time Visual Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1090–1097.
49. 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.
50. Wang, N.; Zhou, W.; Tian, Q.; Hong, R.; Wang, M.; Li, H. Multi-cue Correlation Filters for Robust Visual Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4844–4853.
51. Bertinetto, L.; Valmadre, J.; Golodetz, S.; Miksik, O.; Torr, P.H.S. Staple: Complementary Learners for Real-Time Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1401–1409.

52. Li, Y.; Zhu, J. A Scale Adaptive Kernel Correlation Filter Tracker with Feature Integration. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 5–12 September 2014; pp. 254–265.
53. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 583–596. [CrossRef]
54. Bhat, G.; Danelljan, M.; Gool, L.V.; Timofte, R. Know your surroundings: Exploiting scene information for object tracking. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2020; pp. 205–221.
55. Bhat, G.; Danelljan, M.; Gool, L.V.; Timofte, R. Learning Discriminative Model Prediction for Tracking. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 6182–6191.
56. Danelljan, M.; Bhat, G. PyTracking: Visual Tracking Library Based on PyTorch. Available online: https://github.com/visionml/pytracking (accessed on 8 January 2020).
57. Danelljan, M.; Gool, L.V.; Timofte, R. Probabilistic Regression for Visual Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 7183–7192.
58. Valmadre, J.; Bertinetto, L.; Henriques, J.; 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, Honolulu, HI, USA, 21–26 July 2017; pp. 2805–2813.
59. Cao, Z.; Fu, C.; Ye, J.; Li, B.; Li, Y. HiFT: Hierarchical Feature Transformer for Aerial Tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 15457–15466.
60. Li, X.; Ma, C.; Wu, B.; He, Z.; Yang, M.H. Target-Aware Deep Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 1369–1378.
61. 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]
62. 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]
63. Ma, C.; Huang, J.B.; Yang, X.; Yang, M.H. Hierarchical Convolutional Features for Visual Tracking. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 3074–3082.
64. Li, F.; Yao, Y.; Li, P.; Zhang, D.; Zuo, W.; Yang, M.H. Integrating Boundary and Center Correlation Filters for Visual Tracking with Aspect Ratio Variation. In Proceedings of the IEEE International Conference on Computer Orkshops (ICCVW), Venice, Italy, 22–29 October 2017; pp. 2001–2009.
65. Song, Y.; Ma, C.; Gong, L.; Zhang, J.; Lau, R.; Yang, M.H. CREST: Convolutional Residual Learning for Visual Tracking. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2555–2564.
66. Yun, S.; Choi, J.; Yoo, Y.; Yun, K.; Young Choi, J. Action-decision networks for visual tracking with deep reinforcement learning. In Proceedings of the CVPR, Honolulu, HI, USA, 21–28 July 2017; pp. 2711–2720.
67. 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.* **2020**, *23*, 810–822. [CrossRef]
68. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H.S. Fully-Convolutional Siamese Networks for Object Tracking. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016; pp. 850–865.
69. Danelljan, M.; Robinson, A.; Khan, F.; Felsberg, M. Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016; pp. 472–488.