Real-time long-term tracking with reliability assessment and object recovery

Jun Liu1,2 | Ping Ye1 | Xingchen Zhang1✉ | Gang Xiao1

1 School of Aeronautics and Astronautics, Shanghai Jiao Tong University, China
2 School of Automation and Information Engineering, Sichuan University of Science and Engineering, China

Correspondence
Xingchen Zhang, School of Aeronautics and Astronautics, Shanghai Jiao Tong University, Shanghai 200240, China.
Email: xingchenzhang1990@gmail.com

Funding information
National Basic Research Program of China (973 Program), Grant/Award Number: 2014CB744903; National Natural Science Foundation of China, Grant/Award Numbers: 61973212, 61673270

Abstract
In recent years, many visual tracking algorithms based on discriminative correlation filters have been proposed and proved to be successful in short-term tracking. However, most algorithms do not handle long-term tracking well due to factors such as occlusion and deformation. The aim of this paper is to propose a long-term tracking method with reliability assessment and object recovery. First, the relationship between the overlap rate and the response value is extensively studied, and then from the perspective of the time axis, the tracking process is evaluated by the fluctuation trend of the continuous response value. In the object recovery mechanism, we propose to alternately use local search and global search to improve the efficiency of detection. To this end, a sliding window is designed for cyclic shifting in the local region to achieve dense sampling within the region of interest, and EdgeBox is used in the global search to achieve target detection. Further, flexible switching between local search and global search is achieved by the difference in displacement of the object. Extensive experimental results on several benchmark datasets demonstrate that the proposed long-term tracker can achieve state-of-the-art accuracy with real-time speed of about 27 frames per second.

1 | INTRODUCTION

Visual object tracking has been applied to many fields, such as human-computer interaction, security and surveillance, and auto-control systems. Existing object tracking algorithms have made great achievements, but due to factors such as illumination variation, occlusion, deformation, scale variation and background clutter, there are still many problems to be further investigated.

According to the appearance model, visual object tracking approaches can be broadly classified into two categories, namely generative and discriminative methods. The generative methods [1–9] transform the tracking problem as a nearest-neighbor searching task for the target model. The insufficiency of the generative method is that the background information of the target is not considered, so the image information cannot be applied well. The discriminative methods [10–19] formulate the target tracking as a binary classification problem. The extracted target and background information are used to train the classifier and then separate the target from the background. Since the background information is additionally utilized, the discriminative methods will be more advantageous than the generative methods in practical applications.

Recently, correlation filter based trackers have sparked a lot of interests due to their real-time speed and state-of-the-art accuracy [11, 17, 18, 20, 21]. In general, these trackers learn a correlation filter online to localize the object in consecutive frames. However, the correlation filter based tracker still faces two serious problems. One major issue is boundary effects due to the circulant assumption. The other is that the target search region only contains a small local neighbourhood to limit drift. Therefore, the trackers easily drift in cases of fast motion, occlusion or background clutter. It is important to note that occlusion can have a serious adverse effect on the performance of trackers. When occlusion occurs, the similarity of desired target region degrades and cluttered background distracts the template to match with other mistaken regions. To make it worse, the template is updated with the false information, which further
brings unsatisfactory templates to the subsequent frames. Therefore, how to overcome the performance degradation caused by occlusion is a key concern when designing a long-term tracker.

The aim of this paper is to develop a long-term tracker based on correlation filters to alleviate the problem of tracking failure caused by factors such as deformation and occlusion. The main difficulties in the overall framework of the tracker are reflected in the process evaluation and object recovery. To this end, we designed a criterion to achieve an accurate assessment of the reliability of the tracking process, and also established a complete object recovery mechanism. The main contributions of this paper are as follows:

- Different from the methods using peak to sidelobe ratio (PSR) [20], median flow (MF) [59] and average peak-to-correlation energy (APCE) [49], we regard the reliability of the tracking results as correlated back and forth on the time axis, and design a new tracking result evaluation criterion based on continuous response values.
- We propose a new object recovery mechanism in which the sliding window performs a local search and the Edgebox [61] performs a global search. The recovery mechanism can adaptively select an appropriate search mode to find the missing object based on the average displacement of the object.
- The proposed algorithm is validated on several widely used large-scale benchmark datasets [22, 24–26] and the experimental results show that our long-term tracker is able to achieve state-of-the-art accuracy and real-time speed.

The remaining of this paper is organized as follows. Previous related works are reviewed briefly in Section 2. Section 3 presents our approach for robust object tracking which includes short-term trackers based on correlation filtering, reliability assessment of tracking process, object recovery mechanism and model update strategy. Section 4 summarizes the overall workflow of the algorithm and illustrates the performance of the proposed tracker. Finally, the conclusions and future research directions are discussed in Section 5.

2 | RELATED WORK

2.1 | Tracking-by-detection methods

Detection-based tracking framework has attracted widespread attention due to the fact that classifiers are suitable for online learning, such as support vector machines (SVM) [27], Random Forest classifiers [28] or boosting variants [29, 30], and can be used to achieve real-time tracking. For each frame, a set of positive and negative training samples are collected for incrementally learning a discriminative classifier to separate a target from its background. However, extracting samples to learn online classifiers can cause sample ambiguity problems, while slight inaccuracies in labelling samples can affect the classifier and gradually lead to tracker drift. In order to alleviate the problem of model updating caused by sample ambiguity, considerable efforts have been made in previous studies, such as multiple instance learning (MIL) [29], Struck [31], semi-supervised learning [30, 32], and P-N learning [33]. In addition, a robust appearance model containing both holistic templates and local representations was proposed in [34] to handle drastic appearance changes. Similarly, a learning framework in [35] with 1-class SVM and structured output SVM was designed for collaborating a descriptive component and a discriminative component for tracking. Instead of learning only one single classifier, multiple classifiers with different learning rates were constructed in [36] to address the model drift problem. By alleviating the problem of sample ambiguity, above methods have achieved considerable performance on the recent benchmark dataset [22]. Wang et al. [23] proposed a three-step procedure to convert a general object detector into a tracker. They considered the tracking problem as a special type of object detection problem, which called instance detection. With proper initialization, a detector can be quickly converted into a tracker by learning the new instance from a single image.

2.2 | Correlation filter-based methods

High computational efficiency and state-of-the-art accuracy are two significant advantages of the correlation filter based tracker. By transferring the model into the Fourier domain, the matrix algebra can be solved by element-wise operation. In [20], Bolme et al. proposed the minimum output sum of squared error (MOSSSE) method and learned a minimum output sum of squared error filter on gray-scale images, which used intensity features to represent the object. Heriques et al. [17] proposed the circulant structure of tracking by detection with kernel (CSK) algorithm which generated correlation filters by exploiting the circulant structure of training samples and transferred filters into the Fourier domain. In their work, the CSK achieved a highest tracking speed with considerable performance on benchmark dataset. As an extension to CSK, the histogram of oriented gradients (HOG) features, Gaussian kernels, and ridge regression were used in the kernelized correlation filter (KCF) [18]. The discriminative scale space correlation filter tracker (DSST) [13] learned discriminative correlation filters with a scale pyramid representation to handle the scale change of target objects. They learned two kinds of filters, one for translation estimation and the other for scale estimation. Besides, the scale adaptive with multiple features tracker (SAMF) [14] estimated the scale variation by training the KCF to search the target around the latest estimate position on various sizes of the image patch. Huang et al. [37] proposed the aberrance repressed correlation filter. By introducing a regularization term to restrict the response map, their method was capable of suppressing aberrances that was caused by both background noise information and appearance changes of the tracked objects.

Hand-crafted features have been widely used in correlated filter based tracking, and deep features have recently begun to be combined with correlated filtering to exploit their enormous potential. Following the end-to-end philosophy, the recent work [38] integrated DCF into the Siamese framework [39], and [40] employed DCF as a one-layer convolutional neural network (CNN) for end-to-end training. Some other DCF methods [41–43] focused on integrating convolutional features from
a fixed pre-trained deep network. Ma et al. [41] proposed a hierarchical ensemble method of independent DCF trackers to combine multiple convolutional layers. Danelljan et al. [42] proposed the continuous convolution operator tracker (C-COT) to efficiently integrate multi-resolution shallow and deep feature maps. The efficient convolution operators (ECO) tracker [43] further optimized the speed and accuracy of the C-COT tracker at three levels: model size, training set size, and model update. Due to the high complexity of CNN features, trackers with deep-features are not superior in speed to traditional handcrafted feature trackers. Sun et al. [44] proposed the region-of-interest (ROI) pooled correlation filters for visual tracking. They showed that the ROI-based pooling can be equivalently achieved by enforcing additional constraints on the learned filter weights, which made the ROI-based pooling feasible on the virtual circular samples.

2.3 Long-term tracking methods

Correlation filters with short-term memory have significant limitations in dealing with object occlusion and disappearance. Therefore, many methods aim to make correlation filters with long-term memory to achieve more advanced tracking. The combination of a short-term tracker and detector is a commonly used long-term tracker architecture which was first used in tracking-learning-detection (TLD) [33]. The seminal work of [33] proposes a memory-less flock of flows as a short-term tracker and a template-based detector. Another paradigm was pioneered in [45], where the authors casted localization as local keypoint descriptors matching with a weak geometrical model. Ma et al. [46, 47] proposed using KCF as a short-term tracker and random fern classifier as a detector to construct a long-term tracker. Similarly, Hong et al. [48] combined a KCF tracker with a SIFT-based detector which was also used to detect occlusions. Dai et al. [49] established a simplified long-term tracker by employing assistant re-detection, which combined the short-term components with the SVM classifier in constructing a long-term component. In [4], the target model was decomposed into a grid of cells and an occlusion classifier was learned for each cell. In [50], a Bayes filter was developed for target loss detection and re-detection for multi-target tracking. Alan et al. [51] utilized the novel DCF constrained filter learning method to design a detector that was able to re-detect the target in the whole image efficiently. Moreover, they proposed to use the correlation response value and PSR to evaluate tracking failure. Sun et al. [52] treated the filter as the element-wise product of a base filter and a reliability term to mitigate model degradation. Dai et al. [53] proposed an adaptive spatial constraint correlation filtering algorithm to optimize both filter weights and spatial constraint matrices. In addition, Goutam et al. [54] leveraged the complementary properties of deep and shallow features to improve both robustness and accuracy. Their work has played a significant role in unlocking the potential of deep features for tracking. Recently, some methods have tried to combine deep learning-based proposal networks with correlation filter tracking. For example, Yang et al. [55] proposed a novel long-term correlation filtering tracking method that applied a DCF-based tracking model with a novel target-aware detector in a collaborative way. In their method, they evaluated the reliability of the tracking results based on the enhanced PSR, and used region proposal network (RPN) based detectors to perform object detection under tracking failure. Yan et al. [56] proposed a novel robust and real-time long-term tracking framework based on the proposed skimming and perusal modules. The perusal module aims to precisely locate the tracked object in a local search region using the offline-trained regression and verification networks. The skimming module focuses on efficiently selecting the most possible regions from densely sampled sliding windows.

3 OUR APPROACH

3.1 The basic correlation filter tracking component

Let $x$ be the base sample of size M×N, and the training sample $x_i$ can be obtained after cyclic shift. The decomposition of circulant matrix can be employed to simplify the solution of linear regression. The objective function of linear ridge regression can be formulated as follows:

$$
\min_w \sum_i (f(x_i) - y_i)^2 + \lambda ||w||^2
$$

where $f(x) = w^T x$ is used to minimize the squared error over samples $x_i$ and their regression targets $y_i$. In the case of non-linear regression, kernel trick, $f(z) = w^T z = \sum_{i=1}^m \alpha_i x_i(z, x_i)$, is applied to allow more powerful classifier. For the most commonly used kernel functions, the circulant matrix trick can also be used [18]. The dual space coefficients $\alpha$ can be learnt as below:

$$
\tilde{\alpha}^* = \frac{\tilde{\gamma}}{k^x + \lambda}
$$

where $k^x$ is defined as kernel correlation in [18]. In this paper, we adopt the Gaussian kernel which can be applied with the circulant matrix trick as below:

$$
k_{xx'} = \exp\left(-\frac{1}{\delta^2} \left(||x||^2 + ||x'||^2 - 2F^{-1}(\tilde{x} \odot \tilde{x}')\right)\right)
$$

where $F^{-1}$ denotes the inverse FFT transform. The patch $z$ at the same location in the next frame is treated as the base sample to compute the response in Fourier domain,

$$
f(z) = (k_{xx})^* \odot \tilde{\alpha}
$$

where $\tilde{\alpha}$ denotes the data to be learnt in the model. When we transform $f(z)$ back into the spatial domain, the translation with respect to the maximum response is considered as the movement of the tracked target.
For the scale variation of the objects, we apply the method of the scale pool [14]. The template size is fixed as \( s_T = (s_x, s_y) \), and the scaling pool is defined as \( S = \{ t_1, t_2, ..., t_k \} \). Suppose that the target window size is \( s_t \) in the original image space. The samples can be resized into the fixed template size \( s_T \) by using bilinear-interpolation, and the final response is calculated by

\[
\arg \max_{z^T} F^{-1}(z^T)
\]

where \( z^T \) is the sample patch with the size of \( s_T \), which is resized to \( s_T \). To adapt to changes in the appearance of objects, we linearly combine the new filter with the old one as below:

\[
\tilde{T} = \theta T_{aw} + (1 - \theta) T
\]

where \( T = [x^T, \tilde{x}^T]^T \) is the template to be updated. It is worth noting that \( T \) is very useful for object recovery because it contains the most recent appearance information of the object.

### 3.2 Reliability assessment of tracking process

Accurate assessment of the tracking process is an effective guarantee for the intervention of object recovery mechanism. Since the correlation filtering tracking locates the target according to the response map, the intensity of the response can be used to characterize the accuracy of target positioning. Figure 1 shows the fluctuation trend of the response value during object tracking. To explore a good tracking process evaluation method, we use the groundtruth given by the dataset to investigate the distribution of response values at different overlap rates. The distribution of response values in two presentation cases is shown in Figure 1, where the response value is less than \( \xi \cdot d \cdot \tau \). The distribution of response values at different overlap rates is defined as

\[
O(B_1, B_0) = \frac{|B_1 \cap B_0|}{|B_1 \cup B_0|}
\]

where \( B_1 \) and \( B_0 \) indicates the predicted bounding box and groundtruth, respectively. Since the tracker predicts the new position based on the response map, we can obtain not only the predicted bounding box \( B_1 \) but also the response value corresponding to \( B_1 \). For this reason, we can get the distribution of response values at different overlap rates. In the one-pass evaluation [22, 24], for the tracking result of each frame, when \( O(B_1, B_0) \geq 0.5 \), it is considered that the tracker works well. According to the visual sense, when \( O(B_1, B_0) = 0.5 \), the bounding box still falls on the tracked object. However, as the overlap rate decreases, the bounding box will gradually deviates from the object. When the overlap rate is about 0.4, the tracker almost drifts, and when the overlap rate drops to about 0.2, the tracker completely loses the object. Therefore, we can investigate the relationship between the response values and the reliability of the tracking results by examining the distribution of response values at different overlap rates.

### 3.3 Object recovery mechanism

#### 3.3.1 Template pool

During the tracking process, the appearance of the object changes over time. An important issue in long-term tracking is how to make the tracker have the ability to recover the object successfully in the case of tracking failure caused by significant changes in the object appearance. However, using the...
The schematic diagram of the local search strategy

![Figure 2](https://example.com/figure2.png)

target sample of the initial frame as a fixed template is not well suited for long-term tracking because the fixed template cannot be adapted to the appearance changes of the target during the tracking process [39]. In [60], the short-term classifier-pool uses the last $M$ samples as templates and partially occluded objects are also used as samples, so the classifier-pool is to some extent contaminated by samples containing error information. Different from the methods in [39] and [60], the availability of each template will be examined by its corresponding response value when building a template pool in this work. Assuming that the object recovery mechanism intervenes at the $K$th frame, the template pool $\Theta$ is constructed as a collection of the last 15 frame templates.

$$\Theta = \{T_{K-14}, T_{K-13}, \ldots, T_{K-2}, T_{K-1}, T_{K}\}$$

where $T_K$ is the template of the target at the $K$th frame. According to Equations (4) and (5), the template $T_i$ with the largest response value in $\Theta$ can be determined. The template $T_i$ will be used to recover objects from tracking failures.

### 3.3.2 Local search strategy

The boundary effect is a significant flaw in correlation filter tracking, and the search area is usually set to 1.5 times the size of the object. When the object part moves out of the search area, the positioning of the target will be inaccurate. In particular, fast-moving objects tend to move out of the search area easily, which eventually leads to tracking failure. Therefore, a search area of 1.5 times the object size is usually insufficient. For this reason, SRDCF [21] limits the boundary effect and expands the search area to 4 times the size of the object, but greatly sacrifices the running speed. Generally, when the object appears again after losing, it is usually located in the surrounding area of the losing position. Therefore, the local search method can detect the object in the surrounding area of interest directly with high efficiency. In order to improve the tracking speed and accuracy, we propose a local search strategy which is activated after Equations (7) and (8) are reached. When tracking fails, the tracker will record the position of the object at the current frame, and then the detector searches the target in the local region of the position. To this end, we designed a sliding window to perform a cyclic shift in the local region for dense sampling while extracting image features to generate a sample corresponding to each window. This is equivalent to dividing the expanded search area into many blocks and performing correlation measurement on each block separately, which can effectively reduce the influence of boundary effects. The specific search strategy is shown in Figure 2. The red dot is used to represent the center coordinate of the object’s bounding box. When the criterion is triggered, the tracker will record the coordinates of the red dot. The red box indicates the search region centered on the red dot, where $W_{\text{search}}$ is the width and $H_{\text{search}}$ is the height. The green box with a green dot is a sliding window that will perform a cyclic shift within the search region.

Let $W_K$ and $H_K$ denote the width and height of the bounding box at $K$ frame, respectively. In our algorithm, we use $W_{\text{search}}$ and $H_{\text{search}}$ to represent the search region, i.e. $W_{\text{search}} = \varphi \cdot W_K$ and $H_{\text{search}} = \varphi \cdot H_K$. Therefore, the search region $S_{\text{search}}$ can be expressed as

$$S_{\text{search}} = W_{\text{search}} \cdot H_{\text{search}} = \varphi^2 \cdot (W_K \cdot H_K)$$

where $\varphi$ is a positive integer greater than 1. The larger the values of $\varphi$, the larger the search region $S_{\text{search}}$. Then, the sliding step size of the sliding window in the direction of the $x$ coordinate axis and the $y$ coordinate axis can be written as follows

$$\Delta_x = W_{\text{search}} / \gamma$$

$$\Delta_y = H_{\text{search}} / \gamma$$

where $\gamma$ is the number of sample points along the $x$ and $y$ axes. This means that a total of $\gamma^2$ samples are generated throughout the local search region.

### 3.3.3 Global search strategy

In Section 4, extensive experiments demonstrate the effectiveness of the proposed local search strategy. However, local search
One of the most straightforward ways to alleviate this flaw is to increase $\phi$ and $\gamma$, but this can cause serious degradation to the real-time performance of the tracker. This is mainly because the number of samples increases quadratically with an increase in $\gamma$, and such a computational burden is unaffordable for the tracker. Considering that some computationally efficient target detection algorithms have been applied to the object tracking field, we use the EdgeBox [61] to perform global search on images. In simple terms, the core idea of EdgeBox is to calculate a score for each sliding window that represents the likelihood that the area corresponding to the sliding window contains objects. If the score is large, the area is likely to contain objects. On the contrary, if the score is small, the possibility that the area contains objects is small. EdgeBox can quickly generate object bounding box proposals directly from the edges and it typically takes only about 0.35s to generate 1000 proposals in a 480 $\times$ 720 image. Given the advantages of EdgeBox, using this method to perform global search on images does not have a significant adverse effect on tracking speed. More details about EdgeBox can be found in the literature [61].

As can be seen from Equations (11), (12) and (13), we will get $\gamma^2$ and $N'$ samples respectively in the local search and the global search. Then, these samples and $T_i$ are substituted into

$$\rho^{-1} \cdot \frac{b_w}{\rho \cdot b_h} < b_w \cdot b_h \cdot \frac{b_w}{\rho} \cdot \frac{b_h}{\rho}$$

(13)

where $b_w$ and $b_h$ are the width and height of the proposals, respectively. The $\frac{b_w}{\rho}$ and $\frac{b_h}{\rho}$ are the width and height of the object bounding box at $K^{th}$ frame, $\rho$ is a constraint coefficient.

Since the value of $\rho$ determines the number of candidate samples, choosing an appropriate $\rho$ can improve the running speed without deteriorating the tracking accuracy. We show a case in Figure 3, where the number of candidate samples $N'$ corresponding to different $\rho$ is marked in each subgraph. It can be clearly seen from Figure 3 that when the value of $\rho$ gradually decreases, the number of candidate samples $N'$ will significantly decrease. This means that appropriately reducing the value of $\rho$ can effectively reduce the computational burden in the object recovery mechanism. In order to better balance the running speed and tracking accuracy, we set the limit factor $\rho$ to 1.2. Of course, there are advantages and disadvantages to doing this, but in most cases the gains are acceptable.

### 3.3.4 Target determination

As can be seen from Equations (11), (12) and (13), we will get $\gamma^2$ and $N'$ samples respectively in the local search and the global search. Then, these samples and $T_i$ are substituted into
TABLE 1 The accuracy of the tracker at different threshold coefficients $\varepsilon$

| $\varepsilon$ | 0.0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|--------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|
| Biker        | 0.7500 | 0.7500 | 0.7611 | 0.7111 | 0.4500 | 0.4500 | 0.4500 | 0.4500 | 0.4500 | 0.4500 | 0.4500 |
| Bird         | 0.5152 | 0.5152 | 0.5152 | 0.5152 | 0.4500 | 0.4500 | 0.4500 | 0.4500 | 0.4500 | 0.4500 | 0.4500 |
| Bird1        | 0.3799 | 0.3309 | 0.3309 | 0.3309 | 0.3627 | 0.5025 | 0.3211 | 0.3211 | 0.3211 | 0.3211 | 0.3211 |
| Bluerwal     | 0.8558 | 0.8859 | 0.8859 | 0.8843 | 0.8843 | 0.8843 | 0.8590 | 0.8304 | 0.3978 | 0.2298 | 0.2298 |
| Boc          | 0.8020 | 0.8470 | 0.8490 | 0.8580 | 0.8230 | 0.4410 | 0.4410 | 0.4410 | 0.4410 | 0.4410 | 0.4410 |
| Busstation   | 0.1047 | 0.1047 | 0.1047 | 0.2534 | 0.2534 | 0.1047 | 0.1047 | 0.1047 | 0.1047 | 0.1047 | 0.1047 |
| Carehase1    | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.5426 | 0.5426 | 0.5426 | 0.5426 | 0.5426 | 0.5426 |
| Carehase2    | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.5426 | 0.5426 | 0.5426 | 0.5426 | 0.5426 |
| Cake         | 0.8076 | 0.8591 | 0.8729 | 0.8935 | 0.8935 | 0.8935 | 0.8935 | 0.8935 | 0.8935 | 0.8935 | 0.8935 |
| Couple       | 0.6429 | 0.7643 | 0.7357 | 0.7857 | 0.6000 | 0.5929 | 0.5357 | 0.3286 | 0.3286 | 0.3286 | 0.3286 |
| Girl2        | 0.0720 | 0.9187 | 0.9187 | 0.9027 | 0.9293 | 0.0853 | 0.0707 | 0.0707 | 0.0707 | 0.0707 | 0.0707 |
| Group1       | 0.1528 | 0.9583 | 0.9583 | 0.9583 | 0.9861 | 0.9861 | 0.9583 | 0.9444 | 0.9167 | 0.1667 | 0.1667 |
| Human5       | 0.9705 | 0.9705 | 0.9705 | 0.9705 | 0.9705 | 0.9705 | 0.2454 | 0.2454 | 0.2454 | 0.2454 | 0.2454 |
| Jogging      | 0.9642 | 0.9935 | 0.9935 | 0.9935 | 0.9707 | 0.9511 | 0.1824 | 0.1824 | 0.1824 | 0.1824 | 0.1824 |
| Jumping      | 0.9361 | 0.9361 | 0.9361 | 0.9480 | 0.9480 | 0.9480 | 0.9480 | 0.9480 | 0.9480 | 0.9480 | 0.9480 |
| Lemmating    | 0.2485 | 0.6594 | 0.6287 | 0.8226 | 0.5981 | 0.4484 | 0.4484 | 0.2515 | 0.2515 | 0.2515 | 0.2515 |
| Messi        | 0.3824 | 0.3824 | 0.3824 | 0.3860 | 0.4191 | 0.2610 | 0.2610 | 0.2610 | 0.2610 | 0.2610 | 0.2610 |
| Motorbike    | 0.9509 | 0.9509 | 0.9509 | 0.9509 | 0.9509 | 0.2380 | 0.2380 | 0.2380 | 0.2380 | 0.2380 | 0.2380 |
| Motorcycle   | 0.9929 | 0.9929 | 0.9929 | 0.9929 | 0.9929 | 0.9929 | 0.2199 | 0.2199 | 0.2199 | 0.2199 | 0.2199 |
| Panda        | 0.5970 | 0.8710 | 0.9450 | 0.9500 | 0.9720 | 0.5850 | 0.5850 | 0.5850 | 0.5850 | 0.5850 | 0.5850 |
| Read         | 0.4828 | 0.4828 | 0.4828 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| Shaking      | 0.9068 | 0.9068 | 0.9068 | 0.9068 | 0.8966 | 0.0110 | 0.0110 | 0.0110 | 0.0110 | 0.0110 | 0.0110 |
| Topplane     | 0.4173 | 0.4173 | 0.4691 | 0.4296 | 0.0617 | 0.0617 | 0.0617 | 0.0617 | 0.0617 | 0.0617 | 0.0617 |
| Wandering    | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.9913 | 0.3450 | 0.3450 | 0.3450 | 0.3450 | 0.3450 |

Equations (3) and (4) to obtain a response matrix. For local search, the response matrix is represented as $L$, which contains $\gamma^2$ sub-matrices. For global search, the response matrix is written as $G$, which contains $N'N'$ sub-matrices. Generally speaking, the object corresponding to the sample with the maximum response is the most likely target lost by the tracker at $K$th frame. Therefore, it is only necessary to find the maximum response in the $L$ and $G$ respectively. However, the response of the similar object is relatively large, which may lead to the detector recovering to a wrong target. To alleviate this dilemma, we set a threshold $\varepsilon$ on the time domain to constrain the maximum response value. In Table 1, experiments were conducted on 24 video sequences [22, 24, 25] to investigate the effect of $\varepsilon$ on the object recovery mechanism. In order to observe the accuracy performance under different $\varepsilon$ more intuitively, we averaged the accuracy corresponding to the 24 sequences under each $\varepsilon$, and the average accuracy results under different $\varepsilon$ are shown in Figure 4.

From the results in Figure 4, the average accuracy increases monotonically when $\varepsilon$ takes a value of 0 to 0.3, but monotonically decreases when $\varepsilon$ takes 0.3 to 1. In general, the object recovery mechanism has optimal performance when $\varepsilon$ is 0.3, and the average accuracy is improved by 0.1485 compared to the unconstrained case where $\varepsilon$ is 0. Therefore, it is necessary to set a threshold to reduce the probability of false detection.

3.4 Model update

During tracking, the appearance of the object changes significantly due to factors such as rotation and deformation. Therefore, the target template should be updated appropriately during
The whole flowchart of our tracker is shown in Figure 5. The tracker is mainly composed of five parts, including short-term correlation filter tracking module, reliability evaluation module, template pool module, object recovery module and model update module. After the initial object is confirmed, the short-term tracker is used to perform basic tracking of the object, and at the same time establish a template pool to maintain the appearance model of the object. Then, the reliability evaluation criterion is used to discriminate the tracking result. If the tracking result is unreliable, the object recovery mechanism is activated to recover the lost object from subsequent images. Otherwise, the short-term tracker continues to perform basic tracking. Finally, the template is appropriately updated to adapt to changes in the appearance of objects during tracking.

4 | EXPERIMENTAL RESULTS

4.1 | Implementation details

We validated the proposed algorithm on the OTB [22, 24], Temple-Color-128 (TC-128) [23] and UAV [26] datasets. Among them, OTB contains two subsets of OTB-2013 and OTB-2015, which are mainly used for short-term tracking. The UAV contains two subsets of UAV123 and UAV20L, of which UAV20L is specially for long-term tracking. In addition, video sequences in OTB and UAV are annotated with different attributes. Under the one-pass evaluation (OPE) protocols, we use the distance precision and overlap success rates to evaluate our tracker. Considering that HoG feature can effectively capture local shape information and CN feature can accurately describe the surface properties of objects, we use HoG feature and CN feature to perform object tracking in experiments. In our tracker, $S = \{0.985, 0.99, 0.995, 1, 1.005, 1.01, 1.015\}$, the padding is set to 1.5. Note that when running the UAV benchmark datasets, the padding is set to 2.5, because most of the video sequences in the UAV are aerial scenes and the target is small. The benefit of this operation is that it can alleviate
boundary effects, especially for small targets. The remaining parameters use the default values in KCF. The experimental environment is Intel Core i5 2.3 GHz CPU with 8.00G RAM, MATLAB 2017b.

### 4.2 Hyper parameters analysis of criterion

We conducted extensive experiments on three datasets, OTB [24], Temple-Color [25] and UAV [26], to get general conclusions in our tracking method. In order to more fully display the distribution of the response values, we mark the target by the ratio of the target size to the image size and then use 0.005 as an interval to count the response values of all the targets in each interval and average them. The distribution of response values is shown in Figure 6. According to the observations in Figure 6, we found that when the response value falls to the range of 30% to 40% of the initial frame, the $O(B_1, B_0)$ will be around 0.4, and when the response value further drops to 20% to 30% of the initial frame, the $O(B_1, B_0)$ will be around 0.2. Therefore, taking the values of $d$ and $\xi$ as 0.35 and 0.7 respectively can make the criterion have a better accuracy. To further verify the effectiveness of the proposed method, we also tested the reliability evaluation criterion in twelve typical scenarios. It can be seen intuitively from Figure 7 that when the tracker fails or drifts, the response value of the object drops significantly compared to the initial value $\tau$. Besides, the fluctuation of the response value shown in Figure 7 meets the trigger conditions of Equations (7) and (8). The test results demonstrate that the proposed criterion can accurately determine whether the tracker fails in different scenarios. Since our method uses the primitive correlation filter and does not adjust the response map, the response values are distributed between 0 and 1. As long as the object is disturbed by occlusion, deformation and other factors, the response value will drop significantly, which means that the proposed criterion is suitable for all datasets under the primitive correlation filtering tracking framework.

### 4.3 Ablation study

To investigate the contribution of different components to the tracker, we modified the proposed tracker to form multiple variants. There are five main components to consider, including continuous response value criterion (CRV), peak to sidelobe ratio criterion (PSR), median flow criterion (MF), local search strategy (LS) and global search strategy (GS). It should be noted that the baseline tracker only refers to the basic architecture of SAMF tracker. We manually combine a basic correlation filter with a scale pool to form a baseline tracker instead of directly using the original SAMF tracking code. The LGS represents that the baseline tracker only refers to the basic architecture of SAMF tracker. We conducted extensive experiments on three datasets, OTB [24], Temple-Color [25] and UAV [26], to get general conclusions in our tracking method.
FIGURE 8  Ablation results of our tracker and other variants on OTB-2015. Among them, the sub-figures of the first row mainly verify the effectiveness of the search modes, while the sub-figures in the second row examine the accuracy of the reliability evaluation criteria.

The tracking results of our tracker and the other nine variants on the OTB2015 dataset are shown in Figure 8. As can be seen from the first row of Figure 8, the LGS search mode is more effective than LS and GS under different reliability evaluation criteria. This can be attributed to the way LGS fully considers the motion information of the target, so the proposed tracker can flexibly adopt the appropriate search method in different scenarios. It can also be seen from the first row of Figure 8 that the effectiveness of LS is better than that of GS. This is mainly because LS only performs object search on the area of interest, while GS performs object search on the entire area. Therefore, compared with the GS search mode, LS can more effectively locate objects and avoid the interference of similar objects in other regions. For example, as shown in Figure 3, the detector detects multiple similar faces in GS mode, which will increase the probability of the detector recovering to a wrong object. From the comparison results in the second row of Figure 8, it is obvious that the performance of CRV is better than PSR and MF in different search modes. This is because the PSR evaluates the response map for a single frame and the MF considers the displacement of pixels within the bounding box in two adjacent frames. It is not comprehensive to assess the reliability of the tracking process only by the metrics obtained in a single frame or two adjacent frames. Since CRV considers the reliability of tracking results from a continuous time perspective, it is more comprehensive and effective than PSR and MF.

In terms of overall performance, the proposed tracker is significantly higher than the baseline tracker by 10.43% and 6.85%, respectively, in precision rate and success rate. It is worth noting that compared to the baseline tracker, our proposed tracker did not show significant speed degradation, with a drop of only 3.8 frames per second. This is because in the proposed method, the tracker can adaptively select the appropriate detection mode and the number of proposals is set to 200, so the time required for object detection is further reduced. Besides, the short-term correlation filter tracker has good real-time performance and the main time-consuming part is the recovery of the object. Therefore, when the object recovery mechanism is not enabled, the proposed tracker can be regarded as a baseline tracker. In general, when tracking objects, the phenomenon of tracking failure does not occur frequently, which means that the object recovery mechanism will not intervene frequently, so the real-time performance of the tracker will only be slightly affected. For these reasons, although the overall running speed of the tracker will deteriorate, the reduction is not significant.

Through the results of ablation experiments, we can see that each component contributes to improve the performance of the tracker in different degrees. In terms of the strength of the contribution, the proposed components are superior to others in enhancing the tracking performance, thus demonstrating the effectiveness of the proposed method.

### TABLE 2  Details of the nine variants of our tracker

| Trackers  | CRV | PSR | MF | LS | GS |
|-----------|-----|-----|----|----|----|
| CRV-LGS   | √   | ×   | ×  | √  | √  |
| CRV-LS    | √   | ×   | ×  | √  |    |
| CRV-GS    | √   | ×   |   |    | √  |
| PSR-LGS   | ×   | √   | ×  | √  | √  |
| PSR-LS    | ×   | √   |   |    | √  |
| PSR-GS    | ×   | ×   | √  |    | √  |
| MF-LGS    | ×   | ×   | √  | √  | √  |
| MF-LS     | ×   | ×   | √  |   |    |
| MF-GS     | ×   | ×   | √  |    |    |
| Baseline  | ×   | ×   |   |    |    |

4.4 Comparison with state-of-the-arts

We use 16 state-of-the-art trackers to make a comprehensive comparison with our approach on OTB dataset, including SANet [57], MCPF [58], SiamFC [39], ECO [43], ACFN [62], BACF [10], LMCF [63], SRDCF [21], MEEM [36], DLSSVM [64], DCFNet [65], Staple [66], SAMF [14], LCT [46], DSST [13]
and KCF [18] by using OPE. Among them, SANet incorporates recurrent neural network (RNN) into CNN, MCPF combines particle filters and correlation filters, DLSVM is based on the structured support vector machine method, ECO, BACF, LMCF, SDRCF, DSST, LCT, SAMF, Staple and KCF are based on the correlation filter. MEEM is developed based on regression and multiple trackers. DCFNet integrates the Siamese network and ACFN uses the attentional correlation filter network. The OPE performance comparison on OTB-2013 and OTB-2015 is shown in the first and second rows of Figure 9. It can be seen from the comparison results that even though the precision rate and success rate of our tracker are not as good as SANet, MCPF and ECO trackers, the proposed tracker still achieves considerable performance. In particular, the running speed of SANet and MCPF is only about 1 frame per second, which is far from meeting the real-time tracking requirements. Since our tracker only uses hand-crafted features, the performance of the proposed tracker still has great potential for improvement, such as the use of CNN features in our tracking framework. With the aid of reliability assessment and object recovery mechanism, the proposed method significantly improves the performance in both distance precision (7.62%) and overlap success (5.34%) compared with the basic short-term tracker SAMF on the OTB-2015 dataset.

On the TC-128 dataset, we validated our algorithm with 11 state-of-the-art trackers, including MEEM [36], SAMF [14], Struck [31], ASLA [67], KCF [18], MIL [29], DCF-CA [68], OCT-KCF [69], FCT [16], L1APG [70] and L2-CF [71]. Among them, OCT-KCF, L1APG and DCF-CA have the ability to handle occlusion. In particular, OCT-KCF is recently developed specifically to solve the occlusion and drifting problems. The comparison results of the tracking are shown in the third row of Figure 9. It can be seen that the overall performance of our tracker is the best, and the precision rate and success rate are 5.79% and 3.21% higher than SAMF, respectively.

Further, we also implemented a performance evaluation of the proposed tracker on the UAV dataset and compared the tracking results with the 14 state-of-the-art trackers, including DSLS [72], MEEM [36], SRDCF [21], MUSTER [73], Struck [31], SAMF [14], DSST [13], TLD [33], DCF [18], KCF [18], CSK [17], MOSSE [20], ASLA [67] and IVT [74]. DSLS is a high performance tracker based on optimized structure learning and dense image representation. In MUSTER, a dual-component approach consisting of short-term and long-term memory stores are used to process target appearance memories. IVT uses incremental PCA in a particle-based filtering framework. The tracking results are presented in the fourth and fifth rows of Figure 9. As can be seen from the comparison, our tracker has a very significant advantage in long-term tracking and is 3.1% and 0.3% higher than DSLS in terms of precision rate and success rate, respectively. However, the overall performance of our tracker on the UAV123 dataset is not as good as DSLS, SRDCF, and MEEM. The cause of this phenomenon is mainly the boundary effect, because DSLS and MEEM use a detection-based architecture and implement tracking through an SVM classifier, so compared with DCF-based trackers, these trackers will not be subject to boundaries effect.

FIGURE 9 Performance evaluation of the proposed tracker with other state-of-the-art algorithms on several benchmark datasets, including OTB-2013, OTB-2015, TC-128, UAV20L and UAV123. In the title of each sub-figure, we indicate the relevant information of the comparison results.
As for SRDCF, the focus of this tracker is to solve the boundary effect, so its basic tracking framework is more advantageous than our method. It is worth mentioning that the objects in the UAV dataset are mostly small targets and the sampling range of the basic correlation filter tracking is related to the object size, which severely limits the search area and brings serious boundary effect. In terms of performance gains, compared with the basic tracker SAMF, the proposed tracker improves by 11.50% in distance precision and 8.20% in overlap success.

4.5  |  Attribute-based evaluation

4.5.1  |  Evaluation on OTB

To further reveal the performance of the tracker, we validated the proposed method using 11 annotated attributes in the OTB-2015 dataset, including illumination variation (IV), scale variation (SV), in-plane rotation (IPR), occlusion (OCC), motion blur (MB), fast motion (FM), background clutters (BC), out-of-plane rotation (OPR), deformation (DEF), out-of-view (OV), and low resolution (LR). Figures 10 and 11 present the comparison of the top 10 DCF-based trackers based on attributes.

In the distance precision comparison of Figure 10, our tracker ranks first on the DEF and second on the three challenges of OPR, SV and OCC. In the comparison of the overlap success of Figure 11, our tracker ranks second on the four challenges of OPR, OCC, DEF and BC. Compared with the best-performing ECO tracker, although the performance of our method in terms of OCC, IPR and OPR is not as good as ECO, the performance difference is slight. In particular, the distance precision of our tracker on DEF is 0.92% higher than that of ECO. It can be seen from the attribute-based tracking results that our tracker has considerable performance in short-term tracking. This means that the proposed reliability assessment criterion and object recovery mechanisms can effectively enhance the ability of the tracker to cope with different scenarios. In addition, the proposed tracker performs well in DEF, OCC and OPR, which also provides a strong guarantee for solving long-term tracking problems, because the target always experiences severe deformation and occlusion in long-term tracking.

4.5.2  |  Evaluation on UAV20L

We tested the proposed tracker on the UAV20L dataset to further verify the performance in long-term tracking. All sequences in the UAV dataset are annotated with 12 attributes, including aspect ratio change (ARC), background clutter (BC), camera motion (CM), fast motion (FM), full occlusion (FO), illumination variation (IV), low resolution (LR), out-of-view (OV), partial occlusion (PO), similar object (SO), scale variation (SV) and viewpoint change (VC). Figure 12 shows the success rate of the trackers under different challenging attributes.

As can be seen from Figure 12, the proposed tracker has achieved considerable performance in most of the attribute scenarios. Among them, especially in the challenges of occlusion, background clutter and aspect ratio change, our tracker is far superior to other tracking algorithms. Compared with the second best tracker DSLS, the success rate of our method is
3.0%, 7.1% and 5.9% higher in the ARC, BC and FO scenarios, respectively. There are two main reasons why the proposed tracker is more competitive than other trackers. First, the valid criterion is used in the tracker to fully assess the reliability of the tracking process, which prevents the tracker from continuously tracking the wrong target. Secondly, an object recovery mechanism with resistance to similar object interference is embedded in the tracker. When the object tracking fails, the mechanism can find the most reliable target to reinitialize the tracker. Therefore, these favourable factors make the proposed tracker more robust.

4.6 Qualitative evaluation on OTB, TC-128 and UAV

We visualized the tracking results on the OTB, TC-128 and UAV datasets to more intuitively demonstrate the advantages of the proposed tracker. In all of these sequences, the target experienced severe deformation and occlusion. The detailed qualitative tracking examples for our tracker and other state-of-the-art trackers are reported in Figures 13, 14 and 15. From the comparison of the results on the OTB in Figure 13, only our trackers closely followed the targets in all the sequences, while other trackers experienced varying degrees of drift or even lost targets. In particular, in the sequence Girl2, the target experienced severe full occlusion and appearance changes, and our tracker successfully recovered the target with the help of the reliability assessment and object recovery mechanism, but all other remaining trackers failed. As shown in Figure 14, only our tracker works well in the Busstation-ce1, Carchase1, Face-ce sequences, while other trackers lose their targets. It is worth noting that there are many similar faces in the sequence Face-ce, but our algorithm does not recover a wrong target, so this demonstrates that the detection coefficient \( \varepsilon \) has a significant effect in eliminating similar interference. Regarding to the sequences Motorbike-ce, Panda, Toyplane, the proposed tracker can track the target stably in the whole process, while most of the other trackers fail to track after the target is occluded. From the comparison of the long-term tracking results in Figure 15, it can be seen that the proposed tracker has significant advantages over other trackers. In the sequences Bike, Car16 and Person7, the appearance and trajectory of the target have undergone dramatic changes, and our tracker can be adapted...
FIGURE 13  Comparison of our tracker and other methods on OTB sequences. From left to right and top to bottom, they are Couple, Girl2, Human3, Human5, Panda, Sylvester, respectively.

FIGURE 14  Comparison of our tracker and other methods on TC-128 sequences. From left to right and top to bottom, they are Busstation-ce1, Carchase1, Face-ce, Motorbike-ce, Panda, Toyplane, respectively.

FIGURE 15  Comparison of our tracker and other methods on UAV sequences. From left to right and top to bottom, they are Bike, Car16, Group1, Group2, Group3, Person7, respectively.
well to these changes with the help of effective model update strategy and object recovery mechanism. However, other trackers have caused tracking drift or failure due to the inability to update the model in time. As for sequences Group1, Group2 and Group3, our tracker is very steadily keeping track of the target, while other trackers have almost lost their targets. Especially in Group2 and Group3, due to occlusion, deformation, and other factors, all the other trackers fail, and our tracking is still very robust. It can be seen from the visual comparison results that the proposed tracker has a strong ability to deal with the occlusion and deformation of the target.

4.7 Performance comparison of other DCF-based trackers with the proposed method

In order to further explore the applicability of the proposed method, we integrated reliability assessment and object recovery mechanism into DCF, KCF and OCT-KCF trackers. The comparison results of the improved trackers are shown in Figure 16 and Table 3. It is obvious that the performance of these three trackers has been greatly improved with the help of reliability assessment and recovery mechanism. In terms of test results on OTB2015, the success rates of DCF-RERM, KCF-RERM and OCT-KCF-RERM are 2.68%, 4.05% and 0.67% higher than their original versions, respectively. In addition, the evaluation results on UAV20L also show that the proposed method has a significant contribution to the performance improvement of trackers, among which the performance of DCF-RERM, KCF-RERM and OCT-KCF-RERM has increased by 12.1%, 13.2% and 3.9% respectively in the success rate. It is worth noting that

| Trackers       | OTB2015 | UAV20L |
|----------------|---------|--------|
|                | Precision | Success | Speed | Precision | Success |
| OCT-KCF-RERM   | 0.7754   | 0.5266  | 59.80  | 0.6420    | 0.4190   |
| OCT-KCF        | 0.7690   | 0.5199  | 74.30  | 0.5730    | 0.3800   |
| Gain           | ↑ 0.64%  | ↑ 0.67% | ↓ 14.5 | ↑ 6.9%    | ↑ 3.9%   |
| KCF-RERM       | 0.7423   | 0.5176  | 63.70  | 0.5980    | 0.4030   |
| Gain           | ↑ 4.66%  | ↑ 4.05% | ↓ 49.7 | ↑ 17.2%   | ↑ 13.2%  |
| DCF-RERM       | 0.7170   | 0.5052  | 70.60  | 0.5900    | 0.3940   |
| Gain           | ↑ 2.48%  | ↑ 2.68% | ↓ 195.7| ↑ 16.0%   | ↑ 12.1%  |
| DCF            | 0.6922   | 0.4784  | 266.3  | 0.4360    | 0.2730   |
| Gain           | ↑        |         |        | ↑        |         |

FIGURE 16 The applicability of the proposed method in other DCF-based trackers. In the title of each sub-figure, we indicate the relevant information of the comparison results.
although the proposed method can greatly improve the performance of the tracker, it also leads to a significant deterioration of running speed. Compared with the original version, the speed of OCT-KCF-RERM dropped by about 14 frames per second, while KCF-RERM dropped by 50 frames per second, and DCF-RERM dropped the most seriously by about 196 frames per second. The main time-consuming part comes from the object recovery process after tracking failure, which involves a relatively large amount of calculation. In particular, KCF and DCF only use the Hog feature, so it is easier to lose objects during the tracking process, resulting in frequent intervention of the object recovery mechanism, which ultimately makes the real-time performance significantly degraded. For the generality of the proposed method, we find that the reliability evaluation method is suitable for the tracking method using the most basic correlation filter, that is, the response value is distributed between 0 and 1. However, it is not suitable for trackers that have adaptively adjusted the response map, such as CF-CA [68] and CF-AT [75], because the adjusted target response score will no longer be distributed between 0 and 1. This means that the setting of reliability evaluation criterion needs to fully consider the structural characteristics of the tracker itself. As for the proposed object recovery mechanism, the method is mainly to find interested candidates first, and then measure the similarity between the candidates and the template to obtain the objects with the highest confidence. This implementation idea is also adopted by most recent long-term trackers and is not limited by the structure of the DCF-based tracker. Therefore, the object recovery mechanism has better versatility than the reliability assessment criterion based on continuous response values.

4.8 | Failure cases

In the experiment, we also discover a few failure cases as shown in Figure 17. For the Skiing sequence, the reason for tracking failure is mainly due to the small search area of the correlation tracking components. The small search area cannot fully cover the distribution range of moving objects. Once the object moves outside the search area, it means that the tracker will lose the object, especially the small object with fast movement speed. In the Surf-car1 sequence, the boundary effect makes the background information also have a strong response, so the tracker gradually drifts, which also causes the template to be contaminated by the background information and eventually causes the tracker to lose the target. Regarding the car2-v sequence, traditional hand-crafted features have limitations in characterizing the appearance model of the object. Therefore, when the object is highly similar to the background, the expression of the feature cannot make the object distinguishable, which causes the tracker to drift into the background.

5 | CONCLUSION

The aim of this paper is to present a new tracker with reliability assessment and object recovery mechanisms to address the challenges of long-term tracking. The proposed tracker uses a correlation filter as a short-term tracker to ensure real-time tracking, and then embeds the reliability assessment and object recovery mechanism to enable the tracker to recover from tracking failure. Considering the appearance and motion characteristics of the target, we build a template pool to maintain the appearance model of the target, and adopt a search strategy that can be flexibly switched between local and global to detect the target. Extensive experimental results on several benchmark datasets show that our tracker achieves impressive performance in both accuracy and speed, and significantly outperforms some related state-of-the-art trackers. Since our approach uses traditional hand-crafted features, the overall performance of the proposed tracker is limited to some extent. In the future, we will explore the application of deep features to further improve the performance.

ACKNOWLEDGEMENTS

This paper is sponsored by National Basic Research Program of China (2014CB744903), National Natural Science Foundation of China (61973212, 61673270), Shanghai Science and Technology Committee Research Project (17DZ1204304).

ORCID

Xingchen Zhang https://orcid.org/0000-0002-8012-0103

REFERENCES

1. Avidan, S.: Ensemble tracking. IEEE Trans. Pattern Anal. Mach. Intell. 29(2), 261–271 (2007)
2. Li, X., et al.: Visual tracking via incremental Log-Euclidean Riemannian subspace learning. Computer Vision and Pattern Recognition, Anchorage, Alaska, USA 1–8 (2008)
3. Babenko, B., et al.: Visual tracking with online Multiple Instance Learning. Computer Vision and Pattern Recognition, Miami, FL, USA 983–990 (2009)
4. Kwak, S., et al.: Learning occlusion with likelihoods for visual tracking. International Conference on Computer Vision, Barcelona, Spain 1551–1558 (2011)
5. Kwon, J., Lee, K.M.: Tracking by Sampling Trackers. International Conference on Computer Vision, Barcelona, Spain 1195–1202 (2011)
6. Liu, B., et al.: Robust tracking using local sparse appearance model and K-selection. Computer Vision and Pattern Recognition, Colorado Springs, CO, USA 1313–1320 (2011)
7. Belagiannis, V., et al.: Segmentation based particle filtering for real-time 2d object tracking. European Conference on Computer Vision, Florence, Italy, 842–855 (2012)
8. Oron, S., et al.: Locally Orderless Tracking. Int. J. Comput. Vision 111(2), 213–228 (2015)
9. Wang, D., et al.: Inverse Sparse Tracker With a Locally Weighted Distance Metric. IEEE Trans. Image Processing 24(9), 2646–2657 (2015)
32. Gao, J., et al.: Transfer Learning Based Visual Tracking with Gaussian Processes Regression. European Conference on Computer Vision, Zurich, Switzerland 188–203 (2014)

33. Kalal, Z., et al.: Tracking-Learning-Detection. IEEE Trans. Pattern Anal. Mach. Intell. 34(7), 1409–1422 (2012)

34. Zhong, W., et al.: Robust object tracking via sparsity-based collaborative model. Computer Vision and Pattern Recognition, Rhode Island, USA 1838–1845 (2012)

35. Chen, D., et al.: Description-discrimination collaborative tracking. European Conference on Computer Vision, Zurich, Switzerland 345–360 (2014)

36. Zhang, J., et al.: MEEEM: Robust Tracking via Multiple Experts Using Entropy Minimization. European Conference on Computer Vision, Zurich, Switzerland 188–203 (2014)

37. Huang, Z., et al.: Learning Aberrance Repressed Correlation Filters for Real-Time UAV Tracking. International Conference on Computer Vision, Seoul, Korea, South 2891–2900 (2019)

38. Valmadre, J., et al.: End-to-end representation learning for correlation filter based tracking. Computer Vision and Pattern Recognition, Honolulu, HI, USA 5000–5008 (2017)

39. Bertinetto, L., et al.: Fully-convolutional siamese networks for object tracking. European Conference on Computer Vision, Amsterdam, The Netherlands, 850–865 (2016)

40. Song, Y., et al.: CREST: Convolutional Residual Learning for Visual Tracking. International Conference on Computer Vision, Venice, 2017, pp. 2574–2583

41. Ma, C., et al.: Hierarchical convolutional features for visual tracking. International Conference on Computer Vision, Santiago 3074–3082 (2015)

42. Danelljan, M., et al.: Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking. European Conference on Computer Vision, Amsterdam, The Netherlands 472–488 (2016)

43. Danelljan, M., et al.: ECO: Efficient Convolution Operators for Tracking. Computer Vision and Pattern Recognition, Honolulu, HI, USA 6931–6939 (2017)

44. Sun, Y., et al.: ROI Pooled Correlation Filters for Visual Tracking. Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA 5776–5784 (2019)

45. Pernici, F., Del Bimbo, A.: Object tracking by oversampling local features. IEEE Trans. Pattern Anal. Mach. Intell. 36(12), 2538–2551 (2013)

46. Dai, W., et al.: Long-term correlation tracking. Computer Vision and Pattern Recognition, Boston, MA, USA, 5388–5396 (2015)

47. Ma, C., et al.: Adaptive correlation filters with long-term and short-term memory for object tracking. Int. J. Comput. Vision 126(6), 771–796 (2018)

48. Hong, Z., et al.: Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking. Computer Vision and Pattern Recognition, Boston, MA, USA, 749–758

49. Dai, W., et al.: Long-term adaptive tracking via complementary trackers. IET Image Processing 13(9), 1569–1577 (2019)

50. Beyer, L., et al.: Towards a Principled Integration of Multi-camera re-identification and Tracking Through Optimal Bayes Filters. Computer Vision and Pattern Recognition, Honolulu, HI 1444–1453 (2017)

51. Lukezic, A., et al.: FuCoLoT - A Fully-Correlational Long-Term Tracker. Asian Conference on Computer Vision, Perth, Australia 595–611 (2018)

52. Sun, C., et al.: Correlation Tracking via Joint Discrimination and Reliability Learning. Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT 489–497 (2018)

53. Dai, K., et al.: Visual Tracking via Adaptive Spatially-Regularized Correlation Filters. Computer Vision and Pattern Recognition, Long Beach, CA 4670–4679 (2019)

54. Bhat, G., et al.: Unveiling the Power of Deep Tracking. European Conference on Computer Vision, Munich, Germany 493–509 (2018)

55. Yang, F., et al.: Robust visual tracking based on global-and-local search with incrementally learning. Computer Vision and Pattern Recognition, Boston, MA, USA, 5388–5396 (2015)

56. Li, F., et al.: Integrating Boundary and Center Correlation Filters for Visual Tracking with Aspect Ratio Variation. International Conference on Computer Vision, Venice, Italy 2001–2009 (2017)

57. Huang, Z., et al.: Learning Aberrance Repressed Correlation Filters for Real-Time UAV Tracking. International Conference on Computer Vision, Seoul, Korea, South 2891–2900 (2019)

58. Valmadre, J., et al.: End-to-end representation learning for correlation filter based tracking. Computer Vision and Pattern Recognition, Honolulu, HI, USA 5000–5008 (2017)

59. Bertinetto, L., et al.: Fully-convolutional siamese networks for object tracking. European Conference on Computer Vision, Amsterdam, The Netherlands, 850–865 (2016)

60. Song, Y., et al.: CREST: Convolutional Residual Learning for Visual Tracking. International Conference on Computer Vision, Venice, 2017, pp. 2574–2583

61. Liu, E., et al.: Hierarchical convolutional features for visual tracking. International Conference on Computer Vision, Santiago, Chile 4310–4318 (2015)

62. Wu, Y., et al.: Online object tracking: a benchmark. Computer Vision and Pattern Recognition, Portland, OR, USA, 2411–2418 (2013)

63. Gao, J., et al.: Robust object tracking with online multiple instance learning. IEEE Trans. Pattern Anal. Mach. Intell. 35(1), 1619–1632 (2011)

64. Grabner, H., et al.: Semi-supervised on-line boosting for robust tracking. European Conference on Computer Vision, Marseille, France, 2008, pp. 334–347

65. Li, F., et al.: Description-discrimination collaborative tracking. European Conference on Computer Vision, Zurich, Switzerland 188–203 (2014)
62. Choi, J., et al.: Attentional Correlation Filter Network for Adaptive Visual Tracking. Computer Vision and Pattern Recognition, Honolulu, HI, USA 4828–4837 (2017)
63. Wang, M., et al.: Large Margin Object Tracking With Circulant Feature Maps. Computer Vision and Pattern Recognition, Honolulu, HI, USA 4800–4808 (2017)
64. Ning, J., et al.: Object Tracking via Dual Linear Structured SVM and Explicit Feature Map. Computer Vision and Pattern Recognition, Las Vegas, NV 4266–4274 (2016)
65. Wang, Q., et al.: DCFNet: Discriminant Correlation Filters Network for Visual Tracking. Computer Vision and Pattern Recognition, 2017, arXiv preprint arXiv:1704.04057
66. Bertinetto, L., et al.: Staple: Complementary learners for real-time tracking. Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 1401–1409 (2016)
67. Jia, X., et al.: Visual tracking via adaptive structural local sparse appearance model. Computer Vision and Pattern Recognition, Providence, RI 1822–1829 (2012)
68. Mueller, M., et al.: Context-Aware Correlation Filter Tracking. Computer Vision and Pattern Recognition, Honolulu, HI 1387–1395 (2017)
69. Zhang, B., et al.: Output constraint transfer for kernelized correlation filter in tracking. IEEE Trans. Systems, Man, and Cybernetics: Systems. 47(4), 693–703 (2017)
70. Bao, C., et al.: Real time robust L1 tracker using accelerated proximal gradient approach. Computer Vision and Pattern Recognition, Providence, RI 1830–1837 (2012)
71. Zhang, B., et al.: Latent constrained correlation filter. IEEE Trans. Image Processing. 27(3), 1038–1048 (2018)
72. Yu, X., et al.: Dense structural learning for infrared object tracking at 200+ Frames per Second. Pattern Recognition Letters. 152–159 (2017)
73. Hong, Z., et al.: Multi-Store Tracker (MUSTer): A cognitive psychology inspired approach to object tracking. Conference on Computer Vision and Pattern Recognition, Boston, MA 749–758 (2015)
74. Ross, D.A., et al.: Incremental learning for robust visual tracking. Int. J. Comput. Vision. 77(1), 125–141 (2008)
75. Bibi, A., et al.: Target Response Adaptation for Correlation Filter Tracking. European Conference on Computer Vision, Scottsdale, AZ, United States 419–433 (2016)

How to cite this article: Liu J, Ye P, Zhang X, Xiao G. Real-time long-term tracking with reliability assessment and object recovery. IET Image Process. 2021;15:918–935. https://doi.org/10.1049/ietipr2.12072