

# Robust Correlation Filter Learning With Continuously Weighted Dynamic Response for UAV Visual Tracking

Yang Zhang<sup>✉</sup>, Yu-Feng Yu<sup>✉</sup>, Member, IEEE, Long Chen<sup>✉</sup>, Senior Member, IEEE,  
and Weiping Ding<sup>✉</sup>, Senior Member, IEEE

**Abstract**—Unmanned aerial vehicles (UAVs) visual tracking has always been a challenging task. Existing correlation filter tracking algorithms typically utilize the histograms of oriented gradients (HOGs) and color names (CNs) method to directly incorporate the extracted target features into the model updating process. However, low-resolution (LR) video quality leads to unstable target feature values. To address this limitation, we propose a novel preprocessing technique involving Gaussian denoising. This preprocessing step is designed to enhance the stability of the target's feature values and make the target's scale information clearer, thereby improving the tracker's recognition capability for the target and effectively reducing noise interference. Furthermore, in contrast to other UAV trackers that rely on a singular representation of contextual information, this article aims to enhance the utilization of historical information. Therefore, we introduce a context-based approach that integrates continuously weighted dynamic response maps from both temporal and spatial perspectives. Our tracker has the ability to adapt to rapid environmental changes during the tracking process while simultaneously reducing the potential risks of model overfitting and distortion. Extensive experiments are conducted on authoritative datasets, including DTB70, UAV123@10fps, and UAVDT, comparing our model against other advanced trackers. The experimental results validate the superior tracking performance and robustness of our tracker.

**Index Terms**—Correlation filter, Gaussian denoising, visual tracking, weighted dynamic response.

## I. INTRODUCTION

THROUGH continuous exploration and research by multiple generations of scientists, the field of computer vision has progressively matured. Nevertheless, visual tracking remains a highly prominent and actively studied topic. In particular, the widespread use of unmanned aerial vehicles

Manuscript received 26 June 2023; revised 15 September 2023; accepted 13 October 2023. Date of publication 17 October 2023; date of current version 27 October 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62006056 and Grant 61976120, in part by the Natural Science Foundation of Guangdong Province under Grant 2022A1515011595, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20231337, and in part by the Natural Science Key Foundation of Jiangsu Education Department under Grant 21KJA510004. (*Corresponding author: Yu-Feng Yu*)

Yang Zhang and Yu-Feng Yu are with the Department of Statistics, Guangzhou University, Guangzhou 510006, China (e-mail: 2112164137@e.gzhu.edu.cn; yuyufeng220@163.com).

Long Chen is with the Department of Computer and Information Science, University of Macau, Macau, China (e-mail: longchen@umac.mo).

Weiping Ding is with the School of Information Science and Technology, Nantong University, Nantong 226019, China (e-mail: ding.wp@ntu.edu.cn).

Digital Object Identifier 10.1109/TGRS.2023.3325337

(UAVs) in major cities around the world has made UAV tracking technology highly valuable for research and application. If UAVs could replace human eyes to identify and track targets in various sectors in reality, this would greatly improve the efficiency of society's operations, such as in agriculture [1], forestry [2], wildlife conservation [3], and road traffic [4].

Looking back at the developments in the field of visual tracking, it is evident that discriminative filtering methods have achieved significant success. In the tracking process, discriminative correlation filters (DCF)-based trackers [5], [6], [7] record the initial frame position of the target and extract histograms of oriented gradients (HOGs) [8] and color names (CNs) features [9] from the target and background regions. Trackers [10], [11] that integrate deep learning also extract convolutional neural network (CNN) features [12] from the target. Using a Gaussian function to set the label, the trackers learn the corresponding filters based on the extracted feature channels. The filter is convolved with the feature of the target in the next frame, and the highest value area in the generated response map is searched for as the predicted position of the target. The correlation operation is transformed to the frequency domain via fast Fourier transform (FFT) [13] and iteratively solved by using the alternating direction method of multipliers (ADMMs) algorithm [14]. This enables the tracker to achieve excellent tracking accuracy while also maintaining fast tracking speed. DCF methods have the advantage of being able to deliver convincing performance without requiring expensive graphics processing unit (GPU) equipment.

In order to evaluate the performance of trackers, tracking experiments are typically conducted on authoritative datasets. In traditional benchmark dataset testing [15], [16], trackers such as spatially regularized DCFs (SRDCF) [17], spatial-temporal regularized correlation filters (STRCFs) [18], and learning adaptive DCFs (LADCF) [19] have successively refreshed records of tracking accuracy by improving the mechanisms of their models. This suggests that DCF methods still have the potential to increase robustness and have untapped room for improvement. Compared to traditional benchmarks, the benchmark datasets [20], [21], [22] for UAV tracking are more complex and specific. These datasets, which are captured by UAVs in multiple locations and weather conditions, contain diverse scenes such as cities, countryside, highways, forests, and more than 20 target categories. Moreover, most video sequences consist of more than a thousand frames. Therefore,

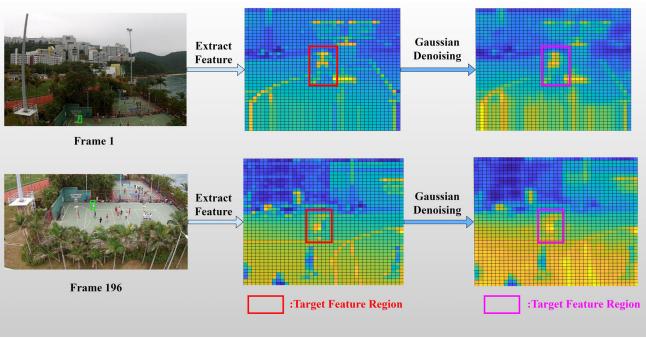


Fig. 1. Visualization of raw features and Gaussian denoised features.

UAV trackers not only require real-time tracking capability, but also need to address the challenges of target occlusion, rapid motion, scale variation (SV), and viewpoint changes (VCs). In existing research, methods [23] based on improved discriminative filtering models have achieved excellent performance. They are no longer limited to using the content of the current frame to train the tracker but also consider the importance of historical information for filter learning. The idea of incorporating features, filters, and response maps from a specific number of past frames into the tracking framework has proven effective. However, these approaches have a single and insufficient use of historical information. In the real-time tracking of a large number of video sequences, the target may undergo unpredictable transformations in each frame. Therefore, when retrospectively utilizing historical information, it is difficult to ensure that the selected specific frame contains accurate information, and there is a risk of decreased robustness in the tracker. To address this issue, we explore how to more comprehensively and accurately utilize historical information and emphasize the effectiveness of denoising features.

In this article, we propose a novel approach to denoise features using Gaussian functions. As shown in Fig. 1, for different frames of the same video sequence, we apply Gaussian denoising to the extracted raw features. The red and pink boxes represent the target feature areas before and after denoising, respectively. According to the visual visualization display, although Gaussian denoising smooths out the edges and details of the target to some extent, the benefits outweigh the drawbacks as it makes the feature values of the target more stable. After denoising, the target retains more position and scale information, which improves the overall recognition of the target by the tracker. This approach can significantly enhance the tracking performance of the model. Furthermore, in order to address the issue of limited utilization of historical information, we increase the depth of retrospective analysis of historical information. As shown in Fig. 2, during the training stage, the model records the response values of the previous  $k$  frames and sparsifies the target area response by applying a background highlight matrix. Considering the uncertainty of potential changes in the target in different scenarios, equal weighting of past response values may lead to overfitting of the tracker. Therefore, we introduce an exponential smoothing model to reassign weights to the response values based on time, such that historical frame response values closer in time to the current frame have greater weight. Given that

drone tracking commonly involves aerial photography from a top-down perspective, and drones are in a state of rapid motion, the background environment within the region of interest for the target in each frame is prone to substantial variations. If the tracker fails to effectively adapt to these background environment changes, it is highly likely that when background areas exhibit objects resembling the target, the drone might struggle to achieve accurate recognition and discernment, consequently resulting in the loss of the correct target. To address this issue and facilitate the tracker's swift adaptation to intricate scenarios, we employ a background highlight matrix. This matrix serves to sparsify the weighted historical response values related to the target, thereby accentuating the impact of the background region. Such a manipulation assists the tracker in more effectively assimilating the dynamics of the background environment, thereby enhancing the target's discriminative performance. Then, the weighted historical response values are summed and embedded into the filter for learning. With the aforementioned improvements, the model is capable of adaptively adjusting itself in real-time tracking. The tracker has enhanced robustness and anti-jamming capability, even in complex scenarios. The proposed tracker makes the following contributions.

- 1) We employ a Gaussian function for denoising as a preprocessing step on the features. The denoised target feature values become more stable and exhibit improved balance and discriminability. Particularly in low-resolution (LR) image quality, the performance of the tracker is significantly enhanced.
- 2) Our model deepens its utilization of historical information by learning from the past continuous weighted response values, thereby reducing the impact of complex environmental disturbances on track. The tracker enhances its ability to resist fluctuations and increases the stability of tracking.
- 3) Our method, RCFL, and other existing trackers have undergone extensive experimental comparisons on the DTB70, UAV123@10FPS, and UAVDT datasets. The experimental results demonstrate that RCFL achieves the best tracking performance.

The subsequent sections of this article are outlined as follows. In Section II, we present a review of related works on correlation filter algorithms and deep-learning algorithms. Section III provides a detailed description of the model of our RCFL tracker. Extensive experimental analyses are presented in Section IV. Finally, we summarize the conclusion of our work in Section V.

## II. RELATED WORK

### A. Correlation Filters Methods

The significant achievements of correlation filter models in the past decade are largely attributed to the proposal of various heuristic methods. Minimum output sum of squared error (MOSSE) filter [24] is one of the pioneers who uses the least squares method to compute filter parameters. The kernelized correlation filters (KCFs) [25] propose the idea of using circulant matrices to convert convolution operations into dot products, thus greatly improving computational efficiency.

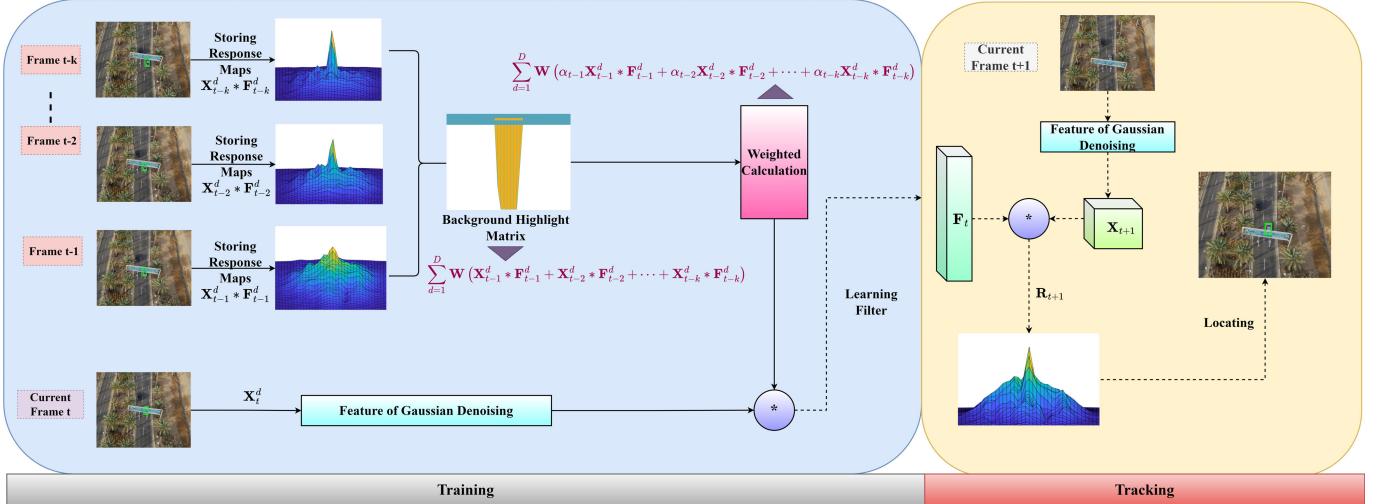


Fig. 2. Framework of the proposed tracker.

The success of KCF spurs further research and development of discriminative filtering algorithms. Subsequent researchers mainly focus on addressing issues such as boundary effects and filter degradation in correlation filter algorithms.

Danelljan et al. [17] introduce a spatial matrix to penalize background features. Dai et al. [26] endow spatial matrix with adaptivity to adjust them in each frame. Background-aware correlation filters (BACFs) [6] creatively use a spatial cropping matrix to separate the target region from the surrounding background area, which not only reduces computational complexity but also allows for learning from more samples. The generation method of cropping matrix can also be flexibly adapted to different scenes, making it highly scalable in application. Li et al. [18] proposed a temporal regularization model based on the SRDCF method to reduce filter degradation. LADCF [19] imposes spatial sparsity constraints on the filter. Dual regularized correlation filters (DRCFs) [27] and enhanced feature selection correlation filters (EFSCFs) [28] expand the target's feature channels, extract more types of features to learn the filter and increase the robustness of the tracker. Dirac-weighted cascading correlation filters (DWCCFs) [29] and channel attentional correlation filters (CACFs) [30] focus on the contribution of each channel feature to the model by adding weight regularization terms, effectively improving the tracking efficiency. However, this leads to an increase in the computational complexity of the model, resulting in suboptimal tracking speed.

Although traditional correlation filter trackers demonstrate respectable tracking accuracy, their performance on UAV datasets is not satisfactory, primarily due to their difficulty in effectively tracking fast-moving targets.

### B. Deep-Learning Methods

Deep correlation filter models and traditional correlation filter models exhibit consistency in their approaches to filter learning. Filter learning is an iterative process wherein the filter from the previous frame is convolved with the features extracted from the current frame. Subsequently, it adapts its parameters by minimizing the discrepancy between the

generated response and Gaussian labels. This enables the acquisition of the filter's parameters for the next frame. Filter learning serves the purpose of enhancing the robustness and precision of a tracker, facilitating the effective differentiation of the target from the background or other interfering elements during the tracking process. In contrast to traditional correlation filter algorithms, which typically employ manually crafted features such as HOG and CN for filter learning, deep learning achieves the extraction of a broader spectrum of features by increasing the depth of the network and adjusting its structure. For example, hierarchical convolutional features (HCFs) [31] use the deep network structure of very deep convolutional networks (VGG) [32] to extract the outputs of conv-3, conv-4, and conv-5 convolution layers as features to train correlation filters. Then, the three confidence maps obtained are fused with weights to obtain the final predicted position of the target. Features extracted from different convolution layers carry different types of information about the target. Features from shallower convolution layers embody more precise details about the target, such as contour information. Features from deeper convolution layers contain more semantic information about the target. In contrast, handcrafted features can only extract features at a single level and often cannot effectively represent the multilevel structure and features of objects. Efficient convolution operators (ECOs) [33] is a highly successful method that combines CNN with filtering techniques. It eliminates filters that have relatively minor contributions to the model and reduces redundant samples during model updates. This reduces the computational cost of feature extraction and model updates, thereby improving tracking speed. Deep hyperspectral kernelized correlation filters (DeepHKCFs) [34] combine KCF with deep CNN features and employ an effective regions-of-interest (ROI) mapping strategy to enhance the speed of deep feature extraction. It is applied in aerial tracking within the hyperspectral domain. Background-aware and spatial-temporal correlation filters (BSTCFs) [35] introduce a background constraint and spatial-temporal regularization and combine hand-crafted features and deep features extracted from pre-trained convolutional networks to enhance object appearance

representation. Unsupervised deep tracking (UDT) [36] is an unsupervised visual tracking method based on the Siamese correlation filter network, which differs from training on annotated tracking datasets. During the training stage, the model learns using a large-scale unlabeled video dataset to train CNNs. This method is capable of achieving effective forward and backward tracking predictions without complete and accurate labels.

In addition to the approach of combining CNN with traditional correlation filter models, another deep-learning model based on regression is typically used to construct a deep-learning framework for predicting the position and state of the target in the next frame. Action-decision networks (ADNets) [37] combines deep reinforcement learning and supervised learning to construct a deep network, which is pretrained on training sequences. During the tracking process, ADNet adapts the fine-tuning based on variations in the target and background to reduce computational complexity. Compared to previous deep trackers that rely on pretrained CNNs for extracting deep features, target-aware deep tracking (TADT) [38] incorporates a regression loss and a ranking loss to guide the generation of target-active and scale-sensitive features. In addition, TADT determines the importance of each convolutional filter by leveraging back-propagated gradients and selects target-aware features based on activations to represent the target, thereby enhancing the overall tracking accuracy of the tracker.

While deep trackers are capable of extracting deep features that capture more intricate details, training complex convolutional networks inevitably requires expensive GPUs to increase computational power and longer computation time. Hence, there is still a need to reduce the computation time of deep tracking models to meet the real-time tracking requirements of UAVs.

### C. UAV Trackers

In UAV tracking tests, the targets exhibit two prominent characteristics. First, the targets have a relatively small size. Second, the targets are in a state of fast motion (FM). Particularly in real-time tracking, the rapid variations in both the target and the background can easily lead to model degradation and tracking drift. Therefore, in recent years, several researchers have proposed dedicated strategies for DCF-based tracking frameworks tailored specifically for UAV targets. Xue et al. [39] combined semantic segmentation and shared feature detection modules to propose a semantic-aware related filter framework for UAV target tracking. Huang et al. [40] proposed a new method called the distortion-suppressed related filter (ARCF), which suppresses distortion during the detection process by limiting the variation rate of the response map generated during detection. Ye et al. [41] effectively improve the performance of UAV tracking and localization by adjusting the response value and channel weight distribution to adapt to target appearance changes and enhance discriminability.

Most of the target tracking methods for UAVs share a commonality: the model continuously learns from past historical information during model updating. Although these trackers achieve good tracking performance, their use of

historical information is typically limited to a single frame in the past. The utilization of historical information is not sufficient, as it is difficult to ensure that the information contained in a particular frame is accurate due to the stochastic nature of tracking. If the target undergoes a significant change in that frame, the learning process not only becomes ineffective but also has a detrimental impact on the tracking performance of the tracker. To address this potential issue, this article proposes the RCFL tracker.

## III. METHODOLOGY

The proposed RCFL tracker is based on the BACF tracker and incorporates improvements from two perspectives. From the first perspective, we observe that the quality of target features diminishes when video frames have low visual quality. To address this issue, we employ a Gaussian function to denoise the features, replacing the original features with the denoised ones as inputs. In the second perspective, we extend the tracker's ability to learn from historical information. In addition, we introduce an exponential smoothing model to construct weights for historical information, thereby constraining the extent of learning for the tracker.

The essence of historical information is the response values generated by past frames, where these responses are obtained through convolutional calculations between input features and filters. The learning of filters is also intrinsically linked to the input features. Consequently, the quality of these features plays a pivotal role in the overall tracking model. When Gaussian denoising is applied to the extracted features, the feature values become more stable, and the features within the target region exhibit more location and scale details. As a result, the learning of filters becomes more precise and effective. The response values obtained through convolutional calculations between features and filters also become of higher quality, becoming more representative and valuable for learning. In tandem, this progression perpetuates the iterative refinement of filters by leveraging historical information during the update phase, thus culminating in the reinforcement of the tracker's robustness. In summation, Gaussian denoising significantly contributes to the advancement of historical information learning, fostering a favorable impact on the process.

### A. Baseline Model

Before presenting our model, this section provides a brief overview of the baseline tracker, BACF, to better understand our approach. The model extracts multichannel features  $\mathbf{X}_t \in \mathbb{R}^{B \times N \times D}$ , where  $D$  denotes the total number of feature channels. Each feature channel has the same size of  $B \times N$ .  $\mathbf{Y} \in \mathbb{R}^{B \times N}$  is a preset class label defined by the Gaussian formula. The task of BACF is to minimize the following loss function to learn the filter  $\mathbf{F}_t \in \mathbb{R}^{B \times N \times D}$  corresponding to each feature channel:

$$E(\mathbf{F}) = \frac{1}{2} \left\| \sum_{d=1}^D \mathbf{F}_t^d * \mathbf{P} \mathbf{X}_t^d - \mathbf{Y} \right\|_2^2 + \frac{\lambda}{2} \sum_{d=1}^D \|\mathbf{F}_t^d\|_2^2 \quad (1)$$

where a binary matrix  $\mathbf{P}$  is utilized to crop the central region of the feature  $\mathbf{X}_t^d$ . The correlation operator is denoted by  $*$ . The regularization hyper-parameter is represented by  $\lambda$ .

### B. Gaussian Denoising

In the field of object tracking, the correlation filter algorithm is widely used to achieve efficient and accurate tracking. Common correlation filter algorithms use a feature extraction method that combines CN and HOG and directly inputs the extracted features into the DCF model framework for processing. However, when the video quality is low, the extracted features may contain noise, which affects the accuracy and stability of the tracker. To solve this problem, we propose to apply denoising techniques to the feature extraction stage of the correlation filter algorithm to improve the tracker's overall recognition ability. Denoising techniques can effectively reduce noise in the image, making the target's position and scale information more stable, thereby improving the performance and robustness of the tracker. The classical denoising techniques include wavelet-based denoising methods [42] and sparse coding-based denoising methods [43]. In this article, to ensure effective denoising and avoid excessive computational costs, we adopt a Gaussian denoising method [44].

From a mathematical modeling perspective, the Gaussian denoising method exhibits remarkable simplicity, relying solely on two parameters: the convolution kernel size and the standard deviation. In contrast, wavelet-based denoising methods necessitate the decomposition of an image into different frequency components using wavelet basis functions, while sparse coding-based denoising methods require the construction of a dictionary of basis functions for representing the image as a linear combination with sparse coefficients to suppress noise. Regarding solution methodologies, Gaussian denoising achieves denoising through linear filtering, implemented via convolution operations. This approach involves straightforward mathematical formulas, yielding the denoised result directly. In contrast, wavelet-based denoising methods involve determining appropriate thresholds to apply to wavelet coefficients, followed by an inverse transformation to reconstruct the denoised image. Similarly, sparse coding-based denoising methods initially encode the image using the learned dictionary and then employ iterative optimization algorithms to find sparse coefficients, which are subsequently used to reconstruct the denoised image. The key distinction lies in the computational complexity introduced by the more intricate mathematical operations of wavelet-based and sparse coding-based denoising methods.

Gaussian noise reduction is a common signal processing technique used to remove noise from data such as images and sounds, making the signals clearer. The basic idea is to replace the value of each pixel with a weighted average of the surrounding pixels, where the weighting coefficients follow a Gaussian distribution such that pixels farther away from the center pixel have smaller weights. The fundamental principle of Gaussian noise reduction lies in Gaussian filtering. Suppose  $\mathbf{X}_t^d \in \mathbb{R}^{B \times N}$  denotes the  $d$ th feature channel of size  $B \times N$  extracted from the center of the object in the  $t$ th frame. The mathematical principle of performing Gaussian denoising on the features can be expressed as follows:

$$g(x, y) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{(x-x_0)^2+(y-y_0)^2}{2\sigma^2}} * f(x, y) \quad (2)$$

where the feature value of the pixel  $(x, y)$  in the extracted object region is denoted by  $f(x, y)$ , and the feature value of the pixel after Gaussian filtering is denoted by  $g(x, y)$ .  $(x_0, y_0)$  represents the center of the Gaussian kernel,  $\sigma$  represents the standard deviation of the Gaussian kernel, and  $*$  represents convolution.

### C. Continuously Weighted Dynamic Response

Learning from historical information plays an important role in UAV target tracking. First, by learning the historical information of a target, it helps the tracker to memorize the target's previous appearance features and semantic information. During model updates, the tracker can quickly construct a target model and gradually improve its performance. Second, targets in real-world scenarios often encounter various changes, such as lighting variations, partial occlusions (POs), and deformations. By learning from historical information, the UAV can better adapt to these challenges, ensuring the stability and accuracy of target tracking. Therefore, learning historical information provides the necessary context and prior knowledge for UAV target tracking, ultimately enhancing the performance and effectiveness of tracking algorithms.

Taking the ARCF [40] as an example, the regularization term in (3) of the model incorporates the learning of the previous frame's response values

$$\left\| \sum_{d=1}^D \mathbf{X}_t^d * \mathbf{F}_t^d - \sum_{d=1}^D \mathbf{X}_{t-1}^d * \mathbf{F}_{t-1}^d \right\|_2^2. \quad (3)$$

The improvements introduced by ARCF have mitigated the potential degradation and distortion of filters. However, the utilization of historical information in ARCF has inherent limitations. This is due to the stochastic nature of both target and environmental changes during target motion. For instance, if the target becomes occluded in the previous frame, the generated response map would contain erroneous content. Consequently, the filter learns incorrect contextual information and prior knowledge, which not only fails to enhance accuracy and robustness but also diminishes discriminability. Therefore, in this section, as shown in Fig. 3, we propose a method for learning the values of past continuously dynamic responses. The formula can be represented as follows:

$$\sum_{d=1}^D \mathbf{X}_{t-1}^d * \mathbf{F}_{t-1}^d + \mathbf{X}_{t-2}^d * \mathbf{F}_{t-2}^d + \cdots + \mathbf{X}_{t-k}^d * \mathbf{F}_{t-k}^d \quad (4)$$

where  $\mathbf{X}_{t-k}^d$  and  $\mathbf{F}_{t-k}^d$  represent the  $d$ th channel feature and filter from the previous  $k$ th frame, respectively.

In addition, we employ a time-weighted learning method to reduce the probability of future filter distortion and prevent overfitting. Specifically, we enable the model to learn the response values from the past  $k$  frames and utilize a strategy of allocating weights based on temporal proximity to balance the utilization of these past response values. This weight allocation mechanism ensures that response values closer to the current frame are assigned higher weights, allowing the filter to focus more on the most relevant and representative response values. Moreover, by constraining the model's learning of excessive past response values, we effectively mitigate

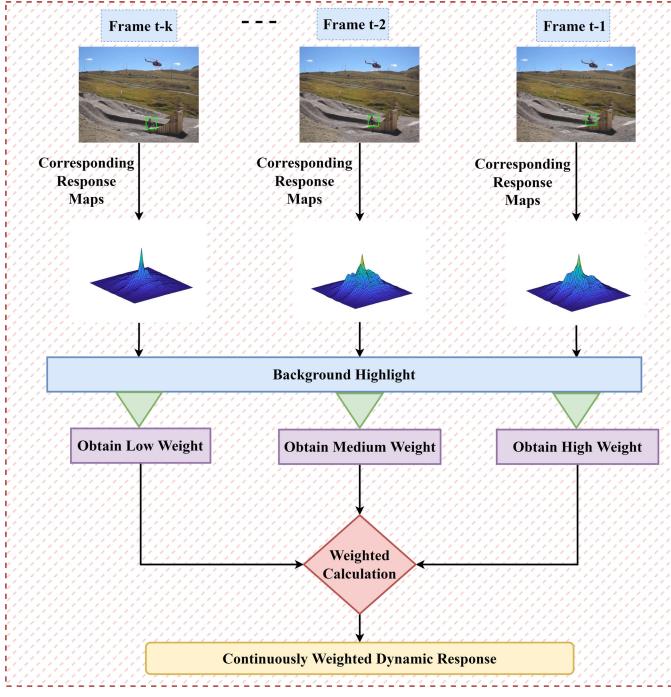


Fig. 3. Historical information utilization.

the risk of overfitting, resulting in a more robust model with enhanced generalization capabilities. By comprehensively considering time weighting and overfitting control, we can achieve more accurate and stable results in target tracking. The time-weighted learning can be formulated as

$$\mathbf{M}^d = \sum_{d=1}^D \mathbf{WP}(\alpha_{t-1} \mathbf{X}_{t-1}^d * \mathbf{F}_{t-1}^d + \alpha_{t-2} \mathbf{X}_{t-2}^d * \mathbf{F}_{t-2}^d + \dots + \alpha_{t-k} \mathbf{X}_{t-k}^d * \mathbf{F}_{t-k}^d) \quad (5)$$

where  $\alpha_{t-k} = \lambda^k$ , the matrix  $\mathbf{W}$  represents the binary matrix of background highlight, where a stronger penalty is applied to the target region. This approach effectively highlights the transformation of the background by emphasizing the differences between the target and the background.

Furthermore, the estimation of the weight  $\alpha_{t-k}$  can be approximated by solving the following equation for  $\lambda$

$$\lambda^1 + \lambda^2 + \dots + \lambda^k = 1. \quad (6)$$

Finally, we propose the continuously weighted dynamic response regularization as follows:

$$\left\| \sum_{d=1}^D \mathbf{X}_t^d * \mathbf{PF}_t^d - \mathbf{M}^d \right\|_2^2. \quad (7)$$

Based on the improvements discussed in the previous sections, we propose our RCFL model

$$\begin{aligned} E(\mathbf{F}) = & \frac{1}{2} \left\| \sum_{d=1}^D \mathbf{F}^d * \mathbf{PX}_t^d - \mathbf{Y} \right\|_2^2 + \frac{\zeta}{2} \sum_{d=1}^D \|\mathbf{F}^d\|_2^2 \\ & + \frac{\mu}{2} \left\| \sum_{d=1}^D \mathbf{F}^d * \mathbf{PX}_t^d - \mathbf{M}^d \right\|_2^2 \end{aligned} \quad (8)$$

where  $\mathbf{X}_t^d$  represents the  $d$ th feature channel of the  $t$ th frame after Gaussian denoising.  $\zeta$  and  $\mu$  represent hyperparameters.

#### D. Optimization of Model

To simplify calculations, we introduce the variable  $\mathbf{Z}^d = \mathbf{PX}_t^d$ . From this, (8) can be converted to

$$\begin{aligned} E(\mathbf{F}) = & \frac{1}{2} \left\| \sum_{d=1}^D \mathbf{F}^d * \mathbf{Z}^d - \mathbf{Y} \right\|_2^2 + \frac{\zeta}{2} \sum_{d=1}^D \|\mathbf{F}^d\|_2^2 \\ & + \frac{\mu}{2} \left\| \sum_{d=1}^D \mathbf{F}^d * \mathbf{Z}^d - \mathbf{M}^d \right\|_2^2. \end{aligned} \quad (9)$$

We introduce an auxiliary variable  $\mathbf{G} \in \mathbb{R}^{B \times N \times D}$ , such that it satisfies the condition  $\mathbf{F} = \mathbf{G}$ . By doing so, we can formulate the augmented Lagrangian expression of (9) as follows:

$$\begin{aligned} E(\mathbf{F}) = & \frac{1}{2} \left\| \sum_{d=1}^D \mathbf{F}^d * \mathbf{Z}^d - \mathbf{Y} \right\|_2^2 + \frac{\zeta}{2} \sum_{d=1}^D \|\mathbf{G}^d\|_2^2 \\ & + \frac{\mu}{2} \left\| \sum_{d=1}^D \mathbf{F}^d * \mathbf{Z}^d - \mathbf{M}^d \right\|_2^2 + \frac{\gamma}{2} \sum_{d=1}^D \|\mathbf{F}^d - \mathbf{G}^d\|_2^2 \\ & + \sum_{d=1}^D \text{Tr}((\mathbf{F}^d - \mathbf{G}^d)^T \hat{\mathbf{S}}^d). \end{aligned} \quad (10)$$

Let  $\mathbb{S} = [\mathbf{S}^1, \mathbf{S}^2, \dots, \mathbf{S}^D] \in \mathbb{R}^{B \times N \times D}$  represent the Lagrange multipliers. Furthermore,  $\gamma$  denotes the step size employed in the augmented Lagrangian formulation.

In order to reduce the computational complexity, and by utilizing Parseval's theorem, we assume  $\mathbb{H} = (1/\gamma)\mathbb{S}$ . Subsequently, (10) can be rewritten as the following form:

$$\begin{aligned} E(\hat{\mathbf{F}}) = & \frac{1}{2} \left\| \sum_{d=1}^D \hat{\mathbf{F}}^d \cdot \hat{\mathbf{Z}}^d - \hat{\mathbf{Y}} \right\|_2^2 + \frac{\zeta}{2} \sum_{d=1}^D \|\hat{\mathbf{G}}^d\|_2^2 \\ & + \frac{\mu}{2} \left\| \sum_{d=1}^D \hat{\mathbf{F}}^d \cdot \hat{\mathbf{Z}}^d - \hat{\mathbf{M}}^d \right\|_2^2 \\ & + \frac{\gamma}{2} \sum_{d=1}^D \|\hat{\mathbf{F}}^d - \hat{\mathbf{G}}^d + \hat{\mathbf{H}}^d\|_2^2. \end{aligned} \quad (11)$$

By employing the ADMM algorithm [45], the optimization problem can be decomposed into a set of subproblems

$$\begin{aligned} \hat{\mathbf{F}}^{(i+1)} = & \underset{\hat{\mathbf{F}}}{\operatorname{argmin}} \frac{1}{2} \left\| \sum_{d=1}^D \hat{\mathbf{F}}^d \cdot \hat{\mathbf{Z}}^d - \hat{\mathbf{Y}} \right\|_2^2 \\ & + \frac{\mu}{2} \left\| \sum_{d=1}^D \hat{\mathbf{F}}^d \cdot \hat{\mathbf{Z}}^d - \hat{\mathbf{M}}^d \right\|_2^2 \\ & + \frac{\gamma}{2} \sum_{d=1}^D \|\hat{\mathbf{F}}^d - \hat{\mathbf{G}}^d + \hat{\mathbf{H}}^d\|_2^2 \end{aligned} \quad (12)$$

$$\hat{\mathbf{G}}^{(i+1)} = \underset{\hat{\mathbf{G}}}{\operatorname{argmin}} \frac{\zeta}{2} \sum_{d=1}^D \|\hat{\mathbf{G}}^d\|_2^2 + \frac{\gamma}{2} \sum_{d=1}^D \|\hat{\mathbf{F}}^d - \hat{\mathbf{G}}^d + \hat{\mathbf{H}}^d\|_2^2 \quad (13)$$

$$\hat{\mathbf{H}}^{(i+1)} = \hat{\mathbf{H}}^{(i)} + \hat{\mathbf{F}}^{(i+1)} - \hat{\mathbf{G}}^{(i+1)}. \quad (14)$$

Solving  $F$ :

$$J(\hat{\mathbf{F}}) = \frac{1}{2} \left\| \sum_{d=1}^D \hat{\mathbf{F}}^d \cdot \hat{\mathbf{Z}}^d - \hat{\mathbf{Y}} \right\|_2^2 + \frac{\mu}{2} \left\| \sum_{d=1}^D \hat{\mathbf{F}}^d \cdot \hat{\mathbf{Z}}^d - \hat{\mathbf{M}}^d \right\|_2^2 + \frac{\gamma}{2} \sum_{d=1}^D \left\| \hat{\mathbf{F}}^d - \hat{\mathbf{G}}^d + \hat{\mathbf{H}}^d \right\|_2^2. \quad (15)$$

By setting the derivative of (15) with respect to the variable  $\hat{\mathbf{F}}$  to zero, we can obtain the closed-form solution as follows:

$$[(1 + \mu)\hat{\mathbf{Z}}^\top \hat{\mathbf{Z}} + \gamma \mathbf{I}] \hat{\mathbf{F}} = \mu \hat{\mathbf{Z}}^\top \hat{\mathbf{M}}^\top + \hat{\mathbf{Z}}^\top \hat{\mathbf{Y}} + \gamma (\hat{\mathbf{H}} - \hat{\mathbf{G}}). \quad (16)$$

Subsequently, the closed-form solution for the variable  $\hat{\mathbf{F}}$  can be expedited by employing the Sherman–Morrison formula [46]

$$\hat{\mathbf{F}}^{(i+1)} = \frac{1}{\gamma} \left( I - \frac{\hat{\mathbf{Z}} \hat{\mathbf{Z}}^\top}{\frac{\gamma}{1+\mu} + \hat{\mathbf{Z}}^\top \hat{\mathbf{Z}}} \right) \mathbf{q} \quad (17)$$

where  $\mathbf{q} = \mu \hat{\mathbf{Z}}^\top \hat{\mathbf{M}}^\top + \hat{\mathbf{Z}}^\top \hat{\mathbf{Y}} + \gamma (\hat{\mathbf{H}} - \hat{\mathbf{G}})$ .

Solving  $G$ :

$$J(\hat{\mathbf{G}}) = \frac{\zeta}{2} \sum_{d=1}^D \left\| \hat{\mathbf{G}}^d \right\|_2^2 + \frac{\gamma}{2} \sum_{d=1}^D \left\| \hat{\mathbf{F}}^d - \hat{\mathbf{G}}^d + \hat{\mathbf{H}}^d \right\|_2^2. \quad (18)$$

By setting the derivative of (18) with respect to the variable  $\hat{\mathbf{G}}$  to zero, we can obtain the closed-form solution as follows:

$$\hat{\mathbf{G}}^{(i+1)} = \frac{\gamma(\hat{\mathbf{F}} + \hat{\mathbf{H}})}{(\zeta + \gamma)\mathbf{I}}. \quad (19)$$

Solving  $H$ :

$$\hat{\mathbf{H}}^{(i+1)} = \hat{\mathbf{H}}^{(i)} + \gamma(\hat{\mathbf{F}}^{(i+1)} - \hat{\mathbf{G}}^{(i+1)}). \quad (20)$$

Updating Stepsize:

$$\gamma^{(i+1)} = \min(\gamma^{\max}, \beta \gamma^{(i)}) \quad (21)$$

where  $\gamma^{\max}$  represents the maximum value that can be assigned to the parameter  $\gamma$ . In addition, the scale factor is denoted by  $\beta$ .

Model Update:

$$\hat{\mathbf{X}}_t^{\text{model}} = (1 - \eta) \hat{\mathbf{X}}_{t-1}^{\text{model}} + \eta \hat{\mathbf{X}}_t \quad (22)$$

where  $\eta$  denotes the learning rate.

The alternating optimization details of ADMM are shown in Algorithm 1.

#### IV. EXPERIMENTS

##### A. Datasets

In this section, we present a comprehensive evaluation of our RCFL tracker by comparing it with state-of-the-art trackers on multiple challenging datasets. The goal of these experiments is to demonstrate the effectiveness and superiority of our tracker in the field of UAV tracking.

To evaluate the performance of our tracker, we employ several authoritative drone benchmarks, namely, DTB70, UAV123@10fps, and UAVDT. These benchmarks have been widely used in the research community and provide a diverse

##### Algorithm 1 RCFL Algorithm

- 
- Record: The response maps of previous K frames.  
Initialization: values  $\mathbf{G}^{(0)}, \mathbf{H}^{(0)}$ .  
1: Input Feature  $\mathbf{X}_t$ ;  
2: Apply Gaussian denoising to the features  $\mathbf{X}_t$ ;  
3: Construct the Gaussian shaped label  $\mathbf{Y}$ ;  
4: Generate the background highlight matrix  $\mathbf{W}$ ;  
5: Compute weights;  
6: Obtain continuously weighted dynamic response  $\mathbf{M}^d$ ;  
7: iteration = 1;  
8: Update  $\hat{\mathbf{F}}$  via Eq. (17);  
9: Update  $\mathbf{G}$  via Eq. (19);  
10: Update  $\mathbf{H}$  via Eq. (20);  
11: repeat step 7-step 10  
12: Until converged  
13: Output correlation filters  $\mathbf{F}$  by applying Inverse Discrete Fourier Transform(IDFT).  
14: Target localization.  
15: Model Update.
- 

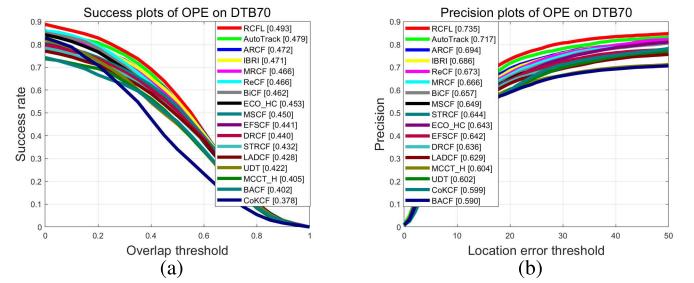


Fig. 4. (a) Success plots and (b) precision plots of the proposed RCFL and other trackers on the DTB70 database.

range of challenging scenarios for evaluating drone tracking algorithms.

DTB70 is a highly diverse video dataset consisting of more than 70 drone videos. This dataset covers a wide range of challenging scenarios, including different weather conditions, lighting variations, and target appearances. UAV123@10fps is a dataset that consists of 123 high-definition video sequences captured from a low-altitude aerial view. These sequences are specifically designed to challenge the tracking algorithms with various factors such as occlusions, FM, SVs, and background clutter (BC). Finally, we evaluate our tracker on the UAVDT dataset which contains 100 video sequences with a total of approximately 20 320 video frames.

##### B. Implementation Detail

The software platform used to evaluate the RCFL tracker in this study's benchmark is MATLAB 2021a, running on a computer equipped with a 12th Gen Intel<sup>®</sup> Core<sup>™</sup> i7-12700H Central Processing Unit (CPU) operating at 2.70 GHz. The evaluation follows the one-pass evaluation (OPE) standard to measure the tracking precision and success rate of RCFL.

The parameter settings have a crucial impact on the performance of the model. In this article, we set the total number of

<sup>1</sup>Registered trademark.

<sup>2</sup>Trademarked.

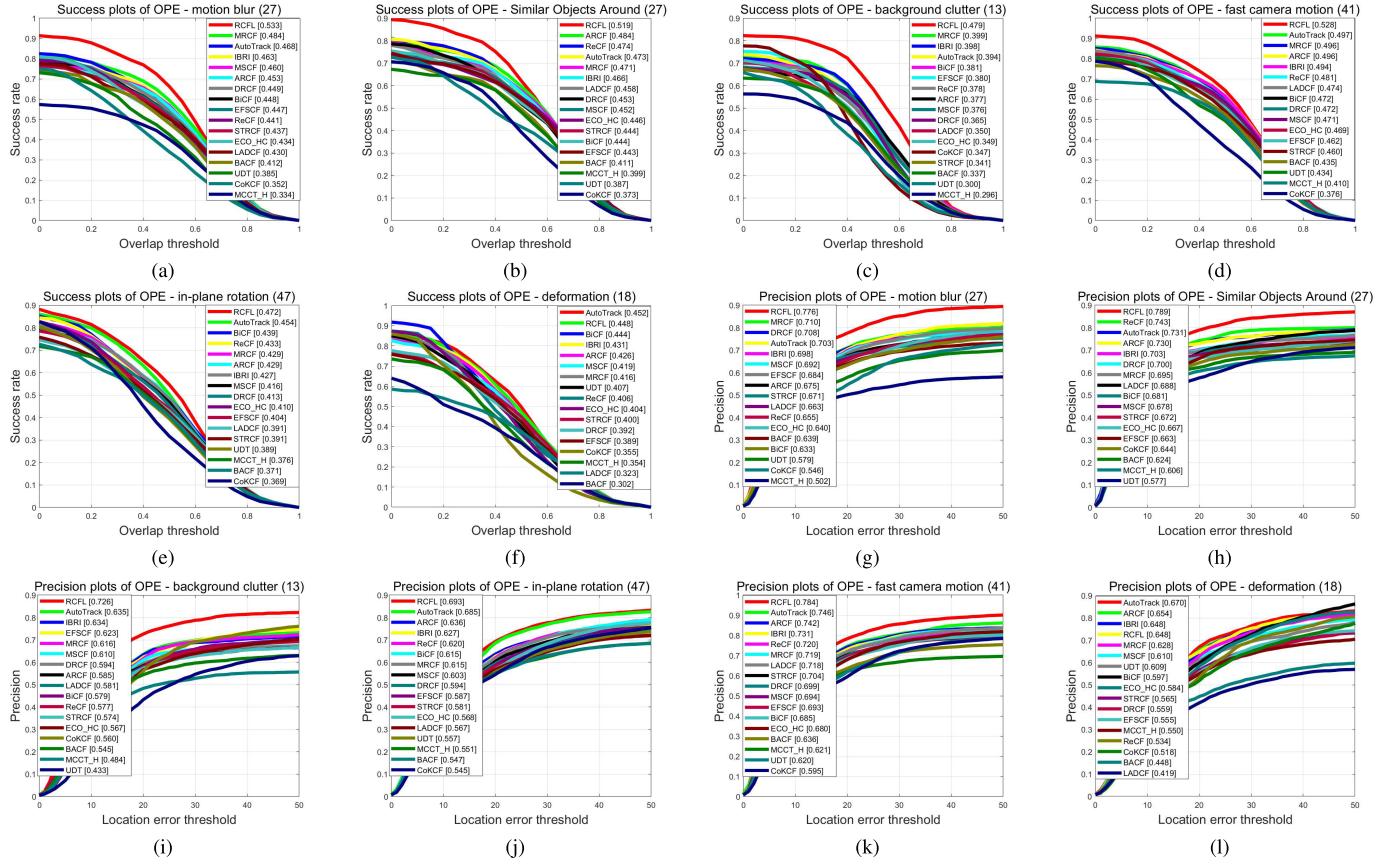


Fig. 5. Comparisons between RCFL and the state-of-the-art trackers specifically focusing on challenging attributes. (a)–(f) Overlap threshold. (g)–(l) Location error threshold.

iterations for the AMDD algorithm to 2. The total number of past frames  $k$  considered for learning is set to 3. The initial step size, denoted by  $\gamma$ , is set to 0.001, while  $\beta$  is set to 10, the maximum step size,  $\gamma^{\max}$ , is set to 10 000. The parameter  $\mu$  is set to 1, and the learning rate  $\eta$  is set to 0.064. The size of the Gaussian kernel is set to 3, and the parameter  $\sigma$  is set to 0.3.

### C. Quantitative Analysis of Experimental Results

1) *Experimental Results on the DTB70*: On the DTB70 dataset, we compare our proposed RCFL tracker with 16 other state-of-the-art trackers to demonstrate the effectiveness of our improvements. The set of trackers included in the comparison comprises multiregularized correlation filter (MRCF) [41], automatic spatio-temporal regularization tracking (AutoTrack) [47], disruptor-aware interval-based response inconsistency (IBRI) [48], mutation sensitive correlation filter (MSCF) [49], ARCF [40], DRCF [27], bidirectional incongruity-aware correlation filter (BiCF) [50], EFSCF [28], response reasoning for correlation filters (ReCFs) [51], STRCF [18], ECO [33], LADCF [19], BACF [6], UDT [36], co-trained kernelized correlation filters (CokCFs) [52], and multicue correlation filters tracking (MCCT) [53].

The experimental results are shown in Fig. 4. The proposed method, RCFL, demonstrates the best performance in terms of tracking success rate and precise rate. In the comparison of tracking success rate, RCFL outperforms the second-ranked

tracker, AutoTrack, by 1.4% points. Moreover, compared to the baseline method BACF, RCFL achieves a substantial improvement of 9.1% points in tracking success rate. In the comparison of tracking precise rate, RCFL surpasses AutoTrack by 1.8% points and outperforms the baseline method by 14.5% points, indicating a significant enhancement in tracking performance.

In addition, to provide a more detailed evaluation of our proposed method, we compare RCFL with the other 16 algorithms across challenging attributes in tracking performance. As shown in Fig. 5, across six specific attributes including motion blur (MB), similar objects around (SOA), BC, fast camera motion (FCM), in-plane rotation (IPR), and deformation (DEF), RCFL outperforms the others in terms of tracking performance in all attributes except DEF.

2) *Experimental Results on the UAV123@10fps*: On the UAV123@10fps dataset, we compare the proposed RCFL tracker with 16 other state-of-the-art trackers to demonstrate the effectiveness of our proposed improvements. The set of trackers included in the comparison comprises MRCF [41], AutoTrack [47], IBRI [48], MSCF [49], ARCF [40], DRCF [27], BiCF [50], EFSCF [28], ReCF [51], STRCF [18], ECO [33], LADCF [19], BACF [6], UDT [36], CokCF [52], and MCCT [53].

The experimental results are shown in Fig. 6. Our method, RCFL, demonstrates the best performance in terms of tracking success rate and precise rate. In the comparison of tracking success rate, RCFL outperforms the second-ranked tracker,

TABLE I

BASED PRECISION PLOT OF THE PROPOSED RCFL AND 16 TRACKERS IN DIFFERENT ATTRIBUTES ON THE UAV123@10FPS DATABASE

	SV	ARC	LR	FM	FO	PO	OV	BC	IV	VC	CM	SO
MRCF	0.623	0.575	0.553	0.510	0.446	<b>0.594</b>	0.544	0.460	0.541	0.577	0.622	0.675
AutoTrack	<b>0.629</b>	<b>0.598</b>	0.532	<b>0.525</b>	0.444	0.584	<b>0.554</b>	0.502	0.550	<b>0.588</b>	<b>0.647</b>	0.664
IBRI	<b>0.631</b>	<b>0.592</b>	<b>0.559</b>	<b>0.547</b>	0.427	<b>0.595</b>	0.539	0.443	0.548	<b>0.590</b>	0.623	0.656
MSCF	0.606	0.567	0.521	0.517	0.415	0.582	<b>0.545</b>	0.431	0.541	0.578	0.598	0.650
ARCF	0.623	0.580	<b>0.561</b>	0.516	0.428	0.582	0.513	0.445	0.552	0.573	0.610	0.657
DRCF	0.604	0.553	0.522	0.520	0.456	0.591	0.533	0.486	0.556	0.566	0.610	0.620
BiCF	0.619	0.578	0.534	0.484	<b>0.469</b>	0.591	0.533	<b>0.515</b>	<b>0.581</b>	0.581	0.625	<b>0.705</b>
EFSCF	0.581	0.532	0.515	0.479	0.408	0.560	0.518	0.475	0.494	0.526	0.592	0.624
ReCF	0.621	0.584	0.543	0.512	<b>0.465</b>	0.590	0.526	<b>0.514</b>	<b>0.592</b>	0.581	0.632	<b>0.687</b>
STRCF	0.580	0.524	0.509	0.488	0.426	0.559	0.523	0.477	0.493	0.537	0.502	0.630
ECO	0.587	0.558	0.527	0.487	0.454	0.556	0.522	0.511	0.507	0.548	0.609	0.637
LADCF	0.608	0.561	0.549	0.503	0.432	0.564	0.537	0.443	0.481	0.548	<b>0.637</b>	0.653
UDT	0.536	0.486	0.481	0.448	0.399	0.498	0.486	0.378	0.411	0.453	0.527	0.611
CoKCF	0.559	0.522	0.480	0.408	<b>0.462</b>	0.559	0.483	0.485	0.522	0.532	0.569	0.627
MCCT	0.547	0.493	0.455	0.361	0.421	0.542	0.493	0.469	0.477	0.484	0.544	0.627
BACF	0.525	0.478	0.431	0.407	0.336	0.467	0.421	0.425	0.430	0.491	0.532	0.605
RCFL	<b>0.647</b> ↑	<b>0.624</b> ↑	<b>0.573</b> ↑	<b>0.552</b> ↑	0.444 ↑	<b>0.609</b> ↑	<b>0.556</b> ↑	<b>0.533</b> ↑	<b>0.593</b> ↑	<b>0.607</b> ↑	<b>0.655</b> ↑	<b>0.705</b> ↑

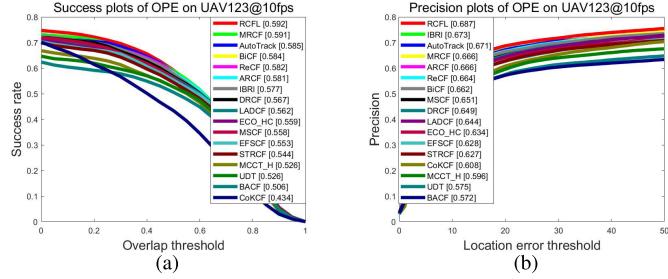


Fig. 6. (a) Success plots and (b) precision plots of the proposed RCFL and other trackers on the UAV123@10fps database.

MRCF, by 0.1% points. Moreover, compared to the baseline method BACF, RCFL achieves a substantial improvement of 8.6% points in tracking success rate. In the comparison of tracking precise rate, RCFL surpasses IBRI by 1.4% points and outperforms the baseline method by 11.5% points, indicating a significant enhancement in tracking performance.

To further evaluate the robustness of our proposed model, we compare RCFL with the other trackers across 12 challenging attributes, including SV, aspect ratio change (ARC), LR, FM, full occlusion (FO), PO, out-of-view (OV), BC, illumination variation (IV), VC, camera motion (CM), and similar object (SO). In Table I, we show the experimental results, where the top three methods in each attribute are color-coded: red, green, and blue. An upward arrow (↑) signifies superior performance compared to the baseline. Except for the FO attribute, RCFL achieves the top performance in terms of tracking accuracy among the other 11 attributes. These experimental results strongly demonstrate the high robustness of our RCFL model, highlighting its adaptability to various complex tracking environments.

3) *Comparison With Deep-Based Trackers*: To compare the performance of our method with both correlation filter-based and deep-learning-based approaches in UAV tracking, we evaluate it on the UAVDT dataset against the state-of-the-art deep-learning trackers. These deep-learning methods include HCF [31], fully convolutional networks tracking (FCNT) [54], hedged deep tracking (HDT) [55], continuous convolution operators for visual tracking (C-COT) [56], end-to-end representation learning for correlation filter (CFNet) [57], ADNet [37], multitask correlation particle filter (MCPF) [58], integrating boundary and center correlation filters (IBC-CFs) [59], CoKCF [52], DeepSTRCF [18], MCCT [53], ASRCF [26], TADT [38], UDT [36], UDT+ [36], correlation filters with weighted convolution responses (CFWCRs) [60], online filtering training samples (MN\_ECO) [61], MN\_MDNet [61], real-time correlation tracking via joint model compression and transfer (fECO) [62], unsupervised deep representation tracking (LUDT) [63], LUDT+ [63], BSTCF [35], multiexpert visual tracking (MEVT) [64], scale equivariance improves Siamese tracking (SE-SiamFC) [65], learning spatio-temporal transformer for visual tracking (STARK-ST101) [66], and robust visual tracking using a very deep generative model (RTDG) [67].

In Table II, we present our experimental findings, with the top three methods for each attribute highlighted in color: red, green, and blue. The experimental results demonstrate that RCFL surpasses the second-ranked tracker, RTDG, by a margin of 0.3% points in terms of precision score. While RCFL's ranking in terms of tracking success rate is not in the first place, it's important to note that our approach relies solely on CPU computation and makes use of manually designed features. In contrast, deep-learning methods require expensive

TABLE II  
COMPARISON OF PERFORMANCE ON THE UAVDT BENCHMARK  
BETWEEN RCFL AND OTHER EXCELLENT DEEP TRACKERS

Tracker	Venue	Prec	Succ	FPS
HCFT	15'ICCV	0.602	0.355	19.0
FCNT	15'ICCV	0.656	0.245	3.2
HDT	16'CVPR	0.596	0.303	9.0
C-COT	16'ECCV	0.656	0.406	1.1
CFNet	17'CVPR	0.681	0.429	44.7
ADNet	17'CVPR	0.683	0.429	7.5
MCPF	17'CVPR	0.675	0.403	0.6
IBCCF	17'ICCVW	0.603	0.389	3.0
CFWCR	17'ICCVW	0.691	0.435	9.5
CoKCF	17'PR	0.605	0.319	20.1
DeepSTRCF	18'CVPR	0.667	0.437	6.8
MCCT	18'CVPR	0.671	0.437	7.9
ASRCF	19'CVPR	0.700	0.437	22.2
TADT	19'CVPR	0.677	0.431	32.3
UDT	19'CVPR	0.674	0.442	73.3
UDT+	19'CVPR	0.696	0.415	56.9
MN_ECO	20'ACM	0.691	0.435	30.6
MN_MDN	20'ACM	0.672	0.440	30.6
fECHO	20'TIP	0.699	0.415	20.6
LUDT	21'JJCVC	0.631	0.418	78.8
LUDT+	21'IJCV	0.701	0.406	59.4
MEVT	21'IS	0.691	0.448	3
SE-SiamFC	21'WACA	0.626	0.363	5.6
STARK-ST101	21'ICCV	0.704	0.469	37.9
BSTCF	23'AI	0.685	0.441	19
RTDG	23'JBD	0.728	0.458	3.2
RCFL	Ours	0.731	0.451	77.5

GPUs for extensive pretraining. However, despite this gap, RCFL not only maintains excellent tracking performance but also excels in tracking speed, achieving a high-speed tracking rate of 77.5 frames/s, meeting the requirements for real-time tracking. In contrast, most deep-learning tracking methods struggle with speed due to the significant time required for model training, rendering them suboptimal in speed performance and often inadequate for real-time tracking demands.

4) *Ablation Study*: In this section, we evaluate whether Gaussian denoising improves the tracking performance of the model in terms of accuracy rate, success rate, and various attributes. We compare three different methods using the DTB70 dataset for evaluation. The Baseline refers to the BACF method, Baseline<sub>wr</sub> represents the baseline method with the addition of a continuously weighted response regularization term, and Baseline<sub>wr+gd</sub> includes Gaussian denoising of the features along with the previous enhancements.

According to Table III, Baseline<sub>wr</sub> shows a decrease of 1.1% in tracking precision and 0.5% in success rate compared to Baseline<sub>wr+gd</sub>. In addition, applying Gaussian denoising to the features leads to improvements in the success rates of the tracker across six attributes: occlusion (OCC), FCM, IPR, BC, SOA, and MB. The tracking results presented in Table III provide evidence for the effectiveness of the Gaussian denoising approach.

5) *Tracking Analysis*: To investigate the performance of the proposed method in real-time tracking at each frame, we compare RCFL with three representative methods: BACF, ARCF, and ReCF.

In the video sequence Car2\_1 shown in Fig. 7(a), the UAV camera is positioned far from the target and undergoes rapid VCs. During subsequent tracking, BACF, ARCF, and

ReCF gradually lose track of the target. Fig. 8 shows the real-time tracking accuracy. We can observe that despite a certain decrease in tracking accuracy during the rapid VCs, RCFL maintains a stable performance and consistently locks onto the target.

In the video sequence ChasingDrones\_1 shown in Fig. 7(b), a UAV is tracking another UAV in the air. The target exhibits fast and erratic movements while being relatively small in size. In addition, the camera undergoes a 360° rotation. Due to the high speed of the target, tracking becomes particularly challenging. However, RCFL effectively leverages historical information by learning from past response maps. This allows it to achieve precise tracking performance without relying on appropriate learning rates to update target features. In terms of real-time tracking accuracy, as shown in Fig. 9, RCFL demonstrates consistent robustness compared to BACF, which experiences two instances of target loss.

In the video sequence Gull\_1 shown in Fig. 7(c), the target's distinctive appearance is not clearly defined, and its semantic information is similar to the surrounding environment. These challenges introduce interference in the tracking process. In terms of real-time tracking accuracy, as shown in Fig. 10, both BACF and RCFL exhibit significant fluctuations during the first half of the tracking. However, RCFL maintains its ability to track the target without losing it, indicating its stronger adaptability.

In the video sequence RcCar6\_1 shown in Fig. 7(d), starting from the tracking live view in the tenth frame, we can clearly observe that the target undergoes drifting and becomes significantly blurred. In such circumstances, it becomes challenging to extract effective features based on the target, and the subsequent tracking using conventional filters is prone to distortion and failure. However, RCFL succeeds in tracking the target. In terms of real-time tracking accuracy, as shown in Fig. 11, when the target encounters the aforementioned conditions, BACF experiences a sharp decline in tracking accuracy, reaching 0. On the other hand, RCFL's tracking accuracy also undergoes a significant decrease but progressively recovers in the subsequent tracking frames. This indicates the capability of RCFL to successfully track targets in complex and challenging environments.

#### D. Parameter Analysis

In this section, based on the DTB70 dataset, we perform a sensitivity analysis on two critical parameters of the model: the parameter  $\sigma$  affecting the effectiveness of Gaussian denoising, and the parameter  $\mu$  governing the learning of the continuously weighted dynamic response.

As shown in Fig. 12, a significant correlation can be observed between the performance of the tracker and the parameter  $\sigma$ , indicating a notable sensitivity of the tracker's performance to this parameter. However, there is also a discernible pattern within this correlation. Specifically, when the parameter  $\sigma$  increases within the range from 0.1 to 0.3, the tracker's performance demonstrates a continuous improvement trend, reaching its optimal performance at the  $\sigma$  value of 0.3. As the parameter  $\sigma$  continues to increase beyond this point,

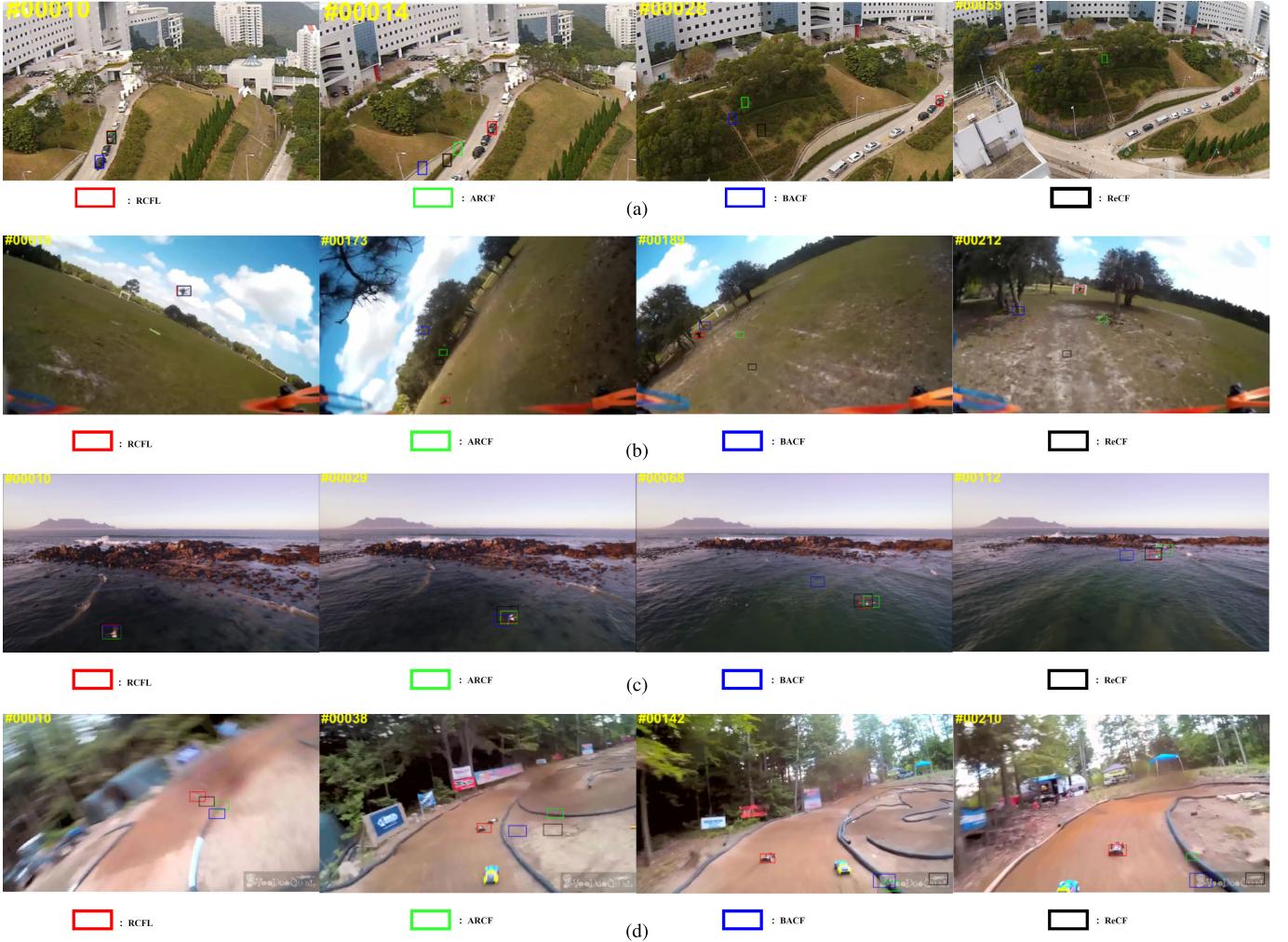


Fig. 7. Performance of different algorithms in real-time tracking. (a) Car2\_1. (b) ChasingDrones\_1. (c) Gull1\_1. (d) ReCar6\_1.

TABLE III  
ABLATION STUDY OF RESULTS FROM THREE METHODS BASED ON BASELINE

Tracker	Prec.	Succ.	OCC	FCM	IPR	BC	SOA	MB
Baseline	0.590	0.402	0.348	0.435	0.371	0.337	0.411	0.412
Baseline <sub>wr</sub>	0.724	0.488	0.397	0.506	0.466	0.445	0.495	0.503
Baseline <sub>wr+gd</sub>	0.735↑	0.493↑	0.422↑	0.528↑	0.472↑	0.479↑	0.519↑	0.533↑



Fig. 8. Real-time accuracy of the tracking video sequence Car2\_1.

the tracker's performance starts to exhibit a noticeable decline. This can be attributed to the fact that larger  $\sigma$  values lead to increased intensity of Gaussian noise, resulting in greater blurring of features.

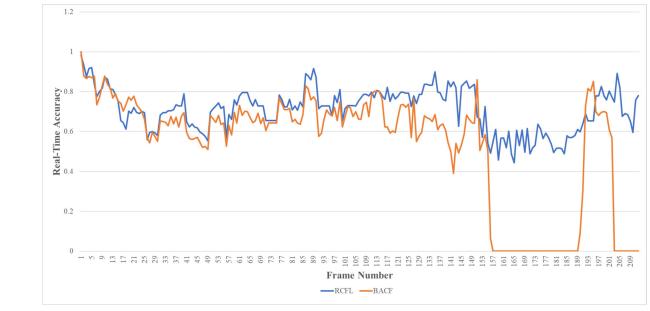


Fig. 9. Real-time accuracy of the tracking video sequence ChasingDrones\_1.

Hence, to ensure the stability and accuracy of the tracker's performance, we avoid setting the parameter  $\sigma$  to excessively large values. As a result, the adjustment of parameter  $\sigma$  only needs to be confined to a smaller range for fine-tuning, thus

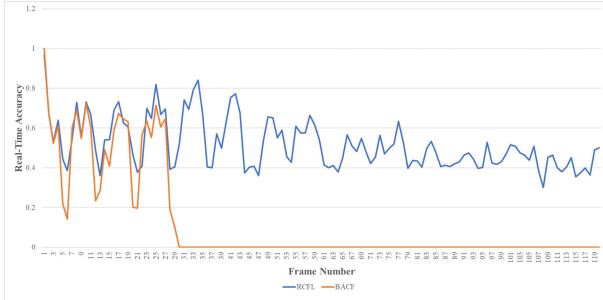


Fig. 10. Real-time accuracy of the tracking video sequence Gull\_1.

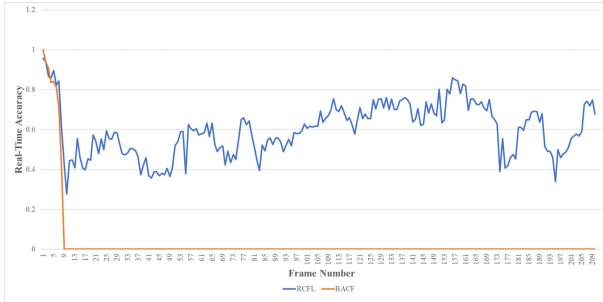


Fig. 11. Real-time accuracy of the tracking video sequence RcCar6\_1.

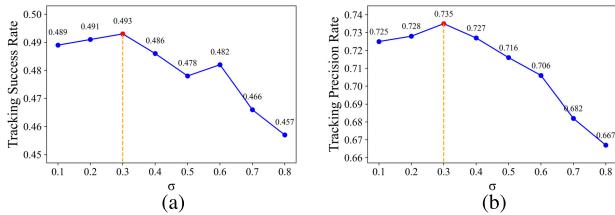


Fig. 12. Impact of different  $\sigma$  values on tracking success rate and tracking precision rate. (a) Impact of different  $\sigma$  values on tracking success rate. (b) Impact of different  $\sigma$  values on tracking precision rate.

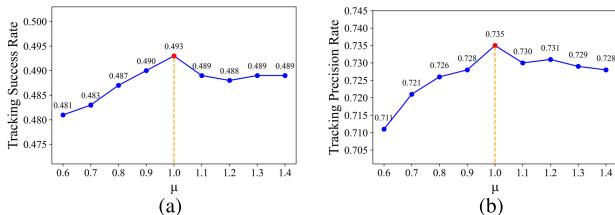


Fig. 13. Impact of different  $\mu$  values on tracking success rate and tracking precision rate. (a) Impact of different  $\mu$  values on tracking success rate. (b) Impact of different  $\mu$  values on tracking precision rate.

making the parameter-tuning process relatively straightforward.

As shown in Fig. 13, a clear pattern emerges between the values of parameter  $\mu$  and the performance of the tracker. As the parameter  $\mu$  varies within the range from 0.6 to 1.0, the tracker's performance consistently improves, culminating in its highest level when  $\mu$  reaches 1.0. Subsequently, as the parameter  $\mu$  continues to increase within the range from 1.0 to 1.4, the tracker's performance maintains a consistently high standard. This is evident from the sustained success rate and accuracy of the tracking, which do not exhibit any significant decline but rather remain stable. Therefore, the parameter  $\mu$  does not exhibit a pronounced sensitivity in relation to the tracker's performance when it is greater than

or equal to 1.0. This indicates that our proposed tracker may not require extensive tuning experiments.

## V. CONCLUSION

In this article, we address a significant challenge observed in existing correlation-filter-based tracking algorithms, wherein the direct integration of CN + HOG features into the model update leads to a substantial degradation in the tracking performance of the tracker when operating in scenarios involving low-quality video. To mitigate this issue, we present an innovative solution involving the application of a Gaussian denoising method for preprocessing the extracted features. Importantly, this preprocessing step is executed independently of the iterative update process. Consequently, its impact on the tracking speed of the tracker is minimal, while it concurrently enhances the stability of target features and provides a clearer representation of target position and scale information.

Diverging from the limitations present in other UAV tracking algorithms with respect to historical information utilization, our approach encompasses a broader utilization of historical information by incorporating both temporal and spatial dimensions. Moreover, we introduce a learning-based continuously weighted dynamic response model. Through empirical evaluation, we demonstrate that this enhanced model exhibits remarkable adaptability, enables dynamic adjustments, and showcases heightened robustness. The effectiveness of the proposed model is substantiated through extensive experimentation on authoritative datasets, including DTB70, UAV123@10fps, and UAVDT, where it outperforms various state-of-the-art tracking algorithms in terms of tracking performance.

In terms of future work, we aim to delve deeper into the feature input stage. Specifically, we seek to explore how to enable the filter to learn more suitable features. The recent adoption of structured sparse norms for representing anomalous pixels in hyperspectral anomaly detection [68] and the utilization of clustering techniques for identifying and distinguishing background categories in the clustering-based background learning (CBL) method [69] have served as a source of inspiration. This inspiration has motivated us to address the denoising problem within features from a mathematical standpoint. We believe that addressing this issue remains a challenge. In addition to training the tracker on traditional RGB datasets, there is a growing research interest in utilizing RGB-T datasets for recognition and tracking. Therefore, we will also investigate the construction and enhancement of DCF methods on such datasets.

## REFERENCES

- [1] A. D. Boursianis et al., "Internet of Things (IoT) and agricultural unmanned aerial vehicles (UAVs) in smart farming: A comprehensive review," *Internet Things*, vol. 18, May 2022, Art. no. 100187.
- [2] S. Sudhakar, V. Vijayakumar, C. S. Kumar, V. Priya, L. Ravi, and V. Subramaniyam, "Unmanned aerial vehicle (UAV) based forest fire detection and monitoring for reducing false alarms in forest-fires," *Comput. Commun.*, vol. 149, pp. 1–16, Jan. 2020.
- [3] L. Gonzalez, G. Montes, E. Puig, S. Johnson, K. Mengersen, and K. Gaston, "Unmanned aerial vehicles (UAVs) and artificial intelligence revolutionizing wildlife monitoring and conservation," *Sensors*, vol. 16, no. 1, p. 97, Jan. 2016.

- [4] L. Wang, F. Chen, and H. Yin, "Detecting and tracking vehicles in traffic by unmanned aerial vehicles," *Autom. Construct.*, vol. 72, pp. 294–308, Dec. 2016.
- [5] Z. Hou, W. Li, J. Zhou, and R. Tao, "Spatial-spectral weighted and regularized tensor sparse correlation filter for object tracking in hyperspectral videos," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5541012.
- [6] H. K. Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1135–1143.
- [7] Y. Li, C. Bian, and H. Chen, "Object tracking in satellite videos: Correlation particle filter tracking method with motion estimation by Kalman filter," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5630112.
- [8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2005, pp. 886–893.
- [9] J. van de Weijer, C. Schmid, J. Verbeek, and D. Larlus, "Learning color names for real-world applications," *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1512–1523, Jul. 2009.
- [10] X.-F. Zhu, X.-J. Wu, T. Xu, Z.-H. Feng, and J. Kittler, "Robust visual object tracking via adaptive attribute-aware discriminative correlation filters," *IEEE Trans. Multimedia*, vol. 24, pp. 301–312, 2022.
- [11] H. Zhu, H. Peng, G. Xu, L. Deng, Y. Cheng, and A. Song, "Bilateral weighted regression ranking model with spatial-temporal correlation filter for visual tracking," *IEEE Trans. Multimedia*, vol. 24, pp. 2098–2111, 2022.
- [12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
- [13] K. P. Schwarz, M. G. Sideris, and R. Forsberg, "The use of FFT techniques in physical geodesy," *Geophys. J. Int.*, vol. 100, no. 3, pp. 485–514, Mar. 1990.
- [14] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, "Optimal parameter selection for the alternating direction method of multipliers (ADMM): Quadratic problems," *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 644–658, Mar. 2015.
- [15] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2411–2418.
- [16] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.
- [17] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4310–4318.
- [18] F. Li, C. Tian, W. Zuo, L. Zhang, and M.-H. Yang, "Learning spatial-temporal regularized correlation filters for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4904–4913.
- [19] T. Xu, Z.-H. Feng, X.-J. Wu, and J. Kittler, "Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual object tracking," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5596–5609, Nov. 2019.
- [20] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for UAV tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 445–461.
- [21] S. Li and D.-Y. Yeung, "Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models," in *Proc. AAAI Conf. Artif. Intell.*, 2017, vol. 31, no. 1, pp. 4140–4146.
- [22] D. Du et al., "The unmanned aerial vehicle benchmark: Object detection and tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 370–386.
- [23] F. Zhang, S. Ma, Y. Zhang, and Z. Qiu, "Perceiving temporal environment for correlation filters in real-time UAV tracking," *IEEE Signal Process. Lett.*, vol. 29, pp. 6–10, 2022.
- [24] D. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2544–2550.
- [25] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.
- [26] K. Dai, D. Wang, H. Lu, C. Sun, and J. Li, "Visual tracking via adaptive spatially-regularized correlation filters," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4670–4679.
- [27] C. Fu, J. Xu, F. Lin, F. Guo, T. Liu, and Z. Zhang, "Object saliency-aware dual regularized correlation filter for real-time aerial tracking," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 12, pp. 8940–8951, Dec. 2020.
- [28] J. Wen, H. Chu, Z. Lai, T. Xu, and L. Shen, "Enhanced robust spatial feature selection and correlation filter learning for UAV tracking," *Neural Netw.*, vol. 161, pp. 39–54, Apr. 2023.
- [29] C. Peng, F. Liu, J. Yang, and N. Kasabov, "Robust visual tracking via dirac-weighted cascading correlation filters," *IEEE Signal Process. Lett.*, vol. 25, no. 11, pp. 1700–1704, Nov. 2018.
- [30] Y. Zhang, Y.-F. Yu, K.-K. Huang, and Y. Wang, "Channel attentional correlation filters learning with second-order difference for UAV tracking," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2023, doi: 10.1109/LGRS.2023.3311441.
- [31] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3074–3082.
- [32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–12.
- [33] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6638–6646.
- [34] B. Uzkent, A. Rangnekar, and M. J. Hoffman, "Tracking in aerial hyperspectral videos using deep kernelized correlation filters," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 1, pp. 449–461, Jan. 2019.
- [35] J. Zhang, Y. He, W. Feng, J. Wang, and N. N. Xiong, "Learning background-aware and spatial-temporal regularized correlation filters for visual tracking," *Appl. Intell.*, vol. 53, no. 7, pp. 7697–7712, Apr. 2023.
- [36] N. Wang, Y. Song, C. Ma, W. Zhou, W. Liu, and H. Li, "Unsupervised deep tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1308–1317.
- [37] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Y. Choi, "Action-decision networks for visual tracking with deep reinforcement learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2711–2720.
- [38] X. Li, C. Ma, B. Wu, Z. He, and M.-H. Yang, "Target-aware deep tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1369–1378.
- [39] X. Xue, Y. Li, X. Yin, C. Shang, T. Peng, and Q. Shen, "Semantic-aware real-time correlation tracking framework for UAV videos," *IEEE Trans. Cybern.*, vol. 52, no. 4, pp. 2418–2429, Apr. 2022.
- [40] Z. Huang, C. Fu, Y. Li, F. Lin, and P. Lu, "Learning aberrance repressed correlation filters for real-time UAV tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2891–2900.
- [41] J. Ye, C. Fu, F. Lin, F. Ding, S. An, and G. Lu, "Multi-regularized correlation filter for UAV tracking and self-localization," *IEEE Trans. Ind. Electron.*, vol. 69, no. 6, pp. 6004–6014, Jun. 2022.
- [42] F. Ali, H. Kumar, S. Patil, A. Ahmed, A. Banjar, and A. Daud, "DBP-DeepCNN: Prediction of DNA-binding proteins using wavelet-based denoising and deep learning," *Chemometric Intell. Lab. Syst.*, vol. 229, Oct. 2022, Art. no. 104639.
- [43] L. Liu, C. L. P. Chen, X. You, Y. Y. Tang, Y. Zhang, and S. Li, "Mixed noise removal via robust constrained sparse representation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 9, pp. 2177–2189, Sep. 2018.
- [44] M. Wang, S. Zheng, X. Li, and X. Qin, "A new image denoising method based on Gaussian filter," in *Proc. Int. Conf. Inf. Sci., Electron. Electr. Eng.*, vol. 1, Apr. 2014, pp. 163–167.
- [45] E. Wei and A. Ozdaglar, "Distributed alternating direction method of multipliers," in *Proc. IEEE 51st IEEE Conf. Decis. Control (CDC)*, Dec. 2012, pp. 5445–5450.
- [46] K. B. Petersen et al., "The matrix cookbook," *Tech. Univ. Denmark*, vol. 7, no. 15, p. 510, 2008.
- [47] Y. Li, C. Fu, F. Ding, Z. Huang, and G. Lu, "AutoTrack: Towards high-performance visual tracking for UAV with automatic spatio-temporal regularization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11923–11932.
- [48] C. Fu, J. Ye, J. Xu, Y. He, and F. Lin, "Disruptor-aware interval-based response inconsistency for correlation filters in real-time aerial tracking," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 8, pp. 6301–6313, Jul. 2021.
- [49] G. Zheng, C. Fu, J. Ye, F. Lin, and F. Ding, "Mutation sensitive correlation filter for real-time UAV tracking with adaptive hybrid label," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 503–509.

- [50] F. Lin, C. Fu, Y. He, F. Guo, and Q. Tang, “BiCF: Learning bidirectional incongruity-aware correlation filter for efficient UAV object tracking,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 2365–2371.
- [51] F. Lin, C. Fu, Y. He, W. Xiong, and F. Li, “ReCF: Exploiting response reasoning for correlation filters in real-time UAV tracking,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 10469–10480, Aug. 2022.
- [52] L. Zhang and P. N. Suganthan, “Robust visual tracking via co-trained kernelized correlation filters,” *Pattern Recognit.*, vol. 69, pp. 82–93, Sep. 2017.
- [53] N. Wang, W. Zhou, Q. Tian, R. Hong, M. Wang, and H. Li, “Multi-cue correlation filters for robust visual tracking,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4844–4853.
- [54] L. Wang, W. Ouyang, X. Wang, and H. Lu, “Visual tracking with fully convolutional networks,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3119–3127.
- [55] Y. Qi et al., “Hedged deep tracking,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4303–4311.
- [56] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, “Beyond correlation filters: Learning continuous convolution operators for visual tracking,” in *Computer Vision—ECCV 2016*. Amsterdam, The Netherlands: Springer, Oct. 2016, pp. 472–488.
- [57] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr, “End-to-end representation learning for correlation filter based tracking,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2805–2813.
- [58] T. Zhang, C. Xu, and M.-H. Yang, “Multi-task correlation particle filter for robust object tracking,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4335–4343.
- [59] F. Li, Y. Yao, P. Li, D. Zhang, W. Zuo, and M.-H. Yang, “Integrating boundary and center correlation filters for visual tracking with aspect ratio variation,” in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 2001–2009.
- [60] Z. He, Y. Fan, J. Zhuang, Y. Dong, and H. Bai, “Correlation filters with weighted convolution responses,” in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 1992–2000.
- [61] J. Zhao, K. Dai, D. Wang, H. Lu, and X. Yang, “Online filtering training samples for robust visual tracking,” in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 1488–1496.
- [62] N. Wang, W. Zhou, Y. Song, C. Ma, and H. Li, “Real-time correlation tracking via joint model compression and transfer,” *IEEE Trans. Image Process.*, vol. 29, pp. 6123–6135, 2020.
- [63] N. Wang, W. Zhou, Y. Song, C. Ma, W. Liu, and H. Li, “Unsupervised deep representation learning for real-time tracking,” *Int. J. Comput. Vis.*, vol. 129, no. 2, pp. 400–418, Feb. 2021.
- [64] S. Moorthy and Y. H. Joo, “Multi-expert visual tracking using hierarchical convolutional feature fusion via contextual information,” *Inf. Sci.*, vol. 546, pp. 996–1013, Feb. 2021.
- [65] I. Sosnovik, A. Moskalev, and A. Smeulders, “Scale equivariance improves Siamese tracking,” in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 2765–2774.
- [66] B. Yan, H. Peng, J. Fu, D. Wang, and H. Lu, “Learning spatio-temporal transformer for visual tracking,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10448–10457.
- [67] E. R. AlBasiouny, A.-F. Attia, H. E. Abdelmunim, and H. M. Abbas, “Robust visual tracking using very deep generative model,” *J. Big Data*, vol. 10, no. 1, pp. 1–26, Jan. 2023.
- [68] Y.-P. Zhao, H. Li, Y. Chen, Z. Wang, and X. Li, “Hyperspectral anomaly detection via structured sparsity plus enhanced low-rankness,” *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5515115.
- [69] M. E. Aghili, M. Imani, and H. Ghassemian, “Clustering based background learning for hyperspectral anomaly detection,” *Egyptian J. Remote Sens. Space Sci.*, vol. 26, no. 3, pp. 477–489, Dec. 2023.



**Yang Zhang** received the B.Sc. degree in applied mathematics from Guangzhou University, Guangzhou, China, in 2020, where he is currently pursuing the M.Sc. degree in statistics.

His research interests include visual tracking and machine learning.



**Yu-Feng Yu** (Member, IEEE) received the Ph.D. degree in statistics from Sun Yat-sen University, Guangzhou, China, in 2017.

From 2016 to 2017, he was a Visiting Scholar with the Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV, USA. From 2017 to 2018, he was a Senior Research Associate with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong. He is currently an Associate Professor with the Department of Statistics, Guangzhou University, Guangzhou. His research interests include image processing, statistical optimization, pattern recognition, machine learning, and computer vision.



**Long Chen** (Senior Member, IEEE) received the B.S. degree in information sciences from Peking University, Beijing, China, in 2000, the M.S.E. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 2003, the M.S. degree in computer engineering from the University of Alberta, Edmonton, AB, Canada, in 2005, and the Ph.D. degree in electrical engineering from The University of Texas at San Antonio, San Antonio, TX, USA, in 2010.

From 2010 to 2011, he was a Post-Doctoral Fellow with The University of Texas at San Antonio. He is an Associate Professor with the Department of Computer and Information Science, University of Macau, Macau, China. His research interests include computational intelligence, Bayesian methods, and other machine learning techniques and their applications.



**Weiping Ding** (Senior Member, IEEE) received the Ph.D. degree in computer science from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2013.

In 2016, he was a Visiting Scholar with the National University of Singapore, Singapore. From 2014 to 2015, he was a Post-Doctoral Researcher with the Brain Research Center, National Chiao Tung University, Hsinchu, Taiwan. From 2017 to 2018, he was a Visiting Professor with the University of Technology Sydney, Ultimo, NSW, Australia. He is a Full Professor with the School of Information Science and Technology, Nantong University, Nantong, China, and also the Supervisor of the Ph.D. postgraduate at the Faculty of Data Science, City University of Macau, Macau, China. He has published over 250 articles, including over 90 IEEE TRANSACTIONS articles. He holds 28 approved invention patents, including two U.S. patents and one Australian patent. He has coauthored four books. His 16 authored/coauthored articles have been selected as ESI Highly Cited Papers. His main research directions involve deep neural networks, multimodal machine learning, and medical image analysis.

Dr. Ding serves as an Associate Editor/Editorial Board Member for IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON FUZZY SYSTEMS, IEEE/CAA JOURNAL OF AUTOMATICA SINICA, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON INTELLIGENT VEHICLES, IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE, IEEE TRANSACTIONS ON ARTIFICIAL INTELLIGENCE, *Information Fusion*, *Information Sciences*, *Neurocomputing*, and *Applied Soft Computing*. He is the Leading Guest Editor of Special Issues in several prestigious journals, including IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON FUZZY SYSTEMS, and *Information Fusion*. Now, he is the Co-Editor-in-Chief of *Journal of Artificial Intelligence and Systems* and *Journal of Artificial Intelligence Research and Advances*. He ranked within the top 2% Ranking of Scientists in the World by Stanford University (2020–2023).