Electronic Research Archive

https://www.aimspress.com/journal/era

ERA, 33(3): 1446–1475.
DOI: 10.3934/era.2025068
Received: 20 November 2024
Revised: 18 February 2025
Accepted: 07 March 2025
Published: 14 March 2025

Research article

UAV object tracking algorithm based on spatial saliency-aware correlation filter

Changhui Wu[1,†], Jinrong Shen[2,†], Kaiwei Chen[1], Yingpin Chen[1,*]and Yuan Liao[1]

[1] School of Physics and Information Engineering, Minnan Normal University, Zhangzhou, China
[2] School of Computer Science, Minnan Normal University, Zhangzhou, China

† The authors contributed equally to this work.

* **Correspondence:** Email: 110500617@163.com.

Abstract: Recently, correlation filter-based tracking methods have been widely adopted in UAV target tracking due to their outstanding performance and excellent tracking efficiency. However, existing correlation filter-based tracking methods still face issues such as redundant visual features with weak discriminative ability, inadequate spatio-temporal information mining, and filter degradation. In order to overcome these challenges, this paper proposes a spatial saliency-aware strategy that reduces redundant information in spatial and channel dimensions, thus improving the discriminative ability between the target and background. Also, this paper proposes a position estimation mechanism under spatio-temporal joint constraints to fully mine spatio-temporal information and enhance the robustness of the model in complex scenarios. Furthermore, this paper establishes a positive expert group using historical positive samples to assess the reliability of candidate samples, thereby effectively mitigating the filter degradation issue. Ultimately, the effectiveness of the proposed method is demonstrated through the evaluation of multiple public datasets. The experimental results reveal that this method outperforms others in tracking performance under various challenging conditions.

Keywords: UAV object tracking; spatial saliency perception; positive expert group; spatio-temporal joint constraints

1. Introduction

UAV object tracking technology is widely used in harsh environments unsuitable for human visual positioning, such as high-altitude operations [1], military surveillance [2], personnel rescue in fire environments [3], and object anchoring in nuclear radiation-polluted areas [4]. This technology primarily involves real-time positioning and continuous monitoring capabilities for specific objects.

UAV object tracking needs to cope with common tracking challenges and tackle some unique challenges as follows: 1) Due to the broad aerial-to-ground perspective, background interference is significantly increased compared to conventional tracking tasks; 2) The small size of UAVs results in minimal carrying capacity, power, and computational resources; 3) Flight-induced vibrations caused by aerial turbulence can lead to motion blur in the captured images, making robust feature extraction more challenging. Consequently, UAV object-tracking algorithms still have significant potential for improvement and research value.

UAV object tracking algorithms can be broadly divided into discriminative correlation filter (DCF)-based methods [5–10] and deep learning-based approaches [11–15]. While deep learning techniques achieve outstanding object-tracking performance, they are unsuitable for deployment on UAV systems with constrained computational resources. DCF algorithms have emerged as the leading framework for UAV object tracking, owing to their computational efficiency and robust performance on single-core CPUs. This character makes them ideal for resource-constrained edge computing platforms, including UAVs and autonomous vehicles. Although DCF algorithms have progressed in UAV object tracking, existing DCF trackers still exhibit limitations in feature extraction, filter degradation, and spatiotemporal feature fusion.

Initial correlation filter algorithms primarily depended on handcrafted features [16], such as histogram of oriented gradient (HOG) [17] and color names (CN) [18], for object representation. These algorithms demonstrated outstanding tracking performance and efficient computational capabilities, achieving a state-of-the-art level [19–21]. Nonetheless, these handcrafted features often fail to capture subtle differences among similar objects in complex environments, hindering effective differentiation between the object and background. As a result, some researchers adopted deep features obtained from convolutional neural networks (CNN) to augment the tracking accuracy of correlation filters [5, 22–24]. However, these high-dimensional features still contain redundant and potentially harmful information. To address these problems, researchers have focused on feature reduction and selection. For instance, some trackers employ principal component analysis (PCA) [22, 25] to reduce the dimensionality of deep features. Some researchers introduced attention mechanisms [26] and dynamic weight allocation strategies [9, 27] to filter adequate spatial and channel information, suppressing invalid or harmful information. Likewise, the GFS-DCF [8] method introduced group sparsity statistical priors to select more significant spatial and channel information. LADCF [28] achieves joint spatio-temporal filter learning on a low-dimensional discriminative manifold, leveraging adaptive spatial feature selection and temporal consistency constraints. Moreover, due to the DCF algorithm's use of cyclic shift operators to form training samples, discontinuities arise at the shift junctions, creating boundary effects.

To enhance the discrimination of perspective features and remove redundant or detrimental information, scholars have started integrating saliency detection into the DCF tracking framework, developing advanced algorithms, and making notable progress. For example, in [29], object saliency maps are incorporated into regularization weights to suppress background noise dynamically. Similarly, DRCF [30] integrates cascading discriminative correlation filters with spatio-temporal saliency to enhance the robustness and precision of object tracking. SDCS-CF [31] uses a lightweight, fully convolutional network to produce saliency maps that serve as differential weights for features in the search area, thereby boosting the tracker's resilience to background disturbances. Alternative methods use image saliency data to establish spatial or temporal regularization constraints [32, 33] or rely on spatio-temporal saliency maps to strengthen object feature representation [34, 35]. Yet, most methods focus on reducing spatial or channel information, neglecting their interconnections. To bolster the link between spatial and channel data, we introduce a feature reduction strategy that is aware of saliency channels, directing the selection of channel features and allocating channel attention through spatial saliency information.

The integration of spatio-temporal information is crucial in object tracking. Yet, traditional UAV DCF approaches have not thoroughly explored this aspect. In recent years, scholars have increasingly recognized the importance of spatio-temporal information, leading to fruitful explorations. For example, STRCF [36] adopts a spatio-temporal regularization method to fuse temporal and spatial information. Unlike the fixed parameters of STRCF, AutoTrack [37] leverages local and global data to devise an adaptive method for tuning regularization parameters. Similarly, DeepABCF [38] employs spatio-temporal anomaly suppressors to mitigate adverse effects from intraclass disruptions and complex backdrops. Specific deep learning techniques, such as CSWinTT [39] and STARK [14], utilize dynamic templates to delve into spatio-temporal information. Drawing inspiration from these methods, we have formulated boundary suppression factors, spatial interference suppressors, and spatiotemporal anomaly suppressors to develop a precise model for object localization.

Filter degradation significantly contributes to tracker failures, especially under partial or complete occlusion, where degraded filters might incorrectly identify occluding objects as the object. To tackle this issue, several researchers have incorporated multiple historical positive samples into the learning process of filters, like SRDCFdecon [40], C-COT [41], STSL [42], and VALACF [43]. Inspired by these methods, we present a strategy to assess the reliability of optimal candidate objects through a panel of optimistic experts. This involves forming a group with multiple historical positive samples, evaluating the reliability of candidate samples in each frame, and updating the filter with the most reliable samples.

In summary, this article introduces a UAV object tracking algorithm utilizing spatial saliency-aware correlation filtering, dubbed SSACF. The key contributions of this study are outlined as follows:

(i) We introduce a spatial saliency-aware strategy employing the object's color statistical histograms to build a non-standard saliency-aware mask. This method replaces "symmetric" sampling with "asymmetric" sampling, leveraging the object's shape characteristics to filter background interference and enhance spatial feature discrimination. Based on this, spatial saliency is utilized to guide the reduction of channel information.

(ii) We introduce a positioning estimation mechanism under joint spatio-temporal constraints, employing boundary suppression factors to reduce boundary effects and spatial interference suppressors to diminish intraclass interference. Furthermore, the mechanism incorporates spatio-temporal anomaly suppression regularizers that analyze response differences between adjacent frames to regulate filter outputs in abnormal regions. These actions thoroughly leverage spatio-temporal information.

(iii) We utilize historical positive samples to form a panel of optimistic experts to assess the reliability of candidate samples. If reliability falls below a predefined threshold, indicating contamination, filter learning is paused, effectively mitigating filter degradation during occlusion scenarios.

The subsequent sections of this paper are organized as follows: Section 2 provides the necessary preliminary background. Section 3 elaborates on the details of implementing the proposed tracking methods. Section 4 presents the evaluation results, comparing our algorithm against state-of-the-art (SOTA) trackers on four benchmark datasets: OTB100 [44], DTB70 [45], UAV123 [46], and UAV20L. Finally, the paper concludes with a summary of key findings and future research directions.

2. Preliminary knowledge

2.1. Review of DCF

Accurately localizing an object in continuous video frames is essential for visual object tracking. The correlation filter (DCF)-based method seeks to predict the initial location of the object in a video. This process assumes that the expected location of the tracked object in frame t+1 is determined by training a filter $\mathcal{W}t \in \mathbb{R}^{H\times W\times C}$, represented as a $W \times H$ matrix with C-dimensional channel features. The training sample $\boldsymbol{Xt}$ from frame t and the corresponding Gaussian-shaped expected response map $\boldsymbol{Y}$ are utilized. To obtain multi-channel correlation filters, DCF formulates the tracking task as a regularized least squares problem:

$$\mathcal{W}_t = \mathop{\mathbf{argmin}}_{\mathcal{W}_t} \frac{1}{2}\left\|\sum_{c=1}^{C} \boldsymbol{X}_t^{\{c\}} \star \boldsymbol{W}_t^{\{c\}} - \boldsymbol{Y}\right\|_2^2 + \frac{\lambda}{2}\sum_{c=1}^{C}\left\|\boldsymbol{W}_t^{\{c\}}\right\|_F^2 = \mathop{\mathbf{argmin}}_{\mathcal{W}_t} \frac{1}{2}\left\|\sum_{c=1}^{C} \boldsymbol{X}_t^{\{c\}} * \bar{\boldsymbol{W}}_t^{\{c\}} - \boldsymbol{Y}\right\|_2^2 + \frac{\lambda}{2}\sum_{c=1}^{C}\left\|\boldsymbol{W}_t^{\{c\}}\right\|_F^2, \tag{2.1}$$

where $\star$ represents the cyclic correlation operator, and $*$ is the cyclic convolution operator. λ is the regularization parameter, $\boldsymbol{W}_t^{\{c\}} \in \mathbb{R}^{H\times W}$ is the corresponding discriminative correlation filter, and $\boldsymbol{X}_t^{\{c\}} \in \mathbb{R}^{H\times W}$ denotes the feature of the c-th channel. $\bar{\boldsymbol{W}}_t^{\{c\}}$ is obtained by first reversing $\boldsymbol{W}_t^{\{c\}}$ row-wise, then performing a one-unit cyclic shift, then reversing it column-wise and shifting it cyclically. If $\boldsymbol{W}_t^{\{c\}} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$, then $\bar{\boldsymbol{W}}_t^{\{c\}} = \begin{pmatrix} 9 & 7 & 8 \\ 2 & 3 & 1 \\ 6 & 4 & 5 \end{pmatrix}$.

The least squares problem can be subsequently converted into the Fourier domain. By deriving the closed-form solution to Eq (2.1), the solution for the filter $\mathcal{W}_t \in \mathbb{R}^{H\times W\times C}$ is obtained through Fourier transform and complex conjugate operations, as follows:

$$\hat{\boldsymbol{W}}_t^{\{c\}} = \frac{\hat{\boldsymbol{X}}_t^{\{c\}} \odot \hat{\boldsymbol{Y}}^*}{\sum_{c=1}^{C}\left(\hat{\boldsymbol{X}}_t^{\{c\}}\right)^* \odot \hat{\boldsymbol{X}}_t^{\{c\}} + \lambda}, \tag{2.2}$$

where $\hat{.}$ denotes the discrete Fourier transform, $\odot$ represents entry-wise multiplication, and $.^*$ stands for the complex conjugate operator.

In the following frame, the response map $\boldsymbol{R} \in \mathbb{R}^{H\times W}$ is calculated by extracting the feature vector $\mathcal{Z} \in \mathbb{R}^{H\times W\times C}$. This is done using the inverse Fourier transform, where the features from all channels are multiplied element-wise with the filter in the Fourier domain and summed, and then an inverse Fourier transform is performed to obtain the response map $\boldsymbol{R}$, as described below:

$$\boldsymbol{R} = \mathbf{real}\left(\mathcal{F}^{-1}\left(\sum_{c=1}^{C}\hat{\boldsymbol{Z}}^c \odot \left(\hat{\boldsymbol{W}}_t^{\{c\}}\right)^*\right)\right), \tag{2.3}$$

where $\mathcal{F}^{-1}$ represents the inverse Fourier transform, and the object position in frame t+1 is determined by the peak location in the response map $\boldsymbol{R}$. **real** refers to the real part operator.

2.2. Spatial saliency awareness

Leveraging feature extraction, this study employs an "asymmetric" sampling mechanism to dynamically allocate visual spatial attention within the tracker, significantly improving the differentiation between the object and the background. In addressing spatial saliency, a spatial saliency perception matrix $\boldsymbol{M}$ facilitates asymmetric background awareness. The design specifics of the matrix $\boldsymbol{M}$ are detailed as follows:

Assuming the test sample is $\mathcal{Z} \in \mathbb{R}^{R_h\times C_w\times 3}$, based on Bayes' posterior probability theorem, the probability that pixel $\boldsymbol{z}_p$=$(r, c)^T$ belongs to the object is:

$$p\left(m = 1|\, z_p\right) \propto p\left(z_p|m = 1\,\right)p\left(m = 1\right), \tag{2.4}$$

where m is an element of the spatial saliency perception matrix $\boldsymbol{M} \in \mathbb{R}^{R_h\times C_w}$, $m = 1$ indicates that pixel z_p belongs to the object, and $m = 0$ indicates that the pixel originates from the contextual background. $p\left(z_p\right) = \frac{1}{R_hC_w}$ is a constant probability, $\propto$ denotes proportionality, and $p\left(z_p|m = 1\right)$=$\mathrm{e}^{-\frac{||z_p-\boldsymbol{p}^{(t-1)}||_2^2}{2\sigma^2}}$ signifies the spatial prior probability of the object, with σ representing the standard deviation of the Gaussian window, and $\boldsymbol{p}^{(t-1)}$ indicating the object's location in the previous frame. $p\left(m = 1\right)$ signifies the probability associated with color likelihood, defined as:

$$p\left(m = 1\right) = \boldsymbol{a}^T\boldsymbol{e}_{k(z_p)}, \tag{2.5}$$

where the color information from each pixel is converted into a vector $\boldsymbol{e}_{k(z_p)} \in \mathbb{R}^{N_j\times 1}$ (where N_j is the number of color categories). The vector is a one-hot vector (with a value of 1 at the $k(z_p)$-th position and 0 elsewhere, where $k(\boldsymbol{z}_p)$ represents the color index at the pixel z_p). $\boldsymbol{d} = \left\{\boldsymbol{d}^o, \boldsymbol{d}^b\right\}$ is the color histogram, and $p\left(m = 1\right)$ is calculated after back-projecting to spatial pixels. $\boldsymbol{d}^o \in \mathbb{R}^{N_j\times 1}$ represents the object's color histogram, and $\boldsymbol{d}^b \in \mathbb{R}^{N_j\times 1}$ represents the background's color histogram. $\boldsymbol{a} \in \mathbb{R}^{N_j\times 1}$ is the regression filter for color histograms, with its solving function as follows:

$$\mathcal{L}_a = \mathbf{min}_{a} \sum_{j=1}^{N_j}\left[\boldsymbol{d}_j^o\left(\boldsymbol{a}_j - 1\right)^2 + \boldsymbol{d}_j^b\left(\boldsymbol{a}_j\right)^2\right] + \lambda\, ||\boldsymbol{a}||_2^2, \tag{2.6}$$

where λ is a hyper-parameter of the ridge regression. $||\boldsymbol{a}||_2^2 = \sqrt{\sum_{j=1}^{N_j} \boldsymbol{a}_j^2}$ is the L_2 norm of the vector $\boldsymbol{a} \in \mathbb{R}^{N_j\times 1}$, and $\boldsymbol{a}_j$ represents the entry of $\boldsymbol{a}$. Similarly, $\boldsymbol{d}_j^o$ and $\boldsymbol{d}_j^b$ represent the entries of $\boldsymbol{d}^o$ and $\boldsymbol{d}^b$, respectively. The answer to Eq (2.6) is provided:

$$\boldsymbol{a}_j = \frac{\boldsymbol{d}_j^o}{\boldsymbol{d}_j^o + \boldsymbol{d}_j^b + \lambda}. \tag{2.7}$$

Note: In Eq (2.7), division is element-wise.

Through binarization of the probabilities detailed in Eq (2.4), the spatial saliency perception matrix $\boldsymbol{M}$ is formulated. The matrix elements m take values of either 0 or 1.

$$m = \begin{cases} 1, & p\left(m = 1|\, z_p\right) > \alpha \\ 0, & \quad others \end{cases}, \tag{2.8}$$

where $\alpha \in (0, 1)$ serves as the posterior probability threshold that dictates whether a pixel is part of the object. If α is set too low, asymmetric sampling may capture too much background detail; conversely, if it is set too high, vital object information may be omitted. In real-world settings, manual adjustment of this parameter is typically necessary to fine-tune tracking performance. Once $\boldsymbol{M}$ is acquired, it needs to be reshaped to $\mathbb{R}^{H\times W}$ to facilitate correlation filtering.

Additionally, color information is derived by transitioning the image from the RGB to the HSV color space. The hue component is segmented into N_j intervals ranging from 0 to 1, followed by computing a color histogram for the hue component of each pixel within the object. Likewise, the color histogram for the hue component of each pixel in the background is calculated. This method allows for estimating the color type $k(z_p)$ of each pixel by directly determining the interval to which its hue component corresponds. However, as this approach is not directly applicable to grayscale images, intervals from 0 to 1 are defined for such images, and histograms for the grayscale values of object and background pixels are computed within each interval. This method permits the estimation of each pixel's color from its grayscale value. However, as color information is essential for object tracking, the effectiveness of this approach is reduced on grayscale images compared to color images.

3. Method

3.1. Spatial saliency-based feature reduction strategy

The effectiveness of deep learning in object tracking largely stems from the capability of neural networks to extract superior and more refined deep features. Although numerous methods incorporate image saliency information to develop spatial/temporal regularization terms [29, 32, 33] for reducing boundary effects or managing object appearance variations through reinforcement learning, the limited training samples in visual tracking pose challenges. This scarcity often leads to overlooked connections between multi-channel features and object saliency information. Employing a deep network trained in a particular object to extract its multi-channel features can result in the inclusion of numerous interfering channels. When the DCF tracker extracts features from the search region and generates the response map according to the object's location, it should prioritize analyzing the energy levels of feature channels specifically within the object region. This paper is dedicated to allocating attention to object channels and selecting channel features based on their saliency within the feature space.

As shown in Figure 1, to quantify the confidence of feature channels, we use the asymmetric sampling from Section 2.2 to obtain the spatial saliency of the object. We subsequently apply weights to the extracted feature maps of the object to generate the object-perception and background-perception region feature maps. Finally, by calculating the average energy of these two parts, we use the FR (Feature Reliability) index as an evaluation metric to allocate object channel attention and select channel features. The FR [47] index is defined as:

$$FR^{\{c\}} = \frac{E_O\left(\boldsymbol{X}_t^{\{c\}}\right)}{E_B\left(\boldsymbol{X}_t^{\{c\}}\right)}, c = 1, 2, \cdots, C, \tag{3.1}$$

where $FR^{\{c\}}$ denotes the FR value of the c-th channel, and $E_O\left(\boldsymbol{X}_t^{\{c\}}\right)$ denotes the average energy of the object-perception region, calculated as:

$$E_O\left(\boldsymbol{X}_t^{\{c\}}\right) = \frac{\sum\limits_{(i,j)\in O} \boldsymbol{X}_t^{\{c\}}(i, j)}{A_O}, \tag{3.2}$$

where $\boldsymbol{X}_t^{\{c\}}(i,\ j)$ denotes the feature located at position $(i,\ j)$ in the current frame, and A_O indicates the area of the object-perception region.

Similarly, $E_B\left(\boldsymbol{X}_t^{\{c\}}\right)$ denotes the average energy of the background region:

$$E_B\left(\boldsymbol{X}_t^{\{c\}}\right) = \frac{\sum\limits_{(i,j)\in S} \boldsymbol{X}_t^{\{c\}}(i,j) - \sum\limits_{(i,j)\in O} \boldsymbol{X}_t^{\{c\}}(i,j)}{A_S - A_O}, \tag{3.3}$$

where A_S signifies the area of the object search region.

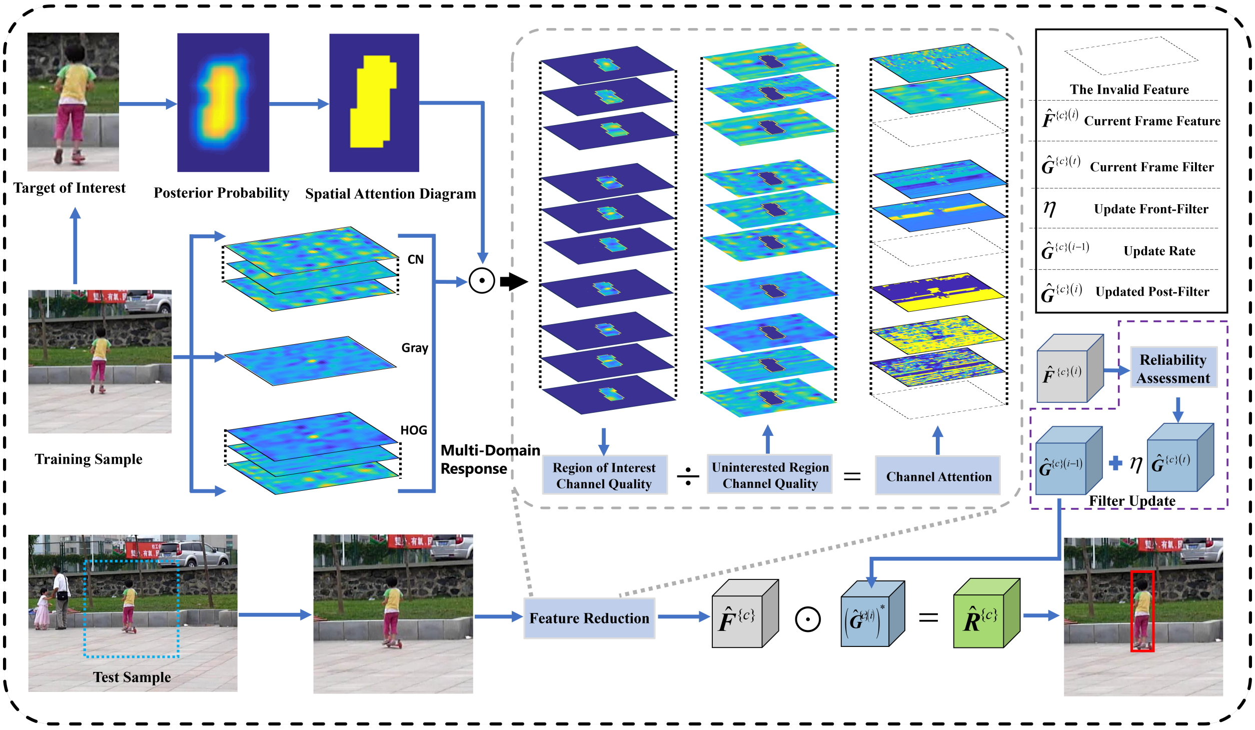

Target of Interest
Posterior Probability
Spatial Attention Diagram
Training Sample
CN
Gray
HOG
Multi-Domain Response
Region of Interest Channel Quality
÷
Uninterested Region Channel Quality
=
Channel Attention
The Invalid Feature
$\hat{F}^{\{c\}(t)}$ Current Frame Feature
$\hat{G}^{\{c\}(t)}$ Current Frame Filter
η Update Front-Filter
$\hat{G}^{\{c\}(t-1)}$ Update Rate
$\hat{G}^{\{c\}(t)}$ Updated Post-Filter
$\hat{F}^{\{c\}(t)}$
Reliability Assessment
$\hat{G}^{\{c\}(t-1)}$ + η $\hat{G}^{\{c\}(t)}$
Filter Update
Test Sample
Feature Reduction
$\hat{F}^{\{c\}}$ ⊙ $(\hat{G}^{\{c\}})^*$ = $\hat{R}^{\{c\}}$

Figure 1. Spatial saliency-based feature reduction strategy.

Based on the earlier equations, it is evident that a high FR score signifies that the feature channel contains substantial object-related information, while a low FR score indicates the presence of more background noise in the channel. Thus, this section evaluates the FR scores for all feature channels and strategically selects channels with higher FR scores for filter training using established weights. This approach minimizes the negative impact of low-confidence channels during filter learning. The specific methodology employs the FR index to determine the importance of each channel, subsequently assigning differential weights to channels based on their assessed importance. The calculation procedure is outlined as follows:

$$s^{\{c\}} = 1 + \frac{1}{2} \times \frac{FR^{\{c\}} - \mathbf{min}(FR)}{\mathbf{max}(FR) - \mathbf{min}(FR)}, \tag{3.4}$$

where $s^{\{c\}}$ denotes the weight of the c-th feature channel, min(FR) is the minimum FR score across all channels, and the max(FR) is the maximum FR score across all channels.

This method effectively improves the precision of feature channel selection during the object-tracking process, thereby enhancing the robustness and accuracy of tracking.

3.2. Position estimation mechanism with spatio-temporal joint constraints

In this strategy, to achieve precise object localization and tracking, we design boundary suppression regularization factors, spatial interference suppression regularization factors, and spatio-temporal outlier suppression regularization factors to integrate both spatial and temporal constraints. The boundary suppression regularization term utilizes a fixed inverse Gaussian function, as detailed in [48], to alleviate boundary effects resulting from the assumption of periodic boundary conditions. The spatial interference suppression regularization factors analyze response map variations between consecutive frames to effectively capture the position of interference sources in the current frame, thus suppressing their impact on the object tracking process. The spatio-temporal anomaly suppression regularization factors aim to suppress pixels with notable changes in adjacent frames, mitigating tracking drift resulting from out-of-plane rotation and significant object deformations.

By incorporating the spatial saliency sampling scheme, we introduce the spatial saliency perception matrix $\boldsymbol{M}$ to build a spatial saliency correlation filtering framework, and apply spatial saliency constraints $\boldsymbol{G}^{\{c\}} = \boldsymbol{M} \odot \boldsymbol{W}^{\{c\}}$, to each channel filter $\boldsymbol{W}^{\{c\}}$. The objective function for the tracking model is then formulated as:

$$\begin{aligned} &\mathcal{L}\left(\boldsymbol{G}^{\{c\}}, \boldsymbol{W}^{\{c\}} \mid \boldsymbol{M}\right) = \tfrac{1}{2} \left\| \sum_{c=1}^{C} \boldsymbol{X}_t^{\{c\}} \star \boldsymbol{G}^{\{c\}} - \boldsymbol{Y} \right\|_2^2 + \tfrac{\lambda}{2} \sum_{c=1}^{C} \left\| \tilde{\boldsymbol{W}} \odot \boldsymbol{M} \odot \boldsymbol{W}^{\{c\}} \right\|_2^2 , \\ &\text{s.t. } \boldsymbol{G}^{\{c\}} = \boldsymbol{M} \odot \boldsymbol{W}^{\{c\}} \end{aligned} \tag{3.5}$$

where $\tilde{\boldsymbol{W}}$ represents the spatio-temporal outlier suppression regularization factors, and the factor is defined as:

$$\tilde{\boldsymbol{W}} = \boldsymbol{S}_B + \theta_1 \boldsymbol{S}_T + \theta_2 \boldsymbol{S}_S , \tag{3.6}$$

where θ_1 and θ_2 represent the balance parameters, $\boldsymbol{S}_B$ represents the fixed-shape inverse Gaussian spatial regularization factors used to suppress boundary effects, $\boldsymbol{S}_T = \frac{\left|\boldsymbol{R}^{(t)}[\Delta_t] - \boldsymbol{R}^{(t-1)}[\Delta_{t-1}]\right|}{\boldsymbol{R}^{(t-1)}[\Delta_{t-1}]}$ represents the spatio-temporal anomaly suppression regularization factors, $\boldsymbol{R}^{(t-1)}\,[\Delta_{t-1}]$ denotes the map after the peak value of $\boldsymbol{R}^{(t-1)}$ is shifted to the center of the search space by the shift operator $[\Delta_{t-1}]$, and $\boldsymbol{R}^{(t)}\,[\Delta_t]$ represents the map after the peak value of $R^{(t)}$ is shifted to the center of the search space by the cyclic shift operator $[\Delta_t]$, with the shift distance Δ_t calculated based on the relative distance between the peak value position of $\boldsymbol{R}^{(t)}$ and the center of the region; $\boldsymbol{S}_S = \boldsymbol{I}_S\,[\Delta_t]$ is the spatio-temporal anomaly suppression regularization factors, and $\boldsymbol{I}_S$ represents the interference object detection matrix, with elements:

$$\boldsymbol{I}_S(x, y) = \begin{cases} 1, & \text{if } (x, y) \text{ is detected as the peak position} \\ 0, & \text{others} \end{cases} , \tag{3.7}$$

where $\boldsymbol{I}_s\,[\Delta_t]$ represents the matrix obtained by shifting $\boldsymbol{I}_s$ to the center of the search space using the cyclic shift operator $[\Delta_t]$.

3.3. Evaluation of optimal candidate object reliability based on the positive expert group

Traditional correlation filter tracking methods often treat the sample with the highest response as the object appearance in the current frame, disregarding the reliability of this prime candidate. This practice can lead to filter degradation in cases of occlusion. In response to this challenge, this paper introduces an evaluation method for the reliability of optimal candidate objects using a positive expert group. This approach determines the reliability of various samples by archiving object appearance slices from different historical periods and selecting the most reliable one for tracking. Without historical data for the first frame, the positive expert group comprises image segments of the object captured from different perspectives or states within that frame.

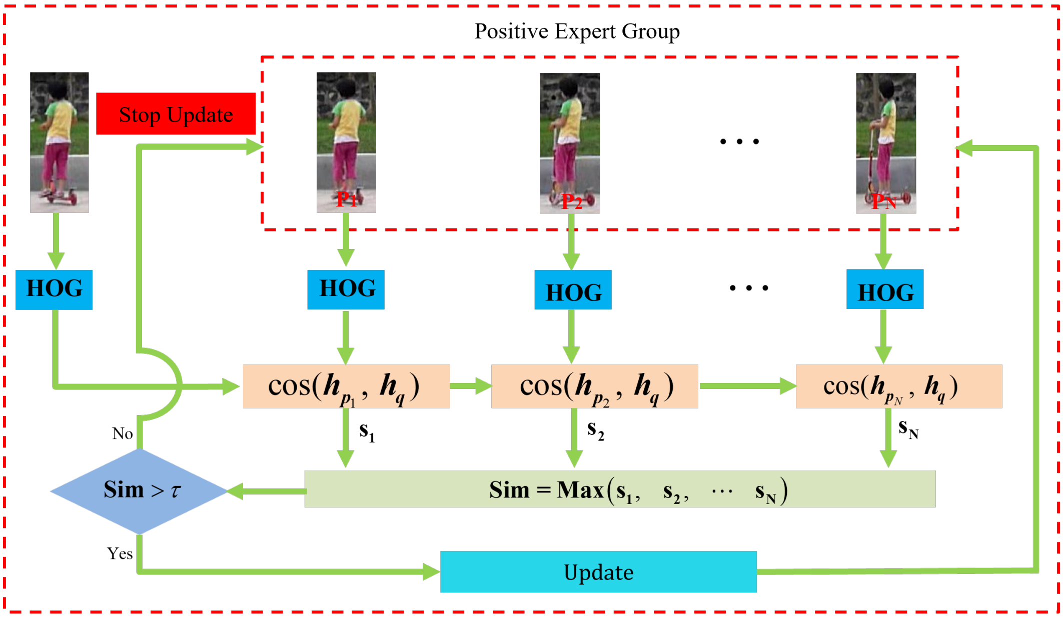


Figure 2. Positive expert group process flow diagram.

As illustrated in Figure 2, these slices serve as reference templates in subsequent frames. They enable the filter to choose the most reliable samples from the positive expert group during occlusions or disturbances. In processing the following frames, the filter first identifies the sample with the highest response value and extracts its features. Subsequently, the extracted features are compared with those from the positive expert group to determine the sample that most closely resembles a positive sample from the past. If the similarity exceeds a specified threshold, the sample is regarded as reliable and adopted as the object appearance for updating the filter in the current frame. Conversely, if the sample fails to meet the similarity threshold, it is not added to the positive expert group, thereby preventing filter degradation. The detailed implementation process is as follows:

For the initial frame, since there is no historical data, the positive expert group is entirely composed of patches from the first frame, i.e., $\boldsymbol{p}_n$=$\mathbf{vec}(\boldsymbol{T}^{(1)}(n{=}1,2,\cdots,N)$, where $\boldsymbol{p}_n$ represents the n-th column vector in the positive expert group matrix $\boldsymbol{P}$, and $\boldsymbol{T}^{(1)} \in \mathbb{R}^{u_H^{(1)} \times u_W^{(1)}}$ represents the patches of the object in the first frame (a four-dimensional tensor, where the first frame's object template $\mathcal{T}^{(1)} \in \mathbb{R}^{u_H^{(1)} \times u_W^{(1)} \times 3}$ is reformulated into matrix form $\boldsymbol{T}^{(1)}$ to reduce computation). Beginning with the second frame, it is assumed that the optimal candidate object derived from the correlation response is $\mathcal{Q}$ (for convenience in subsequent storage and computation, $\mathcal{Q}$ is converted to a grayscale image $\boldsymbol{Q} \in \mathbb{R}^{u_h^{(1)} \times u_w^{(1)}}$, and its column vector $\boldsymbol{q}$ = $vec(\boldsymbol{Q})$), from which HOG features are extracted from both $\boldsymbol{p}_n$ and $\boldsymbol{q}$ as:

$$\begin{cases} \boldsymbol{h}_{p_n} = \dfrac{\mathbf{HOG}(\boldsymbol{p}_n)}{\mathbf{max}(\mathbf{HOG}(\boldsymbol{p}_n))} \\ \boldsymbol{h}_q = \dfrac{\mathbf{HOG}(\boldsymbol{q})}{\mathbf{max}(\mathbf{HOG}(\boldsymbol{q}))} \end{cases}, \tag{3.8}$$

where $\mathbf{HOG}(\boldsymbol{q})$ represents the extracted $\mathbf{HOG}$-features of $\boldsymbol{q}$, and $\mathbf{max}(\mathbf{HOG}(\boldsymbol{q}))$ represents the peak value of the **HOG**-features. Finally, the maximum cosine similarity is computed to determine whether the object is occluded.

When $\mathbf{max}(\cos(\boldsymbol{h}_{p_n},\boldsymbol{h}_q))$ exceeds a certain threshold τ, the most dissimilar positive expert template from the second to the N-th patches in the positive expert group is eliminated. Subsequently, $\boldsymbol{q}$ replaces the eliminated positive expert template. If the similarity falls below the threshold, it is presumed that the object is occluded; consequently, updates to the training samples, filter, positive expert group, and background color histograms are withheld.

3.4. Solving the objective function filter for the tracking model

In this section, for resolving the tracking model, we draw upon [10], employ the Lagrange multiplier $\boldsymbol{L}^c$ along with a quadratic penalty constraint term, and develop the augmented Lagrangian function representation of the objective function as:

$$\begin{aligned}\mathcal{L}\left(\boldsymbol{G}^{\{c\}},\ \boldsymbol{W}^{\{c\}},\ \boldsymbol{L}^{\{c\}}|\boldsymbol{M}\right) = \frac{1}{2}\left\|\sum_{c=1}^{C}\boldsymbol{X}^{\{c\}} * \bar{\boldsymbol{G}}^{\{c\}} - \boldsymbol{Y}\right\|_2^2 + \frac{\lambda}{2}\sum_{c=1}^{C}\left\|\tilde{\boldsymbol{W}}\odot\boldsymbol{M}\odot\boldsymbol{W}^{\{c\}}\right\|_2^2 \\ +\beta\sum_{c=1}^{C}\mathbf{tr}\left\{\boldsymbol{L}_{\boldsymbol{W}^{\{c\}}}\left(\boldsymbol{G}^{\{c\}} - \boldsymbol{M}\odot\boldsymbol{W}^{\{c\}}\right)\right\} + \frac{\beta}{2}\sum_{c=1}^{C}\left\|\boldsymbol{G}^{\{c\}} - \boldsymbol{M}\odot\boldsymbol{W}^{\{c\}}\right\|_2^2,\end{aligned} \tag{3.9}$$

where $\mathbf{tr}\,(\boldsymbol{X})$ signifies the trace of the matrix $\boldsymbol{X}$, and β represents the coefficient associated with the quadratic penalty function.

Through the application of the spatial convolution theorem, Eq (3.9) is reformulated as:

$$\mathcal{L}\left(\hat{\boldsymbol{G}}^{\{c\}},\ \hat{\boldsymbol{W}}^{\{c\}},\ \hat{\boldsymbol{L}}^{\{c\}}|\boldsymbol{M}\right) = \underset{\hat{\boldsymbol{L}}^{\{c\}}}{\mathbf{max}}\ \underset{\hat{\boldsymbol{G}}^{\{c\}},\,\hat{\boldsymbol{W}}^{\{c\}}}{\mathbf{min}}\left\{\begin{aligned}&\frac{1}{2}\left\|\sum_{c=1}^{C}\hat{\boldsymbol{X}}^{\{c\}}\odot\left(\hat{\boldsymbol{G}}^{\{c\}}\right)^{*} - \hat{\boldsymbol{Y}}\right\|_2^2 + \frac{\lambda}{2}\sum_{c=1}^{C}\left\|\tilde{\boldsymbol{W}}\odot\boldsymbol{M}\odot\boldsymbol{W}^{\{c\}}\right\|_2^2 \\ &+\beta\sum_{c=1}^{C}\mathrm{tr}\left\{\hat{\boldsymbol{L}}_{\boldsymbol{W}^{\{c\}}}\left(\hat{\boldsymbol{G}}^{\{c\}} - \mathbf{fft2}\left(\boldsymbol{M}\odot\boldsymbol{W}^{\{c\}}\right)\right)\right\} \\ &+\frac{\beta}{2}\sum_{c=1}^{C}\left\|\hat{\boldsymbol{G}}^{\{c\}} - \mathbf{fft2}\left(\boldsymbol{M}\odot\boldsymbol{W}^{\{c\}}\right)\right\|_2^2\end{aligned}\right\} \tag{3.10}$$

where $\hat{\boldsymbol{X}}$ denotes the frequency domain signal of $\boldsymbol{X}$, and **fft2** represents the 2D Fourier transform operator.

Rewriting Eq (3.10) as a vector form, we obtain:

$$\mathcal{L}\left(\hat{\boldsymbol{g}}^{\{c\}},\ \boldsymbol{w}^{\{c\}},\ \hat{\boldsymbol{l}}^{\{c\}}|\boldsymbol{M}\right) = \underset{\hat{\boldsymbol{l}}^{\{c\}}}{\mathbf{max}}\left\{\underset{\hat{\boldsymbol{g}}^{\{c\}},\,\boldsymbol{w}^{\{c\}}}{\mathbf{min}}\left\{\begin{aligned}&\frac{1}{2}\left\|\sum_{c=1}^{C}(\hat{\boldsymbol{g}}^{\{c\}})^{*}\odot\hat{\boldsymbol{x}}^{\{c\}} - \hat{\boldsymbol{y}}\right\|_2^2 + \frac{\lambda}{2}\sum_{c=1}^{C}\left\|\mathbf{Diag}(\tilde{\boldsymbol{w}})\boldsymbol{P_m}\boldsymbol{w}^{\{c\}}\right\|_2^2 \\ &+\beta\sum_{c=1}^{C}\left\{\hat{\boldsymbol{l}}_{W^{\{c\}}}\left[\hat{\boldsymbol{g}}^{\{c\}} - \left(\boldsymbol{F}\boldsymbol{P_m}\boldsymbol{w}^{\{c\}}\right)\right]\right\} + \frac{\beta}{2}\sum_{c=1}^{C}\left\|\hat{\boldsymbol{g}}^{\{c\}} - \left(\boldsymbol{F}\boldsymbol{P_m}\boldsymbol{w}^{\{c\}}\right)\right\|_2^2\end{aligned}\right\}\right\}, \tag{3.11}$$

where $\hat{\boldsymbol{l}}^{\{c\}}$, $\hat{\boldsymbol{g}}^{\{c\}}$, and $\hat{\boldsymbol{w}}^{\{c\}}$ denote the vector forms of matrices $\hat{\boldsymbol{L}}^{\{c\}}$, $\hat{\boldsymbol{G}}^{\{c\}}$, and $\hat{\boldsymbol{W}}^{\{c\}}$; $\hat{\boldsymbol{x}}$=$\boldsymbol{F}\boldsymbol{x}$=**vec**(**fft2**($\boldsymbol{X}$)) (with $\boldsymbol{F} = \boldsymbol{F}_{\sqrt{D}}\otimes\boldsymbol{F}_{\sqrt{D}}$, $\otimes$ representing the Kronecker product operator, $\boldsymbol{F}_{\sqrt{D}} \in \mathbb{C}^{\sqrt{D}\times\sqrt{D}}$ representing the discrete Fourier transform matrix with dimensions $\sqrt{D}\times\sqrt{D}$, where D=$H\times W$), $\boldsymbol{P_m} \in \mathbb{R}^{D\times D}$ is a diagonal matrix with diagonal elements $\boldsymbol{P_m}(i,i)$=$\begin{cases}1 & \boldsymbol{m}(i)\neq 0\\ 0 & others\end{cases}$ i=$1, 2, \cdots, D$, and $\mathbf{Diag}(\tilde{\boldsymbol{w}})$ refers to the operator that inserts the vector elements $\tilde{\boldsymbol{w}}$ into the diagonal positions of a zero matrix.

The sub-problems are solved by iterative minimization using the alternating direction method:

$$\begin{cases} (\hat{\boldsymbol{g}}^{\{c\}})^{(i+1)} = \underset{\hat{\boldsymbol{g}}^{\{c\}}}{\mathbf{argmin}} \mathcal{L}_{\hat{\boldsymbol{g}}^{\{c\}}} \left(\hat{\boldsymbol{g}}^{\{c\}} \middle| (\boldsymbol{w}^{\{c\}})^{(i+1)}, (\hat{\boldsymbol{l}}^{\{c\}})^{(i)}, \boldsymbol{M} \right) \\ (\boldsymbol{w}^{\{c\}})^{(i+1)} = \underset{\boldsymbol{w}^{\{c\}}}{\mathbf{argmin}} \mathcal{L}_{\boldsymbol{w}^{\{c\}}} \left(\boldsymbol{w}^{\{c\}} \middle| (\hat{\boldsymbol{g}}^{\{c\}})^{(i+1)}, (\hat{\boldsymbol{l}}^{\{c\}})^{(i)}, \boldsymbol{M} \right) \\ (\hat{\boldsymbol{l}}^{\{c\}})^{(i+1)} = \underset{\hat{\boldsymbol{l}}^{\{c\}}}{\mathbf{argmin}} \mathcal{L}_{\hat{\boldsymbol{l}}^{\{c\}}} \left(\hat{\boldsymbol{l}}^{\{c\}} \middle| (\hat{\boldsymbol{g}}^{\{c\}})^{(i+1)}, (\boldsymbol{w}^{\{c\}})^{(i+1)}, \boldsymbol{M} \right) \end{cases} \tag{3.12}$$

where $\mathcal{L}_{\hat{\boldsymbol{g}}^{\{c\}}} \left(\hat{\boldsymbol{g}}^{\{c\}} \middle| (\boldsymbol{w}^{\{c\}})^{(i+1)}, (\hat{\boldsymbol{l}}^{\{c\}})^{(i)}, \boldsymbol{M} \right)$, $\mathcal{L}_{\boldsymbol{w}^{\{c\}}} \left(\boldsymbol{w}^{\{c\}} \middle| (\hat{\boldsymbol{g}}^{\{c\}})^{(i+1)}, (\hat{\boldsymbol{l}}^{\{c\}})^{(i)}, \boldsymbol{M} \right)$, and $\mathcal{L}_{\hat{\boldsymbol{l}}^{\{c\}}} \left(\hat{\boldsymbol{l}}^{\{c\}} \middle| (\hat{\boldsymbol{g}}^{\{c\}})^{(i+1)}, (\boldsymbol{w}^{\{c\}})^{(i+1)}, \boldsymbol{M} \right)$ denote the objective functions of the individual sub-problems.

3.4.1. Solving the $\hat{\boldsymbol{g}}^{\{c\}}$-subproblem

The components pertaining to $\hat{\boldsymbol{g}}^{\{c\}}$ within the augmented Lagrangian function establish the sub-objective function for $\hat{\boldsymbol{g}}^{\{c\}}$, as outlined below:

$$\begin{aligned} &\mathcal{L}_{\hat{\boldsymbol{g}}^{\{c\}}} \left(\hat{\boldsymbol{g}}^{\{c\}} | (\boldsymbol{w}^{\{c\}})^{(i)},\ (\hat{\boldsymbol{l}}^{\{c\}})^{(i)}, \boldsymbol{M} \right) \\ &= \underset{\hat{\boldsymbol{g}}^{\{c\}}}{min} \frac{1}{2} \left\| \sum_{c=1}^{C} (\hat{\boldsymbol{g}}^{\{c\}})^{*} \odot \hat{\boldsymbol{x}}^{\{c\}} - \hat{\boldsymbol{y}} \right\|_2^2 + \frac{\beta}{2} \left\| (\hat{\boldsymbol{l}}^{\{c\}})^{(i)} \right\|_2^2 + \beta \sum_{c=1}^{C} \left\{ \hat{\boldsymbol{l}}^{H} \left[\hat{\boldsymbol{g}}^{\{c\}} - \left(\boldsymbol{F}\boldsymbol{P}_{\boldsymbol{m}}(\boldsymbol{w}^{\{c\}})^{(i)} \right) \right] \right\} + \frac{\beta}{2} \sum_{c=1}^{C} \left\| \hat{\boldsymbol{g}}^{\{c\}} - \left(\boldsymbol{F}\boldsymbol{P}_{\boldsymbol{m}}(\boldsymbol{w}^{\{c\}})^{(i)} \right) \right\|_2^2 - \frac{\beta}{2} \left\| (\hat{\boldsymbol{l}}^{\{c\}})^{(i)} \right\|_2^2 \\ &= \underset{\hat{\boldsymbol{g}}^{\{c\}}}{min} \frac{1}{2} \left\| \sum_{c=1}^{C} (\hat{\boldsymbol{g}}^{\{c\}})^{*} \odot \hat{\boldsymbol{x}}^{\{c\}} - \hat{\boldsymbol{y}} \right\|_2^2 + \frac{\beta}{2} \sum_{c=1}^{C} \left\| \hat{\boldsymbol{g}}^{\{c\}} - \left(\boldsymbol{F}\boldsymbol{P}_{\boldsymbol{m}}(\boldsymbol{w}^{\{c\}})^{(i)} \right) + (\hat{\boldsymbol{l}}^{\{c\}})^{(i)} \right\|_2^2 - \frac{\beta}{2} \left\| (\hat{\boldsymbol{l}}^{\{c\}})^{(i)} \right\|_2^2 \\ &= \underset{\hat{\boldsymbol{g}}^{\{c\}}}{min} \frac{1}{2} \left\| \sum_{c=1}^{C} \hat{\boldsymbol{g}}^{\{c\}} \odot \left(\hat{\boldsymbol{x}}^{\{c\}} \right)^{*} - \hat{\boldsymbol{y}}^{*} \right\|_2^2 + \frac{\beta}{2} \sum_{c=1}^{C} \left\| \hat{\boldsymbol{g}}^{\{c\}} - \left(\boldsymbol{F}\boldsymbol{P}_{\boldsymbol{m}}(\boldsymbol{w}^{\{c\}})^{(i)} \right) + (\hat{\boldsymbol{l}}^{\{c\}})^{(i)} \right\|_2^2 - \frac{\beta}{2} \left\| (\hat{\boldsymbol{l}}^{\{c\}})^{(i)} \right\|_2^2 . \end{aligned} \tag{3.13}$$

Differentiating the objective function concerning $\hat{\boldsymbol{g}}^{\{c\}}$ and setting it equal to zero, we obtain:

$$\sum_{c=1}^{C} \hat{\boldsymbol{x}}^{\{c\}} \odot (\hat{\boldsymbol{x}}^{\{c\}})^{*} \odot \hat{\boldsymbol{g}}^{\{c\}} - \hat{\boldsymbol{x}}^{\{c\}} \odot \hat{\boldsymbol{y}}^{*} + \beta \hat{\boldsymbol{g}}^{\{c\}} - \beta \boldsymbol{F}\boldsymbol{P}_{\boldsymbol{m}}(\boldsymbol{w}^{\{c\}})^{(i)} + \beta (\hat{\boldsymbol{l}}^{\{c\}})^{(i)} = \mathbf{0}. \tag{3.14}$$

By combining similar terms for $\hat{\boldsymbol{g}}^{\{c\}}$, we get:

$$\left\{ \hat{\boldsymbol{g}}^{\{c\}} \right\}^{(i+1)} = \frac{\hat{\boldsymbol{x}}^{\{c\}} \odot \hat{\boldsymbol{y}}^{*} + \beta \boldsymbol{F}\boldsymbol{P}_{\boldsymbol{m}}(\boldsymbol{w}^{\{c\}})^{(i)} - \beta (\hat{\boldsymbol{l}}^{\{c\}})^{(i)}}{\sum_{c=1}^{C} \hat{\boldsymbol{x}}^{\{c\}} \odot (\hat{\boldsymbol{x}}^{\{c\}})^{*} + \beta}, \tag{3.15}$$

where the division sign in the above equation indicates element-wise division.

Writing Eq (3.15) in matrix form gives:

$$\left\{ \hat{\boldsymbol{G}}^{\{c\}} \right\}^{(i+1)} = \frac{\hat{\boldsymbol{X}}^{\{c\}} \odot \hat{\boldsymbol{Y}}^{*} + \beta \mathbf{mat} \left(\boldsymbol{F}\boldsymbol{P}_{\boldsymbol{m}}(\boldsymbol{w}^{\{c\}})^{(i)} \right) - \beta (\hat{\boldsymbol{L}}^{\{c\}})^{(i)}}{\sum_{c=1}^{C} \hat{\boldsymbol{X}}^{\{c\}} \odot (\hat{\boldsymbol{X}}^{\{c\}})^{*} + \beta}, \tag{3.16}$$

where **mat** denotes the operator responsible for converting a vector into a matrix.

3.4.2. Solving the $\boldsymbol{w}^{\{c\}}$-subproblem

The components associated with $\boldsymbol{w}^{\{c\}}$ in the augmented Lagrangian function constitute the sub-objective function for $\boldsymbol{w}^{\{c\}}$, detailed below:

$$\begin{aligned}
&\mathcal{L}_{\boldsymbol{w}^{\{c\}}}\left(\boldsymbol{h}^{\{c\}}|(\boldsymbol{g}^{\{c\}})^{(i+1)},\ (\hat{\boldsymbol{l}}^{\{c\}})^{(i)},\boldsymbol{M}\right)\\
&=\min_{\boldsymbol{w}^{\{c\}}}\left\{\frac{\lambda}{2}\sum_{c=1}^{C}\left\|\mathbf{Diag}(\tilde{\boldsymbol{w}})\boldsymbol{P_m}\boldsymbol{w}^{\{c\}}\right\|_2^2+\beta\sum_{c=1}^{C}\left\{\left((\hat{\boldsymbol{l}}^{\{c\}})^{(i)}\right)^{H}\left[(\hat{\boldsymbol{g}}^{\{c\}})^{(i+1)}-\left(\boldsymbol{F}\boldsymbol{P_m}\boldsymbol{w}^{\{c\}}\right)\right]\right\}+\frac{\beta}{2}\sum_{c=1}^{C}\left\|(\hat{\boldsymbol{g}}^{\{c\}})^{(i+1)}-\left(\boldsymbol{F}\boldsymbol{P_m}\boldsymbol{w}^{\{c\}}\right)\right\|_2^2\right\}\\
&=\min_{\boldsymbol{w}^{\{c\}}}\left\{-\frac{\beta}{2}\left\|(\hat{\boldsymbol{l}}^{\{c\}})^{(i)}\right\|_2^2+\frac{\lambda}{2}\sum_{c=1}^{C}\left\|\mathbf{Diag}(\tilde{\boldsymbol{w}})\boldsymbol{P_m}\boldsymbol{w}^{\{c\}}\right\|_2^2+\frac{\beta}{2}\sum_{c=1}^{C}\left\|\left(\boldsymbol{F}\boldsymbol{P_m}\boldsymbol{w}^{\{c\}}\right)-(\hat{\boldsymbol{g}}^{\{c\}})^{(i+1)}-(\hat{\boldsymbol{l}}^{\{c\}})^{(i)}\right\|_2^2\right\}.
\end{aligned} \tag{3.17}$$

Differentiating the objective function concerning $\boldsymbol{w}^{\{c\}}$ and setting it equal to zero, we obtain:

$$\begin{aligned}
\frac{\partial\mathcal{L}_{\boldsymbol{w}^{\{c\}}}}{\partial\boldsymbol{w}^{\{c\}}}&=\lambda(\mathbf{Diag}(\tilde{\boldsymbol{w}})\boldsymbol{P_m})^{H}\mathbf{Diag}(\tilde{\boldsymbol{w}})\boldsymbol{P_m}\boldsymbol{w}^{\{c\}}+\beta(\boldsymbol{F}\boldsymbol{P_m})^{W}\left[\left(\boldsymbol{F}\boldsymbol{P_m}\boldsymbol{w}^{\{c\}}\right)-(\hat{\boldsymbol{g}}^{\{c\}})^{(i+1)}-(\hat{\boldsymbol{l}}^{\{c\}})^{(i)}\right]\\
&=\lambda\boldsymbol{P_m}\mathbf{Diag}(\tilde{\boldsymbol{w}})\mathbf{Diag}(\tilde{\boldsymbol{w}})\boldsymbol{P_m}\boldsymbol{w}^{\{c\}}+\beta(\boldsymbol{F}\boldsymbol{P_m})^{H}\left[\left(\boldsymbol{F}\boldsymbol{P_m}\boldsymbol{w}^{\{c\}}\right)-(\hat{\boldsymbol{g}}^{\{c\}})^{(i+1)}-(\hat{\boldsymbol{l}}^{\{c\}})^{(i)}\right]\\
&=\lambda\mathbf{Diag}(\tilde{\boldsymbol{w}}\odot\tilde{\boldsymbol{w}}\odot\boldsymbol{m})\boldsymbol{w}^{\{c\}}+\beta\boldsymbol{P_m}F^{H}\left[\left(\boldsymbol{F}\boldsymbol{P_m}\boldsymbol{w}^{\{c\}}\right)-(\hat{\boldsymbol{g}}^{\{c\}})^{(i+1)}-(\hat{\boldsymbol{l}}^{\{c\}})^{(i)}\right]=\mathbf{0},
\end{aligned} \tag{3.18}$$

where $\boldsymbol{F}$ satisfies $\boldsymbol{F}^{H}\boldsymbol{F}{=}D\boldsymbol{I}_D$, $\boldsymbol{F}^{H}$ is the conjugate transpose matrix of the original matrix, and $\boldsymbol{I}_D \in \mathbb{R}^{D\times D}$ is the identity matrix.

Therefore, we have:

$$\lambda\tilde{\boldsymbol{w}}\odot\tilde{\boldsymbol{w}}\odot\boldsymbol{m}\odot\boldsymbol{w}^{\{c\}}{+}\beta D\boldsymbol{P_m}\boldsymbol{w}^{\{c\}}{=}\beta\boldsymbol{P_m}\boldsymbol{F}^{H}((\hat{\boldsymbol{g}}^{\{c\}})^{(i+1)}+(\hat{\boldsymbol{l}}^{\{c\}})^{(i)}). \tag{3.19}$$

$\boldsymbol{F}^{H}$ satisfies $\boldsymbol{x}{=}\frac{\boldsymbol{F}^{H}\hat{\boldsymbol{x}}}{D}{=}\mathbf{vec}(\mathbf{ifft2}(\hat{\boldsymbol{X}}))$ (where **ifft2** represents the 2D inverse Fourier transform operator), then we obtain:

$$\lambda\tilde{\boldsymbol{w}}\odot\tilde{\boldsymbol{w}}\odot\boldsymbol{m}\odot\boldsymbol{w}^{\{c\}}{+}\beta D\boldsymbol{m}\odot\boldsymbol{w}^{\{c\}}{=}\beta D\boldsymbol{m}\odot((\boldsymbol{g}^{\{c\}})^{(i+1)}+(\boldsymbol{l}^{\{c\}})^{(i)}) \tag{3.20}$$

We obtain:

$$\boldsymbol{w}^{\{c\}}{=}\frac{\beta D\boldsymbol{m}\odot(\boldsymbol{g}^{\{c\}})^{(i+1)}+(\boldsymbol{l}^{\{c\}})^{(i)})}{\lambda\tilde{\boldsymbol{w}}\odot\tilde{\boldsymbol{w}}\odot\boldsymbol{m}{+}\beta D\boldsymbol{m}}. \tag{3.21}$$

3.4.3. Solving the $\hat{\boldsymbol{l}}^{\{c\}}$-subproblem

The objective function of the $\hat{\boldsymbol{l}}^{\{c\}}$-subproblem is:

$$\mathcal{L}_{\hat{\boldsymbol{l}}^{\{c\}}}\left(\hat{\boldsymbol{l}}^{\{c\}}|(\hat{\boldsymbol{g}}^{\{c\}})^{(i+1)},\ (\boldsymbol{w}^{\{c\}})^{(i+1)},\boldsymbol{M}\right){=}\max_{\hat{\boldsymbol{l}}^{\{c\}}}\left\{\hat{\boldsymbol{l}}_{W^{\{c\}}}\left((\hat{\boldsymbol{g}}^{\{c\}})^{(i+1)}-\boldsymbol{F}\boldsymbol{P_m}(\boldsymbol{w}^{\{c\}})^{(i+1)}\right)\right\}. \tag{3.22}$$

Using gradient ascent, we obtain:

$$(\hat{\boldsymbol{l}}^{\{c\}})^{(i+1)}{=}(\hat{\boldsymbol{l}}^{\{c\}})^{(i)}{+}\mu\left((\hat{\boldsymbol{g}}^{\{c\}})^{(i+1)}-\boldsymbol{F}\boldsymbol{P_m}(\boldsymbol{w}^{\{c\}})^{(i+1)}\right). \tag{3.23}$$

where the value of μ is set to 0.02.

The algorithm is summarized in **Algorithm 1** .

Algorithm 1: The algorithm proposed in this paper

```
Input: First-frame training sample $\mathcal{X}^{(t)} \in \mathbb{R}^{R_h \times C_w \times 3}$, object template $\boldsymbol{T}^{(1)} \in \mathbb{R}^{u_h^{(1)} \times u_w^{(1)}}$,
       current-frame test sample $\mathcal{Z}$, positive expert group $\boldsymbol{P}$, linear interpolation learning rate $\eta$,
       object color histogram $(\boldsymbol{c}^o)^{(t)}$, background color histogram $\left(\boldsymbol{c}^b\right)^{(t)}$.
Output: Predicted object position and optimal object size $\left[u_h^{(t)} \times u_w^{(t)}\right]$.
Compute $\hat{\boldsymbol{G}}^{\{c\}}$ using Eq (3.16), build the filter group $\mathcal{G}^{(1)} = \{\hat{\boldsymbol{G}}^{\{c\}}\}\big|_{c=1,2,\cdots,C}$, and determine the
  spatial saliency matrix using Eq (2.8);
for $t = 2$ to $T$ do
    Extract features $\hat{\boldsymbol{Z}}^{\{c\}}\big|_{\mathcal{Z}}$ from $\mathcal{Z} \in \mathbb{R}^{R \times C \times 3}$;
    Estimate the spatial saliency perception matrix $\boldsymbol{M}$ for the current frame sample based on
      $(\boldsymbol{c}^o)^{(t-1)}$ and $\left(\boldsymbol{c}^b\right)^{(t-1)}$;
    Learn weights $s^{\{c\}}$ according to Eq (3.4), and zero out channel features with low channel
      weights as shown in Figure 1;
    Determine the response $\boldsymbol{R} = \mathbf{real}\left\{\sum_{c=1}^{C} \mathbf{ifft2}\left\{\left(s^{\{c\}}\hat{\boldsymbol{Z}}^{\{c\}}\big|_{\mathcal{Z}}\right) \odot \left(\hat{\boldsymbol{G}}^{\{c\}}\right)^{*}\right\}\right\}$ for sample $\mathcal{Z}$;
    Identify the optimal position using the maximum response value;
    Determine the optimal scale $s_b$ based on the DSST algorithm [49];
    Set the object size $u_h^{(t)} = s_b u_h^{(t-1)}, u_w^{(t)}=s_b u_w^{(t-1)}$;
    Determine the best candidate samples $\boldsymbol{Q}$ and $\boldsymbol{q} = \mathbf{vec}(\boldsymbol{Q})$ based on the optimal position and
      $u_h^{(t)}, u_w^{(t)}$;
    for $n = 1$ to $N$ do
        $Simi(n) = \cos(\boldsymbol{h}_{\boldsymbol{p}_n}, \boldsymbol{h}_{\boldsymbol{q}})$;
    end
    if $\max(Simi) < \tau$ then
        Do not update foreground and background color histograms: $(\boldsymbol{c}^o)^{(t)} = (\boldsymbol{c}^o)^{(t-1)}$,
          $\left(\boldsymbol{c}^b\right)^{(t)} = \left(\boldsymbol{c}^b\right)^{(t-1)}$;
        Do not update training sample: $\mathcal{X}^{(t)} = \mathcal{X}^{(t-1)}$;
        Do not update the correlation filter: $\mathcal{G}^{(t)} = \mathcal{G}^{(t-1)}$;
    else
        Retrieve the current frame sample $\tilde{\mathcal{X}}$, centered on $\boldsymbol{Q}$, and sized $R \times C$;
        Compute the foreground and background histograms $\tilde{\boldsymbol{c}}^o$ and $\tilde{\boldsymbol{c}}^b$ for the best training
          sample $\mathcal{X}^{(t)}$;
        Refresh the foreground and background histograms:
          $(\boldsymbol{c}^o)^{(t)}= (1-\eta)(\boldsymbol{c}^o)^{(t-1)}+\eta\tilde{\boldsymbol{c}}^o, \left(\boldsymbol{c}^b\right)^{(t)}= (1-\eta)\left(\boldsymbol{c}^b\right)^{(t-1)}+\eta\tilde{\boldsymbol{c}}^b$;
        Update the spatial saliency matrix $\boldsymbol{M}$;
        Update filter $\hat{\boldsymbol{G}}^{\{c\}}$ based on the current frame sample and Eq (3.16), and form the filter
          group tensor $\tilde{\mathcal{G}} = \{\hat{\boldsymbol{G}}^{\{c\}}\}\big|_{c=1,2,\cdots,C}$ with $\hat{\boldsymbol{G}}^{\{c\}}$;
        Update the filter $\mathcal{G}^{(t)}= (1-\eta)\mathcal{G}^{(t-1)}+\eta\tilde{\mathcal{G}}$;
        $\boldsymbol{p}_i = \boldsymbol{q}$ (where $i$ represents the template with the lowest similarity to $\boldsymbol{q}$);
    end
end
```

4. Experiments

In the experimental section of this paper, we chose four representative benchmark datasets: OTB100, DTB70, UAV123, and UAV20L. These datasets encompass tracking challenges from various perspectives, complexities, and application scenarios, both from the ground and aerial views, allowing for a thorough and systematic evaluation of the proposed object-tracking algorithm. First, we performed a detailed comparison of the performance of the SSACF algorithm and other mainstream trackers on these four datasets. Subsequently, we selected several typical video sequences from the UAV123 and UAV20L datasets to qualitatively analyze SSACF's tracking performance, highlighting its characteristics in various environments. Ultimately, we conducted a series of ablation studies on the OTB100 dataset to assess the distinct contributions of each component of SSACF towards enhancing tracking performance.

The experiment was conducted on a platform with an AMD Ryzen 7 7735H processor (3.20 GHz base frequency, integrated Radeon graphics), 16.0 GB of system memory, and a 64-bit x64 architecture. All algorithms were implemented using MATLAB R2023a. The key parameter configurations are as follows: the balance coefficients θ_1 and θ_2 in Eq (3.6) are set to 0.00009 and 50, respectively; the reliability evaluation threshold τ in Section 3.3 is set to 0.70; in the objective function in Section 3.4, the regularization parameter λ is set to 0.05, the quadratic penalty coefficient β is set to 3, and the update step size μ is set to 0.02.

4.1. Quantitative analysis

4.1.1. A quantitative analysis of the UAV123 dataset

The UAV123 dataset is a comprehensive, large-scale dataset created specifically for tracking in aerial videos, composed of 123 video sequences with over 110,000 frames, making it one of the most biggest aerial tracking datasets available. The sequences in UAV123 cover a variety of objects, including vehicles, pedestrians and buildings, filmed from multiple angles and heights, with complex conditions such as dynamic backgrounds, occlusion, and rotation. Unlike traditional ground-based datasets, UAV123 simulates natural aerial surveillance and tracking tasks from a drone perspective, which enables a better assessment of tracking algorithm performance in aerial environments. All sequences in the dataset come with detailed bounding box annotations and are categorized according to various attribute challenges, enabling researchers to test algorithm performance under specific conditions.

In this experiment, we will use the UAV123 dataset to evaluate the robustness and accuracy of the SSACF algorithm in aerial scenarios. UAV123 offers 12 attribute categories for different visual challenges, including Camera Motion (CM), Full Occlusion (FO), Similar Object (SO), Illumination Variation (IV), Viewpoint Change (VC), Partial Occlusion (PO), Scale Variation (SV), Aspect Ratio Change (ARC), Out-of-View (OV), Fast Motion (FM), Background Clutter (BC), and Low Resolution (LR). These attributes cover various visual uncertainties encountered during tracking, providing a comprehensive reference for evaluating tracker performance under different conditions. Furthermore, a qualitative comparison is conducted between the proposed tracker and seven other SOTA trackers, including STRCF [36], AutoTrack [37], BACF [20], MCCT_H [50], ARCF_H [51], Staple [19], and CSR-DCF [9].

As shown in Figure 3, SSACF demonstrates excellent overall performance, ranking first in both accuracy and success rate, with scores of 0.774 and 0.603, respectively. Regarding different attribute challenges, SSACF achieved accuracy scores of 0.718 and 0.701 under VC and BC conditions, significantly outperforming other tracking algorithms. This indicates that SSACF performs excellently in handling viewpoint changes and background clutter, making it especially suitable for object tracking in complex environments from a UAV perspective. Additionally, under SV and SO conditions, SSACF exhibited high accuracy, highlighting the algorithm's stability in handling dynamic object scale adjustments. SSACF similarly maintained high success rates under multiple challenging conditions in UAV123, particularly under SO and VC conditions, with success rates of 0.612 and 0.506, respectively. This performance demonstrates SSACF's adaptability and stability in object clutter and viewpoint changes. SSACF demonstrated stable performance across most dataset attributes, outperforming other competing trackers.

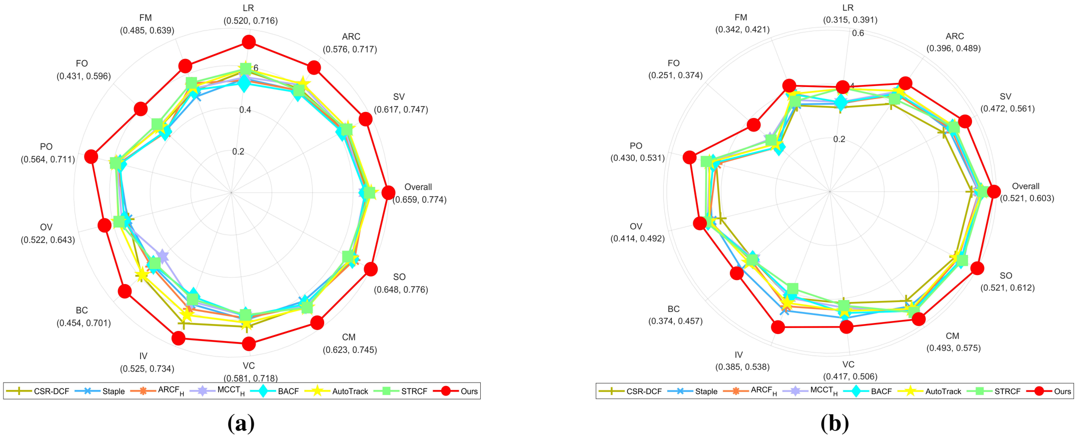
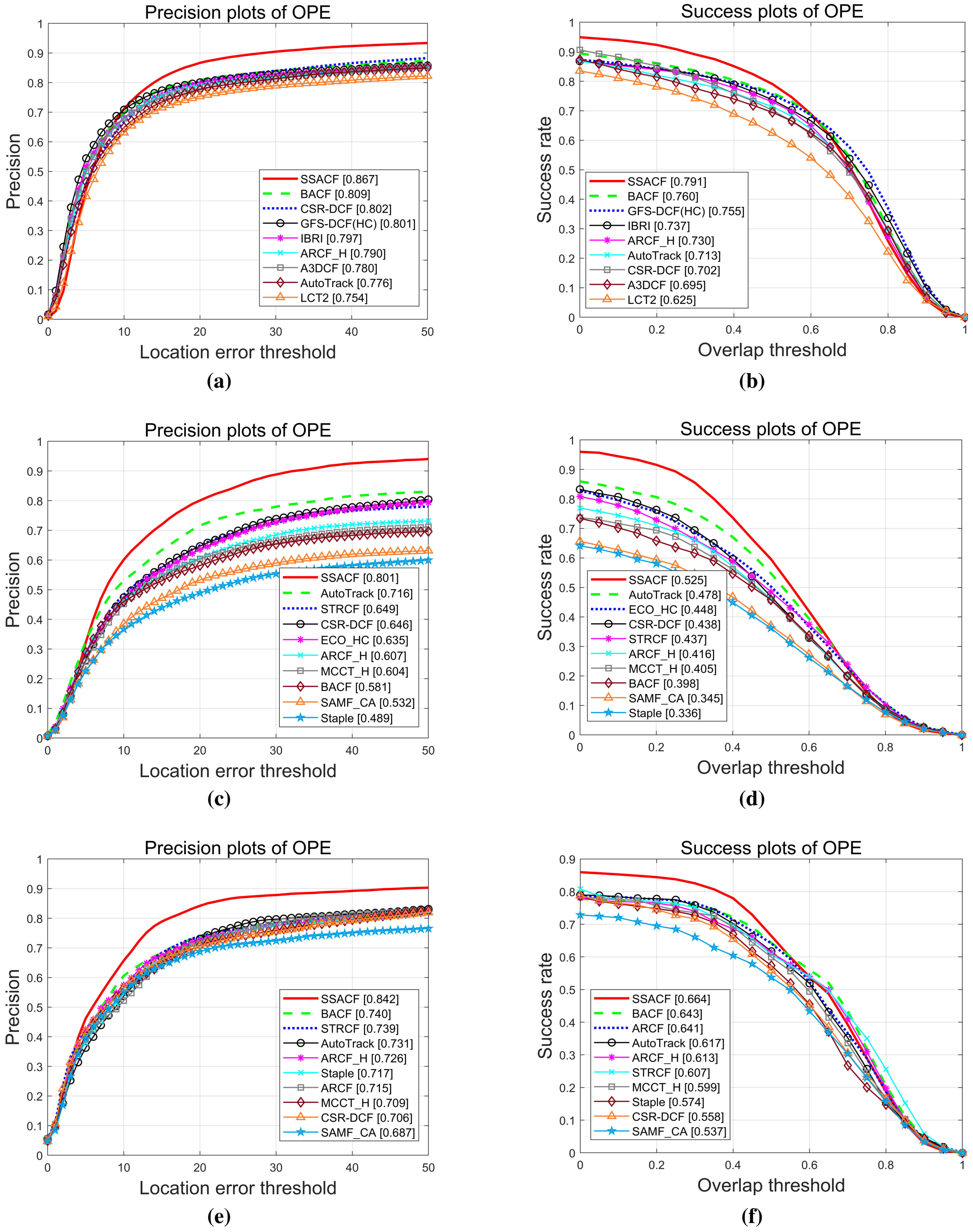


Figure 3. Radar chart of precision and success rate for SSACF and other trackers on the UAV123 dataset. (a) The precision. (b)The success rate.

4.1.2. A quantitative analysis of the OTB100 dataset

The OTB100 dataset ranks as one of the earliest and most extensively used benchmark datasets in the field of object tracking. It comprises 100 video sequences that vary in length and present a variety of typical object-tracking challenges. Originating primarily from ground-level perspectives, these sequences showcase a wide array of scenes and object types such as pedestrians, animals, vehicles, and handheld items. The dataset serves as a comprehensive benchmark for testing the efficacy of different tracking algorithms across various scenarios, establishing OTB100 as an essential tool for evaluating the robustness, precision, and flexibility of these algorithms. Detailed annotations are provided along with the dataset, and the primary evaluation metrics include the success rate and accuracy per attribute. In the attribute analysis, the OTB100 benchmark categorizes video sequences into 11 challenging attributes based on visual interference factors, namely Scale Variation (SV), Low Resolution (LR), Motion Blur (MB), Out-of-View (OV), Background Clutter (BC), Deformation (DEF), In-Plane Rotation (IPR), Illumination Variation (IV), Occlusion (OCC), Fast Motion (FM), and Out-of-Plane Rotation (OPR). Furthermore, we conducted a detailed comparison of the proposed SSACF with nine other SOTA trackers, including BACF [20], CSR-DCF [9], GFS-DCF(HC) [8], IBRI [52], ARCF_H [51], A3DCF [53], AutoTrack [37], and LCT2 [54].



(a) **(b)**

(c) **(d)**

(e) **(f)**

Figure 4. Comprehensive comparison of algorithms's precision and success rate. (a) The precision on the OTB100 dataset. (b) The success rate on the OTB100 dataset. (c) The precision on the DTB70 dataset. (d) The success rate on the DTB70 dataset. (e) The precision on the UAV20L dataset. (f) The success rate on the UAV20L dataset.

Table 1. The precision of 11 challenging attributes on the OTB100 dataset.

Attribute	SSACF	BACF	CSR-DCF	GFS-DCF (HC)	IBRI	ARCF_H	A3DCF	AutoTrack	LCT2
IV	0.853	0.782	0.779	0.745	0.768	0.769	0.765	0.745	0.721
OPR	0.831	0.767	0.760	0.753	0.747	0.737	0.742	0.734	0.733
SV	0.823	0.755	0.739	0.784	0.744	0.736	0.749	0.715	0.665
OCC	0.860	0.714	0.700	0.735	0.710	0.676	0.735	0.693	0.661
DEF	0.849	0.747	0.777	0.693	0.748	0.740	0.689	0.724	0.666
MB	0.856	0.716	0.741	0.765	0.713	0.718	0.753	0.703	0.641
FM	0.848	0.787	0.766	0.772	0.727	0.758	0.782	0.761	0.681
IPR	0.817	0.777	0.781	0.746	0.741	0.750	0.743	0.741	0.765
OV	0.821	0.748	0.691	0.772	0.652	0.674	0.743	0.736	0.593
BC	0.876	0.801	0.778	0.767	0.788	0.803	0.761	0.761	0.734
LR	0.796	0.741	0.677	0.708	0.741	0.692	0.759	0.763	0.537

Table 2. The success rate of 11 challenging attributes on the OTB100 dataset.

Attribute	SSACF	BACF	CSR-DCF	GFS-DCF (HC)	IBRI	ARCF_H	A3DCF	AutoTrack	LCT2
IV	0.803	0.756	0.726	0.720	0.730	0.746	0.687	0.726	0.592
OPR	0.749	0.695	0.644	0.691	0.674	0.649	0.622	0.653	0.602
SV	0.722	0.686	0.605	0.723	0.664	0.642	0.643	0.633	0.464
OCC	0.783	0.676	0.632	0.700	0.662	0.626	0.649	0.650	0.561
DEF	0.760	0.671	0.681	0.632	0.679	0.663	0.560	0.670	0.564
MB	0.829	0.710	0.711	0.752	0.685	0.705	0.674	0.683	0.617
FM	0.797	0.759	0.704	0.747	0.694	0.730	0.724	0.708	0.613
IPR	0.714	0.697	0.638	0.670	0.671	0.654	0.644	0.644	0.629
OV	0.735	0.698	0.582	0.727	0.621	0.622	0.656	0.678	0.531
BC	0.801	0.771	0.705	0.731	0.748	0.762	0.660	0.722	0.663
LR	0.618	0.663	0.434	0.632	0.666	0.568	0.700	0.669	0.295

Tables 1 and 2 show the performance of SSACF and the other nine advanced trackers in accuracy and success rate evaluations based on attributes. The best three performances are distinguished by the colors red, green, and blue. As shown in Table 1, SSACF exhibited outstanding performance in most attributes, particularly in BC and MB, where its accuracy reached 0.876 and 0.858, respectively. SSACF maintained a high tracking accuracy compared to other algorithms in these specific interference conditions. Moreover, SSACF's performance was slightly lower under LR conditions, reaching only 0.796. SSACF's accuracy metrics outperformed other algorithms in most attributes, showcasing its strong adaptability to different environments. As shown in Table 2, SSACF also performed exceptionally well in terms of success rate in OTB100, particularly under MB conditions, with a success rate of 0.803. This result suggests that, compared to other trackers, SSACF can more effectively handle common issues, such as object variations in the scene. Combining the visual rankings shown in Figure 4(a),(b), SSACF ranks highly in accuracy and success rate, achieving 0.867 and 0.631, respectively, fully showcasing its stability and reliability on the OTB100 benchmark.

4.1.3. A quantitative analysis of the DTB70 dataset

The DTB70 dataset is a benchmark specifically designed for drone viewpoints, containing 70 challenging sequences focused on UAV tracking tasks. The video sequences in DTB70 encompass a range of complex factors, simulating the high-dynamic environments commonly encountered in real drone tracking scenarios. This dataset is particularly apt for evaluating tracking algorithms' performance in handling high-frequency motion, environmental vibrations, and changes in viewpoint, thereby confirming their suitability for UAV applications. All sequences in DTB70 are accurately annotated using a similar evaluation method to the OTB dataset, enabling direct comparison with results from other datasets.

In this experiment, we utilize the DTB70 dataset to assess the robustness and flexibility of the algorithm in a UAV environment. The attribute annotations in DTB70 cover 11 visual challenges, slightly differing from OTB100, including Background Clutter (BC), Motion Blur (MB), Fast Camera Motion (FCM), Out-of-View (OV), In-Plane Rotation (IPR), Deformation (DEF), Aspect Ratio Variation (ARV), Scale Variation (SV), Occlusion (OCC), Small Object Appearance (SOA), and Out-of-Plane Rotation (OPR). The trackers compared include ECO_H [22], AutoTrack [37], BACF [20], ARCF_H [51], CSR-DCF [9], MCCT_H [50], SAMF_CA [55], Staple [19], and STRCF [36].

Table 3. The precision of 11 challenging attributes on the DTB70 dataset.

Attribute	SSACF	ECO_H	AutoTrack	BACF	ARCF_H	CSR-DCF	MCCT_H	SAMF_CA	Staple	STRCF
SV	0.725	0.530	0.688	0.533	0.560	0.663	0.643	0.490	0.489	0.568
ARV	0.686	0.494	0.605	0.392	0.431	0.551	0.495	0.428	0.430	0.492
OCC	0.795	0.648	0.631	0.515	0.546	0.617	0.570	0.560	0.528	0.617
DEF	0.728	0.557	0.670	0.448	0.427	0.561	0.550	0.408	0.419	0.554
FCM	0.851	0.677	0.744	0.622	0.654	0.711	0.621	0.537	0.494	0.713
IPR	0.750	0.557	0.684	0.534	0.547	0.602	0.551	0.447	0.457	0.586
OPR	0.486	0.418	0.439	0.266	0.262	0.449	0.383	0.209	0.371	0.385
OV	0.736	0.534	0.690	0.567	0.671	0.689	0.573	0.629	0.420	0.652
BC	0.825	0.553	0.635	0.499	0.555	0.612	0.484	0.419	0.393	0.611
SOA	0.860	0.660	0.731	0.624	0.679	0.614	0.606	0.554	0.529	0.677
MB	0.835	0.632	0.703	0.617	0.590	0.637	0.502	0.492	0.332	0.689

Table 4. The success rate of 11 challenging attributes on the DTB70 dataset.

Attribute	SSACF	ECO_H	AutoTrack	BACF	ARCF_H	CSR-DCF	MCCT_H	SAMF_CA	Staple	STRCF
SV	0.510	0.429	0.493	0.392	0.406	0.476	0.439	0.336	0.349	0.417
ARV	0.448	0.373	0.405	0.273	0.314	0.396	0.334	0.299	0.314	0.347
OCC	0.520	0.432	0.415	0.348	0.354	0.407	0.377	0.341	0.349	0.400
DEF	0.478	0.389	0.452	0.302	0.308	0.396	0.354	0.279	0.283	0.390
FCM	0.546	0.464	0.496	0.429	0.444	0.455	0.410	0.347	0.331	0.467
IPR	0.489	0.401	0.454	0.365	0.383	0.414	0.376	0.310	0.318	0.393
OPR	0.387	0.311	0.343	0.203	0.228	0.339	0.243	0.157	0.283	0.257
OV	0.490	0.387	0.407	0.382	0.424	0.445	0.349	0.388	0.278	0.424
BC	0.490	0.332	0.394	0.316	0.354	0.376	0.296	0.264	0.231	0.369
SOA	0.532	0.446	0.473	0.411	0.434	0.394	0.399	0.348	0.346	0.447
MB	0.539	0.426	0.468	0.402	0.395	0.416	0.334	0.312	0.217	0.447

As shown in Table 3, the attribute accuracy on the DTB70 dataset indicates that SSACF performs significantly under FCM and OCC conditions, with accuracy scores of 0.851 and 0.795, respectively. This result demonstrates that SSACF can handle high-speed motion and dynamically changing backgrounds while maintaining high accuracy even in scenes with frequent occlusions. Table 4 presents the success rate performance under various challenging attributes. SSACF achieved success rates of 0.510 and 0.490 in the SV and BC environments, respectively. SSACF outperforms other algorithms in complex background conditions, highlighting its strong resistance to interference. SSACF shows an exceptional ability to adapt compared to other trackers, making it especially suitable for dynamic UAV environments. Other algorithms show considerable fluctuations in complex scenes, whereas SSACF maintains stable scores in various scenarios, highlighting its strong generalization ability. The rankings in Figure 4(c),(d) further illustrate SSACF's leading position in accuracy and success rate, achieving

0.801 and 0.525, respectively.

4.1.4. A quantitative analysis of the UAV20L dataset

UAV20L is a subset of UAV123, consisting of 20 long sequence videos designed to evaluate long-duration tracking tasks. Long-duration tracking tasks require the algorithm to deal with more frequent challenges, such as occlusion, background clutter, and object changes, especially in long-range tracking from a UAV perspective. The design of the UAV20L dataset is aimed at testing the stability and continuity of tracking algorithms in long-duration scenarios, assessing their robustness and processing capability for extended sequences. To provide a more detailed analysis of visual uncertainties, the UAV20L dataset also annotates sequences with 12 different attributes, with challenges similar to those in UAV123.

Table 5. The precision of 12 challenging attributes on the UAV20L dataset.

Attribute	SSACF	ARCF	AutoTrack	BACF	ARCF_H	CSR-DCF	MCCT_H	SAMF_CA	Staple	STRCF
SV	0.835	0.701	0.717	0.726	0.713	0.693	0.696	0.672	0.703	0.727
ARC	0.832	0.713	0.734	0.729	0.713	0.718	0.732	0.676	0.721	0.711
LR	0.866	0.671	0.706	0.710	0.702	0.668	0.681	0.673	0.681	0.705
FM	0.878	0.746	0.812	0.827	0.806	0.846	0.829	0.742	0.826	0.804
FO	0.886	0.807	0.795	0.810	0.798	0.762	0.786	0.681	0.773	0.774
PO	0.846	0.714	0.739	0.746	0.738	0.698	0.699	0.671	0.717	0.753
OV	0.833	0.675	0.726	0.730	0.728	0.698	0.686	0.649	0.712	0.769
BC	0.906	0.958	0.921	0.940	0.923	0.860	0.913	0.894	0.887	0.881
IV	0.746	0.692	0.649	0.636	0.605	0.626	0.666	0.644	0.628	0.602
VC	0.861	0.712	0.728	0.727	0.708	0.753	0.761	0.697	0.746	0.727
CM	0.838	0.707	0.723	0.732	0.719	0.698	0.699	0.678	0.709	0.732
SO	0.793	0.648	0.625	0.635	0.617	0.589	0.594	0.635	0.605	0.639

Table 6. The success rate of 12 challenging attributes on the UAV20L dataset.

Attribute	SSACF	ARCF	AutoTrack	BACF	ARCF_H	CSR-DCF	MCCT_H	SAMF_CA	Staple	STRCF
SV	0.655	0.622	0.597	0.626	0.599	0.571	0.580	0.558	0.587	0.587
ARC	0.695	0.672	0.643	0.674	0.649	0.622	0.632	0.599	0.639	0.606
LR	0.734	0.693	0.657	0.689	0.678	0.647	0.639	0.623	0.651	0.639
FM	0.760	0.748	0.744	0.772	0.733	0.710	0.742	0.684	0.738	0.700
FO	0.757	0.723	0.696	0.749	0.730	0.691	0.683	0.625	0.696	0.686
PO	0.679	0.654	0.633	0.665	0.647	0.578	0.620	0.549	0.585	0.635
OV	0.622	0.581	0.581	0.602	0.588	0.513	0.577	0.488	0.520	0.609
BC	0.902	0.928	0.872	0.943	0.933	0.879	0.835	0.859	0.881	0.847
IV	0.603	0.611	0.554	0.595	0.559	0.520	0.518	0.531	0.546	0.472
VC	0.670	0.636	0.593	0.613	0.572	0.584	0.617	0.532	0.607	0.578
CM	0.676	0.650	0.626	0.661	0.636	0.566	0.609	0.554	0.582	0.610
SO	0.608	0.557	0.521	0.566	0.542	0.485	0.487	0.526	0.507	0.489

As shown in Table 5, SSACF demonstrated overall high accuracy, excelling in most attributes. For instance, under BC conditions, SSACF achieved an accuracy of 0.906, markedly outperforming other algorithms and showcasing its exceptional ability to manage complex background scenarios. In addition, SSACF also performed well under FO and LR conditions, achieving accuracy scores of 0.886 and 0.866, respectively. As shown in Table 6, SSACF's success rate under BC conditions reached 0.902, continuing to demonstrate its strong tracking ability in complex backgrounds. SSACF showed outstanding success rates across various challenging attributes, highlighting its high reliability

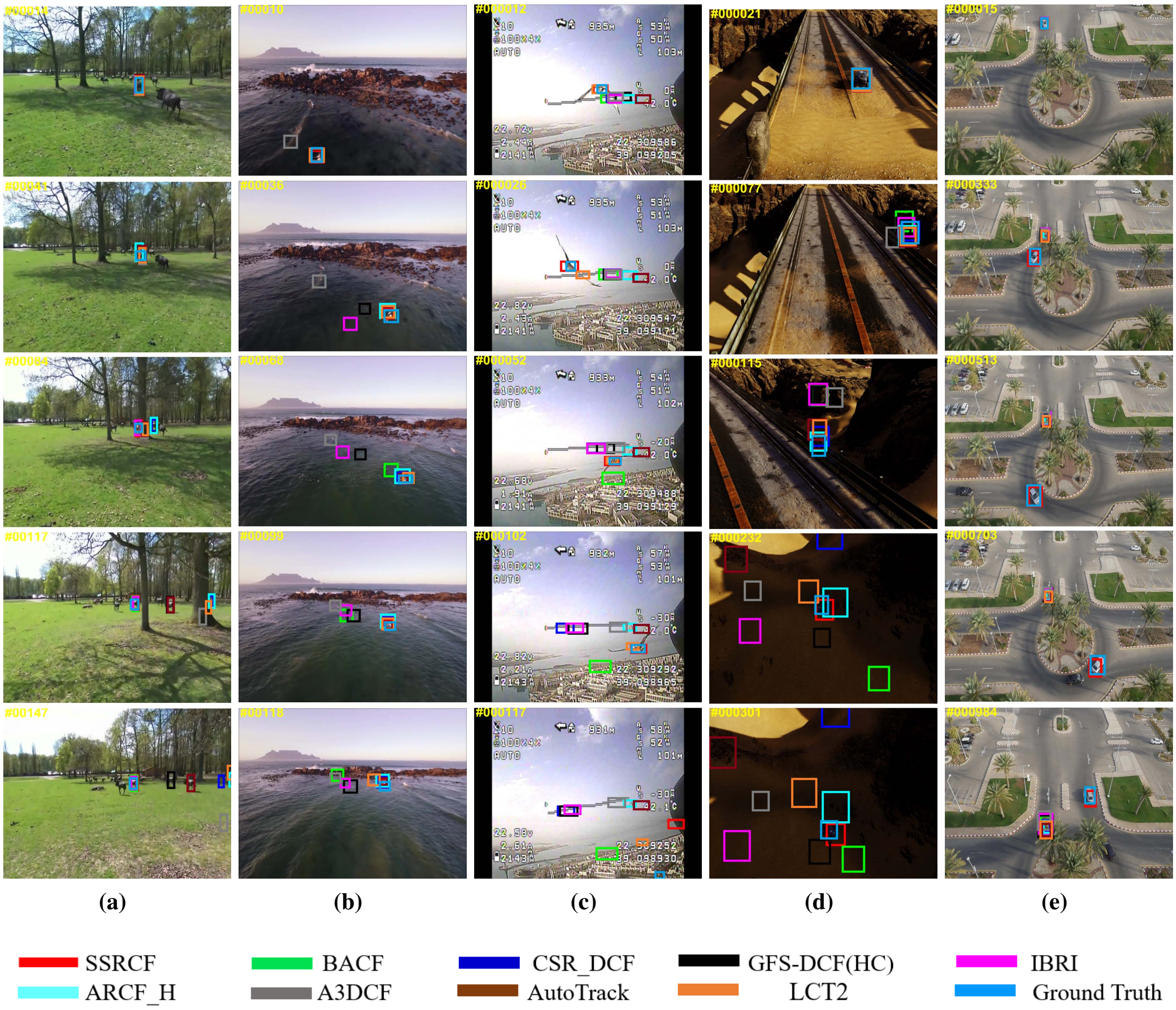


Figure 5. Visualization of tracking performance on different video sequences. (a) Horse1. (b) Gull1. (c) bird1_1. (d) car2_s. (e) car7.

in different practical application scenarios. Figure 4(e),(f) presents the ranking of SSACF against other SOTA algorithms on the UAV20L dataset, showcasing SSACF's leading position in complex dynamic environments. Its accuracy and success rate reached 0.842 and 0.664, confirming its stability and adaptability in long-duration tasks.

4.2. *Qualitative analysis*

To more intuitively evaluate the tracking performance, typical video sequences were selected from the DTB70 and UAV123 datasets, including the "Horse1" and "Gull1" sequences from DTB70, and the "bird1_1", "car2_s", and "car7" sequences from UAV123. Frame-by-frame comparisons were made between SSACF and nine SOTA trackers. The lighter blue box indicates the ground truth. Figure 5 illustrates the performance of different algorithms across several typical tracking challenges. The following is a detailed analysis of the comparison results for these challenge attributes:

(1) **Similar targets**. In the "Horsel" video sequence shown in Figure 5(a), the object is a group of

horses moving on the grass with similar colors and shapes, causing some algorithms to misidentify other objects. SSACF, BACF, and IBRI algorithms maintain accurate object tracking throughout the sequence, while the other algorithms suffer from varying degrees of object loss and mistracking of interfering objects. Likewise, in the "bird1_1" video sequence shown in Figure 5(c), the color of the object bird is similar to the numbers on the UAV interface, leading most algorithms to misidentify the interface numbers as the object. Only SSACF succeeds in maintaining accurate tracking of the object.

(2) **Background clutter**. In the "car2_s" video sequence shown in Figure 5(d), the car gradually moves into a shadowed area, increasing background complexity. This background interference causes most algorithms to incorrectly identify the shadow as the object, losing track of the original object. SSACF effectively filters out background distractions by capturing the object's shape features, ensuring stable object tracking.

(3) **Occlusion**. In the "car7" video sequence shown in Figure 5(e), the car is occluded by tree branches, and all other algorithms lose the object. However, SSACF, thanks to its robust handling of occlusion features, can continue tracking the occluded object, showcasing the algorithm's strength in dealing with occlusions.

(4) **Fast motion**. In the "Gull1" video sequence shown in Figure 5(b), the rapid movement of the seagull results in motion blur and drastic changes in position, which presents a considerable challenge to tracking algorithms. The BACF algorithm completely loses the object. In contrast, SSACF remains stable in tracking the object despite motion blur and positional changes, exhibiting strong adaptability to fast motion.

The SSACF algorithm shows remarkable robustness and stability when confronting typical tracking challenges such as similar objects, background clutter, occlusion, and fast motion, further affirming its reliability for tracking in complex settings.

4.3. Ablation study

4.3.1. Feature reduction based on spatial saliency

To validate the impact of spatial saliency-based feature reduction on tracking results, this study experiments with SSACF algorithms with and without feature reduction and explains the tracking outcomes. The blue box indicates the ground truth.

Table 7. Comparison of performance metrics between models with and without feature reduction.

Performance metric	With feature reduction	Without feature reduction
Average center point error	7.81	17.23
Average tracking overlap	0.76	0.61

As shown in Figure 6, the red dashed box represents the model with feature reduction, and the green solid box represents the model without feature reduction. The experiment demonstrates that at frame 16, when the difference between the object and background is evident, both models can track the object effectively. In frame 49, the green solid box experiences slight drift when the background changes, while the red dashed box continues to track accurately. By frame 82, the green solid box is misled by nearby interference and drifts, while the red box tracks the object accurately. Table 7 shows that the model with feature reduction outperforms the model without feature reduction, with an average center point error of 7.81 and an average tracking overlap of 0.76, compared to 17.23 and 0.61, respectively.

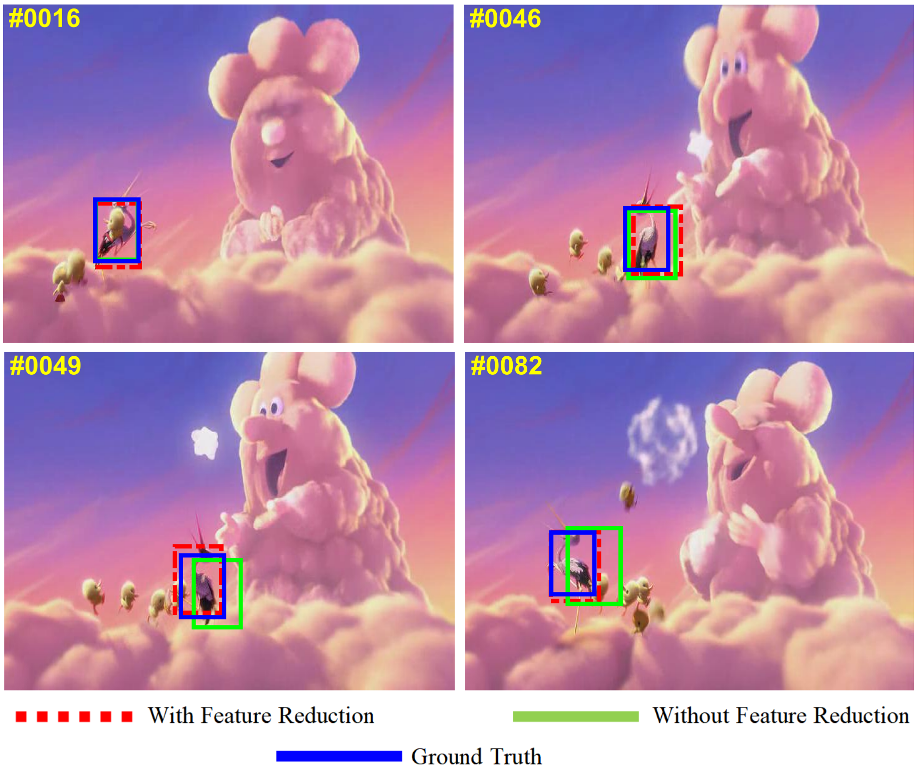


Figure 6. Comparison of models with and without feature reduction.

4.3.2. Regularization factors under temporal-spatial joint constraints

This study examines whether introducing three regularization factors (boundary suppression factor, spatial interference suppression factor, and temporal-spatial anomaly suppression factor- affects) the tracking results and compares the outcomes. The blue box indicates the ground truth.

As shown in Figure 7, the comparison experiment shows the results of models with and without regularization factors. The red dashed box represents the model with regularization factors, while the green solid box represents the model without regularization factors. The experiment shows that from frames 81 to 190, both models can track the object accurately. However, at frame 190, an intra-class interference occurs on the left side, leading to significant displacement of the solid box at frame 201, causing the object to be inaccurately tracked. Similar results are observed in frames 201 to 394. However, the model with regularization factors (indicated by the dashed box) is better at maintaining the accurate tracking of the object. As seen in Table 8, the model with regularization factors has an average center point error of 4.55, significantly lower than the 40.19 error for the model without regularization factors. The average tracking overlap for the model with regularization factors is 0.74, while the model without regularization factors only achieves 0.35. This shows that incorporating regularization factors significantly enhances both tracking accuracy and stability.

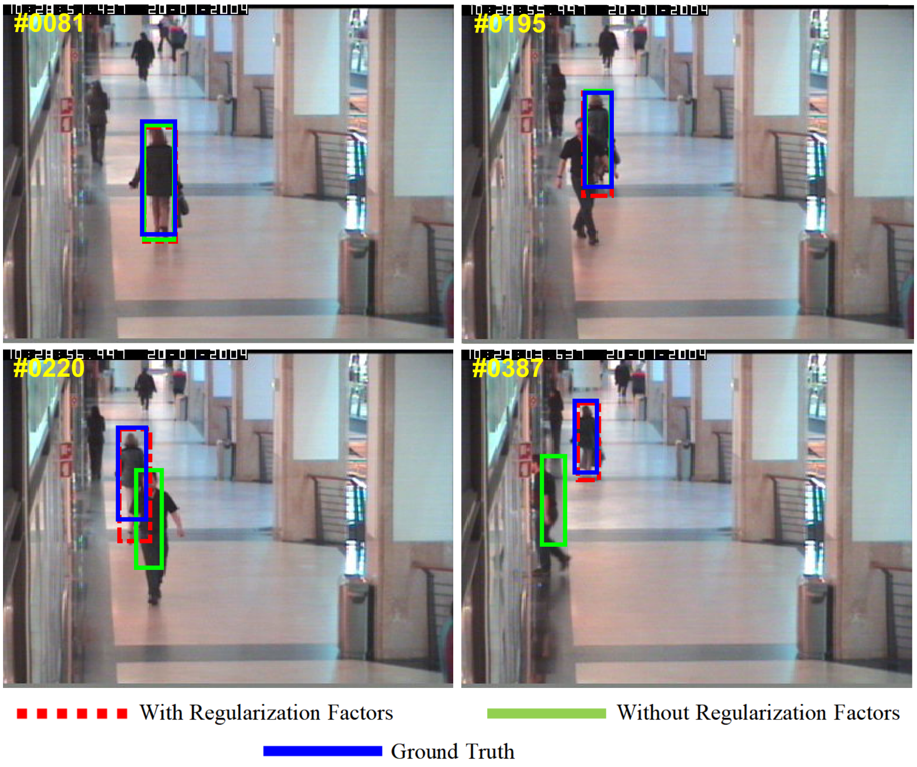


Figure 7. Comparison of models with and without feature reduction.

Table 8. Comparison of performance metrics between models with and without regularization factors.

Performance metric	With regularization factors	Without regularization factors
Average center point error	4.55	40.19
Average tracking overlap	0.74	0.35

4.3.3. Optimal candidate objects based on positive expert group

This study conducts a comprehensive experimental evaluation of the SSACF algorithm with and without the positive expert group to verify whether the optimal candidate object based on the positive expert group influences the tracking results. The blue box indicates the ground truth.

Figure 8 shows an ablation experiment on the optimal candidate object using the positive expert group. In frame 38, both models (with and without the expert group) initially track the object accurately. However, in frame 323, occlusion occurs. In subsequent frames, the model without the expert group fails due to prior learning of the occluder's features, causing a large displacement. In contrast, the model with the expert group retains the object's features and continues tracking accurately. As shown in Table 9, the model with the expert group has an average center point error of 8.57, significantly lower than the 90.38 error without it. Additionally, the tracking overlap for the model with the expert group is 0.75, compared to 0.28 for the model without. This highlights that incorporating the optimal candidate object from the expert group improves tracking accuracy and stability.

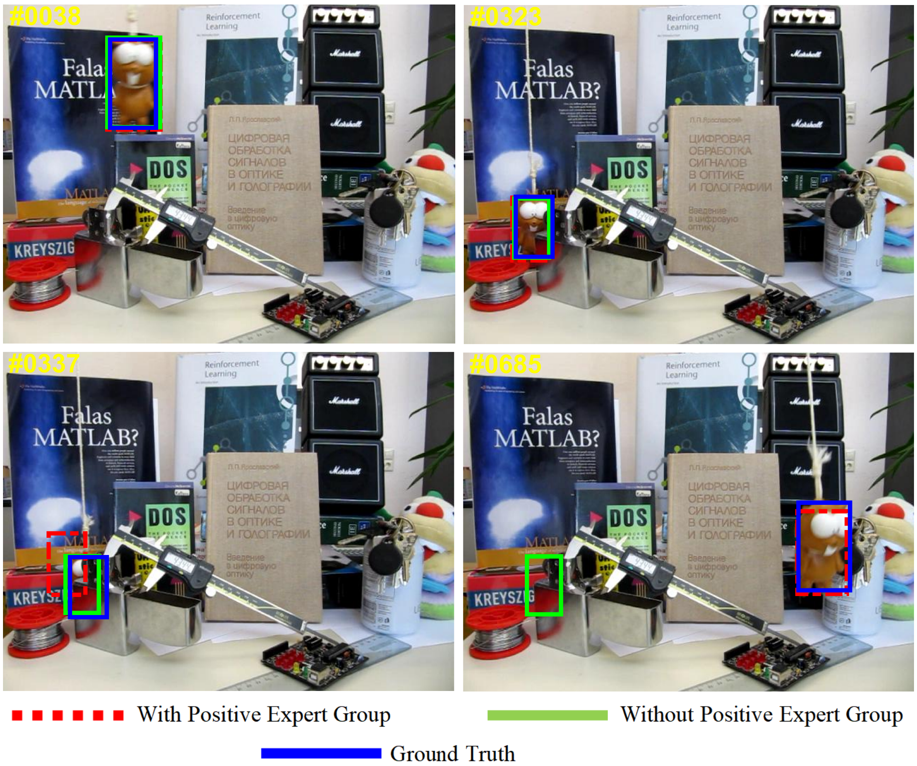


Figure 8. Comparison of models with and without feature reduction.

Table 9. Comparison of performance metrics between models with and without positive expert group.

Performance metric	With positive expert group	Without positive expert group
Average center point error	8.57	90.38
Average tracking overlap	0.75	0.28

4.4. Comparing deep-learning-based trackers

In this subsection, we present a detailed tracking performance comparison experiment between the SSACF algorithm and other deep- learning-based trackers (including SiamFC [11], ATOM [56], CSWinTT [39], TransT [13], and DiMP [57]) on the UAV123 dataset. The quantitative comparative analysis in Table 10 reveals that most deep-learning-based trackers outperform the proposed method. However, their high computational complexity limits their potential deployment on edge devices. The core module of the SSACF algorithm, with its innovative lightweight design, significantly reduces algorithm complexity while maintaining target recognition accuracy. Its modular architecture and hardware adaptation optimization strategies make it more suitable for deployment on UAV platforms.

Table 10. Comparison with algorithms based on deep learning tracker.

Metohd	SiamFC	CSWinTT	TransT	DiMP	ATOM	Ours
Success rate (%)	49.2	68.2	66	64.2	61.7	60.3
Precision (%)	72.7	87.5	85.2	84.9	82.7	80.1

4.5. Real-time tracking comparison

In this subsection, we comprehensively explore the frames per second (fps) and limitations of the SSACF algorithm using the OTB100 dataset. Table 11 compares the real-time performance (measured in FPS) of the SSACF algorithm with traditional handcrafted feature trackers (such as ECO_H, AutoTrack and BACF) across multiple video sequences. The experimental results show that the proposed method is capable of achieving a good balance between speed and accuracy.

Table 11. Frames per second (fps) of each tracking algorithm in some videos come from OTB100.

Video	Ours	CSR-DCF	BACF	GFS-DCF(HC)	IBRI	ARCF H	A3DCF	AutoTrack	LCT2
Girl	23.62	20.77	23.72	23.81	11.45	32.01	20.25	7.40	12.71
Doll	15.33	11.27	16.04	15.40	13.33	28.75	14.25	9.56	28.82
Football1	19.76	18.04	33.11	26.46	17.10	40.80	22.24	8.01	14.60
Boy	22.49	21.75	27.12	27.48	4.68	9.26	24.38	11.40	17.64
Subway	20.53	20.20	38.02	27.10	21.54	44.89	24.50	14.26	18.35

5. Conclusions

In this paper, we proposed the SSACF tracker, which effectively tackles common problems in UAV object tracking, such as visual feature redundancy, limited discriminative power, insufficient exploitation of spatiotemporal information, and filter degradation. This paper refines feature selection on both spatial and channel dimensions by implementing a spatial saliency-aware strategy, substantially improving the discriminative capability between the object and the background. Furthermore, the spatiotemporal joint constraint location estimation mechanism introduced in this paper fully leverages spatiotemporal information, considerably enhancing the model's tracking robustness in complex environments. Additionally, to address filter degradation, this paper successfully mitigates decreases in tracking accuracy during occlusions by employing a reliable expert group evaluation method. The experimental outcomes indicate that the SSACF algorithm performs exceptionally well across various challenging public datasets, confirming its considerable potential for UAV visual object-tracking applications. Future research will concentrate on improving the real-time performance and robustness of the algorithm to accommodate the increasing needs of various UAV applications.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work is supported by the Natural Science Foundation of Fujian Province (2024J01820, 2024J01821, 2024J01822), Natural Science Foundation Project of Zhangzhou City (ZZ2023J37), the Principal Foundation of Minnan Normal University (KJ19019), the High-level Science Research Project of Minnan Normal University (GJ19019), Research Project on Education and Teaching of Undergraduate Colleges and Universities in Fujian Province (FBJY20230083), Guangdong Province Natural Science Foundation (2024A1515011766).

Conflict of interest

The authors declare there is no conflicts of interest.

References

1. H. Zhang, P. He, M. Zhang, C. Daqing, E. Neretin, B. Li, UAV target tracking method based on deep reinforcement learning, in *International Conference on Cyber-Physical Social Intelligence (ICCSI)*, IEEE, (2022), 274–277. https://doi.org/10.1109/ICCSI55536.2022.9970588

2. R. Wu, Y. Liu, X. Wang, P. Yang, Visual tracking based on spatiotemporal transformer and fusion sequences, *Image Vision Comput.*, **148** (2024), 105107. https://doi.org/10.1016/j.imavis.2024.105107

3. J. McGee, S. J. Mathew, F. Gonzalez, Unmanned aerial vehicle and artificial intelligence for thermal target detection in search and rescue applications, in *International Conference on Unmanned Aircraft Systems*, IEEE, (2020), 883–891. https://doi.org/10.1109/ICUAS48674.2020.9213849

4. P. Byukusenge, Y. Zhang, Life detection based on uavs - thermal images in search and rescue operation, in *IEEE 22nd International Conference on Communication Technology*, IEEE, (2022), 1728–1731. https://doi.org/10.1109/ICCT56141.2022.10073136

5. K. Chen, L. Wang, H. Wu, C. Wu, Y. Liao, Y. Chen, et al., Background-aware correlation filter for object tracking with deep cnn features, *Eng. Lett.*, **32** (2024), 1351–1363.

6. J. Wen, H. Chu, Z. Lai, T. Xu, L. Shen, Enhanced robust spatial feature selection and correlation filter learning for uav tracking, *Neural Networks*, **161** (2023), 39–54. https://doi.org/10.1016/j.neunet.2023.01.003

7. J. Lin, J. Peng, J. Chai, Real-time UAV correlation filter based on response-weighted background residual and spatio-temporal regularization, *IEEE Geosci. Remote Sens. Lett.*, **20** (2023), 1–5. https://doi.org/10.1109/LGRS.2023.3272522

8. T. Xu, Z. H. Feng, X. J. Wu, J. Kittler, Joint group feature selection and discriminative filter learning for robust visual object tracking, in *IEEE/CVF International Conference on Computer Vision*, IEEE, (2019), 7949–7959. https://doi.org/10.1109/ICCV.2019.00804

9. A. Lukezic, T. Vojir, L. C. Zajc, J. Matas, M. Kristan, Discriminative correlation filter with channel and spatial reliability, in *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, (2017), 6309–6318. https://doi.org/10.1109/CVPR.2017.515

10. Y. Chen, K. Chen, Four mathematical modeling forms for a correlation filter object tracking algorithm and the fast calculation for the filter, *Electron. Res. Arch.*, **32** (2024), 4684–4714. https://doi.org/10.3934/era.2024213

11. L. Bertinetto, J. Valmadre, J. Henriques, A. Vedaldi, P. Torr, Fully-convolutional siamese networks for object tracking, in *Computer Vision-ECCV 2016 Workshops*, Springer, **9914** (2016), 850–865. https://doi.org/10.1007/978-3-319-48881-3_56

12. P. Voigtlaender, J. Luiten, P. H. S. Torr, B. Leibe, Siam r-cnn: Visual tracking by re-detection, in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, (2020), 6578–6588. https://doi.org/10.1109/CVPR42600.2020.00661

13. X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, H. Lu, Transformer tracking, in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, (2021), 8126–8135. https://doi.org/10.1109/CVPR46437.2021.00803

14. B. Yan, H. Peng, J. Fu, D. Wang, H. Lu, Learning spatio-temporal transformer for visual tracking, in *IEEE/CVF International Conference on Computer Vision*, IEEE, (2021), 10448–10457. https://doi.org/10.1109/ICCV48922.2021.01028

15. Y. Cui, C. Jiang, L. Wang, G. Wu, Mixformer: End-to-end tracking with iterative mixed attention, in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, (2022), 13608–13618. https://doi.org/10.1109/CVPR52688.2022.01324

16. S. Xuan, S. Li, M. Han, X. Wan, G. Xia, Object tracking in satellite videos by improved correlation filters with motion estimations, *IEEE Trans. Geosci. Remote Sens.*, **58** (2020), 1074–1086. https://doi.org/10.1109/TGRS.2019.2943366

17. S. Ma, B. Zhao, Z. Hou, W. Yu, L. Pu, X. Yang, Socf: A correlation filter for real-time uav tracking based on spatial disturbance suppression and object saliency-aware, *Expert Syst. Appl.*, **238** (2024), 122131. https://doi.org/10.1016/j.eswa.2023.122131

18. J. van de Weijer, C. Schmid, J. Verbeek, D. Larlus, Learning color names for real-world applications, *IEEE Trans. Image Process.*, **18** (2009), 1512–1523. https://doi.org/10.1109/TIP.2009.2019809

19. L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, P. Torr, Staple: Complementary learners for real-time tracking, in *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, (2016), 1401–1409. https://doi.org/10.1109/CVPR.2016.156

20. H. K. Galoogahi, A. Fagg, S. Lucey, Learning background-aware correlation filters for visual tracking, in *IEEE International Conference on Computer Vision*, IEEE, (2017), 1135–1143. https://doi.org/10.1109/ICCV.2017.129

21. M. Mueller, N. Smith, B. Ghanem, Context-aware correlation filter tracking, in *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, (2017), 1396–1404. https://doi.org/10.1109/CVPR.2017.152

22. M. Danelljan, G. Bhat, F. S. Khan, M. Felsberg, Eco: Efficient convolution operators for tracking, in *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, (2017), 6638–6646. https://doi.org/10.1109/CVPR.2017.733

23. C. Ma, J. B. Huang, X. Yang, M. H. Yang, Hierarchical convolutional features for visual tracking, in *IEEE International Conference on Computer Vision*, IEEE, (2015), 3074–3082. https://doi.org/10.1109/ICCV.2015.352

24. M. Danelljan, G. Hager, F. S. Khan, M. Felsberg, Convolutional features for correlation filter based visual tracking, in *IEEE International Conference on Computer Vision Workshops*, IEEE, (2015), 58–66. https://doi.org/10.1109/ICCVW.2015.84

25. K. Dai, D. Wang, H. Lu, C. Sun, J. Li, Visual tracking via adaptive spatially-regularized correlation filters, in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, (2019), 4670–4679. https://doi.org/10.1109/CVPR.2019.00480

26. J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, (2018), 7132–7141. https://doi.org/10.1109/CVPR.2018.00745

27. F. Du, P. Liu, W. Zhao, X. Tang, Joint channel reliability and correlation filters learning for visual tracking, *IEEE Trans. Circuits Syst. Video Technol.*, **30** (2019), 1625–1638. https://doi.org/10.1109/TCSVT.2019.2909654

28. T. Xu, Z. H. Feng, X. J. Wu, J. Kittler, Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual object tracking, *IEEE Trans. Image Process.*, **28** (2019), 5596–5609. https://doi.org/10.1109/TIP.2019.2919201

29. W. Feng, R. Han, Q. Guo, J. Zhu, S. Wang, Dynamic saliency-aware regularization for correlation filter-based object tracking, *IEEE Trans. Image Process.*, **38** (2019), 3232–3245. https://doi.org/10.1109/TIP.2019.2895411

30. D. Zhao, L. Xiao, H. Fu, T. Wu, X. Xu, B. Dai, Augmenting cascaded correlation filters with spatial-temporal saliency for visual tracking, *Inf. Sci.*, **470** (2019), 78–93. https://doi.org/10.1016/j.ins.2018.08.053

31. P. Yang, Q. Wang, J. Dou, L. Dou, Sdcs-cf: Saliency-driven localization and cascade scale estimation for visual tracking, *J. Visual Commun. Image Represent.*, **98** (2024), 104040. https://doi.org/10.1016/j.jvcir.2023.104040

32. C. Fu, J. Xu, F. Lin, F. Guo, T. Liu, Z. Zhang, Object saliency-aware dual regularized correlation filter for real-time aerial tracking, *IEEE Trans. Geosci. Remote Sens.*, **58** (2020), 8940–8951. https://doi.org/10.1109/TGRS.2020.2992301

33. X. Yang, S. Li, J. Ma, J. Yang, J. Yan, Co-saliency-regularized correlation filter for object tracking, *Signal Process. Image Commun.*, **103** (2022), 116655. https://doi.org/10.1016/j.image.2022.116655

34. P. Zhang, W. Liu, D. Wang, Y. Lei, H. Wang, H. Lu, Non-rigid object tracking via deep multi-scale spatial-temporal discriminative saliency maps, *Pattern Recognit.*, **100** (2020), 107130. https://doi.org/10.1016/j.patcog.2019.107130

35. L. Gao, B. Liu, P. Fu, M. Xu, J. Li, Visual tracking via dynamic saliency discriminative correlation filter, *Appl. Intell.*, **52** (2022), 5897–5911. https://doi.org/10.1007/s10489-021-02260-2

36. F. Li, C. Tian, W. Zuo, L. Zhang, M. H. Yang, Learning spatial-temporal regularized correlation filters for visual tracking, in *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, (2018), 4904–4913. https://doi.org/10.1109/CVPR.2018.00515

37. Y. Li, C. Fu, F. Ding, Z. Huang, G. Lu, Autotrack: Towards high-performance visual tracking for uav with automatic spatio-temporal regularization, in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, (2020), 11920–11929. https://doi.org/10.1109/CVPR42600.2020.01194

38. Y. Chen, H. Wu, Z. Deng, J. Zhang, H. Wang, L. Wang, et al., Deep-feature-based asymmetrical background-aware correlation filter for object tracking, *Digital Signal Process.*, **148** (2024), 104446. https://doi.org/10.1016/j.dsp.2024.104446

39. Z. Song, J. Yu, Y. P. P. Chen, W. Yang, Transformer tracking with cyclic shifting window attention, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2022), 8791–8800. https://doi.org/10.1109/CVPR52688.2022.00859

40. M. Danelljan, G. Hager, F. S. Khan, M. Felsberg, Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking, in *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, (2016), 1430–1438. https://doi.org/10.1109/CVPR.2016.159

41. M. Danelljan, A. Robinson, F. S. Khan, M. Felsberg, Beyond correlation filters: Learning continuous convolution operators for visual tracking, in *European Conference on Computer Vision*, Springer, (2016), 472–488. https://doi.org/10.1007/978-3-319-46454-1_29

42. Q. Hu, H. Wu, J. Wu, J. Shen, H. Hu, Y. Chen, et al., Spatio-temporal self-learning object tracking model based on anti-occlusion mechanism, *Eng. Lett.*, **31** (2023), 1141–1150.

43. Y. Huang, Y. Chen, C. Lin, Q. Hu, J. Song, Visual attention learning and antiocclusion-based correlation filter for visual object tracking, *J. Electron. Imaging*, **32** (2023), 013023. https://doi.org/10.1117/1.JEI.32.1.013023

44. Y. Wu, J. Lim, M. H. Yang, Online object tracking: A benchmark, in *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, (2013), 2411–2418. https://doi.org/10.1109/CVPR.2013.312

45. S. Li, D. Y. Yeung, Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models, in *AAAI Conference on Artificial Intelligence*, AAAI Press, (2017), 4140–4146.

46. M. Mueller, N. G. Smith, B. Ghanem, A benchmark and simulator for uav tracking, in *European Conference on Computer Vision*, Springer, (2016), 445–461. https://doi.org/10.1007/978-3-319-46448-0_27

47. S. Ma, Z. Zhao, L. Pu, Z. Hou, L. Zhang, X. Zhao, Learning discriminative correlation filters via saliency-aware channel selection for robust visual object tracking, *J. Real-Time Image Process.*, **20** (2023), 51. https://doi.org/10.1007/s11554-023-01306-7

48. M. Danelljan, G. Hager, F. S. Khan, M. Felsberg, Learning spatially regularized correlation filters for visual tracking, in *IEEE International Conference on Computer Vision*, IEEE, (2015), 4310–4318. https://doi.org/10.1109/ICCV.2015.490

49. M. Danelljan, G. Hager, F. S. Khan, M. Felsberg, Discriminative scale space tracking, *IEEE Trans. Pattern Anal. Mach. Intell.*, **39** (2016), 1561–1575. https://doi.org/10.1109/TPAMI.2016.2609928

50. N. Wang, W. Zhou, Q. Tian, R. Hong, M. Wang, H. Li, Multi-cue correlation filters for robust visual tracking, in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, (2018), 4844–4853. https://doi.org/10.1109/CVPR.2018.00509

51. Z. Huang, C. Fu, Y. Li, F. Lin, P. Lu, Learning aberrance repressed correlation filters for real-time UAV tracking, in *IEEE/CVF International Conference on Computer Vision*, IEEE, (2019), 2891–2900. https://doi.org/10.1109/ICCV.2019.00298

52. C. Fu, J. Ye, J. Xu, Y. He, J. Xu, Y. He, Disruptor-aware interval-based response inconsistency for correlation filters in real-time aerial tracking, *IEEE Trans. Geosci. Remote Sens.*, **59** (2021), 6301–6313. https://doi.org/10.1109/TGRS.2020.3030265

53. X. F. Zhu, X. J. Wu, T. Xu, Z. H. Feng, J. Kittler, Robust visual object tracking via adaptive attribute-aware discriminative correlation filters, *IEEE Trans. Multimedia*, **24** (2022), 301–312. https://doi.org/10.1109/TMM.2021.3050073

54. C. Ma, X. Yang, C. Zhang, M. H. Yang, Long-term correlation tracking, in *IEEE Conference on Computer Vision and Pattern Recognition*, (2015), 5388–5396. https://doi.org/10.1109/CVPR.2015.7299177

55. Y. Li, J. Zhu, A scale adaptive kernel correlation filter tracker with feature integration, in *Computer Vision-ECCV 2014 Workshops*, Springer, **8926** (2015), 254–265. https://doi.org/10.1007/978-3-319-16181-5_18

56. M. Danelljan, G. Bhat, F. S. Khan, M. Felsberg, Atom: Accurate tracking by overlap maximization, in *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, (2019), 4660–4669. https://doi.org/10.1109/CVPR.2019.00479

57. G. Bhat, M. Danelljan, L. V. Gool, R. Timofte, Learning discriminative model prediction for tracking, in *IEEE/CVF International Conference on Computer Vision*, IEEE, (2019), 6182–6191. https://doi.org/10.1109/ICCV.2019.00628