Received January 31, 2020, accepted February 16, 2020, date of publication February 20, 2020, date of current version March 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2975224

Visual Tracking With Online Assessment and Improved Sampling Strategy

MENG DING[1], WEN-HUA CHEN[2], (Fellow, IEEE), LI WEI[3], YUN-FENG CAO[4], AND ZHOU-YU ZHANG[4]

[1]School of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
[2]Department of Aeronautical and Automotive Engineering, Loughborough University, Leicestershire LE11 3TU, U.K.
[3]Jincheng College, Nanjing University of Aeronautics and Astronautics, Nanjing 211156, China
[4]School of Astronautics, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

Corresponding author: Meng Ding (nuaa_dm@nuaa.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61673211, Grant U1633105, and Grant 61203170, in part by the Aeronautical Science Foundation of China under Grant 20155152041 and Grant 20170752008, in part by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant 18KJB590002, and in part by the Open Funds for Key Laboratory of Civil Aircraft Health Monitoring and Intelligent Maintenance under Grant NJ2018012.

ABSTRACT The kernelized correlation filter (KCF) is one of the most successful trackers in computer vision today. However its performance may be significantly degraded in a wide range of challenging conditions such as occlusion and out of view. For many applications, particularly safety critical applications (e.g. autonomous driving), it is of profound importance to have consistent and reliable performance during all the operation conditions. This paper addresses this issue of the KCF based trackers by the introduction of two novel modules, namely online assessment of response map, and a strategy of combining cyclically shifted sampling with random sampling in deep feature space. A method of online assessment of response map is proposed to evaluate the tracking performance by constructing a 2-D Gaussian estimation model. Then a strategy of combining cyclically shifted sampling with random sampling in deep feature space is presented to improve the tracking performance when the tracking performance is assessed to be unreliable based on the response map. Therefore, the module of online assessment can be regarded as the trigger for the second module. Experiments verify the tracking performance is significantly improved particularly in challenging conditions as demonstrated by both quantitative and qualitative comparisons of the proposed tracking algorithm with the state-of-the-art tracking algorithms on OTB-2013 and OTB-2015 datasets.

INDEX TERMS Visual tracking, kernelized correlation filter, online assessment, random sampling, deep feature, handcrafted feature.

I. INTRODUCTION

Visual tracking has been studied over several decades, however, it is still an active research topic in the field of computer vision and pattern recognition [1], [2]. Although visual tracking has been found application in a wide ranges, such as intelligent transportation systems (ITS) [3], vision-based navigation [4], surveillance [5] and motion recognition [6], it still remains challenging in the presence of spatiotemporal variation of targets such as occlusion, illumination variation, and out of view. This causes concerns in outdoor applications particularly for safety critical applications such as driverless cars. Reliable visual tracking is essential to perception and decision making for ensuring timely and appropriate response to environment and events under all the possible weather conditions and traffic conditions. Therefore, improving the robustness of tracking algorithms in the face of these challenges has become an urgent problem in their engineering applications. We believe a promising approach to enhance the tracking robustness of the existing algorithms is to first construct evaluation mechanism for online monitoring the tracking performance, and then improve the tracking performance when it is not reliable by modifying the tracking algorithm as appropriate. This motivate the research reported in this paper.

Generally, visual tracking algorithms are categorized as either discriminative or generative. In the past few years, due to the disadvantage that generative approaches do not

The associate editor coordinating the review of this manuscript and approving it for publication was Guitao Cao.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

make effective use of surrounding information that can distinguish a target from its background, discriminative approaches have gradually become the current mainstream in the field of visual tracking. The discriminative approaches treat the tracking problem as a detection task and learn information about the target from each detection online. Consequently, the discriminative approaches are also referred to as *tracking-by-detection*. Discriminative approaches can also be further divided into two categories: feature-to-classifier trackers and deep learning-based trackers.

Feature-to-classifier trackers aim to establish a classifier that distinguishes a target object from its background. To adapt to the changes of target appearance in dynamic scenes, these trackers must meet two requirements: firstly, the feature representing the difference between a target region and background should be robust and discriminative with respect to variations in both the extrinsic and intrinsic environments; secondly, the classifier for detection must be updateable online. Since online updating can be formulated as a process of online learning, the uncertainty of the labels corresponding to the new training samples obtained from the current tracking results may lead to drifting problems. Therefore, for feature-to-classifier trackers, in order to avoid incorrect tracking results contaminating the classifier, it is very crucial to construct an assessment method to online evaluate the reliability of the tracking result.

Deep learning-based trackers also contain two subcategories [7]. The first subcategory is the deep feature-based trackers, which merely use a pre-training deep network to extract features. In this subcategory, the parameters of the network are not adjusted during tracking. For example, the CNT tracker propagates an image forward in a convolutional neural network (CNN) to extract weak features, and then uses these features to construct a classifier to distinguish a target or background [8]. In the framework of spatially regularized discriminative correlation filters (SRDCF), deep features extracted from the first layer of the VGG network was to enhance the performance of the SRDCF tracker [9]. It is worth noting that SRDCF framework can achieve a better tracking performance than the traditional framework of discriminative correlation filter (DCF) because it mitigates the negative boundary effect of the inherent periodic assumption of the standard DCF. This conclusion can also be proved by [10] which investigated and compared the tracking performances of deep features within both the traditional framework of the DCF and SRDCF framework. However, as the SRDCF framework introduces a spatial regularization component to improve tracking performance, the real-time performance of the algorithm is greatly reduced. Compared with handcrafted features, deep features extracted by a pre-training deep network can represent objects more comprehensively and have a stronger ability to classify different objects [7]. The other subcategory is the tracker that specifically constructs a network framework to extract feature and evaluate candidate regions of tracking [11]–[14]. For example, the SiameFC tracker takes the target template and current search region as inputs, and exploits a deep network to generate a response map for the tracked object with a convolution operation [11]. As an excellent representative of this subcategory, the computational speed of the SiameFC is more than 80 FPS on the GPU platform [11], which indicates that it has outstanding real-time performance.

As a high-speed feature-to-classifier tracker, the kernelized correlation filter (KCF) employs high dimensional features, *e.g.*, the histogram of oriented gradients (HOG), and Gaussian kernel regression to compute a response map, to track a target in accordance with the location of the peak of the response map [15]–[17]. In general, the advantages of the KCF come from four aspects. Firstly, it uses cyclically shifted sampling to achieve a large enough number of samples for training. Secondly, since convolution operation in the spatial domain is converted into the element-wise multiplication in the frequency domain by Fourier transforms, the real-time performance of the KCF is improved greatly. Thirdly, the KCF concedes multi-channel features that enable further extension of its high dimensional features to distinguish the target from the background by simply summing them in the frequency domain. Lastly, the regression model based on the kernel method can improve the classification performance. Considering that the KCF has advantages in tracking performance compared with the standard DCF, and has the advantage of computational efficiency compared with the SRDCF, we choose the KCF as the basic tracking framework of the proposed algorithm. However, despite its excellent tracking performance in a normal conditions, the KCF cannot yield a reliable performance when it confronts with challenges such as occlusion, fast motion [18], [19]. In our opinion, this is due to two reasons as discussed below:

(1) Since the majority of correlation filters use a fixed learning rate to update the regression model in each frame, the errors from subsequent frames will accumulate continuously. For example, when occlusion occurs, due to the disappearance of the target in the several consecutive frames, the tracking results with no tracked target, as new training samples, directly contaminates the regression model and lead to tracking failures. Therefore, it is vitally important to evaluate the tracking performance online and then update the regression model according to the evaluation output.

(2) As a dense sampling scheme, cyclically shifted sampling of the KCF limits the scope of target searching and leads to tracking failure when the distance of the target positions in two consecutive frames caused by fast motion exceeds the search scope. However, merely expanding boundary of cyclically shifted sampling may result in amplification of the boundary effect and lead to an inaccurate representation of the image content [9]. Thus, it would be sensible to incorporate a new sampling scheme to increase the search scope whilst keeping the boundary of cyclically shifted sampling unchanged.

Motivated by the above observations, this paper firstly proposes a method of online assessment of response map in the framework of the KCF, and then proposes a strategy that combines cyclically shifted with random sampling in deep feature space. The main contributions of this paper are as follows.

(1) This paper proposes a method in which the response map is used to online evaluate the reliability of the tracking result in each frame. Specifically, this method firstly designs two indexes denoting the response map shape, and then constructs a 2-D Gaussian estimation model by these two indexes for the reliability assessment of the tracking result.

(2) It proposes a scheme to enhance the cyclically shifted sampling of the KCF by adding random sampling which broadens the search scope of candidate regions when the reliability of the tracking results of the basic KCF is insufficient.

(3) To fully take advantages of deep features in performance and of handcrafted features in efficiency, this paper further incorporates a deep feature-based regression model into the proposed hybrid sampling scheme, and then proposes a strategy of combining cyclically shifted sampling with random sampling in deep feature space. Moreover, according to the result of online assessment, handcrafted and deep feature-based regression models are used interchangeably and updated using different learning rates in the proposed algorithm.

We compare the proposed algorithm with the state-of-the-art trackers on large benchmark datasets OTB-2013 [20] and OTB-2015 [21]. Both quantitative and qualitative experimental results demonstrate that the proposed algorithm performs favorably against state-of-the-art tracking algorithms.

The remainder of this paper is organized as follows. Section 2 discusses the related work, and Section 3 presents details of the proposed tracking algorithm. Section 4 presents and analyzes our experimental results and offers related comparisons with other state-of-the-art tracking algorithms. Finally, Section 5 presents our conclusions.

II. RELATED WORK

There are many reviews about visual tracking. This section only discusses some of the most relevant work motivating our tracker, including sampling mode for tracking tasks, and feature representation of targets in tracking tasks.

A. SAMPLING SCHEME FOR TRACKING

The tracking problem can be described as deciding a way to track an object with little a-priori knowledge. For feature-to-classifier trackers, sampling is an indispensable tool to complete online learning and detection. Sampling scheme is used to collect sufficient training samples in the target's neighborhood, where typically each sample characterizes a sub-window of the same size as the target region. Generally, sampling schemes used in tracking algorithms are divided into two types: random and dense sampling.

As a representative of random sampling, particle sampling is based on Monte Carlo methodology. Since both the computational burden and tracking accuracy are proportional to the particle number, real-time performance is always a huge challenge for particle filter-based trackers [22], [23]. In tracking tasks, dense sampling is to collect all the sub-windows with a certain step size in the target's neighborhood. Generally speaking, this scheme leads to a lot of redundancy because most of the samples have a large amount of overlap regions in tracking tasks. Fortunately, *Henriques et al.* associated this redundant structure with the circulant matrix [24]. The property of the circulant matrix and the circulant structure of samples allow the use of fast Fourier transforms to quickly incorporate information from all sub-windows and to obtain a regression model for detection. Therefore, trackers using this sampling exhibit excellent computational efficiency [25]. However, since the neighborhood area of dense sampling is limited, it will be difficult to identify a target whose position is far away from its current position, for example, due to its fast movement, and reappearance after occlusion occurs [26], [27]. Therefore, this paper proposes a scheme of combining cyclically shifted with random sampling to strike a better balance between computational burden and tracking performance particularly in challenging conditions.

B. FEATURE REPRESENTATION FOR TRACKING

For the past few years, diverse methods of features representation have been proposed for tracking tasks [28]. Generally, the features used in tracking tasks can be divided into three levels: primary, intermediate (handcrafted) and advanced. Primary features include edges, contours and color information, which are ubiquitous and widely used in tracking tasks [29]–[31]. Although many primary features, such as the color histogram, can frequently offer a robust defense against noise, they may not perform well when variations occur in illumination. Compared with primary features, intermediate or handcrafted features, such as HOG [32], local Haar-like features [33] and the scale invariant feature transform (SIFT) [34], have more discriminative abilities that can distinguish a target from its background. In general, advanced features fall into two categories. The first category is sparse features that are further extracted from handcrafted features by sparse coding, such as sparse coding spatial pyramid matching (ScSPM) [35]. The second category is referred as deep features that are mainly generated from the outputs of different layers of a pre-training CNN, and have shown strong advantages, e.g. good generalization and migration ability [7], [10], [36]. However, the computational complexity of deep features is much higher than that of handcrafted features. Therefore, the proposed algorithm aims to make use of the advantage of deep features in performance and of handcrafted features in efficiency.

C. ONLINE EVALUATION OF TRACKING RESULTS

The idea of online evaluation of tracking performance originated from the Tracking-Learning-Detection (TLD) tracker [37], where the tracking performance is evaluated to decide online learning or detection progress. Motivated by this work, the parallel tracking and verifying (PTAV) uses

Siamese network to verify the tracking result calculated by the DSST tracker and improve the tracking performance [38]. Although the idea of online assessment in this paper is similar to that in the PTAV tracker, there are several fundamental differences between them. Firstly, the basic tracking framework is different. Our work was developed based on the KCF, but PTAV selects the DSST as the basic tracker. Generally, since the kernel method is used in the KCF framework to estimate the regression model, the performance of the KCF tracker is better than the performance of the translation filter of the DSST. Secondly, our approach is able to adjust trackers every frame through online assessment but PTAV only operates on sampled frames. This is because PTAV uses a Siamese network with substantial computational burden. To ensure running time efficiency, the verification is run only on sampled frames and cannot adjust the tracker every frame. By contrast, the proposed algorithm designs a method of online assessment of response map which can evaluate and verify the tracking performance every frame. Thirdly, the mechanism of increasing the search scope is different where the performance of the tracker is not reliable. PTAV improves tracking performances by decreasing frame sampling interval and increasing the size of the local region to search for the target. In our opinion, in the framework of the DSST used in PTAV, expanding the size of the local region excessively may reinforce the negative effects of boundary effect. Instead, our algorithm broadens the search scope by combining cyclically shifted with random sampling to avoid enlarging these local regions. Lastly, the operation of the tracking part and the verifier/assessment is different. The tracking part and the verifier work in parallel on two separate threads in the PTAV while online evaluation is used as a trigger to switch different features and sampling schemes in a serial manner in our proposed algorithm.

III. METHODOLOGY

The proposed tracking algorithm first uses cyclically shifted sampling and a handcrafted feature HOG to compute the response map of each frame in the basic KCF framework, and then evaluates the reliability of the tracking result of each frame by online assessment of the response map. If it is assessed to be unreliable, the proposed algorithm employs the scheme of combining cyclically shifted with random sampling in deep feature space to improve tracking performance of this frame. The key to realize the switching between the two strategies is the online assessment of the response map. Therefore, the module of online assessment and improved sampling strategy can be regarded as a whole and embedded into an existing tracking framework.

A. FRAMWORK OF KCF

Considering the KCF is the essential framework of the proposed tracker, we firstly introduce this framework briefly in this section. Generally, the KCF framework contains three modules: regression model training, target detection and regression model updating.

1) REGRESSION MODEL TRAINING

Consider a feature map $\mathbf{Y}_t \in \Re^{m\times n\times C}$ representing the target region and its padding, and a Gaussian-shaped label matrix $\mathbf{r} \in \Re^{m\times n}$ where C is the dimension of the feature, $m \times n$ is the size of feature map. For the first frame, based on the given target region and its padding, the parameters of the regression model $\left(\hat{\mathbf{k}}_1^{\mathbf{YY}}, \hat{\alpha}_1\right)$ is computed by [15]

$$\begin{cases} \mathbf{k}_t^{\mathbf{YY}} = \exp\left(-\frac{2}{\sigma^2}\left(\|\mathbf{Y}_t\|^2 - \mathcal{F}^{-1}\left(\sum_{c=1}^{C} \hat{\mathbf{Y}}_t^* \odot \hat{\mathbf{Y}}_t\right)\right)\right) \\ \hat{\alpha}_t = \dfrac{\hat{\mathbf{r}}}{\hat{\mathbf{k}}_t^{\mathbf{YY}} + \lambda} \end{cases} \tag{1}$$

where λ is a regularization parameter, σ is the Gaussian kernel parameter, $\mathbf{k}_t^{\mathbf{YY}} \in \Re^{m\times n}$ is the kernel correlation between $\mathbf{Y}_t$ and $\mathbf{Y}_t$ itself, $\hat{\mathbf{k}}_t^{\mathbf{YY}}$ is its discrete Fourier transform (DFT), $\alpha_t \in \Re^{m\times n}$ is the regression model, $\hat{\alpha}_t$ is the corresponding DFT, $\mathbf{Y}_t^*$ is the complex-conjugate of $\mathbf{Y}_t$, $\mathcal{F}^{-1}$ denotes the inverse of DFT. $\hat{\mathbf{k}}_t^{\mathbf{YY}}$ and $\hat{\alpha}_t$ are the outputs of the training module.

2) TARGET DETECTION

Depending on the target location in the previous frame, the KCF generates the candidate patches in the current frame by cyclically shifted sampling. Given the feature map of the test image patch $\mathbf{Z}_t \in \Re^{m\times n\times C}$ determined by the target location in the previous frame $t-1$, [15]

$$\begin{cases} \mathbf{k}_t^{\mathbf{YZ}} = \exp\left(-\frac{1}{\sigma^2}\left(\|\mathbf{Y}_{t-1}\|^2 + \|\mathbf{Z}_t\|^2 \right.\right. \\ \qquad \left.\left. -2\mathcal{F}^{-1}\left(\sum_{c} \hat{\mathbf{Y}}_{t-1}^* \odot \hat{\mathbf{Z}}_t\right)\right)\right) \\ \qquad \mathbf{R}_t = \mathcal{F}^{-1}\left(\hat{\mathbf{k}}_t^{\mathbf{YZ}} \odot \hat{\alpha}_{t-1}\right) \end{cases} \tag{2}$$

where $\mathbf{R}_t$ is the response map of the current frame, each element of $\mathbf{R}_t$ denotes the possibility of the target being located in the corresponding position. The position of the tracked target is determined by the location with the maximal value of $\mathbf{R}_t \in \Re^{m\times n}$ as

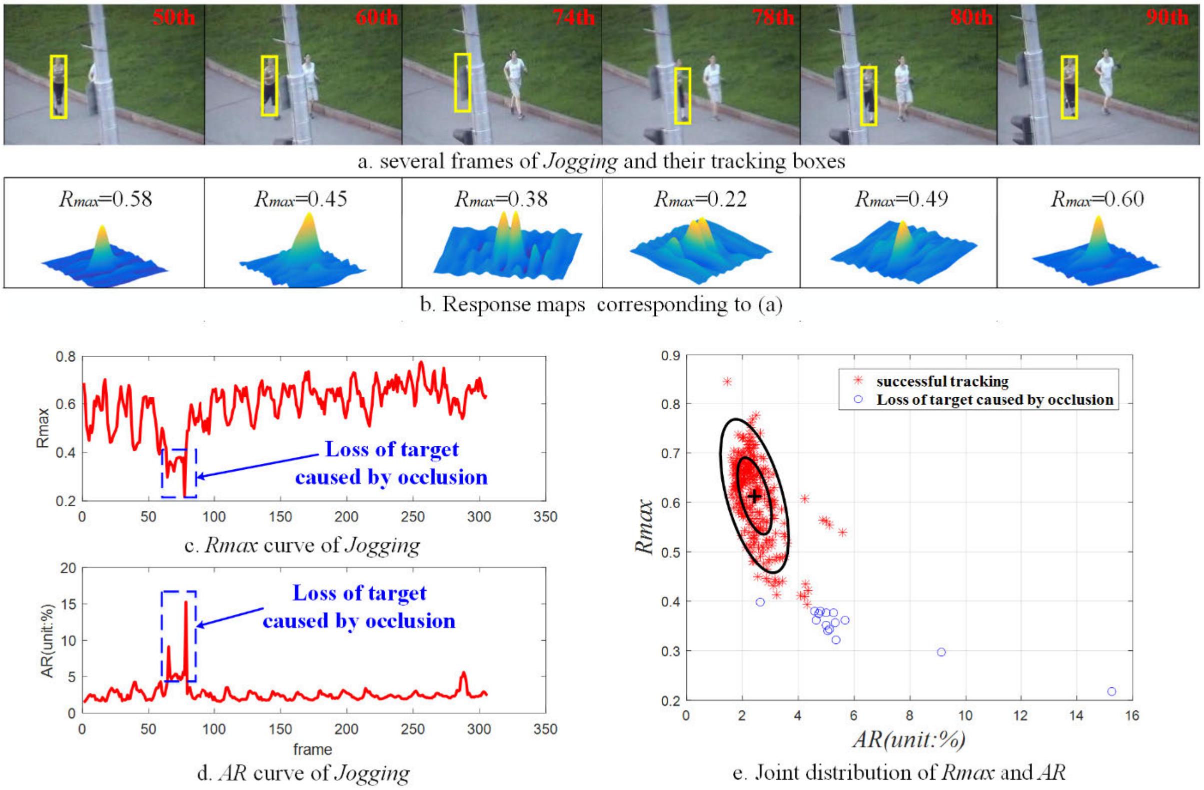
$$[x_t, y_t] = \underset{\substack{i\in 1,2,..,m \\ j\in 1,2,..,n}}{\arg\max}\, \mathbf{R}_t(i,j) \tag{3}$$

where $[x_t, y_t]$ is the position of the detected target.

3) UPDATE

According to the tracking result of each frame, a new feature map of the target $\mathbf{Y}_t$ is produced. In order to learn the latest target appearance, the KCF uses the following scheme to update the existing regression model. $\hat{\alpha}_t$ is first updated in the frequency domain:

$$\hat{\alpha}_t = (1-\delta)\hat{\alpha}_{t-1} + \delta\frac{\hat{\mathbf{r}}}{\hat{\mathbf{k}}_t^{\mathbf{YY}} + \lambda}, \tag{4}$$



a. several frames of *Jogging* and their tracking boxes

b. Response maps corresponding to (a)

c. *Rmax* curve of *Jogging*

d. *AR* curve of *Jogging*

e. Joint distribution of *Rmax* and *AR*

FIGURE 1. Response maps and their assessment indexes for the sequence *Jogging*.

followed by updating $\mathbf{Y}_t$ as

$$\mathbf{Y}_t = \delta \mathbf{Y}_t + (1-\delta)\mathbf{Y}_{t-1} \tag{5}$$

where δ is the learning rate, a fixed value in the KCF.

B. ONLINE ASSESSMENT OF RESPONSE MAP

This section presents a method to online assess the response map $\mathbf{R}_t$ calculated by Eq.(2). Assessment results directly determine whether to employ the strategy of combining cyclically shifted with random sampling in deep feature space.

According to the principle of cyclically shifted sampling, a desirable response map has only one sharp peak and remains smooth in all other regions, because there is only one sample where the target locates at the center. Therefore, the shape of a response map can reveal the reliability of the tracking result. As shown in Figure.1 (a) and (b), the response maps of the 50th and 90th frames of sequence *Jogging* are regular, and there is only one sharp peak and the other regions remain smooth in these two response maps, and their tracking results are reliable. When the target is close to the telegraph pole in 60th and 80th frames, the peaks of response maps become smaller and the other regions of response maps start to fluctuate due to partial occlusion and background clutter. As the target disappears in 74th and 78th frames, two peaks appear and corresponding values decrease further, and the surrounding region fluctuates seriously. Considering the peak and fluctuation degree can denote the reliability of a response map, we design two indexes to assess them:

(1) ***Maximum of response map*** $\mathbf{R}_{\mathbf{max}}$: $R^t_{\max} = \max(\mathbf{R}_t)$, the high of the peak R_{max} indicates the reliability of the tracking result.

(2) ***Area ratio of independence regions AR***, which is defined as follows:

$$AR_t = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}\mathbf{B}(i,j)}{m \times n}$$
$$\mathbf{B}(i,j) = \begin{cases} 1 & \mathit{if}\ \mathbf{R}(i,j) > \tau \\ 0 & \mathit{else} \end{cases} \tag{6}$$

where τ is the threshold of segmentation and estimated by *Ostu* algorithm [39]. As it is well known, *Otsu* algorithm can ideally segment an image where the difference between the foreground and background is outstanding. Furthermore, since a desirable response map is sharp around the peak and smooth in all other regions, the area of the foreground obtained by the *Otsu* segmentation algorithm accounts for a small proportion of the area of the entire desirable response map. Therefore, the lower *AR*, the more reliable the tracking result.

In Figure.1 (c) and (d) respectively shown the changes in the values of R_{max} and *AR* during the tracking process of *Jogging*, we can clearly observe that the values of R_{max} and

AR significantly change between 65th and 79th frames due to the target disappearance caused by occlusion. Figure.1(e) further shows the distribution of the two indices in 2-D space where a blue circle marks the location of the indices corresponding to frames from 65th to 79th with poor tracking performance, and by "$*$" for successfully tracking the target, respectively.

Considering that there is a certain correlation between two parameters, we propose a method to online evaluate the response map by constructing a 2-D Gaussian estimation model (the black ellipses of Figure.1 (e)). Suppose that the tracking results of the first S frames of each tracking sequence are correct, according to the observation vectors containing the two indices $\mathbf{I}^t = [R^t_{\max}, AR_t]^T, t = 2, \ldots, S$, a 2-D Gaussian distribution model can be calculated by maximum likelihood estimation (MLE), its mean vector $\mathbf{u}$ and covariance matrix Δ are expressed as follows:

$$\mathbf{u} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{I}^i,$$
$$\boldsymbol{\Delta} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{I}^i - \mathbf{u})(\mathbf{I}^i - \mathbf{u})^T \qquad (7)$$

where N is the number of observation vector $\mathbf{I}^t$. For the initialization of 2-D Gaussian estimation model, $N = S - 1$. When $t = S + 1$, we can compute the reliability of the response map $\mathbf{R}_t$ according to

$$p(\mathbf{I}^t; \mathbf{u}_{t-1}, \boldsymbol{\Delta}_{t-1}) = \frac{1}{2\pi\sqrt{|\boldsymbol{\Delta}_{t-1}|}}\exp\left(-\frac{1}{2}(\mathbf{I}^t - \mathbf{u}_{t-1})^T\boldsymbol{\Delta}^{-1}_{t-1}(\mathbf{I}^t - \mathbf{u}_{t-1})\right) \qquad (8)$$

If $p(\mathbf{I}^t; \mathbf{u}_{t-1}, \boldsymbol{\Delta}_{t-1}) > \varepsilon$, the tracking result of frame t is reliable and then this vector $\mathbf{I}^t$ representing reliable sample is used to update the 2-D Gaussian distribution model, ε is a threshold. The online assessment method of the response map is summarized as follows.

Online assessment of the response map using the 2-D Gaussian model is one of the main contributions in this paper. It monitors the tracking performance in real time and quantifies the reliability and confidence of the current tracker. It is not only used as a trigger for selecting one of the two tracking strategies presented in this paper, but also as a prerequisite for switching between the detector and the tracker in image understanding system. Furthermore, it can find a much wide range of applications. For example, it could be used as a condition monitoring method for visual tracking, and as an indicator of the level of uncertainty or confidence of the visual sensors in the context of multi-sensor data fusion (e.g. which sensor outcome shall be trusted more in this driving condition) or fed into decision making (*e.g.* reduce vehicle speed or change driving strategy). We will explore this further in our future work.

Algorithm 1 Online Assessment of Response Map

```
Input: the frame index $t$, threshold $\varepsilon$ and the response map $\mathbf{R}_t$
1. Estimate observation vector of 2-D Gaussian model $\mathbf{I}^t$ using $\mathbf{R}_t$ and Eq. (6)
if $t \leq S$
    2. $\mathbf{I} = \{\mathbf{I}^2, \ldots, \mathbf{I}^t\}$ $t = S$
    3. Initialize 2-D Gaussian model $(\mathbf{u}_S, \boldsymbol{\Delta}_S)$ by using $\mathbf{I}$ and Eq.(7);
else
    4. Compute $\mathbf{I}^t$ and $p(\mathbf{I}^t; \mathbf{u}_{t-1}, \boldsymbol{\Delta}_{t-1})$ using Eq.(8);
    if $p(\mathbf{I}^t; \mathbf{u}, \boldsymbol{\Delta}) > \varepsilon$
        5. Update $\mathbf{I} = \{\mathbf{I}, \mathbf{I}^t\}$ and using Eq.(7) to calculate 2-D Gaussian model $(\mathbf{u}_t, \boldsymbol{\Delta}_t)$;
        6. Output: the tracking result of frame $t$ is reliable and 2-D Gaussian model $(\mathbf{u}_t, \boldsymbol{\Delta}_t)$ and $\mathbf{I}$;
    else
        7. $(\mathbf{u}_t, \boldsymbol{\Delta}_t) = (\mathbf{u}_{t-1}, \boldsymbol{\Delta}_{t-1})$ and $\mathbf{I} = \mathbf{I}$;
        8. Output: the tracking result of frame $t$ is not reliable and 2-D Gaussian model $(\mathbf{u}_t, \boldsymbol{\Delta}_t)$ and $I$;
    end if
end if
Return output
```

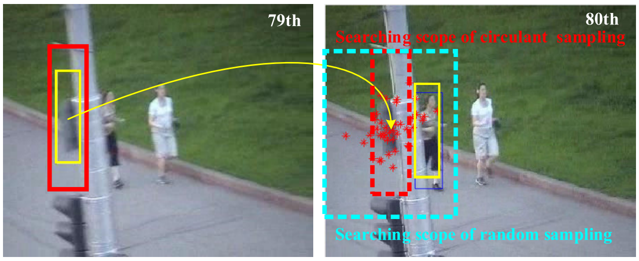

79th
80th
Searching scope of circulant sampling
Searching scope of random sampling

FIGURE 2. Comparison of the search scope for cyclically shifted and random sampling.

C. SCHEME OF COMBINING CYCLICALLY SHIFTED WITH RANDOM SAMPLING

Although cyclically shifted sampling can guarantee the performance of the tracker in real time, the search scope of this sampling mode is limited. In most of KCF-based trackers, the search scope in the current frame is determined by the location of the tracking box in the previous frame. When the occlusion occurs, the target may not be is in the search area, as shown the red dotted line in Figure.2. This may cause the KCF-based trackers using cyclically shifted sampling to fail to track the target successfully. Hence, in order to broaden the search scope of the candidate region for tracking, this paper proposes a scheme of combining cyclically shifted sampling with random sampling, which is used to track the target when the reliability of the tracking results using cyclically shifted sampling is insufficient. This combination scheme contains two modules: sampling and detection.

1) SAMPLING

If $[x_{t-1}, y_{t-1}]$ is the location of the target in frame $t-1$, then

$$\begin{bmatrix} \overline{x}_t^i \\ \overline{y}_t^i \end{bmatrix} = \begin{bmatrix} x_{t-1} + N(0, \Delta) \\ y_{t-1} + N(0, \Delta) \end{bmatrix} \tag{9}$$

where $(\overline{x}_t^i, \overline{y}_t^i)$ is the location of the random sampling in frame t, and each location $i = 1, 2, \cdots, \eta$ represents a tracking candidate region and the corresponding feature map is $\overline{\mathbf{Y}}_t^i \in \Re^{m \times n \times C}$, η is the number of random samples, and $N(0, \Delta)$ is white noise of a Gaussian distribution with standard deviation Δ.

2) DETECTION

If $\{\mathbf{Y}_{t-1}, \hat{\alpha}_{t-1}\}$ is the regression model for the previous frame, and each $\mathbf{Z}_t^i$ denotes a feature map of the test image patch with $(\overline{x}_t^i, \overline{y}_t^i)$ as the center point, we can compute response maps $\mathbf{R}_t^i$ via Eq. (2) and $[x_t^i, y_t^i, score_t^i]$ via Eq. (3), where $score_t^i$ is the maximum value of all elements in the matrix $\mathbf{R}_t^i$. The detection result $[x_t, y_t]$ can be achieved by

$$[x_t, y_t] = \underset{i \in 1,2,..,\eta}{\arg\max} \ (score^i) \tag{10}$$

As shown in Fig 2 where "$*$" denotes the center points of candidate regions obtained by random sampling, the random sampling expands the search scope and can ensure that the target is re-tracked when the occlusion occurs. The proposed algorithm only enables the random sampling if the tracking result of cyclically shifted sampling is unreliable, which indicates that the target may have temporarily disappeared in the image frame. Consequently, we do not need update the module during the process of random sampling to avoid corrupting the regression model.

D. STRATEGY OF COMBINING CYCLICALLY SHIFTED WITH RANDOM SAMPLING IN DEEP FEATURE SPACE

Feature representation plays a significant role in all tracking algorithms. Handcrafted feature, *e.g.* HOG, has been widely used in many KCF-based trackers and achieved good performance. In recent years, with the development of deep learning, it has been shown that the deep features extracted from a pre-training CNN model exhibit a better performance compared with handcrafted features in the same tracking framework. Thus, following the conclusion of Ref. [10], the proposed algorithm employs the activation of the fifth convolutional layer of a pre-trained VGG-2048 network as the deep features to replace the handcrafted features when tracking results based on them are not reliable.

In order to further improve the tracking performance of the proposed algorithm, this paper uses the deep features in the scheme of combining cyclically shifted with random sampling, and then forms a strategy of combining cyclically shifted with random sampling in deep feature space. When the evaluation shows that the result obtained by **Algorithm 1** is unreliable, this strategy is used to improve the tracking performance as described below:

Algorithm 2 Strategy of Combining Cyclically Shifted With Random Sampling in Deep Feature Space

```
**Input:** The location of the target in frame $t-1$ $[x_{t-1}, y_{t-1}]$, deep features-based regression model $\{\mathbf{Y}_{t-1}^D, \hat{\alpha}_{t-1}^D\}$
**1.** Obtaining $\eta$ locations of the random sampling $(\overline{x}_t^i, \overline{y}_t^i)$ in frame $t$ using Eq.(9).
**2.** Computing each deep feature map $\mathbf{Z}_t^{Di}$ of the test patch with $(\overline{x}_t^i, \overline{y}_t^i)$ as the center point.
**3.** Using $\mathbf{Z}_t^{Di}$ and $\{\mathbf{Y}_{t-1}^D, \hat{\alpha}_{t-1}^D\}$ to compute response maps $\mathbf{R}_t^{Di}$ via Eq. (2).
**4.** Using response maps $\mathbf{R}_t^{Di}$ to achieve $[x_t^i, y_t^i, score_t^i]$ via Eq. (3).
**5. Output:** Estimating the tracking result of frame $t$ $[x_t, y_t]$ using Eq.(10).
**Return** output
```

E. PROPOSED ALGORITHM

A new tracking algorithm is proposed by integrating the online assessment of response map and the strategy of combining cyclically shifted with random sampling in deep feature space into the KCF framework.

In the first S frames of a test video sequence, the proposed algorithm trains two regression models based on handcrafted and deep features, respectively, and initializes a 2-D Gaussian estimation model for response map assessment. In the subsequent frames, if the evaluation result of the response map using the handcrafted feature-based regression model in a frame image is reliable, this regression is updated using the fixed learning rate. Otherwise, this model is not updated, and then the strategy of combining cyclically shifted with random sampling in deep feature space is employed to track the target. For the deep feature-based regression, it is updated using a fixed learning rate every k frames if the tracking result of this frame is reliable. It follows that using either of them alone cannot effectively improve the tracking performance of the existing framework. The proposed tracking algorithm is summarized as follows:

IV. EXPERIMENTAL RESULTS

In this section, we conduct experiments to evaluate the proposed tracking algorithm on two challenging public benchmark datasets, containing the OTB-2013 Visual Tracker Benchmark with 50 image sequences [20] and its updated version OTB-2015 with 100 image sequences [21].

OTB datasets involve 11 attributes, including occlusion (OCC) occurred in 48 test sequences, fast motion (FM) in 43 sequences, illumination variation (IV) in 38 sequences, motion blur (MB) in 31 sequences, deformation (DEF) in 45 sequences, out-of plane rotation (OPR) in 63 sequences, scale variation (SV) in 65 sequences, background clutter (BC) in 30 sequences, out-of-view (OV) in 14 sequences, in-plane rotation (IPR) in 51 sequences, low resolution (LR) in 10 sequences. One-pass evaluation (OPE), which is to run the tracker throughout a test sequence with initialization from the

Algorithm 3 The Proposed Tracking Algorithm

```
Input: Test sequence, bounding box $(x_1, y_1)$, $S$ representing the first $S$ frames, $k$ is the update interval of deep feature-based regression model, $L$ denoting the total number of the frames of the test sequence.
1. Initialize the regression models.
   Input the first frame image, $t = 1$, according to bounding box $(x_1, y_1)$, using HOG descriptor and a pretrained VGG-2048 network to calculate handcrafted feature $\mathbf{Y}_1$ and deep feature $\mathbf{Y}_1^D$ respectively, and initializing two regression models $\{\mathbf{Y}_1, \hat{\alpha}_1\}$, $\{\mathbf{Y}_1^D, \hat{\alpha}_1^D\}$ using Eq.(1), $t = t + 1$;
For $t = 2 : L$ do
    if $t \leq S$
2.      According to the frame image $t$ and $(x_{t-1}, y_{t-1})$, calculating handcrafted feature $\mathbf{Y}_t$;
3.      Using Eq. (2) and (3) to compute $\mathbf{R}_t$;
4.      Output: tracking result of frame $t$ $(x_t, y_t)$ using Eq.(3);
5.      Using $(x_t, y_t)$ and Eq. (4) (5) to update regression models $\{\mathbf{Y}_t, \hat{\alpha}_t\}$
6.      Updating a 2-D Gaussian model $(\mathbf{u}_t, \mathbf{\Delta}_t)$ for response map evaluation using Algorithm 1.
7.      Using $(x_t, y_t)$ to calculate deep feature $\mathbf{Y}_t^D$, and using Eq. (1) (4) (5) to update regression model $\{\mathbf{Y}_t^D, \hat{\alpha}_t^D\}$.
    else
8.      Return to Step 2-3.
9.      Computing $\mathbf{I}^t$ using $\mathbf{R}_t$ and $p(\mathbf{I}^t; \mathbf{u}_{t-1}, \mathbf{\Delta}_{t-1})$ using Eq.(8).
        if $p(\mathbf{I}^t; \mathbf{u}_{t-1}, \mathbf{\Delta}_{t-1}) > \varepsilon$
            Return to Step 4-6.
            if mod$((t\text{-}S)/k) = 0$*
                Return to Step.7
            end if
        else
10.         Output: Using Algorithm 2 to achieve tracking result of frame $t$ $(x_t, y_t)$.
        End if
    End if
End for
Return output
```

* mod$((t\text{-}S)/k) = 0$ denotes the result of dividing $t - S$ by k is an integer, and also means that the regression model $\{\mathbf{Y}_t^D, \hat{\alpha}_t^D\}$ is updated using a fixed learning rate every k frames on the premise that the tracking result of this frame is reliable.

ground-truth position in the first frame, is used to objectively evaluate the performance of trackers by two indicators: precision plot and success plot. The precision plot is defined as the percentage of frames whose average Euclidean distance between the center positions of tracked bounding box and the ground-truth is less than the given threshold [20], [21]. The success plot denotes the percentage of successful frames whose overlap rate between the tracked bounding box and the ground-truth bounding box is larger than the given threshold [20], [21]. Evaluated trackers are ranked by the area under the curve (AUC) of each success plot.

The remaining section consists of three parts. The first part is used to describe the details of experimental setup. Secondly, effectiveness of contribution of the proposed algorithm analyzed and compared with the tracker without online assessment of response map and the scheme of combining cyclically shifted with random sampling. In the last part, we compare our tracker with state-of-the-art trackers.

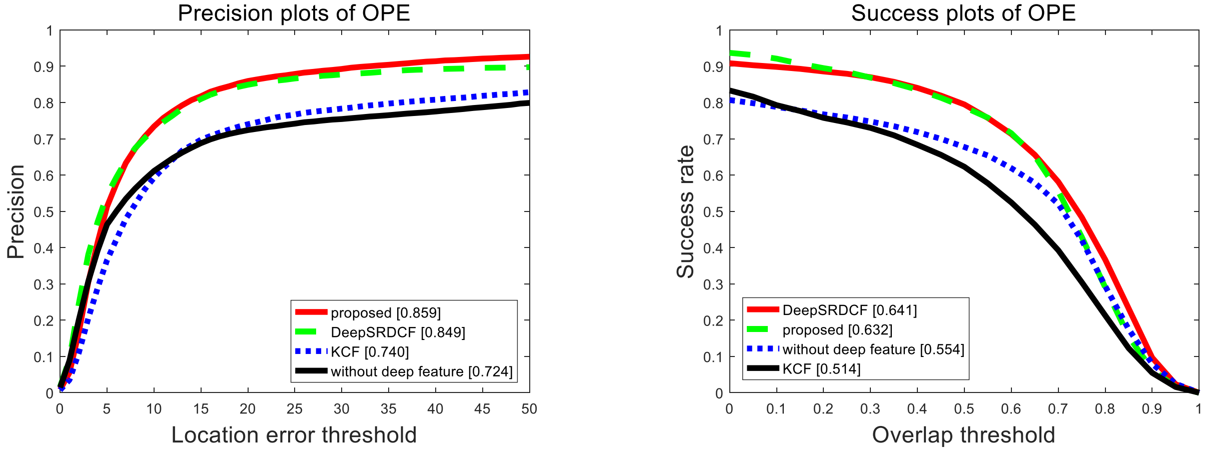
A. EXPERIMENTAL SETUP

We run our proposed tracker in MATLAB 2016a on an Intel i7-7700 CPU (2.8 GHz) PC with 8 GB of memory. All experiments are carried out using the following parameters. For the KCF framework, according to parameter defaults for the KCF, σ of the Gaussian kernel is set to 0.5, the regularization parameter $\lambda = 0.001$ and the learning rate $\delta = 0.01$. VGG-2048 network for deep feature extraction can be download from the MatConvNet toolkit (http://www.vlfeat.org/matconvnet/pretrained/). The size of the image patch for deep feature extraction is expanded to $224 \times 224 \times 3$ by bilinear interpolation. Moreover, our proposed algorithm directly uses the scale detection module of the DSST for scale variation [38].

In **Algorithm 1**, the threshold of ε is set to 0.01. This threshold has a significant influence in the frequency of using **Algorithm 2** during tracking and will be discussed in the ablation study of this section.

For **Algorithm 2**, because the value of η directly affects the computing efficiency of random sampling, to ensure that the function of random sampling can be fully utilized, we set the value of η to 50. The value of Δ directly determines the search range of random sampling. When the value η is determined, the entire search area may not be effectively covered if the value of Δ is too large. On the contrary, random sampling degenerates into an exhaustive search if the value of Δ is too small. Thus, after analyzing the target displacement between adjacent frames of the test data set, Δ of Eq. (9) is set to equal to the width of the ground-truth in the first frame of each test sequence. The combined effect of two parameters on tracking performance will be further discussed in the following ablation study.

In **Algorithm 3**, two parameters, S and k, used for deep feature-based regression model training and update, need to be preset. Considering that the traditional KCF can normally track the target successfully in the first 20 frames of all test sequences used in our experiment, we set S to 20 denoting the first S frames. In our experiments, update interval of deep feature-based regression model k equals to 20. In the ablation study, we will discuss the effects of these two parameters of **Algorithm 3** on tracking performance, respectively.



(a) Precision plots on OTB-2013

(b) Success plots on OTB-2013

FIGURE 3. Performance comparison of the proposed tracker and the other three trackers.

B. FUNCTION OF TWO MODULES

In this section, we analyze the function and impact of these two proposed modules, namely online assessment of response map, and a strategy of combining cyclically shifted sampling with random sampling in deep feature space, by designing comparative experiments between the proposed tracker and the other three trackers on OTB-2013 dataset. Using either of them alone cannot effectively improve the tracking performance of the existing framework. This is because **Algorithm 1** is the trigger for **Algorithm 2** and one module must be followed by the other. As described above, if there are no these two modules, the proposed algorithm degenerates to a KCF tracker. Therefore, adding the KCF to the comparative experiment of this section can test the function of these two modules. To evaluate the influence of online assessment and random sampling without using deep features, we investigate the tracking performance using the handcrafted feature, instead of deep features in **Algorithm 2**, which is referred to as 'without deep feature' in Figure 3. Figure 3 shows the precision plots and success plots of the comparative experiments on the OTB-2013 dataset. From Figure 3, it can be clearly observed that our tracker integrating these two modules has significant advantages in precision plots, compared to the KCF and 'without deep feature' trackers. The results of Figure.3 confirms that integrating these two modules together is beneficial to improve the performance of a tracker. Moreover, it plays an important role to use deep feature in the strategy of combining cyclically shifted with random sampling for improving the tracking performance.

Considering that the deep feature used in this paper comes from the DeepSRDCF tracker, we select it as one of trackers for comparison to evaluate the effectiveness of the deep feature. As mentioned above, the SRDCF framework used by the DeepSRDCF tracker is superior to the standard DCF and KCF because it introduces a spatial regularization component to mitigate the boundary effect [10]. However, Figure.3 shows that the proposed tracker can achieve tracking performance similar as the DeepSRDCF by integrating the online assessment and improved sampling strategy into the KCF framework. As mentioned above, the main contribution of this paper is to introduce two modules into an existing tracking framework. It is not restricted to the KCF framework as discussed in this paper. Therefore it is expected that these two modules can be introduced into the SRDCF framework to further improve its performance. Furthermore, the real-time performance of the SRDCF framework is much worse. Specifically, as shown in table.2, the computational speed of the DeepSRDCF tracker is only about 2 frames per second (FPS) on our experimental platform is far less than the 12 fps of our tracker.

C. ABLATION STUDIES

The threshold of ε in **Algorithm 1** is the most important tuning parameter and directly determines the evaluation result of response map. When the value of ε is chosen to be too small, most of the tracking results are assessed to be reliable, the benefit of the proposed approach cannot be fully realized, and the tracking performance will not be improved significantly. On the contrary, if the threshold of ε is chosen to be large, real-time performance of the algorithm is significantly reduced since the strategy of combining cyclically shifted with random sampling in deep feature space is employed quite frequently. Taking *Jogging* as an example, Figure.4 shows the relationship between the change of ε and the number of the strategy of combining cyclically shifted with random sampling in deep feature space activated. Furthermore, we compare the five different ε on the OTB-2013 dataset and the results are shown in Table 1. Tracking speed in FPS is used to evaluate the real-time performance of the tracker. From Table 1, we can find that FPS decreases rapidly as the value of ε increases. When the value of ε exceeds 0.01, the increase trend of precision and success rates is significantly reduced. Therefore, considering the balance between real-time performance and tracking performance, in this paper we choose 0.01 as the value of ε.

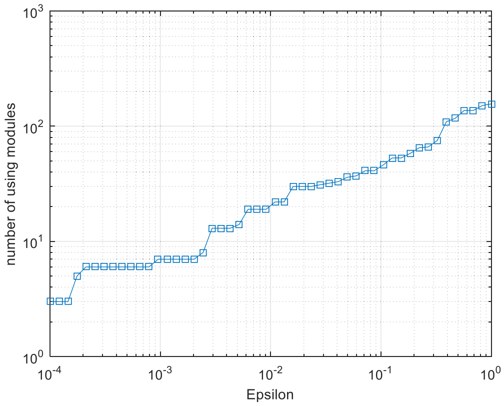


FIGURE 4. relationship between the change of ε and the number of using proposed modules on the test sequence *Jogging*.

TABLE 1. Performance comparison between five different ε on the OTB-2013.

ε	10^{-4}	10^{-3}	10^{-2}	10^{-1}	10^0
Precision rate	0.748	0.785	0.859	0.861	0.860
Success rate	0.529	0.571	0.632	0.631	0.632
FPS	41	30	12	4	0.5

TABLE 2. Performance comparison between three different Δ and three different η on the OTB-2013.

Δ	η	Precision rate	Success rate	FPS
$0.5w$	30	0.749	0.531	14
	50	0.783	0.568	12
	70	0.797	0.603	6
w	30	0.803	0.608	13
	50	0.859	0.632	12
	70	0.860	0.632	6
$2w$	30	0.772	0.566	14
	50	0.803	0.602	11
	70	0.826	0.616	7

w is the width of the ground-truth in the first frame of each test sequence, and ε=0.01.

In **Algorithm 2**, the values of η and Δ directly determine the searching range of random sampling and the computational speed of **Algorithm 2**. In order to analyze the effect of different values of η and Δ on tracking performance, we compare the precision rates, success rates and FPSs corresponding to the different values of η and Δ on the OTB-2013 dataset and the results are shown in Table 2 where $\varepsilon = 0.01$. From Table 2, we can observe two trends: (1) On the premise of the value of η is constant, as the search area expands, the values of precision rate and success rate increase first and then decrease. (2) On the premise of the value of Δ is constant, increasing the number of random samples can improve the tracking performance but reduces the real-time performance. As we all know, the second trend is easy to understand. An increase in the number of samples will inevitably lead to an increase in computational burden and an increase in the search density in a certain area. The former is the cause of the decline in real-time performance, and the latter is the reason for the improvement in tracking performance. We firmly believe that in the first trend, the reason for the improvement in tracking performance at the beginning is that the expansion of the search area can obtain more candidate areas, conversely, when the search area is enlarged to a certain extent, because the number of samples does not increase proportionally with the expansion of the search area, the sampling density decreases, some candidate regions containing the target are ignored, and more background interference is introduced, as a result, tracking performance is degraded.

TABLE 3. Performance comparison between three different S on the OTB-2013.

S	Precision rate	Success rate	FPS
10	0.841	0.624	14
20	0.859	0.632	12
30	0.742	0.533	11

TABLE 4. Performance comparison between three different k on the OTB-2013.

k	Precision rate	Success rate	FPS
10	0.859	0.632	10
20	0.859	0.631	12
30	0.858	0.632	12

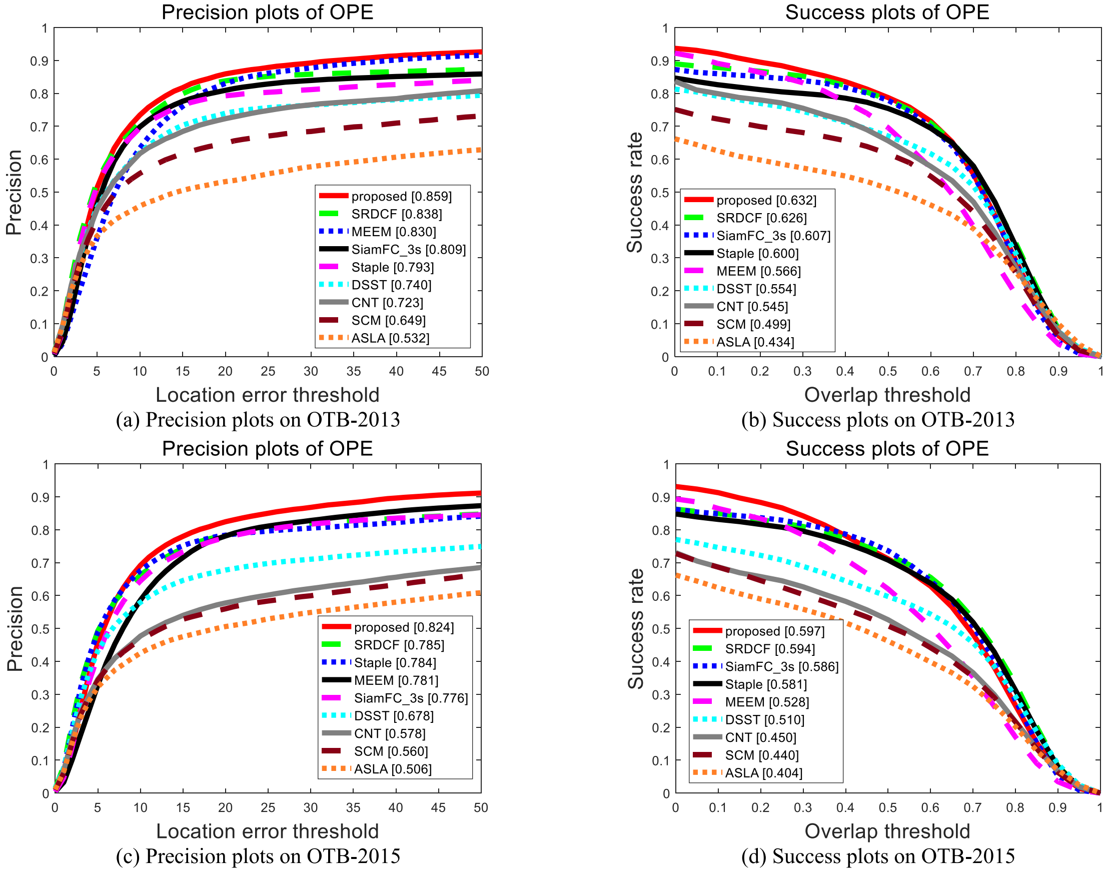
In **Algorithm 3**, the tracking results of the first S frames of each test sequence, which are achieved by the traditional KCF, are used to train a deep feature-based regression model. Since the traditional KCF can successfully track the targets in the first 20 frames of most test sequences, from Table 3 we can find that the tracking performance is not much different when $S = 10$ and $S = 20$. Furthermore, when $S = 30$, some false results (between 20th to 30th frames) from the traditional KCF tracker may contaminate the deep feature-based regression model, resulting in degraded tracking performance. The tracking performance indexes for three different update intervals of deep feature-based regression model are shown in Table 4. Table 4 indicates that changes in the value of k have little effect on tracking performance.

D. COMPARISONS TO STATE-OF-THE-ART TRACKERS

We evaluate our proposed algorithm against eight representative algorithms. These trackers can be divided into three typical categories: (1) Correlation filter-based algorithms (DSST [40], Staple [41] and SRDCF [9]), (2) CNN-based algorithms (CNT [8] and SiamFC [11]), and (3) multiple online classifier-based or sparse coding-based algorithms (ALSA [42], SCM [43] and MEEM [44]). The comparison experiments are conducted quantitatively and qualitatively.

1) QUANTITATIVE EVALUATION

Figure 5 contains the precision and success plots for the OPE test on OTB-2013 and OTB-2015. As shown in Figure 5, the proposed algorithm performs favorably against all the other eight algorithms. Taking comparative results of OTB-2015 as an example as shown in Figure 5 (c), the proposed tracker performs well for the precision rate with 82.4%, which



(a) Precision plots on OTB-2013
(b) Success plots on OTB-2013
(c) Precision plots on OTB-2015
(d) Success plots on OTB-2015

FIGURE 5. Performance comparison of the proposed tracker and state-of-the-art trackers.

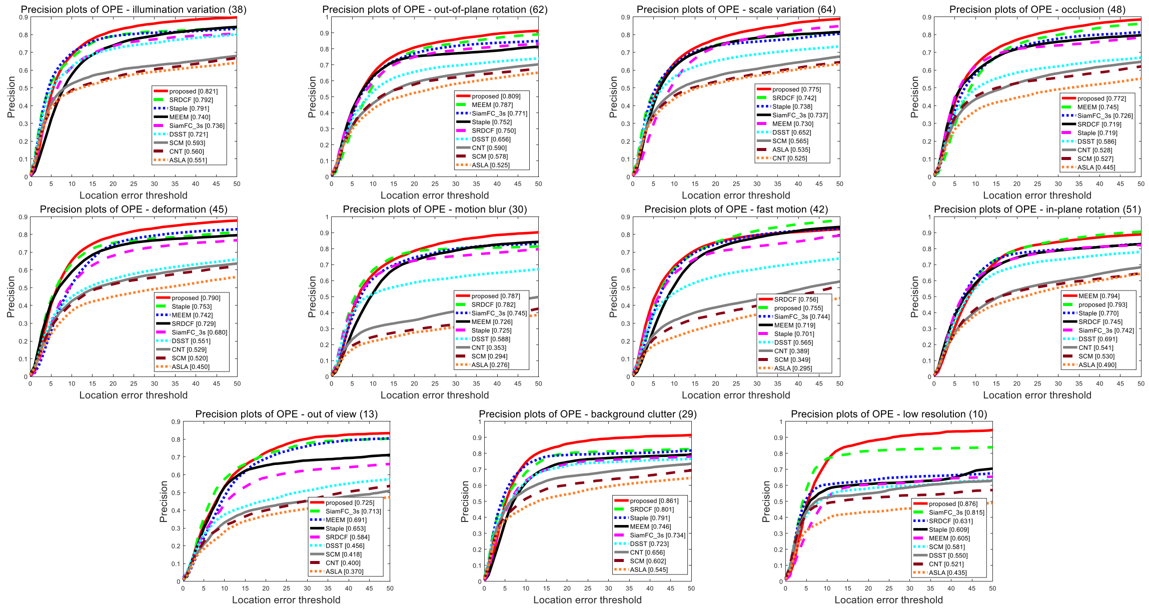


FIGURE 6. Precision plots for attribute-based evaluation on OTB-2015.

is approximate 4% higher than the tracker ranked second. Moreover, the proposed tracker also achieves the best success rate of 59.7% among all the trackers.

We further use the image sequences annotated by 11 attributes to comprehensively evaluate the performance of trackers. Figure 6 and 7 show the precision and success

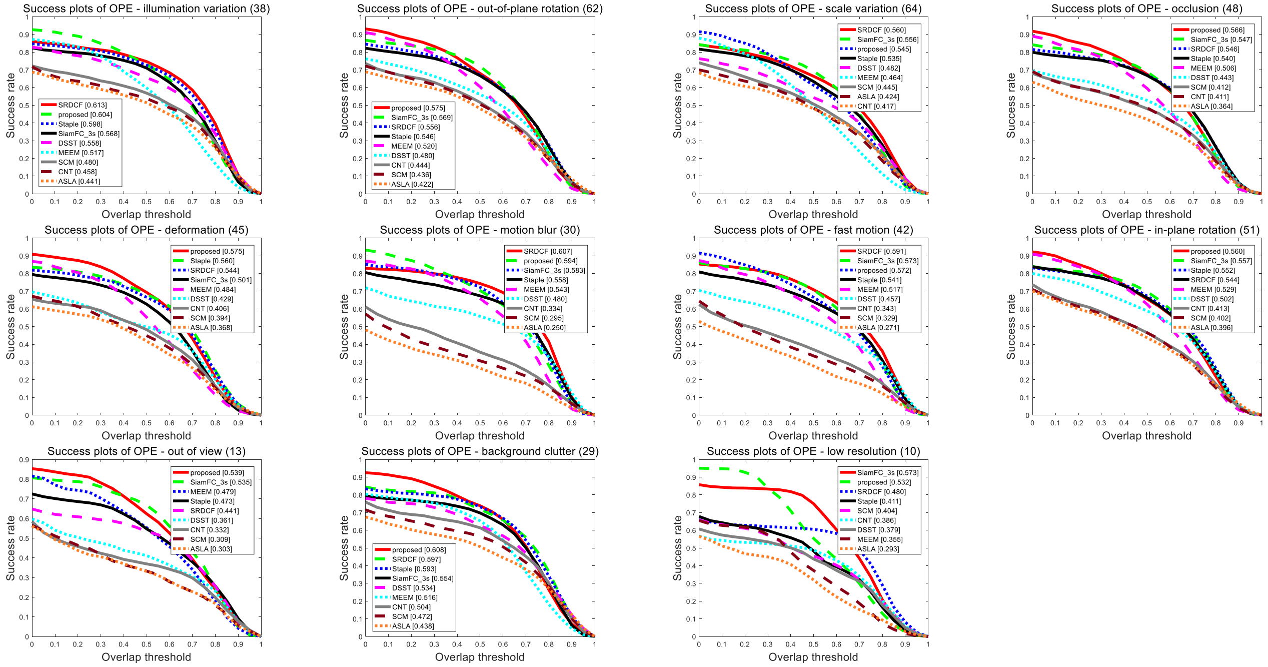


FIGURE 7. Success plots for attribute-based evaluation on OTB-2015.

TABLE 5. Real-time performance comparison of nine trackers on OTB-2015.

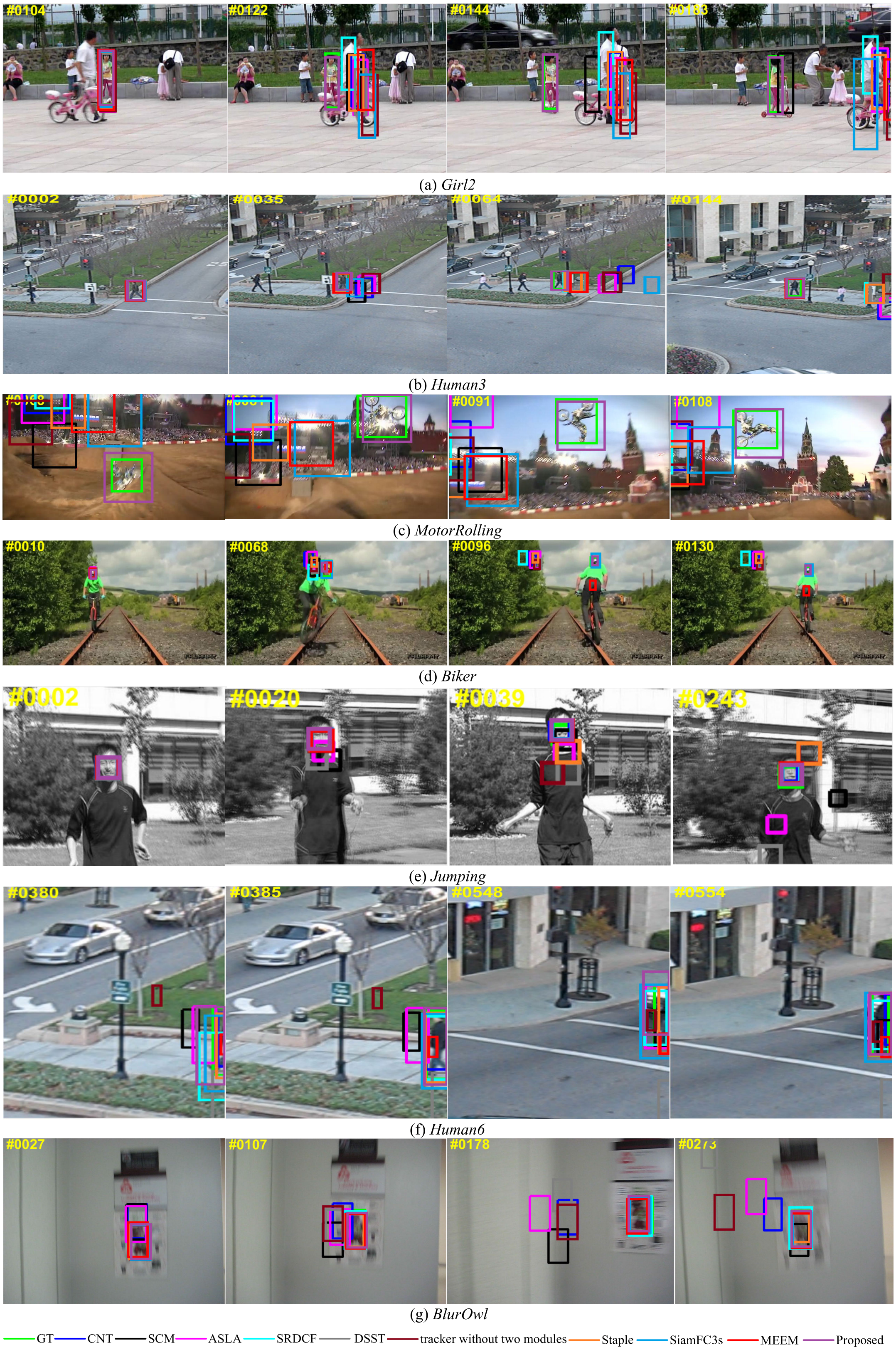
Tracker	DSST	ASLA	Staple	SCM	SRDCF	MEEM	CNT	DeepSRDCF	Proposed
FPS	23	6	44	0.5	5	10	5	2	12

plots of the proposed tracker and other 8 trackers respectively on the OTB-2015. Although there is no tracker that shows excellent performance on 11 attributes, the proposed tracker shows excellent performance on most of attributes. Specifically, the proposed algorithm achieves the best performance on 9 attributes in term of the precision rate, including illumination variation (82.1%), out-of-plane rotation (80.9%), scale variation (77.5%), occlusion (77.2%), deformation (79%), motion blur (78.7%), out of view (72.5%), background clutter (86.1%) and low resolution (87.6%). In term of the success rate, the proposed algorithm significantly outperforms the compared trackers on 6 attributes, including out-of-plane rotation (57.6%), occlusion (56.6%), deformation (57.5%), in-plane rotation (56%), out of view (54.6%) and background clutter (60.9%). It can also be seen that the robustness of the proposed algorithm in the presence of various challenges significantly outperforms the other 8 tracking algorithms.

Moreover, Table 5 shows the speeds of nine algorithms in FPS, obtained from the average values when running OTB-2015 on our computational platform. Considering that the SiamFC needs to run on the GPU, we do not test the computational speed of the SiamFC in this comparison experiment of real-time performance. In the nine trackers, the proposed tracker ranks third. Although the calculation speed of the proposed method is slower than that of the Staple and DSST, Figure 5 shows that the tracking performance of our method is significantly better than these two trackers. Table 5 also demonstrates that although the proposed algorithm uses deep features, its computational speed is still much faster than SRDCF without deep features and CNT which is one of the CNN-based trackers.

2) QUALITATIVE EVALUATION

This section provides a qualitative analysis of the proposed algorithm, the tracker without these two proposed modules and the other eight algorithms, with the tracking results shown in Figure.8. In the *Girl2*, when full occlusion occurs, only our algorithm can track the target successfully at Frame 144. In the *Human3*, when partial occlusions occur that the target crosses a pole and passes by other pedestrians, only our algorithm and MEEM can accomplish the tracking task at Frame 144. This clearly shows that the proposed algorithm exhibits an excellent performance in re-tracking the target when the target reappears after being occluded. In the *MotorRolling*, despite all the other nine algorithms failed to capture the target, the proposed algorithm can capture the rotated target successfully. In the *Biker*, the head of the biker moves quickly from left to right. In the *Jumping*, the player bounces up and down at a high rate. For these two sequences, except for our algorithm, CNT and SiamFC_3S, the other algorithms could not capture the fast-moving target reliably. In the *Human6*, the target moves out of view in Frame 380 and 548, respectively. Our algorithm can re-track the target when it re-entered the field of view in



(a) *Girl2*

(b) *Human3*

(c) *MotorRolling*

(d) *Biker*

(e) *Jumping*

(f) *Human6*

(g) *BlurOwl*

FIGURE 8. Qualitative comparison of ten trackers.

Frame 385 and 554, respectively. Motion blur occurs when the target region is blurred due to the motion of the target or camera. In the *BlurOwl*, our algorithm, SRDCF and MEEM achieve superior performance than the other algorithms in coping with this challenging condition.

V. CONCLUSION

This paper aims to improve robustness and reliability of visual tracking in challenging operation conditions. In the promising KCF framework, two new functional modules have been proposed and developed to further enhance its tracking performance. An online assessment method has been proposed to evaluate tracking performance and reliability based on the response map. To this end, a new criterion was developed by constructing a 2-D Gaussian estimation model based on the peak and the area ration of independence regions in the response map defined in the paper. When the tracking performance is assessed to be unreliable, a strategy of combining cyclically shifted with random sampling was proposed to improve the tracking performance. These two proposed modules are then integrated into the current KCF tracker to constitute a new tracking algorithm. With this framework, deep features have also been exploited to further enhance its tracking performance and reliability. We extensively test our algorithm on two well documented benchmark datasets with very encouraging results. Detailed qualitative and quantitative analysis and comparisons with eight existing competitive tracking algorithms clearly demonstrate attractive tracking performance of our proposed algorithm in terms of accuracy and reliability without a significant increase of the computational burden in coping with a wide range of challenging operations, including illumination variation, out-of-plane rotation, scale variation, occlusion, deformation, motion blur, out of view, background clutter and low resolution. A tuning parameter is introduced to trade off between the reliability and accuracy of the tracking and its real-time performance.

The proposed online performance and reliability assessment method could find a wide range of applications such as real-time tracking performance monitoring, and characterization of the confidence or uncertainty level of the visual tracking information for the purpose of data fusion with other sensing sources, or as an input to follow-on decision making. It is expected that it would have a significant implication in a wider application of visual tracking, particularly for safety critical situations such as autonomous driving. This will be explored in our future work.

REFERENCES

[1] J. Zhang, X. Jin, J. Sun, J. Wang, and K. Li, "Dual model learning combined with multiple feature selection for accurate visual tracking," *IEEE Access*, vol. 7, pp. 43956–43969, Apr. 2019.

[2] D. Ge, J. Song, Y. Qi, C. Wang, and Q. Miao, "Self-paced dense connectivity learning for visual tracking," *IEEE Access*, vol. 7, pp. 37181–37191, Apr. 2019.

[3] J.-C. Tai, S.-T. Tseng, C.-P. Lin, and K.-T. Song, "Real-time image tracking for automatic traffic monitoring and enforcement applications," *Image Vis. Comput.*, vol. 22, no. 6, pp. 485–501, Jun. 2004.

[4] A. Ess, K. Schindler, B. Leibe, and L. Van Gool, "Object detection and tracking for autonomous navigation in dynamic environments," *Int. J. Robot. Res.*, vol. 29, no. 14, pp. 1707–1725, May 2010.

[5] I. S. Kim, H. S. Choi, and K. M. Yi, "Intelligent visual surveillance-a survey," *Int. J. Control, Automat. Syst.*, vol. 8, no. 5, pp. 926–939, Oct. 2010.

[6] M. Ziaeefard and R. Bergevin, "Semantic human activity recognition: A literature review," *Pattern Recognit.*, vol. 48, no. 8, pp. 2329–2345, Aug. 2015.

[7] P. Li, D. Wang, L. Wang, and H. Lu, "Deep visual tracking: Review and experimental comparison," *Pattern Recognit.*, vol. 76, pp. 323–338, Apr. 2018.

[8] K. Zhang, Q. Liu, Y. Wu, and M.-H. Yang, "Robust visual tracking via convolutional networks without training," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1779–1792, Apr. 2016.

[9] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4310–4318.

[10] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Dec. 2015, pp. 58–66.

[11] L. Bertinetto, J. Valmadre, and J. F. Henriques, "Fully-convolutional siamese networks for object tracking," in *Proc. Eur. Conf. Comput. Vis. Workshops*, Amsterdam, The Netherlands, Oct. 2016, pp. 850–865.

[12] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Y. Choi, "Action-decision networks for visual tracking with deep reinforcement learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 2711–2720.

[13] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 4293–4302.

[14] H. Nam, M. Baek, and B. Han, "Modeling and propagating CNNs in a tree structure for visual tracking," 2016, *arXiv:1608.07242*. [Online]. Available: http://arxiv.org/abs/1608.07242

[15] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

[16] F. Liu, C. Gong, X. Huang, T. Zhou, J. Yang, and D. Tao, "Robust visual tracking revisited: From correlation filter to template matching," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2777–2790, Jun. 2018.

[17] Z. Zhao, P. Feng, J. Guo, C. Yuan, T. Wang, F. Liu, Z. Zhao, Z. Cui, and B. Wu, "A hybrid tracking framework based on kernel correlation filtering and particle filtering," *Neurocomputing*, vol. 297, pp. 40–49, Jul. 2018.

[18] L. Zhang and P. N. Suganthan, "Robust visual tracking via co-trained kernelized correlation filters," *Pattern Recognit.*, vol. 69, pp. 82–93, Sep. 2017.

[19] J. Wang, W. Liu, W. Xing, and S. Zhang, "Visual object tracking with multi-scale superpixels and color-feature guided kernelized correlation filters," *Signal Process., Image Commun.*, vol. 63, pp. 44–62, Apr. 2018.

[20] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 2411–2418.

[21] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.

[22] M. Ding, Y. Yao, L. Wei, and Y. Cao, "Visual tracking using locality-constrained linear coding and saliency map for visible light and infrared image sequences," *Signal Process., Image Commun.*, vol. 68, pp. 13–25, Oct. 2018.

[23] M. Ding, L. Wei, Y. Cao, J. Wang, and L. Cao, "Visual tracking using locality-constrained linear coding under a particle filtering framework," *IET Comput. Vis.*, vol. 12, no. 2, pp. 196–207, Mar. 2018.

[24] J. F. Henriques, R. Caseiro, and P. Martins, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. Eur. Conf. Comput. Vis.*, Florence, Italy, Oct. 2012, pp. 702–715.

[25] B. Uzkent and Y. Seo, "EnKCF: Ensemble of kernelized correlation filters for high-speed object tracking," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Lake Tahoe, NV, USA, Mar. 2018, pp. 1133–1141.

[26] M. Wang, Y. Liu, and Z. Huang, "Large margin object tracking with circulant feature maps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 4800–4808.

[27] T. Zhang, S. Liu, C. Xu, B. Liu, and M.-H. Yang, "Correlation particle filter for visual tracking," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2676–2687, Jun. 2018.

[28] Z. Pan, S. Liu, and W. Fu, "A review of visual moving target tracking," *Multimedia Tools Appl.*, vol. 76, no. 16, pp. 16989–17018, Jun. 2016.

[29] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564–577, May 2003.

[30] H. Zhou, Y. Yuan, Y. Zhang, and C. Shi, "Non-rigid object tracking in complex scenes," *Pattern Recognit. Lett.*, vol. 30, no. 2, pp. 98–102, Jan. 2009.

[31] J. Ning, L. Zhang, D. Zhang, and C. Wu, "Robust object tracking using joint color-texture histogram," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 23, no. 7, pp. 1245–1263, Nov. 2011.

[32] S. Sun, Q. Guo, F. Dong, and B. Lei, "On-line boosting based real-time tracking with efficient HOG," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Vancouver, BC, Canada, May 2013, pp. 2297–2301.

[33] C. Xia, S. F. Sun, and P. Chen, "Haar-like and HOG fusion based object tracking," in *Proc. Pacific Rim Conf. Multimedia*, Kuching, Malaysia, Dec. 2014, pp. 173–182.

[34] Z. Wang and K. Hong, "A new method for robust object tracking system based on scale invariant feature transform and camshaft," in *Proc. ACM Res. Appl. Comput. Symp.*, San Antonio, TX, USA, Oct. 2012, pp. 132–136.

[35] S. Zhang, H. Yao, X. Sun, and X. Lu, "Sparse coding based visual tracking: Review and experimental comparison," *Pattern Recognit.*, vol. 46, no. 7, pp. 1772–1788, Jul. 2013.

[36] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, Dec. 2015, pp. 3074–3082.

[37] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-Learning-Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, Jul. 2012.

[38] H. Fan and H. Ling, "Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 5487–5495.

[39] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

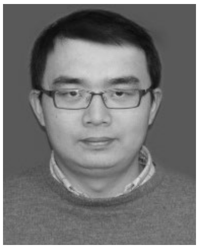
[40] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Discriminative scale space tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1561–1575, Aug. 2017.

[41] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 1401–1409.

[42] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparse collaborative appearance model," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2356–2368, May 2014.

[43] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via coarse and fine structural local sparse appearance models," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4555–4564, Oct. 2016.

[44] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: Robust tracking via multiple experts using entropy minimization," in *Proc. Eur. Conf. Comput. Vis.*, Zurich, Switzerland, Sep. 2014, pp. 188–203.



MENG DING received the Ph.D. degree in control theory and control engineering from the Nanjing University of Aeronautics and Astronautics (NUAA), in 2010.

He then joined the School of Civil Aviation as a Lecturer and became an Associate Professor, in 2013. From 2018 to 2019, as a Visiting Scholar, he did research on vision-based environment perception system of intelligent vehicle at Loughborough University, Loughborough, U.K. He has undertaken several funded research project as the Principle Investigator of the National Natural Science Foundation of China, in 2012 and 2016, the China Postdoctoral Science Foundation, in 2013, and the Aeronautical Science Foundation of China, in 2015 and 2017, respectively. He has published more than 40 articles in international journals and conferences. His current research interests include vision-based target detection and tracking, sense-and-avoid system for unmanned aerial vehicle, and environment perception system of intelligent vehicle. He was honored with the National Defense Science and Technology Progress Award, in 2011 (ranking ninth), 2013 (ranking sixth), and 2017 (ranking second).

WEN-HUA CHEN (Fellow, IEEE) received the M.Sc. and Ph.D. degrees from Northeast University, Shenyang, China, in 1989 and 1991, respectively.

From 1991 to 1996, he was a Lecturer and then an Associate Professor with the Department of Automatic Control, Nanjing University of Aeronautics and Astronautics, Nanjing, China. From 1997 to 2000, he held a research position and then he was a Lecturer in control engineering with the Centre for Systems and Control, University of Glasgow, Glasgow, U.K. In 2000, he moved to the Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, U.K., as a Lecturer, where he was appointed as a Professor, in 2012. His research interests include the development of advanced control, as well as signal processing and decision making methods and their applications in aerospace engineering. As a Professor in autonomous vehicles, he is currently working on the development of unmanned autonomous systems. He is a Chartered Engineer (CEng) in the UK, a Fellow of the Institution of Engineering and Technology (FIET), and a Fellow of the Institution of Mechanical Engineers (FIMechE).

LI WEI received the B.E. and M.E. degrees from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2004, and the Guilin University of Electronic Technology, in 2011, respectively. From 2004 to 2008, she was a Research Engineer of China Unicom. In 2011, she joined the College of Jincheng, Nanjing University of Aeronautics and Astronautics, as a Lecturer. Her current research interests include environment perception system of intelligent vehicle, as well as object detection and tracking.

YUN-FENG CAO received the B.E. and M.E. degrees from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 1985 and 1992, respectively. He then joined the College of Automation Engineering as a Lecturer and became an Associate Professor, in 1997. He joined the College of Astronautics, in 2003, where he became a Professor. He has published more than 100 articles in international journals and conferences. He used to be the Routine Director of the Jiangsu Association of Automation and the Vice Director of the Jiangsu Electro-Technical Society in Control Section.

ZHOU-YU ZHANG received the B.E. and M.E. degrees from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2014 and 2017, respectively, where he is currently pursuing the Ph.D. degree in guidance, navigation and control. His current research interests include sense and avoid, object detection, and object tracking.