Background-Aware Correlation Filter for Object Tracking with Deep CNN Features

Kaiwei Chen, Lingzhi Wang, Huangyu Wu, Changhui Wu, Yuan Liao, Yingpin Chen*, Hui Wang, Jingwen Yan, Jialing Lin, Jiale He

Abstract—**Correlation filter tracking algorithms have garnered significant attention due to their efficiency and outstanding tracking performance. However, these methods face several limitations. Firstly, they rely on periodic boundary conditions, leading to boundary effects. Secondly, most traditional methods only extract hand-crafted features from the image. Nevertheless, these features are insufficient to discriminate in complex scenes. Thirdly, they assume that the maximum position of the correlation response represents the object without further evaluating its reliability. These limitations make it very easy to lose the object in occluded scenes. A background-aware correlation filter algorithm based on an anti-occlusion mechanism and deep features is proposed to solve the above limitations. Primarily, the video frames to be processed are cyclically shifted to crop the cyclically shifted samples in a small window. This operation significantly reduces boundary effects while obtaining background samples from the real world. Then, deep features are extracted through deep neural networks. Subsequently, we propose a background perception tracking framework that synchronously estimates position and scale based on these features. This framework aims to determine the optimal candidate sample position and scale. Finally, an anti-occlusion mechanism is constructed to evaluate the optimal candidate samples obtained in each frame further. This mechanism fully exploits the diversity of objects and effectively solves the tracking drift and failure issues caused by occlusion, fast motion, and so on. Extensive experiments are conducted on the object tracking benchmark (OTB) dataset and compared with industry-leading tracking algorithms to validate the effectiveness of the proposed method. The results show that the method has robust tracking performance.**

Index Terms—**object tracking, correlation filter, background-aware, anti-occlusion mechanism, deep feature**

I. Introduction

OBJECT tracking [1] refers to accurately locating and tracking objects from consecutive frames by modeling their shape, appearance, and other features in a video sequence. As a fundamental issue in the field of computer vision, object tracking is widely used in fields such as unmanned aerial vehicle (UAV) tracking [2], facial recognition systems, and intelligent traffic [3]. During object tracking, the tracker's performance may be easily affected by factors such as background clutter, occlusion, and scale changes.

The method based on discriminative correlation filter [4-6] (DCF) transforms correlation operation in the spatial domain into entry-wise multiplication operation in the frequency domain. This transformation enhances the efficiency of DCF. Thus, DCF has become a mainstream method in visual object tracking. A famous example is the minimum output sum of squared error (MOSSE) [7] tracker, whose efficient tracking benefits from the fast Fourier transform (FFT) and relatively simple object features. Henriques *et al.* proposed the circulant structure of tracking-by-detection with kernels [8] (CSK) tracker. This tracker generalizes the linear regression model to a nonlinear one based on kernel tricks. It trains the classifier by dense sampling to improve the discriminative ability of the tracker effectively. Henriques *et al.* promoted the single-channel gray features to the multi-channel histogram of oriented gradient (HOG) features based on CSK. They proposed the high-speed tracking with kernelized correlation filter [9] (KCF) method to further improve the tracking performance. Although the above work achieves commendable tracking results, the cyclic shift operation constructs virtual training samples at the extended boundary. This can lead to discontinuous image edge splicing, i.e., boundary effects.

To address this issue, Danelljan *et al*. proposed a spatial regularized correlation filter [10] (SRDCF). They punished the background based on spatial location, making the filter likely to focus on information near the center. To fully explore background information [11, 12], Galoogahi *et al*. proposed the background-aware correlation filter [13] (BACF) tracking algorithm. BACF cleverly avoids boundary effects via a small cropping window in a large search area. In this way, BACF expands the search area without high

Manuscript received November 8, 2023; revised July 15, 2024.

This work is supported by national natural science foundation of China (62001199), the natural science foundation project of Fujian Province (2023J01155), the 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), the education research program of Minnan Normal University (202211), college student innovation and entrepreneurship training program project (202310402010, S202310402014) and the Guangdong Provincial university innovation team project (2020KCXTD012).

Kaiwei Chen is a postgraduate student of Minnan Normal University, Zhangzhou, 363000 China (e-mail: 1914931478@qq.com).

Lingzhi Wang is an associate professor of Xiamen City University, Xiamen, 361000 China (*co-first author, e-mail: 64564254@qq.com).

Huanyu Wu is a postgraduate student of Minnan Normal University, Zhangzhou, 363000 China (e-mail: 598851621@qq.com).

Changhui Wu is a postgraduate student of Minnan Normal University, Zhangzhou, 363000 China (e-mail: 1213670742@qq.com).

Yuan Liao is a postgraduate student of Minnan Normal University, Zhangzhou, 363000 China. (e-mail: 782327076@qq.com).

Yingpin Chen is an associate professor of Key Laboratory of Light Field Manipulation and System Integration Applications in Fujian Province, School of Physics and Information Engineering, Minnan Normal University, Zhangzhou, 363000 China (*correspondence author, e-mail: cyp1707@mnnu.edu.cn).

Hui Wang is an associate professor of Minnan Normal University, Zhangzhou, 363000 China (e-mail: wh1953@mnnu.edu.cn).

Jingwen Yan is a professor of Shantou University, Shantou, 515000 China (e-mail: Jwyan@stu.edu.cn).

Jialing Lin is an undergraduate student of Minnan Normal University, Zhangzhou, 363000 China (e-mail: 1400997236@qq.com).

Jiale He is an undergraduate student of Minnan Normal University, Zhangzhou, 363000 China (e-mail: 279995349@qq.com).

computational costs. To speed up the training steps, the ECO [14] tracker uses Gaussian mixture models to simplify the training set. The strategy effectively balances speed and performance. The group feature selection and discriminative filter [15] (GFS-DCF) method selects the extracted features at the spatial and channel levels. This reduces the information redundancy and irrelevance of high-dimensional multi-channel features. Additionally, GFS-DCF employs an effective low-rank approximation method to fuse the historical information of video frames adaptively. Thus, the filter is smoothed across the time dimension (frames).

Scale estimation is an important part of the object-tracking framework [16-19]. Li *et al.* proposed a scale adaptive kernel correlation filter tracker [20] (SAMF), which utilizes a scale pool to determine the object's scale. Similarly, a discriminative scale space tracker [21] (DSST) estimates the object's scale by a feature pyramid with 33 scales and a one-dimensional scale filter. Although the aforementioned methods have made significant progress, they have limited capability in representing the object's scale due to insufficient discriminative information from hand-crafted features.

Deep learning techniques have greatly advanced visual object tracking by providing powerful deep feature learning capabilities. Existing deep learning-based trackers can be categorized into two main streams. One focuses on improving the feature representation in the DCF framework by deep neural networks, such as MDNet [22] and C-COT [23]. Compared with traditional hand-crafted features such as the HOG features and color names [24] (CN), the deep features extracted from the convolutional neural networks (CNN) can be used in various visual applications. Another branch utilizes neural networks for end-to-end learning of object classification and regression tasks. For example, Bertinetto *et al.* introduced the Siamese network for visual tracking, showing the great potential of Siamese neural networks [25-29] trained offline. However, CNN always relies on large-scale image datasets to obtain superior accuracy with a high computational cost.

The traditional DCF adopts hand-crafted features and updates the filter every frame. However, it is easy to generate tracking drift when encountering strong occlusion [30]. To this end, this paper constructs a multimodal template pool utilizing historical positive samples. It aims to capture the object's variability and assess the confidence of the optimal candidate samples, thereby enhancing the algorithm's resistance to occlusions. In summary, this paper proposes a background-aware correlation filter for object tracking with deep features (DeepBACF). The main contributions of the proposed DeepBACF are as follows:

(1) The deep features extracted by the CNN contain high-level object semantic information and exhibit good invariance to changes in object appearance, improving the overall performance of the tracker.

(2) The historical multimodal template pool is proposed to evaluate the optimal candidate samples for each frame. It optimizes the algorithm's update mechanism to enhance the filter's robustness in complex occlusion scenarios.

(3) The proposed filter learning optimization problem can be decomposed into several subproblems and solved efficiently by the alternating direction method of multipliers (ADMM).

(4) Comprehensive experiments on the OTB dataset show that the performance of DeepBACF is superior to other state-of-the-art trackers.

II. PRELIMINARY KNOWLEDGE

A. Correlation Filter Tracking Algorithm

The DCF distinguishes between object and background by training a classifier to use background information as negative samples and objects as positive samples. The candidate sample with the highest response is selected as the prediction result. The correlation form of the correlation filter is shown in Eq. (1):

$$\begin{aligned} E(\boldsymbol{h}) &= \min_{\boldsymbol{h}} \frac{1}{2}\left\|\boldsymbol{y}-\boldsymbol{x}\star\boldsymbol{h}\right\|_2^2+\frac{\lambda}{2}\left\|\boldsymbol{h}\right\|_2^2 \\ &= \min_{\boldsymbol{h}} \frac{1}{2}\left\|\boldsymbol{y}-\boldsymbol{C}\left(\boldsymbol{h}^{\mathrm{T}}\right)\boldsymbol{x}\right\|_2^2+\frac{\lambda}{2}\left\|\boldsymbol{h}\right\|_2^2 \\ &= \min_{\boldsymbol{h}} \frac{1}{2}\sum_{t=0}^{T-1}\left(\boldsymbol{y}(t)-\boldsymbol{x}^{\mathrm{T}}\boldsymbol{h}[\Delta\tau_t]\right)^2+\frac{\lambda}{2}\left\|\boldsymbol{h}\right\|_2^2 \\ &= \min_{\boldsymbol{h}} \frac{1}{2}\left\|\bar{\boldsymbol{y}}-\boldsymbol{C}\left(\boldsymbol{x}^{\mathrm{T}}\right)\boldsymbol{h}\right\|_2^2+\frac{\lambda}{2}\left\|\boldsymbol{h}\right\|_2^2 \\ &= \min_{\boldsymbol{h}} \frac{1}{2}\sum_{t=0}^{T-1}\left(\bar{\boldsymbol{y}}(t)-\boldsymbol{h}^{\mathrm{T}}\boldsymbol{x}[\Delta\tau_t]\right)^2+\frac{\lambda}{2}\left\|\boldsymbol{h}\right\|_2^2 \end{aligned} \tag{1}$$

where $\boldsymbol{x}\in\mathbb{R}^{T\times 1}$ represents the column vector form of the object sample after the weighted cosine window, T represents the size $\boldsymbol{x}$. $\boldsymbol{y}\in\mathbb{R}^{T\times 1}$ is the expected correlation response value. $\bar{\boldsymbol{y}}$ is reverse signal of $\boldsymbol{y}$ by setting the first element of $\boldsymbol{y}$ as the first element of $\bar{\boldsymbol{y}}$ and setting the reversed second to the final elements of $\boldsymbol{y}$ as the second to the final elements of $\bar{\boldsymbol{y}}$. For example, if $\boldsymbol{y}=[1,2,3,4]^{\mathrm{T}}$, then $\bar{\boldsymbol{y}}=[1,4,3,2]^{\mathrm{T}}$. $\boldsymbol{h}\in\mathbb{R}^{T\times 1}$ represents the filter. $\star$ represents the correlation operator, that is $(\boldsymbol{x}\star\boldsymbol{h})(n)=\sum_{m=0}^{T-1}\boldsymbol{x}(m)\boldsymbol{h}(n+m)$. λ represents the balancing parameter for balancing the fidelity term $\frac{1}{2}\left\|\boldsymbol{y}-\boldsymbol{x}\star\boldsymbol{h}\right\|_2^2$ and the ridge regression regularity term $\frac{\lambda}{2}\left\|\boldsymbol{h}\right\|_2^2$. $C\left(\boldsymbol{h}^{\mathrm{T}}\right)$ represents the row circulant matrix, superscript T indicates transpose operation. For example, if $\boldsymbol{h}^{\mathrm{T}}=[1,2,3,4]$, then $C\left(\boldsymbol{h}^{\mathrm{T}}\right)=\begin{pmatrix}1&2&3&4\\4&1&2&3\\3&4&1&2\\2&3&4&1\end{pmatrix}$. $C\left(\boldsymbol{h}^{\mathrm{T}}\right)$ satisfies $C\left(\boldsymbol{h}^{\mathrm{T}}\right)\boldsymbol{x}=\boldsymbol{x}\star\boldsymbol{h}$ and $C\left(\boldsymbol{h}^{\mathrm{T}}\right)\boldsymbol{x}=\overline{C\left(\boldsymbol{x}^{\mathrm{T}}\right)\boldsymbol{h}}$. Let $\boldsymbol{r}=C\left(\boldsymbol{h}^{\mathrm{T}}\right)\boldsymbol{x}$, then $\boldsymbol{r}\in\mathbb{R}^{T\times 1}$ and $\boldsymbol{r}(t)=\boldsymbol{x}^{\mathrm{T}}\boldsymbol{h}[\Delta\tau_t]$ ($\boldsymbol{h}[\Delta\tau_t]=\mathrm{circshift}(\boldsymbol{h},t)$, $\mathrm{circshift}(\boldsymbol{h},t)$ represents the cyclic shift operator that shifts the signal by t steps, $t=0,1,\cdots,T-1$.).

To calculate the correlation filter directly from the frequency domain, Eq. (1) is written in convolutional form:

$$E(\boldsymbol{h})=\min_{\boldsymbol{h}}\frac{1}{2}\left\|\boldsymbol{y}-\bar{\boldsymbol{h}}*\boldsymbol{x}\right\|_2^2+\frac{\lambda}{2}\left\|\bar{\boldsymbol{h}}\right\|_2^2 \tag{2}$$

where $*$ represents the convolution operator and satisfies $\boldsymbol{x} \star \boldsymbol{h} = \boldsymbol{x} * \overline{\boldsymbol{h}}$.

According to the convolution theorem [31] and Parseval's theorem [32], Eq. (2) is written in frequency domain form:

$$E(\boldsymbol{h}) = \min_{\hat{\boldsymbol{h}}^*} \frac{1}{2T} \left\| \hat{\boldsymbol{y}} - \hat{\boldsymbol{h}}^* \odot \hat{\boldsymbol{x}} \right\|_2^2 + \frac{\lambda}{2T} \left\| \hat{\boldsymbol{h}}^* \right\|_2^2 \tag{3}$$

where $\hat{\boldsymbol{h}}$ is the Fourier transform of $\boldsymbol{h}$, and $\hat{\boldsymbol{h}}^*$ is the conjugate signal of $\hat{\boldsymbol{h}}$. $\hat{\boldsymbol{h}}^*$ satisfies $\hat{\boldsymbol{h}}^* = \mathcal{F}\left(\overline{\boldsymbol{h}}\right)$, $\mathcal{F}$ denotes the one-dimensional Fourier operator. The symbol $\odot$ represents the entry-wise multiplication operation.

By taking the first-order partial derivative of $\hat{\boldsymbol{h}}^*$ and setting it to zero, the solution is:

$$\hat{\boldsymbol{h}}^* = \frac{\hat{\boldsymbol{y}} \odot \hat{\boldsymbol{x}}^*}{\hat{\boldsymbol{x}}^* \odot \hat{\boldsymbol{x}} + \lambda} \tag{4}$$

where the division in Eq. (4) is the entry-wise division.

For a new sample $\hat{\boldsymbol{z}}$, the corresponding spatial response is:

$$\boldsymbol{r} = \mathbf{real}\{\mathcal{F}^{-1}(\hat{\boldsymbol{h}}^* \odot \hat{\boldsymbol{z}})\} \tag{5}$$

where **real** represents the real part-taking operator. $\mathcal{F}^{-1}$ represents the Fourier inverse transform operator.

When multi-channel features such as HOG features, CN features, gray features, CNN features, etc., are extracted (assuming the number of feature channels is K), the objective function of learning multi-channel CFs in the spatial domain is:

$$\begin{aligned}
E(\boldsymbol{h}_k) &= \min_{\boldsymbol{h}_k} \frac{1}{2} \left\| \boldsymbol{y} - \sum_{k=1}^{K} \boldsymbol{x}_k \star \boldsymbol{h}_k \right\|_2^2 + \frac{\lambda}{2} \sum_{k=1}^{K} \left\| \boldsymbol{h}_k \right\|_2^2 \\
&= \min_{\boldsymbol{h}_k} \frac{1}{2} \left\| \boldsymbol{y} - \sum_{k=1}^{K} \overline{\boldsymbol{h}}_k * \boldsymbol{x}_k \right\|_2^2 + \frac{\lambda}{2} \sum_{k=1}^{K} \left\| \overline{\boldsymbol{h}}_k \right\|_2^2 \\
&= \min_{\boldsymbol{h}_k} \frac{1}{2} \left\| \boldsymbol{y} - \sum_{k=1}^{K} C\left(\boldsymbol{h}_k^{\mathrm{T}}\right) \boldsymbol{x}_k \right\|_2^2 + \frac{\lambda}{2} \sum_{k=1}^{K} \left\| \boldsymbol{h}_k \right\|_2^2 \\
&= \min_{\boldsymbol{h}_k} \frac{1}{2} \sum_{t=0}^{T-1} \left(\boldsymbol{y}(t) - \sum_{k=1}^{K} \boldsymbol{x}_k^{\mathrm{T}} \boldsymbol{h}_k[\Delta\tau_t] \right)^2 + \frac{\lambda}{2} \sum_{k=1}^{K} \left\| \boldsymbol{h}_k \right\|_2^2 \\
&= \min_{\boldsymbol{h}_k} \frac{1}{2} \left\| \overline{\boldsymbol{y}} - \sum_{k=1}^{K} C\left(\boldsymbol{x}_k^{\mathrm{T}}\right) \boldsymbol{h}_k \right\|_2^2 + \frac{\lambda}{2} \sum_{k=1}^{K} \left\| \boldsymbol{h}_k \right\|_2^2 \\
&= \min_{\boldsymbol{h}_k} \frac{1}{2} \sum_{t=0}^{T-1} \left(\overline{\boldsymbol{y}}(t) - \sum_{k=1}^{K} \boldsymbol{h}_k^{\mathrm{T}} \boldsymbol{x}_k[\Delta\tau_t] \right)^2 + \frac{\lambda}{2} \sum_{k=1}^{K} \left\| \boldsymbol{h}_k \right\|_2^2
\end{aligned} \tag{6}$$

where $\boldsymbol{x}_k \in \mathbb{R}^{T \times 1}$ is the feature image of the k-th channel extracted from the entire image. $\boldsymbol{h}_k \in \mathbb{R}^{T \times 1}$ is the correlation filter of the k-th channel.

Since Eq. (6) can be expressed in convolution form, according to the convolution theorem and Parseval's theorem, its frequency domain expression is:

$$E(\boldsymbol{h}_k) = \min_{\hat{\boldsymbol{h}}_k^*} \frac{1}{2T} \left\| \hat{\boldsymbol{y}} - \sum_{k=1}^{K} \hat{\boldsymbol{h}}_k^* \odot \hat{\boldsymbol{x}}_k \right\|_2^2 + \frac{\lambda}{2T} \left\| \hat{\boldsymbol{h}}_k^* \right\|_2^2 \tag{7}$$

By taking the first-order partial derivative of $\hat{\boldsymbol{h}}_k^*$ and setting it to zero, we have:

$$\hat{\boldsymbol{h}}_k^* = \frac{\hat{\boldsymbol{y}} \odot \hat{\boldsymbol{x}}_k^*}{\sum_{k=1}^{K} \hat{\boldsymbol{x}}_k \odot \hat{\boldsymbol{x}}_k^* + \lambda} \tag{8}$$

where the division in Eq. (8) is the entry-wise division.

For a new sample $\hat{\boldsymbol{z}}_k$ (weighted cosine window function), the corresponding spatial response is:

$$\boldsymbol{r} = \mathbf{real}\{\mathcal{F}^{-1}(\sum_{k=1}^{K} \hat{\boldsymbol{h}}_k^* \odot \hat{\boldsymbol{z}}_k)\} \tag{9}$$

The traditional DCF methods form a circulant matrix through the patches obtained by T cyclic shifts. Then, they utilize the diagonalization property of the circulant matrix to transform spatial domain correlation operations into frequency domain entry-wise multiplication operations, enabling efficient filter training. Due to the object sample containing insufficient background information, the direct cyclic shift of the object sample increases the risk of overfitting. At the same time, the cyclic shift operation based on periodic boundary conditions also causes boundary effects.

B. The Background-Aware Correlation Filter Model

The traditional DCFs often suffer from undesired boundary effects. Several regional expansion methods have been proposed in the past few years to compensate for this shortcoming. BACF extends the circulant matrix sampling region to a larger scope than the object region. It densely samples negative samples that contain background information. In this way, BACF fully explores the samples from the background. This not only increases the number of negative samples but also ensures the circulant structure of the samples. Due to the expansion of the search range, the filter can track the object at a relatively high speed. As shown in Fig. 1, the blue box indicates that the traditional DCFs directly cyclic shift operation on the object sample, and the orange box indicates that the BACF uses the cropping matrix to extract patches from the background densely.

BACF alleviates the boundary effect caused by cyclic shift by densely extracting negative samples from real-world scenes. However, BACF uses the traditional update strategy and simple hand-crafted features. The lack of feature generalization ability leads to the overfitting phenomenon in complex scenes and reduces the robustness of the appearance model. Meanwhile, expanding the search region also introduces more background noise, which can easily cause filter degradation in occluded scenes.

III. Proposed Method

As mentioned before, the cyclic shift in DCF causes boundary effects, which limit the sample area, resulting in the tracker's lack of effective background information. At the same time, DCF trackers based on hand-crafted features cannot accurately identify changes in the object's appearance, limiting the algorithm's performance. However, deep learning models have become an important means of improving tracking accuracy due to their large model capacity and powerful feature learning capabilities. Therefore, based on these considerations, this paper proposes a background-aware correlation filter for object tracking with deep CNN features to enable high-performance visual tracking.

A. Proposed Model

By applying the cropping matrix $\boldsymbol{P}$ on the training

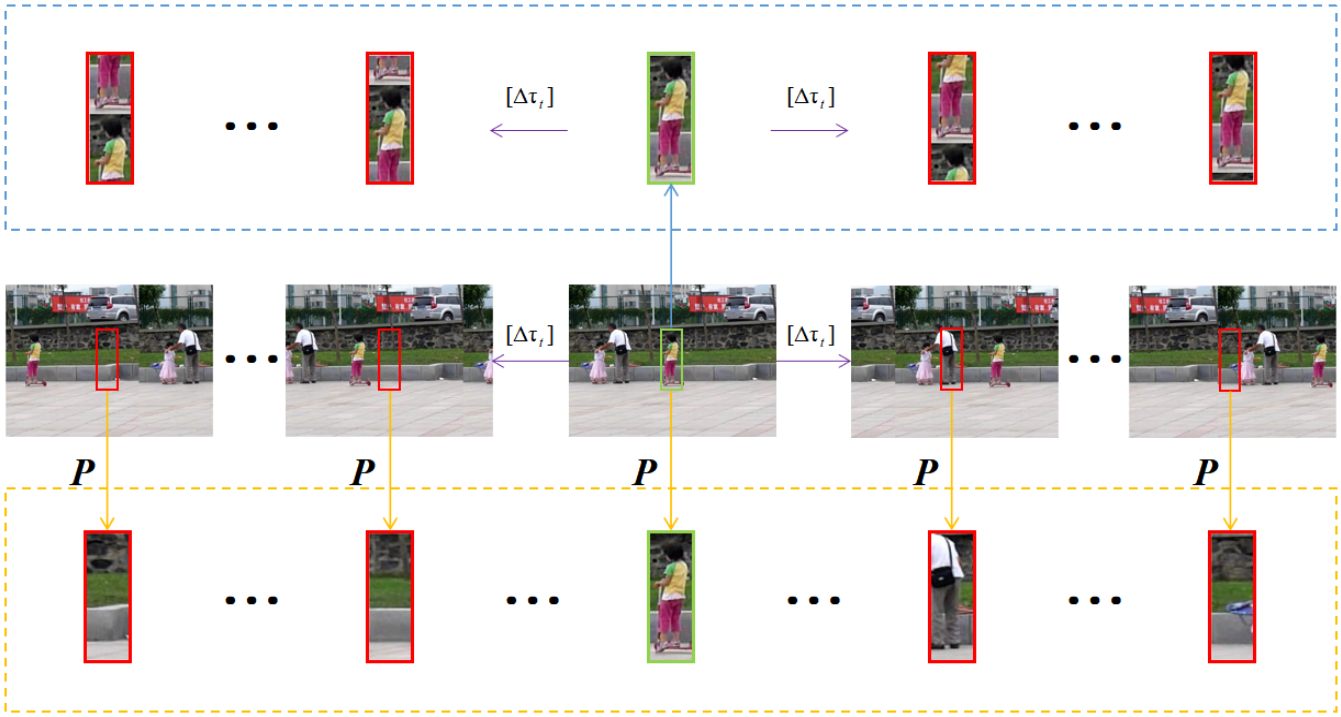


Fig. 1. Schematic of background-aware sampling and object cyclic shift sampling.

samples of the filter, the proposed DeepBACF can employ not only the object information but also the real background information. Meanwhile, multi-channel deep features are introduced to improve the accuracy of object appearance modeling. The cost function can be expressed as:

$$\begin{aligned} E(\boldsymbol{h}_k) &= \min_{\boldsymbol{h}_k} \frac{1}{2}\sum_{t=0}^{T-1}\left(\bar{\boldsymbol{y}}(t) - \sum_{k=1}^{K} \boldsymbol{h}_k^{\mathrm{T}} \boldsymbol{P}\boldsymbol{x}_k[\Delta\tau_t] \right)^2 + \frac{\lambda}{2}\sum_{k=1}^{K}\|\boldsymbol{h}_k\|_2^2 \\ &= \min_{\boldsymbol{h}_k} \frac{1}{2}\sum_{t=0}^{T-1}\left(\boldsymbol{y}(t) - \sum_{k=1}^{K} \boldsymbol{x}_k^{\mathrm{T}} \boldsymbol{P}\boldsymbol{h}_k[\Delta\tau_t] \right)^2 + \frac{\lambda}{2}\sum_{k=1}^{K}\|\boldsymbol{h}_k\|_2^2 \end{aligned} \tag{10}$$

where $\boldsymbol{P} \in \mathbb{R}^{T\times T}$, $\boldsymbol{P}(i,i) = \begin{cases} 1, i \in I \\ 0, others \end{cases}$ is the cropping matrix, and I is represents the set of cropped pixels determined by the size of the object template. Applying the cropping operator $\boldsymbol{P}\boldsymbol{x}_k[\Delta\tau_t]$ to the training samples, the region with the center size of D (D is the size of the foreground object after being pulled into a column vector, where $T \gg D$.) is cropped from the shifted image so that the smaller filter can act on the larger search samples.

To improve computational efficiency, an auxiliary variable $\hat{\boldsymbol{g}}$ is introduced. According to Parseval's theorem, Eq. (10) is transformed into the following equation:

$$\begin{aligned} E(\boldsymbol{h}, \hat{\boldsymbol{g}}) &= \min_{\boldsymbol{h},\hat{\boldsymbol{g}}} \frac{1}{2T}\left\| \hat{\boldsymbol{y}} - \hat{\boldsymbol{X}}^{(d)}\hat{\boldsymbol{g}} \right\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{h}\|_2^2 \\ s.t. \hat{\boldsymbol{g}} &= \sqrt{T}(\boldsymbol{I}_K \otimes \boldsymbol{F}\boldsymbol{P})\boldsymbol{h} \end{aligned} \tag{11}$$

where $\otimes$ operator represents the Kronecker product. $\boldsymbol{I}_K$ is the identity matrix of $K \times K$. $\boldsymbol{F}$ is the discrete normalized Fourier transform coefficient matrix of $T \times T$. The size of $\hat{\boldsymbol{X}}^{(d)} = [\mathbf{Diag}(\hat{\boldsymbol{x}}_1), \mathbf{Diag}(\hat{\boldsymbol{x}}_2), ..., \mathbf{Diag}(\hat{\boldsymbol{x}}_K)]$ is $T \times KT$. **Diag** represents the operator that stacks the column vectors onto the diagonal of the diagonal matrix. $\boldsymbol{h} = \left(\bar{\boldsymbol{h}}_1^{\mathrm{T}}, \bar{\boldsymbol{h}}_2^{\mathrm{T}}, \cdots, \bar{\boldsymbol{h}}_K^{\mathrm{T}}\right)^{\mathrm{T}}$ and $\hat{\boldsymbol{g}} = \left(\hat{\boldsymbol{g}}_1^{\mathrm{T}}, \hat{\boldsymbol{g}}_2^{\mathrm{T}}, \cdots, \hat{\boldsymbol{g}}_K^{\mathrm{T}}\right)^{\mathrm{T}}$ show the $KT \times 1$ over-complete representations by connecting their K vectorized channels.

The workflow of the DeepBACF tracker is shown in Fig. 2. The tracker samples the whole image and crops the image with a small window to get high-quality samples. The object and background patches are weighted by a cosine window and sent to the feature extractor. Given a search window in the $(f+1)$ frame, multi-channel features $\hat{\boldsymbol{Z}}_s^{(d)} = [\mathbf{Diag}(\hat{\boldsymbol{z}}_1^s), \mathbf{Diag}(\hat{\boldsymbol{z}}_2^s), ..., \mathbf{Diag}(\hat{\boldsymbol{z}}_K^s)]$ (where $\hat{\boldsymbol{z}}_k^s$ is the k-th frequency domain feature of the scaled sample $\boldsymbol{z}^s$, $s = 1, 2, \cdots, S$ is the scale index of the muti-scale samples, S is the scale number of the muti-scale samples.) are extracted at different scales to accommodate scale changes. This paper applies various features to characterize the object appearance model, including deep features extracted by the deep convolutional network, HOG features, and CN features. The tracking framework that utilizes deep convolutional features to train correlation filters possesses rich semantic information. Finally, this paper constructs an anti-occlusion mechanism to break the update strategy of traditional correlation filters. It evaluates samples with the largest correlation response, stores reliable samples in a multimodal template pool, and employs a threshold to prevent unreliable samples from contaminating the appearance model and filters. This approach enhances the reliability of the samples.

The detection step is performed by the product of the filter $\hat{\boldsymbol{g}}$ and the new frame sample $\hat{\boldsymbol{Z}}_s^{(d)}$, that is:

$$\boldsymbol{r}_s = \mathbf{real}\{\mathcal{F}^{-1}(\hat{\boldsymbol{Z}}_s^{(d)}\hat{\boldsymbol{g}})\} \tag{12}$$

where the filter is trained independently on each channel. By comparing the channel responses of each scale, the maximum value of the response corresponds to the position and size of the object in this frame. The object's position and scale are finally estimated by selecting the maximum value of the $\boldsymbol{r}_s (s = 1, 2, \cdots, S)$.

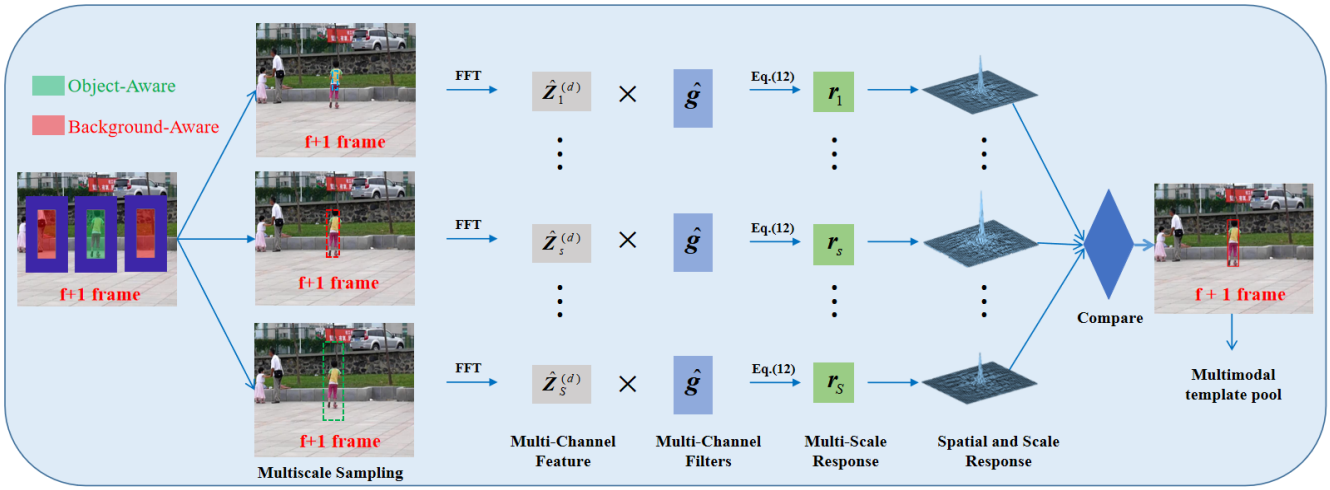


Fig. 2. General structure diagram of the proposed method.

B. Anti-occlusion Strategy

The model updating strategy is crucial in the object-tracking process. If the template is not updated, the apparent changes of the template cannot be perceived in time, while unprincipled overall updating of the template will introduce invalid apparent changes. To effectively solve the above problems, this paper readjusts the update strategy of the appearance model. A multimodal template pool comprises patches judged as positive samples in history. The historical multimodal samples can supervise the current best candidate samples. The sample with the largest response per frame is obtained through the filter, and the feature information of the candidate sample is extracted. If the similarity between the feature information of the candidate sample and that of the historical positive sample exceeds the set threshold, then this sample is deemed reliable. It is then incorporated into the multimodal template pool to update the filter and appearance model. On the contrary, the sample will be judged as unreliable. At this time, it is necessary to avoid introducing the sample into the template pool and stop updating the filter. The following briefly describes an anti-occlusion method that exploits the similarity of a multimodal template pool.

Firstly, the multimodal template pool needs to be constructed. Since there is no historical data for the first frame, the multimodal template pool will be filled with the patches of the first frame, i.e., $\boldsymbol{f}_n = \boldsymbol{x}^{(1)}(n=1,2,\cdots,N)$, where $\boldsymbol{x}^{(1)}$ denotes the patches of the first frame and $\boldsymbol{f}_n$ denotes the n-th column vector of the multimodal template pool $\boldsymbol{M}$. Starting from the second frame, it is assumed that the patch of the optimal sample obtained by the correlation response is $\boldsymbol{b}$, the HOG features of $\boldsymbol{M}(:,n)$ and $\boldsymbol{b}$ are extracted as shown in equations (13) and (14):

$$\boldsymbol{l}_{f_n} = \frac{\mathbf{HOG}(\boldsymbol{f}_n)}{\max(\mathbf{HOG}(\boldsymbol{f}_n))} \tag{13}$$

$$\boldsymbol{l}_b = \frac{\mathbf{HOG}(\boldsymbol{b})}{\max(\mathbf{HOG}(\boldsymbol{b}))} \tag{14}$$

where $\mathbf{HOG}$ represents the directional gradient histogram extraction operator. $\mathbf{HOG}(\boldsymbol{f}_n)$ and $\mathbf{HOG}(\boldsymbol{b})$ represent the HOG features of the multimodal template pool sample and the best candidate sample, respectively.

According to Eq. (15), whether the object is occluded can be determined:

$$\max\left(\cos\left(\boldsymbol{l}_{f_n}, \boldsymbol{b}\right)\right) > \tau \tag{15}$$

where τ is a threshold value in the range $[0,1]$. When $\max\left(\cos\left(\boldsymbol{l}_{f_n}, \boldsymbol{b}\right)\right)$ is greater than the set threshold, it indicates that the object is not occluded. At this point, $\boldsymbol{b}$ can be updated to the object template pool, and the patch with the lowest similarity to $\boldsymbol{b}$ among the 2nd to N-th templates is eliminated. Otherwise, the object is considered occluded, and the information is not updated.

The multimodal template pool captures the variability of the object and evaluates the reliability of the optimal candidate samples. It extracts the HOG visual features of each patch for similarity measurement, aiming to prevent low-confidence samples from misleading the filter and appearance updates. This optimization of the algorithm's update mechanism enhances its anti-occlusion capability.

C. Solver for the Proposed Model

To solve Eq. (11), this paper introduces the Lagrangian vector $\hat{\boldsymbol{\zeta}}$ and constructs the Augmented Lagrangian Method (ALM):

$$\mathcal{L}(\hat{\boldsymbol{g}}, \boldsymbol{h}, \hat{\boldsymbol{\zeta}}) = \max_{\hat{\boldsymbol{\zeta}}} \left\{ \min_{\boldsymbol{h}, \hat{\boldsymbol{g}}} \left\{ \begin{array}{l} \frac{1}{2T}\left\|\hat{\boldsymbol{y}} - \hat{\boldsymbol{X}}^{(d)}\hat{\boldsymbol{g}}\right\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{h}\|_2^2 + \\ \hat{\boldsymbol{\zeta}}^{\mathrm{H}}(\hat{\boldsymbol{g}} - \sqrt{T}(\boldsymbol{I}_K \otimes \boldsymbol{FP})\boldsymbol{h}) + \\ \frac{\mu}{2}\left\|\hat{\boldsymbol{g}} - \sqrt{T}(\boldsymbol{I}_K \otimes \boldsymbol{FP})\boldsymbol{h}\right\|_2^2 \end{array} \right\} \right. \tag{16}$$

where μ is the penalty coefficient. $\hat{\boldsymbol{\zeta}} = [\hat{\boldsymbol{\zeta}}_1^{\mathrm{T}}, ..., \hat{\boldsymbol{\zeta}}_K^{\mathrm{T}}]^{\mathrm{T}}$ is the $KT \times 1$ Lagrangian vector of the Fourier transform. Superscript H indicates conjugate transpose operation. Eq. (16) is solved iteratively by the alternating direction method of multipliers (ADMM) to optimize and accelerate the calculation for each subproblem $\boldsymbol{h}$ and $\hat{\boldsymbol{g}}$.

The subproblem of $\boldsymbol{h}$ is:

$$\boldsymbol{h}=\arg\min_{\boldsymbol{h}}\left\{\begin{array}{l}\frac{\lambda}{2}\|\boldsymbol{h}\|_2^2+\hat{\boldsymbol{\zeta}}^{\mathrm{H}}(\hat{\boldsymbol{g}}-\sqrt{T}(\boldsymbol{I}_K\otimes\boldsymbol{FP})\boldsymbol{h})\\+\frac{\mu}{2}\left\|\hat{\boldsymbol{g}}-\sqrt{T}(\boldsymbol{I}_K\otimes\boldsymbol{FP})\boldsymbol{h}\right\|_2^2\end{array}\right\} \quad (17)$$

Taking the partial derivative of $\mathcal{L}$ as zero, that is:

$$\begin{aligned}&\frac{\partial\mathcal{L}}{\partial\boldsymbol{h}}=\lambda\boldsymbol{h}-\left[\hat{\boldsymbol{\zeta}}^{\mathrm{H}}\sqrt{T}(\boldsymbol{I}_K\otimes\boldsymbol{FP})\right]^{\mathrm{H}}\\&-\mu\sqrt{T}(\boldsymbol{I}_K\otimes\boldsymbol{FP})^{\mathrm{H}}(\hat{\boldsymbol{g}}-\sqrt{T}(\boldsymbol{I}_K\otimes\boldsymbol{FP})\boldsymbol{h})=\boldsymbol{0}\end{aligned} \quad (18)$$

Then, we have:

$$\begin{aligned}&\frac{\partial\mathcal{L}}{\partial\boldsymbol{h}}=\lambda\boldsymbol{h}-\sqrt{T}(\boldsymbol{I}_K\otimes\boldsymbol{PF}^{\mathrm{H}})\hat{\boldsymbol{\zeta}}\\&-\mu\sqrt{T}(\boldsymbol{I}_K\otimes\boldsymbol{PF}^{\mathrm{H}})\hat{\boldsymbol{g}}+\mu T\boldsymbol{h}=\boldsymbol{0}\end{aligned} \quad (19)$$

The closed-form solution of $\boldsymbol{h}$ is:

$$\boldsymbol{h}=\frac{\mu\boldsymbol{g}+\boldsymbol{\zeta}}{\frac{\lambda}{T}+\mu} \quad (20)$$

where $\boldsymbol{g}=\frac{1}{\sqrt{T}}(\boldsymbol{I}_K\otimes\boldsymbol{PF}^{\mathrm{H}})\hat{\boldsymbol{g}}$, $\boldsymbol{\zeta}=\frac{1}{\sqrt{T}}(\boldsymbol{I}_K\otimes\boldsymbol{PF}^{\mathrm{H}})\hat{\boldsymbol{\zeta}}$. The complexity of Eq. (20) is $\mathcal{O}(KT\log_2(T))$. $T\log_2(T)$ is the IFFT complexity for computing a signal of length T.

The subproblem of $\hat{\boldsymbol{g}}$ is:

$$\begin{aligned}\hat{\boldsymbol{g}}=\arg\min_{\hat{\boldsymbol{g}}}\{&\frac{1}{2T}\left\|\hat{\boldsymbol{y}}-\hat{\boldsymbol{X}}^{(d)}\hat{\boldsymbol{g}}\right\|_2^2\\&+\hat{\boldsymbol{\zeta}}^{\mathrm{H}}(\hat{\boldsymbol{g}}-\sqrt{T}(\boldsymbol{I}_K\otimes\boldsymbol{FP})\boldsymbol{h})\\&+\frac{\mu}{2}\left\|\hat{\boldsymbol{g}}-\sqrt{T}(\boldsymbol{I}_K\otimes\boldsymbol{FP})\boldsymbol{h}\right\|_2^2\}\end{aligned} \quad (21)$$

where the complexity of Eq. (21) is $\mathcal{O}(T^3K^3)$, it is difficult to perform real-time tracking because we need to solve $\hat{\boldsymbol{g}}$ at each ADMM iteration. Since $\hat{\boldsymbol{X}}^{(d)}$ is sparsely banded, i.e., the value of each pixel is independent, the subproblem $\hat{\boldsymbol{g}}$ can be decomposed into T independent mini-objectives to reduce the computational cost:

$$\begin{aligned}\hat{\boldsymbol{g}}(t)=\arg\min_{\hat{\boldsymbol{g}}(t)}\{&\frac{1}{2T}\left\|\hat{\boldsymbol{y}}(t)-\hat{\boldsymbol{x}}(t)^{\mathrm{T}}\hat{\boldsymbol{g}}(t)\right\|_2^2\\&+\hat{\boldsymbol{\zeta}}(t)^{\mathrm{H}}(\hat{\boldsymbol{g}}(t)-\hat{\boldsymbol{h}}(t))+\frac{\mu}{2}\left\|\hat{\boldsymbol{g}}(t)-\hat{\boldsymbol{h}}(t)\right\|_2^2\}\end{aligned} \quad (22)$$

where $\hat{\boldsymbol{y}}(t)$ depends on $\hat{\boldsymbol{x}}(t)=[\hat{\boldsymbol{x}}_1(t),\hat{\boldsymbol{x}}_2(t),\cdots,\hat{\boldsymbol{x}}_K(t)]^{\mathrm{T}}$ and $\hat{\boldsymbol{g}}(t)=[\hat{\boldsymbol{g}}_1(t),\hat{\boldsymbol{g}}_2(t),\cdots,\hat{\boldsymbol{g}}_K(t)]^{\mathrm{T}}$. $\hat{\boldsymbol{h}}(t)=\left[\hat{\boldsymbol{h}}_1^*(t),...,\hat{\boldsymbol{h}}_K^*(t)\right]^{\mathrm{T}}$.

Similarly, the closed-form solution of $\hat{\boldsymbol{g}}(t)$ is:

$$\begin{aligned}&\hat{\boldsymbol{g}}(t)=[(\hat{\boldsymbol{x}}^*(t)\hat{\boldsymbol{x}}(t)^{\mathrm{T}}+\mu T\boldsymbol{I}_K)]^{-1}\\&[\hat{\boldsymbol{y}}(t)\hat{\boldsymbol{x}}^*(t)-T\hat{\boldsymbol{\zeta}}(t)+\mu T\hat{\boldsymbol{h}}(t)]\end{aligned} \quad (23)$$

where the complexity of Eq. (23) is $\mathcal{O}(TK^3)$, and although this computation is much smaller than directly solving Eq. (21), it is still difficult to solve in real time.

We use the Sherman-Morrison formula to quickly calculate $(\boldsymbol{uv}^{\mathrm{T}}+\boldsymbol{A})^{-1}=\boldsymbol{A}^{-1}-(1+\boldsymbol{v}^{\mathrm{T}}\boldsymbol{A}^{-1}\boldsymbol{u})^{-1}(\boldsymbol{A}^{-1}\boldsymbol{uv}^{\mathrm{T}}\boldsymbol{A}^{-1})$, where $\boldsymbol{u}=\hat{\boldsymbol{x}}^*(t)$, $\boldsymbol{v}=\hat{\boldsymbol{x}}(t)$, $\boldsymbol{A}=\mu T\boldsymbol{I}_K$. Eq. (23) is rewritten as:

$$\begin{aligned}&\hat{\boldsymbol{g}}(t)=\frac{1}{\mu T}(\hat{\boldsymbol{y}}(t)\hat{\boldsymbol{x}}^*(t)-T\hat{\boldsymbol{\zeta}}(t)+\mu T\hat{\boldsymbol{h}}(t))\\&-\frac{\hat{\boldsymbol{x}}^*(t)}{\mu T\alpha}(\hat{\boldsymbol{y}}(t)\hat{s}_{\hat{\boldsymbol{x}}}(t)-T\hat{s}_{\hat{\boldsymbol{\zeta}}}(t)+\mu T\hat{s}_{\hat{\boldsymbol{h}}}(t))\end{aligned} \quad (24)$$

where $\hat{s}_{\hat{\boldsymbol{x}}}(t)=\hat{\boldsymbol{x}}(t)^{\mathrm{T}}\hat{\boldsymbol{x}}^*(t)$, $\hat{s}_{\hat{\boldsymbol{\zeta}}}(t)=\hat{\boldsymbol{x}}(t)^{\mathrm{T}}\hat{\boldsymbol{\zeta}}(t)$, $\hat{s}_{\hat{\boldsymbol{h}}}(t)=\hat{\boldsymbol{x}}(t)^{\mathrm{T}}\hat{\boldsymbol{h}}(t)$ and $\alpha=\hat{s}_{\hat{\boldsymbol{x}}}(t)+\mu T$. The cost of computing $\hat{\boldsymbol{g}}$ using Eq. (24) is $\mathcal{O}(TK)$, which is much smaller than the computation of Eq. (23) $(\mathcal{O}(TK^3))$.

The Lagrangian parameter $\hat{\boldsymbol{\zeta}}$ is updated according to the following equation:

$$\hat{\boldsymbol{\zeta}}^{(i+1)}=\hat{\boldsymbol{\zeta}}^{(i)}+\mu(\hat{\boldsymbol{g}}^{(i+1)}-\hat{\boldsymbol{h}}^{(i+1)}) \quad (25)$$

where $\hat{\boldsymbol{g}}^{(i+1)}$ and $\hat{\boldsymbol{h}}^{(i+1)}$ are the current solutions of the above subproblems at iteration $i+1$ in ADMM, and $\hat{\boldsymbol{h}}^{(i+1)}=\sqrt{T}(\boldsymbol{I}_K\otimes\boldsymbol{FP})\boldsymbol{h}^{(i+1)}$. A common scheme for updating μ is $\mu^{(i+1)}=\min(\mu_{max},\beta\mu^{(i)})$.

D. Online Updating of Appearance Model

In this paper, the linear interpolation method is used as the model updating strategy:

$$\hat{\boldsymbol{x}}_{model}^{(f)}(t)=(1-\eta)\hat{\boldsymbol{x}}_{model}^{(f-1)}(t)+\eta\hat{\boldsymbol{x}}^{(f)}(t) \quad (26)$$

where η is the online update rate. $\hat{\boldsymbol{x}}_{model}^{(f)}(t)$ is the signal model at frame f, $\hat{\boldsymbol{x}}^{(f)}(t)$ is the image feature at frame f, we use $\hat{\boldsymbol{x}}_{model}^{(f)}(t)$ instead of $\hat{\boldsymbol{x}}(t)$ in Eq. (24) to calculate $\hat{\boldsymbol{g}}(t)$, $\hat{s}_{\hat{\boldsymbol{x}}}(t)$, $\hat{s}_{\hat{\boldsymbol{\zeta}}}(t)$ and $\hat{s}_{\hat{\boldsymbol{h}}}(t)$.

IV. EXPERIMENTS

In this section, we first analyze the impact of each module proposed in the DeepBACF framework on performance. Then, comprehensive experiments are performed on video challenge sequences from the OTB dataset to evaluate the accuracy and robustness of DeepBACF.

A. Ablation Experiment

The Ablation of Anti-occlusion Strategy

To verify whether the anti-occlusion strategy of the multimodal template pool proposed in section 3.2 improves the tracking effect of the algorithm, this section compares the anti-occlusion performance of the DeepBACF algorithm and BACF algorithm. As shown in Fig. 3, both the DeepBACF algorithm with anti-occlusion strategy and the BACF algorithm without occlusion module can achieve good tracking of the object when the object is not occluded in frame 21. In frames 65 to 147, the object is occluded. The green dashed tracking frame representing the BACF algorithm cannot capture the object. In contrast, the red solid tracking frame representing the DeepBACF algorithm is not interfered with by occlusion and can track the object accurately. Experiments show that the multimodal template pool's anti-occlusion strategy can solve object-tracking drift in occluded scenes.

The Ablation of Deep Features

To evaluate the improvement of deep CNN in complex scenarios, this section conducts comparative experiments between the DeepBACF algorithm with the introduction of deep features and the BACF algorithm based on hand-crafted features. As shown in Fig. 4, the red solid tracking frame represents the DeepBACF algorithm, and the green dashed tracking frame represents the BACF algorithm. Both algorithms track the object normally in frame 13. In frames 80 to 362, under complex scenes such as rotation, occlusion, and photometric changes, the BACF algorithm has insufficient discrimination ability, resulting in misjudgment of the object and thus failing to achieve accurate tracking. In contrast, the DeepBACF algorithm has a better tracking effect and can achieve real-time tracking. Experiments show that the deep features extracted by CNN can significantly improve the robustness of the tracker to geometric and photometric changes.

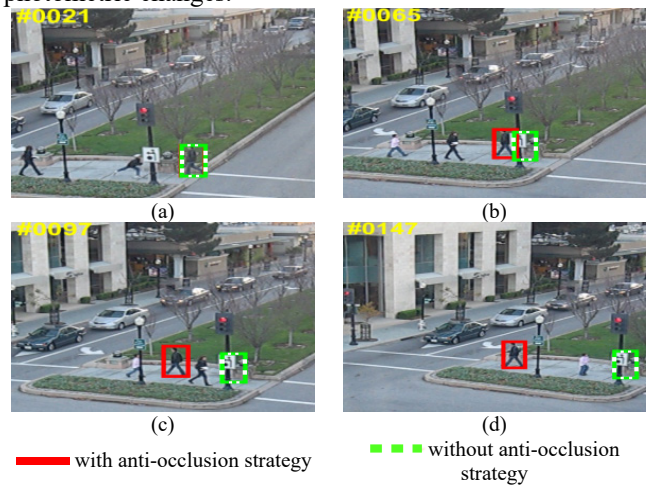


Fig. 3. Comparison experiment with or without anti-occlusion strategy.

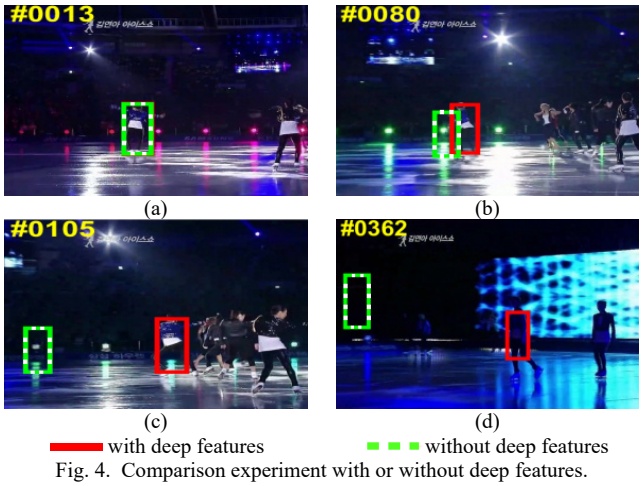


Fig. 4. Comparison experiment with or without deep features.

B. Qualitative Evaluation

To further evaluate the performance of DeepBACF, this section provides an overview of the tracking performance of DeepBACF with nine other state-of-the-art algorithms for different scenarios on the OTB dataset, including CSK [8], KCF [9], DSST [21], SRDCF [10], BACF [13], CSR-DCF [33], LCT2 [34], ARCF_H [35], and Auto Track [36]. Fig. 5 intercepts some frames of five representative video sequences, which include typical challenges such as fast motion, occlusion, rotation, background clutter, and similar objects in object tracking. It shows the tracking performance of this algorithm and other algorithms compared under different tracking challenges.

The following is a qualitative analysis of several typical tracking challenge factors:

(1) Occlusion. In Fig. 5 (b), the DeepBACF algorithm always tracks the object accurately after a passerby occludes the little girl, while all other algorithms misjudge the object, resulting in lost tracking. In Fig. 5 (c), there is a local occlusion of the object, and only the DeepBACF algorithm and the STRCF algorithm track accurately, while the other algorithms show obvious tracking drift. In Fig. 5 (d), after the object is occluded by a utility pole during running, the BACF algorithm shows significant tracking loss. In contrast, the improved DeepBACF algorithm consistently and robustly tracks the object. It can be seen that the multimodal template pool can effectively deal with partial occlusion and complete occlusion situations and improve the reliability of tracking.

(2) Background clutter. Fig. 5 (c) box video sequence has complex background and scale changes. The color of the object and the surrounding background on the image are similar, making it difficult to distinguish between the object and the background. The LCT2 and CSK algorithms mistook the background as the object in frame 312 and followed the wrong object in the subsequent frames. In frame 569, the STRCF algorithm could follow the object, but the scale is poor. In contrast, the DeepBACF algorithm could achieve accurate positioning throughout the process without tracking error or drift phenomena. The DeepBACF algorithm relies on rich background information to reduce the impact of background clutter on the object and greatly improve the accuracy of the tracking algorithm.

(3) Fast motion. In Fig. 5 (e), the object moves fast and has some motion blur accompanied by rotation. The CSK, BACF, ARCF_H, Auto Track, LCT2, DSST, and KCF algorithms lose the object when the lens is strongly blurred, resulting in a certain degree of drift. The apparent state of the object changes in a short time after fast motion. However, the DeepBACF algorithm will make scale-adaptive changes according to the real world and can carry out robust tracking.

(4) Rotation. Rotation is present in the video sequences Biker, Box, and Girl2 in Fig. 5. In Fig. 5 (b), only the DeepBACF algorithm can achieve accurate tracking. In Fig. 5 (c), the LCT2 and CSK algorithms drift more when the object is rotated, and all other algorithms track more stably. In Fig. 5 (a) Biker video sequence at frame 78, when the object rotates 180 degrees, all other algorithms experience different degrees of tracking drift. The DeepBACF algorithm always adapts to the object change and achieves accurate tracking and positioning according to the object's movement.

In summary, the anti-occlusion module introduced by the DeepBACF algorithm can effectively deal with all kinds of occlusion problems. At the same time, it shows that the deep features extracted by the CNN framework are reliable. The tracking performance is minimally affected by factors such as illumination and posture, thereby significantly enhancing the algorithm's precision.

C. Quantitative Evaluation

Comprehensive Evaluation: Table I shows the precision

Fig. 5. Visualize the position error and overlap rate on different video sequences.

TABLE I
COMPREHENSIVE EVALUATION OF OUR PROPOSED TRACKER AND OTHER NINE STATE-OF-THE-ART TRACKERS ON OTB100.

Evaluation	ARCF_H	BACF	KCF	DSST	CSR-DCF	STRCF	Auto Track	CSK	LCT2	DeepBACF
Precision	0.790	0.809	0.698	0.679	0.802	0.857	0.778	0.517	0.753	**0.886**
Success rate	0.730	0.759	0.553	0.600	0.702	0.791	0.713	0.410	0.624	**0.829**

TABLE II
PRECISION OF 11 CHALLENGING ATTRIBUTES ON OTB100.

Attributes	ARCF_H	BACF	KCF	DSST	CSR-DCF	STRCF	Auto Track	CSK	LCT2	DeepBACF
IV	0.769	0.782	0.724	0.715	0.779	0.816	0.752	0.484	0.721	0.847
OPR	0.737	0.767	0.677	0.644	0.760	0.837	0.738	0.485	0.733	0.864
SV	0.736	0.755	0.638	0.633	0.739	0.828	0.718	0.450	0.665	0.841
OCC	0.675	0.714	0.636	0.589	0.700	0.794	0.696	0.430	0.661	0.849
DEF	0.740	0.747	0.619	0.533	0.777	0.822	0.730	0.452	0.666	0.836
MB	0.717	0.715	0.607	0.567	0.741	0.800	0.708	0.351	0.640	0.837
FM	0.758	0.787	0.626	0.552	0.766	0.802	0.758	0.399	0.681	0.870
IPR	0.750	0.777	0.705	0.691	0.781	0.796	0.744	0.512	0.765	0.843
OV	0.674	0.747	0.514	0.481	0.691	0.766	0.729	0.282	0.591	0.880
BC	0.803	0.801	0.719	0.703	0.778	0.871	0.758	0.577	0.733	0.891
LR	0.692	0.741	0.560	0.567	0.677	0.737	0.763	0.422	0.537	0.868

and success rate of the DeepBACF algorithm against nine other state-of-the-art object tracking algorithms on the OTB dataset, including CSK, CSR-DCF, SRDCF, LCT2, KCF, DSST, BACF, Auto Track, and ARCF_H. Bold, underlined, and wavy lines indicate the top three algorithms. As shown in Table I, the DeepBACF algorithm is ranked first with 88.6% precision, which is 2.9% better than the second-ranked STRCF algorithm. While the BACF algorithm is ranked third with 80.9% precision. The DeepBACF algorithm is ranked first with an 82.9% success rate, a 7% improvement over the BACF algorithm. The evaluation results show that the DeepBACF algorithm performs well in overall tracking performance.

Attribute-Based Evaluation: We use 11 challenging attributes (IV denotes illumination variation, OPR denotes out-of-plane rotation, SV denotes scale variation, OCC denotes occlusion, DEF denotes deformation, MB denotes motion blur, FM denotes fast motion, IPR denotes in-plane rotation, OV denotes out of view, BC denotes background clutter, LR denotes low-resolution.) on the OTB100 dataset

TABLE III
SUCCESS RATE OF 11 CHALLENGING ATTRIBUTES ON OTB100.

Attributes	ARCF_H	BACF	KCF	DSST	CSR-DCF	STRCF	Auto Track	CSK	LCT2	DeepBACF
IV	0.746	0.756	0.55	0.649	0.726	0.754	0.729	0.395	0.592	0.814
OPR	0.649	0.695	0.527	0.551	0.645	0.754	0.654	0.376	0.602	0.784
SV	0.641	0.685	0.417	0.525	0.604	0.746	0.632	0.307	0.464	0.759
OCC	0.624	0.675	0.515	0.531	0.631	0.732	0.649	0.351	0.561	0.772
DEF	0.663	0.671	0.503	0.479	0.681	0.711	0.673	0.354	0.564	0.778
MB	0.703	0.707	0.554	0.551	0.709	0.766	0.681	0.342	0.617	0.816
FM	0.728	0.757	0.529	0.517	0.703	0.759	0.704	0.370	0.613	0.826
IPR	0.653	0.695	0.555	0.589	0.638	0.717	0.643	0.412	0.629	0.760
OV	0.618	0.692	0.467	0.442	0.578	0.697	0.667	0.278	0.531	0.796
BC	0.760	0.768	0.613	0.613	0.703	0.803	0.716	0.466	0.660	0.843
LR	0.568	0.663	0.295	0.442	0.434	0.652	0.669	0.277	0.295	0.732

TABLE IV
AVERAGE TRACKING OVERLAP RATE OF EACH TRACKING ALGORITHM IN SOME VIDEOS.

Video	ARCF_H	BACF	KCF	DSST	CSR-DCF	STRCF	Auto Track	CSK	LCT2	DeepBACF
Bird2	0.54	0.58	0.58	0.46	0.61	0.47	0.57	0.58	0.66	**0.70**
Dudek	0.75	0.83	0.73	0.78	0.83	0.83	0.80	0.72	0.74	**0.85**
Girl	0.52	0.66	0.55	0.44	0.16	0.71	0.21	0.37	0.68	**0.74**
Jumping	0.70	0.71	0.27	0.06	0.65	0.69	0.61	0.05	0.71	**0.79**
Subway	0.76	0.76	0.75	0.18	0.17	0.75	0.79	0.19	0.76	**0.82**
Car1	0.67	0.73	0.14	0.64	0.72	**0.87**	0.72	0.10	0.11	0.81
Crowds	0.76	0.75	**0.79**	0.74	0.58	0.71	0.67	0.76	0.74	0.78
Football	0.71	0.72	0.55	0.56	0.61	0.70	0.66	0.55	**0.74**	**0.74**
Average	0.676	0.718	0.545	0.482	0.541	0.716	0.629	0.415	0.643	**0.779**

TABLE V
AVERAGE CENTER POINT ERROR OF EACH TRACKING ALGORITHM IN SOME VIDEOS.

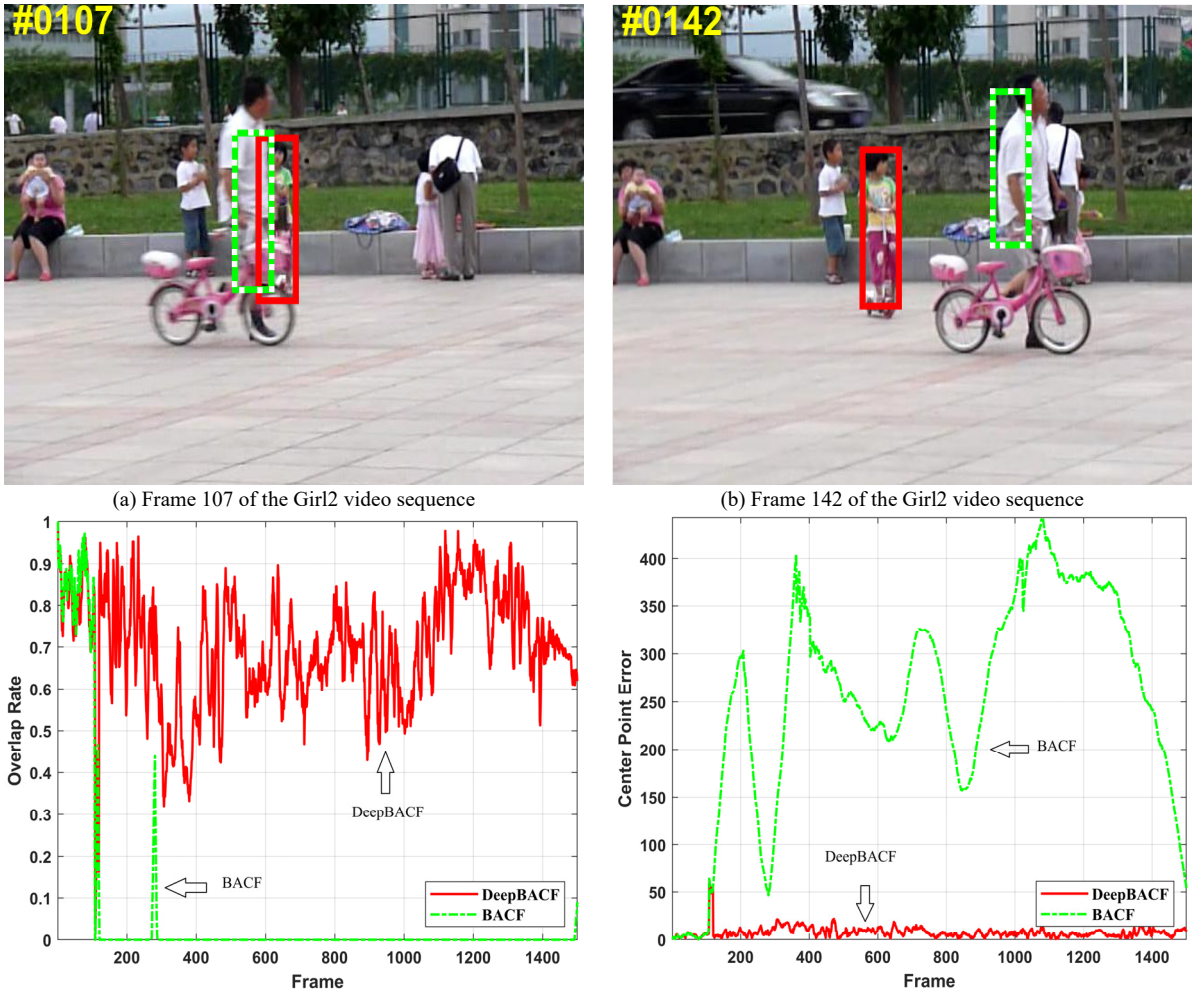
Video	ARCF_H	BACF	KCF	DSST	CSR-DCF	STRCF	Auto Track	CSK	LCT2	DeepBACF
Bird2	23.16	21.59	21.37	55.65	17.23	47.60	21.80	18.30	15.67	**8.89**
Dudek	9.37	9.85	11.38	13.35	9.94	10.91	12.05	13.39	9.97	**9.24**
Girl	9.93	6.32	11.92	10.96	38.58	**2.35**	16.76	19.34	5.10	3.25
Jumping	4.08	4.45	26.12	125.46	4.06	3.68	4.42	85.97	4.85	**3.26**
Subway	2.46	2.51	2.97	146.69	143.85	2.65	2.22	164.37	3.15	**2.16**
Car1	1.22	1.31	42.43	1.41	1.18	**0.96**	1.36	570.44	133.94	1.13
Crowds	3.03	3.43	3.07	3.76	5.82	3.85	4.27	3.69	3.71	**2.83**
Football	5.65	**3.82**	14.60	15.34	13.35	6.08	6.18	16.19	3.97	4.13
Average	7.363	6.660	16.733	46.578	29.251	9.760	8.633	111.461	22.545	**4.361**

TABLE VI
FPS OF EACH TRACKING ALGORITHM IN SOME VIDEOS.

Video	ARCF_H	BACF	KCF	DSST	CSR-DCF	STRCF	Auto Track	CSK	LCT2	DeepBACF
Car2	9.59	29.45	93.21	19.79	20.57	28.49	14.06	520.97	11.30	2.10
Coupon	32.80	27.37	68.26	10.54	18.74	27.89	12.32	293.59	8.84	2.12
Crossing	52.05	40.24	186.49	46.89	15.11	21.59	12.59	570.13	49.17	4.32
Dancer	32.60	26.87	58.77	13.00	20.96	27.53	12.86	312.70	22.22	2.25
Deer	26.92	22.82	71.35	12.54	12.01	12.07	10.29	32.29	8.37	2.40
Dog	30.90	25.44	102.53	22.01	14.56	14.16	9.20	591.17	23.03	2.34
Human7	29.64	24.39	76.98	11.62	14.03	13.17	7.07	303.43	18.14	1.99
Rubik	17.52	13.87	54.77	8.39	11.19	12.98	9.68	225.69	11.81	2.15
Singer1	23.94	20.35	50.24	3.19	10.22	12.16	7.98	86.35	19.92	1.98

to evaluate the accuracy and robustness of DeepBACF in various challenging scenarios. Table II shows the precision of DeepBACF and the other nine state-of-the-art tracking algorithms under each challenging attribute. Table III shows the success rate of DeepBACF and the other nine state-of-the-art tracking algorithms under each challenging attribute. Experiments show that the DeepBACF algorithm achieves satisfactory results on visual attributes, and its precision and success rate rank first. It is worth noting that the DeepBACF significantly outperforms other algorithms in fast motion, occlusion, and background clutter scenes.

Table IV lists the average tracking overlap rate between



(a) Frame 107 of the Girl2 video sequence

(b) Frame 142 of the Girl2 video sequence

(c) Average tracking overlap rate

(d) Average center point error

Fig. 6. Overlap rate and center point error of the Girl2 video sequence.

DeepBACF and other tracking algorithms in eight typical video sequences. A larger average tracking overlap rate indicates better tracking performance. Bold, underlined, and wavy lines indicate the top three algorithms. The average tracking overlap rate of the DeepBACF algorithm has an average value of 0.779, which is ranked first compared to other algorithms. The results show that DeepBACF achieves competitive tracking performance in complex scenes.

Table V lists the average center point error of DeepBACF and other tracking algorithms in eight typical videosequences. The smaller value indicates the smaller error with the real position of the object. Bold, underlined, and wavy lines indicate the top three algorithms. The total mean pixel error of the DeepBACF algorithm is 4.361 pixels, ranked first among the ten algorithms by a significant margin. The results show that DeepBACF has the highest tracking accuracy and achieves good tracking in different challenges.

Fig. 6 demonstrates the overlap rate and center point error of the DeepBACF and the BACF algorithms in the Girl2 video sequence. The red solid tracking frame represents the DeepBACF algorithm, and the green dashed tracking frame represents the BACF algorithm. The results indicate that despite various challenging factors such as occlusion, scale variation, deformation, motion blur, rotation, and more present in the Girl2 video sequence, the DeepBACF algorithm maintains accurate tracking of the object. Its tracking performance notably surpasses that of the BACF algorithm. The above experiments demonstrate that the deep learning mechanism and the anti-occlusion module can effectively improve the algorithm's robustness in tracking the object.

D. Limitation

Although introducing deep features can effectively improve the accuracy of algorithms, the large computational complexity involved in deep learning leads to poor speed. Table VI shows the FPS (Frames Per Second) of DeepBACF versus several traditional hand-crafted trackers in some video sequences. The results show that DeepBACF underperforms in terms of FPS compared to traditional hand-crafted trackers.

V. Conclusions

In order to solve the boundary effect and enhance the algorithm's anti-occlusion performance, this paper proposes a tracking algorithm based on a background-aware correlation filter in the depth-view domain, i.e., DeepBACF. Introducing deep features extracted by deep neural networks improves the robustness of the tracker, which is crucial for accurate object localization. The historical multimodal template pool selects the diversity and high-quality positive training sample sets. It enables the model to fully learn the insensitive features such as occlusion and deformation. This work has the following main advantages:

(1) The deep features extracted by CNN are employed to

train correlation filters, ensuring that the learned model can track robustly under fast motion, illumination variation, and other challenging factors. The algorithm also meets real-time requirements.

(2) The anti-occlusion strategy of the historical multimodal template pool is introduced to change the traditional algorithm update mode. This strategy sets the threshold mechanism during model updates, enabling the algorithm to screen reliable samples to address the challenge of robust tracking in complex occluded scenes.

(3) The ADMM algorithm is employed to optimize the model, effectively enhancing the tracking speed and performance of the algorithm.

Comprehensive experiments show that the DeepBACF algorithm achieves state-of-the-art accuracy, robustness, and efficiency under diverse features and novel anti-occlusion mechanisms. The subsequent work will improve the algorithm by combining spatial regularization and continuous convolution to improve the tracker's performance based on DCF.

REFERENCES

[1] F. Chen, X. Wang, Y. Zhao, S. Lv, and X. Niu, "Visual object tracking: A survey," *Computer Vision and Image Understanding,* vol. 222, p. 103508, 2022.

[2] J. J. Ye, C. H. Fu, Z. Cao, S. An, G. Z. Zheng, and B. W. Li, "Tracker meets night: A transformer enhancer for UAV tracking," *IEEE Robotics and Automation Letters,* vol. 7, no. 2, pp. 3866-3873, 2022.

[3] Y. Qian, L. Yu, W. Liu, and A. G. Hauptmann, "Electricity: An efficient multi-camera vehicle tracking system for intelligent city," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Seattle, 2020, pp. 588-589.

[4] Y. M. Huang, Y. P. Chen, C. Lin, Q. Hu, and J. H. Song, "Visual attention learning and antiocclusion-based correlation filter for visual object tracking," *Journal of Electronic Imaging,* vol. 32, no. 1, p. 13023, 2023.

[5] J. Wen, H. Chu, Z. Lai, T. Xu, and L. Shen, "Enhanced robust spatial feature selection and correlation filter learning for UAV tracking," *Neural Networks,* vol. 161, pp. 39-54, 2023.

[6] J. M. Zhang, Y. Q. He, and S. G. Wang, "Learning adaptive sparse spatially-regularized correlation filters for visual tracking," *IEEE Signal Processing Letters,* vol. 30, pp. 11-15, 2023.

[7] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Conference on Computer Vision and Pattern Recognition*, San Francisco, 2010, pp. 2544-2550.

[8] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *European Conference on Computer Vision*, Florence, 2012, pp. 702-715.

[9] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 37, no. 3, pp. 583-596, 2015.

[10] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *IEEE International Conference on Computer Vision*, Santiago, 2015, pp. 4310-4318.

[11] M. Mueller, N. Smith, and B. Ghanem, "Context-aware correlation filter tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, 2017, pp. 1396-1404.

[12] D. Sharma and Z. A. Jaffery, "Multiple object tracking through background learning," *Computer Systems Science and Engineering,* vol. 44, no. 1, pp. 191-204, 2023.

[13] H. Kiani Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *IEEE International Conference on Computer Vision*, Venice, 2017, pp. 1135-1143.

[14] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg, "Eco: Efficient convolution operators for tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, 2017, pp. 6638-6646.

[15] T. Xu, Z.-H. Feng, X.-J. Wu, and J. Kittler, "Joint group feature selection and discriminative filter learning for robust visual object tracking," in *IEEE International Conference on Computer Vision*, Seoul, 2019, pp. 7950-7960.

[16] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Long-term correlation tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, Boston, 2015, pp. 5388-5396.

[17] R. Yao, S. Xia, Z. Zhang, and Y. Zhang, "Real-time correlation filter tracking by efficient dense belief propagation with structure preserving," *IEEE Transactions on Multimedia,* vol. 19, no. 4, pp. 772-784, 2017.

[18] P. Li, J. Zhang, Z. Zhu, Y. Li, L. Jiang, and G. Huang, "State-aware re-identification feature for multi-target multi-camera tracking," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Long Beach, 2019, pp. 1506-1516.

[19] S. M. Marvasti-Zadeh, H. Ghanei-Yakhdan, and S. Kasaei, "Efficient scale estimation methods using lightweight deep convolutional neural networks for visual tracking," *Neural Computing and Applications,* vol. 33, pp. 8319-8334, 2021.

[20] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *European Conference on Computer Vision*, Zurich, 2014, pp. 254-265.

[21] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Discriminative scale space tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 39, no. 8, pp. 1561-1575, 2017.

[22] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 2016, pp. 4293-4302.

[23] M. Danelljan, A. Robinson, F. Shahbaz Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *European Conference on Computer Vision*, Amsterdam, 2016, pp. 472-488.

[24] J. Van De Weijer, C. Schmid, J. Verbeek, and D. Larlus, "Learning color names for real-world applications," *IEEE Transactions on Image Processing,* vol. 18, no. 7, pp. 1512-1523, 2009.

[25] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 2018, pp. 8971-8980.

[26] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "Siamrpn++: Evolution of siamese visual tracking with very deep networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, 2019, pp. 4282-4291.

[27] K. Yang, H. Song, K. Zhang, and J. Fan, "Deeper siamese network with multi-level feature fusion for real-time visual tracking," *Electronics Letters,* vol. 55, no. 13, pp. 742-745, 2019.

[28] C. Fan, H. Y. Yu, Y. Huang, C. F. Shan, L. Wang, and C. L. Li, "Siamon: Siamese occlusion-aware network for visual tracking," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 33, no. 1, pp. 186-199, 2023.

[29] W. M. Hu, Q. Wang, L. Zhang, L. Bertinetto, and P. H. S. Torr, "Siammask: A framework for fast online object tracking and segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 45, no. 3, pp. 3072-3089, 2023.

[30] X. Dong, J. Shen, D. Yu, W. Wang, J. Liu, and H. Huang, "Occlusion-aware real-time object tracking," *IEEE Transactions on Multimedia,* vol. 19, no. 4, pp. 763-771, 2017.

[31] B. Hunt, "A matrix theory proof of the discrete convolution theorem," *IEEE Transactions on Audio and Electroacoustics,* vol. 19, no. 4, pp. 285-288, 1971.

[32] A. Iwasaki, "Deriving the variance of the discrete Fourier transform test using Parseval's theorem," *IEEE Transactions on Information Theory,* vol. 66, no. 2, pp. 1164-1170, 2020.

[33] A. Lukezic, T. Vojir, L. ˇCehovin Zajc, J. Matas, and M. Kristan, "Discriminative correlation filter with channel and spatial reliability," in *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, 2017, pp. 6309-6318.

[34] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Adaptive correlation filters with long-term and short-term memory for object tracking," *International Journal of Computer Vision,* vol. 126, pp. 771-796, 2018.

[35] Z. Huang, C. Fu, Y. Li, F. Lin, and P. Lu, "Learning aberrance repressed correlation filters for real-time UAV tracking," in *IEEE International Conference on Computer Vision*, Seoul, 2019, pp. 2891-2900.

[36] Y. Li, C. Fu, F. Ding, Z. Huang, and G. Lu, "Autotrack: Towards high-performance visual tracking for UAV with automatic spatio-temporal regularization," in *IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, 2020, pp. 11923-11932.

Date of modification: July 15, 2024
Brief description of the changes:

1. We have revised the date to July 15th.

2. We have modified Eq. (1) on the right-hand side of the second page from

$$
\begin{aligned}
E(\boldsymbol{h}) &= \min_{\boldsymbol{h}} \frac{1}{2}\left\|\boldsymbol{y}-\boldsymbol{h}\star\boldsymbol{x}\right\|_2^2+\frac{\lambda}{2}\left\|\boldsymbol{h}\right\|_2^2 \\
&= \min_{\boldsymbol{h}} \frac{1}{2}\left\|\boldsymbol{y}-\boldsymbol{C}\left(\boldsymbol{h}^{\mathrm{T}}\right)\boldsymbol{x}\right\|_2^2+\frac{\lambda}{2}\left\|\boldsymbol{h}\right\|_2^2 \\
&= \min_{\boldsymbol{h}} \frac{1}{2}\sum_{t=0}^{T-1}\left(\boldsymbol{y}(t)-\boldsymbol{x}^{\mathrm{T}}\boldsymbol{h}[\Delta\tau_t]\right)^2+\frac{\lambda}{2}\left\|\boldsymbol{h}\right\|_2^2 \\
&= \min_{\boldsymbol{h}} \frac{1}{2}\left\|\overline{\boldsymbol{y}}-\boldsymbol{C}\left(\boldsymbol{x}^{\mathrm{T}}\right)\boldsymbol{h}\right\|_2^2+\frac{\lambda}{2}\left\|\boldsymbol{h}\right\|_2^2 \\
&= \min_{\boldsymbol{h}} \frac{1}{2}\sum_{t=0}^{T-1}\left(\overline{\boldsymbol{y}}(t)-\boldsymbol{h}^{\mathrm{T}}\boldsymbol{x}[\Delta\tau_t]\right)^2+\frac{\lambda}{2}\left\|\boldsymbol{h}\right\|_2^2
\end{aligned}
$$

to

$$
\begin{aligned}
E(\boldsymbol{h}) &= \min_{\boldsymbol{h}} \frac{1}{2}\left\|\boldsymbol{y}-\boldsymbol{x}\star\boldsymbol{h}\right\|_2^2+\frac{\lambda}{2}\left\|\boldsymbol{h}\right\|_2^2 \\
&= \min_{\boldsymbol{h}} \frac{1}{2}\left\|\boldsymbol{y}-\boldsymbol{C}\left(\boldsymbol{h}^{\mathrm{T}}\right)\boldsymbol{x}\right\|_2^2+\frac{\lambda}{2}\left\|\boldsymbol{h}\right\|_2^2 \\
&= \min_{\boldsymbol{h}} \frac{1}{2}\sum_{t=0}^{T-1}\left(\boldsymbol{y}(t)-\boldsymbol{x}^{\mathrm{T}}\boldsymbol{h}[\Delta\tau_t]\right)^2+\frac{\lambda}{2}\left\|\boldsymbol{h}\right\|_2^2 . \\
&= \min_{\boldsymbol{h}} \frac{1}{2}\left\|\overline{\boldsymbol{y}}-\boldsymbol{C}\left(\boldsymbol{x}^{\mathrm{T}}\right)\boldsymbol{h}\right\|_2^2+\frac{\lambda}{2}\left\|\boldsymbol{h}\right\|_2^2 \\
&= \min_{\boldsymbol{h}} \frac{1}{2}\sum_{t=0}^{T-1}\left(\overline{\boldsymbol{y}}(t)-\boldsymbol{h}^{\mathrm{T}}\boldsymbol{x}[\Delta\tau_t]\right)^2+\frac{\lambda}{2}\left\|\boldsymbol{h}\right\|_2^2
\end{aligned}
$$

3. We have modified the $\frac{1}{2}\left\|\boldsymbol{y}-\boldsymbol{h}\star\boldsymbol{x}\right\|_2^2$ to $\frac{1}{2}\left\|\boldsymbol{y}-\boldsymbol{x}\star\boldsymbol{h}\right\|_2^2$ on the right-hand side of the second page.

4. We have modified the $C\left(\boldsymbol{h}^{\mathrm{T}}\right)\boldsymbol{x}=\boldsymbol{h}\star\boldsymbol{x}$ to $C\left(\boldsymbol{h}^{\mathrm{T}}\right)\boldsymbol{x}=\boldsymbol{x}\star\boldsymbol{h}$ on the right-hand side of the second page.

5. We have modified Eq. (6) on the left-hand side of the third page from

$$
\begin{aligned}
E(\boldsymbol{h}_k) &= \min_{\boldsymbol{h}_k} \frac{1}{2}\left\|\boldsymbol{y}-\sum_{k=1}^{K}\boldsymbol{h}_k\star\boldsymbol{x}_k\right\|_2^2+\frac{\lambda}{2}\sum_{k=1}^{K}\left\|\boldsymbol{h}_k\right\|_2^2 \\
&= \min_{\boldsymbol{h}_k} \frac{1}{2}\left\|\boldsymbol{y}-\sum_{k=1}^{K}\overline{\boldsymbol{h}}_k*\boldsymbol{x}_k\right\|_2^2+\frac{\lambda}{2}\sum_{k=1}^{K}\left\|\overline{\boldsymbol{h}}_k\right\|_2^2 \\
&= \min_{\boldsymbol{h}_k} \frac{1}{2}\left\|\boldsymbol{y}-\sum_{k=1}^{K}C\left(\boldsymbol{h}_k^{\mathrm{T}}\right)\boldsymbol{x}_k\right\|_2^2+\frac{\lambda}{2}\sum_{k=1}^{K}\left\|\boldsymbol{h}_k\right\|_2^2 \\
&= \min_{\boldsymbol{h}_k} \frac{1}{2}\sum_{t=0}^{T-1}\left(\boldsymbol{y}(t)-\sum_{k=1}^{K}\boldsymbol{x}_k^{\mathrm{T}}\boldsymbol{h}_k[\Delta\tau_t]\right)^2+\frac{\lambda}{2}\sum_{k=1}^{K}\left\|\boldsymbol{h}_k\right\|_2^2 \\
&= \min_{\boldsymbol{h}_k} \frac{1}{2}\left\|\overline{\boldsymbol{y}}-\sum_{k=1}^{K}C\left(\boldsymbol{x}_k^{\mathrm{T}}\right)\boldsymbol{h}_k\right\|_2^2+\frac{\lambda}{2}\sum_{k=1}^{K}\left\|\boldsymbol{h}_k\right\|_2^2 \\
&= \min_{\boldsymbol{h}_k} \frac{1}{2}\sum_{t=0}^{T-1}\left(\overline{\boldsymbol{y}}(t)-\sum_{k=1}^{K}\boldsymbol{h}_k^{\mathrm{T}}\boldsymbol{x}_k[\Delta\tau_t]\right)^2+\frac{\lambda}{2}\sum_{k=1}^{K}\left\|\boldsymbol{h}_k\right\|_2^2
\end{aligned}
$$

to

$$
\begin{aligned}
E(\boldsymbol{h}_k) &= \min_{\boldsymbol{h}_k} \frac{1}{2}\left\|\boldsymbol{y}-\sum_{k=1}^{K}\boldsymbol{x}_k\star\boldsymbol{h}_k\right\|_2^2+\frac{\lambda}{2}\sum_{k=1}^{K}\left\|\boldsymbol{h}_k\right\|_2^2 \\
&= \min_{\boldsymbol{h}_k} \frac{1}{2}\left\|\boldsymbol{y}-\sum_{k=1}^{K}\overline{\boldsymbol{h}}_k*\boldsymbol{x}_k\right\|_2^2+\frac{\lambda}{2}\sum_{k=1}^{K}\left\|\overline{\boldsymbol{h}}_k\right\|_2^2 \\
&= \min_{\boldsymbol{h}_k} \frac{1}{2}\left\|\boldsymbol{y}-\sum_{k=1}^{K}C\left(\boldsymbol{h}_k^{\mathrm{T}}\right)\boldsymbol{x}_k\right\|_2^2+\frac{\lambda}{2}\sum_{k=1}^{K}\left\|\boldsymbol{h}_k\right\|_2^2 . \\
&= \min_{\boldsymbol{h}_k} \frac{1}{2}\sum_{t=0}^{T-1}\left(\boldsymbol{y}(t)-\sum_{k=1}^{K}\boldsymbol{x}_k^{\mathrm{T}}\boldsymbol{h}_k[\Delta\tau_t]\right)^2+\frac{\lambda}{2}\sum_{k=1}^{K}\left\|\boldsymbol{h}_k\right\|_2^2 \\
&= \min_{\boldsymbol{h}_k} \frac{1}{2}\left\|\overline{\boldsymbol{y}}-\sum_{k=1}^{K}C\left(\boldsymbol{x}_k^{\mathrm{T}}\right)\boldsymbol{h}_k\right\|_2^2+\frac{\lambda}{2}\sum_{k=1}^{K}\left\|\boldsymbol{h}_k\right\|_2^2 \\
&= \min_{\boldsymbol{h}_k} \frac{1}{2}\sum_{t=0}^{T-1}\left(\overline{\boldsymbol{y}}(t)-\sum_{k=1}^{K}\boldsymbol{h}_k^{\mathrm{T}}\boldsymbol{x}_k[\Delta\tau_t]\right)^2+\frac{\lambda}{2}\sum_{k=1}^{K}\left\|\boldsymbol{h}_k\right\|_2^2
\end{aligned}
$$