Reliable Re-Detection for Long-Term Tracking

Ning Wang, Wengang Zhou, and Houqiang Li, *Senior Member, IEEE*

***Abstract*— In long-term object tracking, severe occlusion and deformation could happen to the targets. Due to the accumulation and propagation of estimation errors, even a few frames of full occlusion in a video sequence could lead to the failure of the tracking. Recently, correlation filter-based trackers have received lots of attention and gained great success in real-time tracking. However, most of them ignore the reliability of the tracked results and lack an effective mechanism to refine the unreliable results. To cope with these issues, in this paper, we propose a long-term tracking framework composed of both tracking-by-detection and re-detection modules. The tracking-by-detection part is built on the discriminative correlation filter (DCF) integrated with a color-based model. The re-detection module filters a large number of detection candidates and refines the tracking results. With the proposed re-detection refinement, detected results in each frame were re-evaluated and re-detection is carried out when necessary. Besides, the reliability estimation in the re-detection module also helps adaptively update the object detector and keep it from corruption. The proposed re-detection module can be integrated into correlation filter-based trackers to consistently boost the performance. Extensive experiments on the OTB-2015, Temple-Color, and VOT-2015 benchmarks show that the proposed method performs favorably against the state-of-the-art methods while still running faster than 40 f/s.**

***Index Terms*— Long-term tracking, tracking-by-detection, re-detection, feature combination.**

I. Introduction

OBJECT tracking is a fundamental task in computer vision. Although significant progress has been made in the past decades, there still remain many challenges [1]. In this article, we focus on long-term tracking, where the target may go through heavy occlusion, significant appearance changes and even moving out-of-view.

In recent years, tracking-by-detection has become one of the most successful paradigms for object tracking [2]–[4]. Following such a paradigm, a tracker identifies objects through a detector and updates it over time to compensate the appearance and scale changes. In spite of the state-of-the-art performance, existing tracking-by-detection approaches struggle when their detectors are misled by the corrupted results.

In a long-term tracking, when there appears heavy occlusion or significant appearance change, the result of the detection model may be unreliable and the model update is prone to the error propagation. Therefore, the reliability of the detection result must be carefully checked, which will guide the update of the detection model. Limited by few training data and prior knowledge, additional information is expected to enhance the tracker. The process of model update is of vital importance but suffers the risk of drifting to the background. Some algorithms incorporate the template in the first frame or prior knowledge for online updating [5], [6]. However, relying on a fixed model or prior restricts the detector's capability to handle appearance and scale changes. Recent part-based methods [7], [8] have been studied actively due to their robustness to appearance variations and partial occlusion. However, these methods have difficulty in handling full occlusion and objects with a large visually homogeneous region. Other algorithms try to enhance the robustness by combining multiple features but the impact of poor updates still remains [9], [10].

For long-term tracking, another common issue in a tracking-by-detection tracker is how to identify a better result when the current tracking result is unreliable, and thus an effective re-detection module is necessary. Besides, the re-detection technique is also highly related to the practical applications with increasing focus, such as person re-identification [11], [12] and multi-object tracking (MOT) [13]–[15]. However, many long-term trackers heavily depend on the detection model to avoid the training set contamination, and thus are incapable of re-detecting objects [16], [17]. Other algorithms with re-detection model mostly ignore the quality of re-detected objects, which may corrupt the detector further [18], [19]. The reliability of re-detected objects is crucial for the detection model to retain its discriminative power in the long-term tracking, while existing long-term trackers fail to fully explore such reliability problem and usually trust their re-detected targets just like a detector always trusts its own result. Furthermore, in many long-term tracking algorithms [17], [20], the incorporation of multiple trackers leads to an obvious reduction in tracking speed. Due to the efficiency requirement imposed by many practical applications, the capability of real-time processing is also essential for tracking algorithms.

As one of the state-of-the-art tracking-by-detection methods, Discriminative Correlation Filter (DCF) based trackers [4], [21] have gained sustained attention thanks to their impressive robustness and speed. In DCF based trackers, a region of interest (ROI) is usually cropped at the position of the target in the previous frame and with 2.5 times the size of the target itself. However, the detection range

Manuscript received April 21, 2017; revised September 30, 2017; accepted March 10, 2018. Date of publication March 16, 2018; date of current version March 7, 2019. The work of H. Li was supported in part by the 973 Program under Contract 2015CB351803 and in part by NSFC under Contract 61390514. The work of W. Zhou was supported in part by NSFC under Contract 61472378 and Contract 61632019, in part by the Fundamental Research Funds for the Central Universities, and in part by the Young Elite Scientists Sponsorship Program through CAST under Grant 2016QNRC001. This paper was recommended by Associate Editor S. Gong. *(Corresponding authors: Wengang Zhou; Houqiang Li.)*

The authors are with the CAS Key Laboratory of Technology in Geo-spatial Information Processing and Application System, Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230027, China (e-mail: wn6149@mail.ustc.edu.cn; zhwg@ustc.edu.cn; lihq@ustc.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2018.2816570

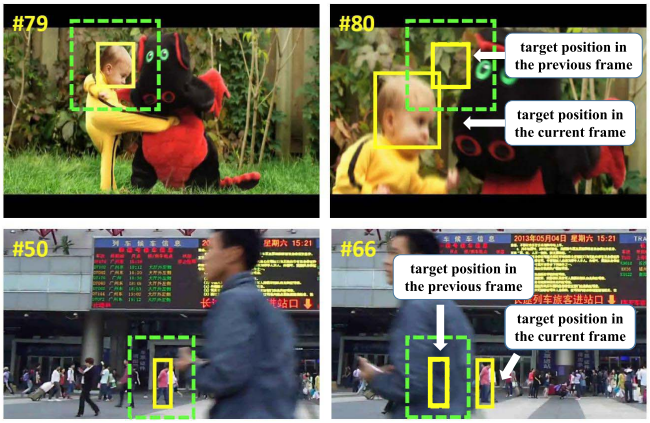

Fig. 1. Illustration of tracking failure in the videos of *DragonBaby* (top) and *Railwaystation* (bottom). Yellow and green boxes denote the groundtruth targets and the searching areas in DCF based trackers, respectively. Searching areas are usually located at the position of the target in the previous frame. The tracker produces low confidence levels in case of motion blur, heavy occlusion and out-of-view.

sometimes restricts the tracking capability of DCF based trackers (Fig. 1). By penalizing filter coefficients outside object boundaries, the recent SRDCF algorithm [22] increases the detection range and the performance at the same time, but the real-time capability is heavily restricted.

In this paper, to address the above issues, we are dedicated to the following problem: how to identify the unreliable tracking results to prevent the model update of the detector from corruption and how to re-detect the target when unreliable result appears. We propose a long-term tracker, which is composed of both tracking-by-detection and re-detection modules. Inspired by the excellent performance of Staple [23] which combines HOG [24] and color features and still leaves much room for improvement (Section IV-C), we take it as a baseline and make use of these features for re-detection to explore the potential of DCF based trackers. The core module of our approach is denoted as a re-detection switch, which estimates the occurrence of contamination with color and HOG features, and makes decision on whether it is necessary to perform re-detection and whether or not to replace the original target with the re-detected object. We apply an recall-dominated "unreliability check" criterion in the re-detection switch to bring potential contaminated detection results into the re-detection process. Further, we set a precision-dominated "reliability check" criterion in the re-detection switch to ensure the quality of the re-detected result before substituting it to the detected unreliable result. Based the measurement on the detection reliability, we adaptively update the detection model. The proposed approach is evaluated by extensive experiments and compared with the state-of-the-art algorithms on three large-scale benchmarks to demonstrate its favorable effectiveness. It is notable that, our framework is ready to be integrated into many DCF based algorithms to realize performance improvement (ranging from 1.5%-10.2% in distance precision and 3.1%-9.1% in overlap precision).

In the rest of the paper, related work is described in Section II. The proposed method is elaborated in Section III. After that, experiment and analysis are provided in Section IV. Finally, Section V concludes the paper.

II. Related Work

In this section, we discuss three categories of trackers closely related to our algorithm: tracking-by-detection, correlation tracking and long-term tracking.

A. Tracking-by-Detection

Tracking-by-detection methods [3], [25], [26] learn an initial discriminative model (*e.g.*, with a support vector machine (SVM) [27], [28]) to detect the target. It typically consists of two phases: tracking and learning. In the former phase, the discriminative model is used to predict the object location, while in the latter the new location is used to update the detector. As a typical representative, TLD [29] decomposes the tracking task into tracking, learning and detection. In TLD, tracking and detection run simultaneously and mutually update each other. Compressive tracking (CT) algorithm [2] extracts features through compressing sensing and constructs classifier using naive Bayes for target detection. Struck [3] uses a structured output formulation to learn and update the detector. The multi-entropy minimisation (MEEM) tracker [30] maintains a collection of snapshots and chooses the best prediction from them based on the framework of SVM. In recent years, tracking algorithm based on particle filter has been extensively studied and different methods have been incorporated to improve the robustness [25], [31]. Combinations of particle filter with sparse representation [6], [32], [33] or correlation filters [34] have demonstrated robust performance. Recently, Zhang *et al.* [35] propose a circulant sparse tracker with high efficiency. However, these tracking-by-detection methods focus on short-term tracking tasks and usually performs poorly in case of heavy occlusion and out-of-view.

B. Correlation Tracking

In DCF based trackers, a filter is trained through minimizing a least-squares loss for all circular shifts of a training sample. The target is tracked by correlating the filter over a region of interest, and the location with the maximum response indicates the new location of the target. Bolme *et al.* [21] propose a tracking algorithm using minimum output sum of squared error (MOSSE) filter. MOSSE tracker is computationally efficient with a speed of several hundred frames per second. Heriques *et al.* exploit the circulant structure of the training patches [36] and propose to use correlation filter in a kernel space with HOG features to achieve better performance [4]. Zhang *et al.* [37] propose a STC algorithm, which incorporates context information into filter learning. The DSST tracker [38] utilizes multi-scale correlation filters to handle scale changes of the object. The SRDCF tracker [22] alleviates the boundary effects by penalizing correlation filter coefficients depending on spatial location, which has demonstrated excellent performance. An improved version of SRDCF is SRDCFdecon [39], which reduces the influence of corrupted samples by computing a joint loss. Bertinetto *et al.* [23] propose a Staple tracker which combines DCF and color-based model to handle color changes and deformation while runs in real-time. Recently, several methods use learned Convolutional

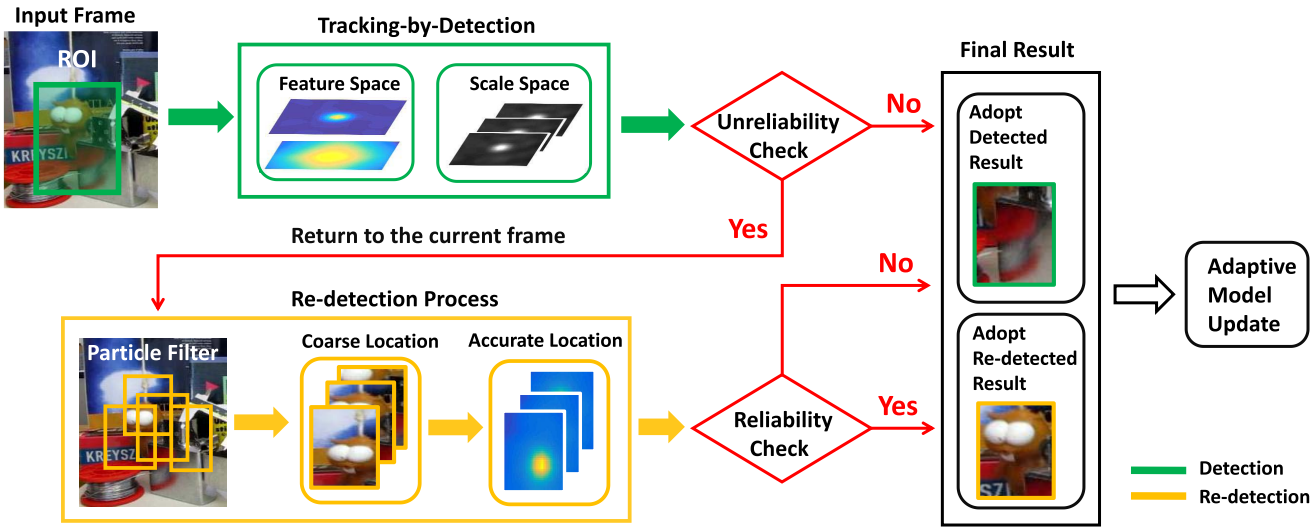

Fig. 2. A system flowchart of the proposed tracking algorithm. The proposed algorithm consists of tracking-by-detection and re-detection process, denoted by green arrow and orange arrow, respectively. After tracking-by-detection, an "unreliability check" criterion judges whether there needs a re-detection process. If the re-detection process is adopted, a "reliability check" criterion judges whether the re-detected result can replace the originally detected result.

Neural Network (CNN) features [40], [41] rather than hand-crafted features in correlation filters to boost the performance further, like HCF [42] and DeepSRDCF [43]. It is worth mentioning that HCF constructs multiple correlation filters on hierarchical convolutional layers to capture both spatial details and semantics, which demonstrates high accuracy. Different from these DCF based methods, our framework focuses on the reliability estimation of the results and re-detection refinement.

C. Long-Term Tracking

To handle the challenging factors in long-term tracking, a self-paced learning scheme [16] is combined to the tracker in which the target appearance can be learned by selecting trustworthy frames to avoid corrupting the training set. The multi-store tracker (MUSTer) [20] maintains a short-term memory for detecting and a long-term memory for outputing control. In [17], a tracking-by-detection approach with occlusion and motion reasoning is proposed to handle occlusion and motion changes. However, both MUSTer [20] and [17] integrate multiple trackers and need to evaluate them in every frame. Long-term correlation tracker (LCT) [18] uses DCF for detection as well as a random forest for re-detection and achieves state-of-the-art results. Different from previous re-detection approaches, our method makes full use of baseline tracker for re-detection and does not need to incorporate extra features or trackers. We aim to explore the potential of DCF based trackers by designing a better searching strategy. In long-term tracking, besides re-detection, another main task is identifying the state (*e*.g., partial or full occlusion) of the object as well as avoiding confusing updates. In [44], Wang *et al.* propose an algorithm to track interacting and invisible targets using intertwined flows. MUSTer predicts the state of the target by counting the number of SIFT keypoints [45]; LCT considers the maximum value of the correlation response map to judge occlusion.

Different from algorithms mentioned above: (i) we check the reliability of the target and update tracking model adaptively through multiple features; (ii) through an "unreliability and reliability check" criterion, we can generate reliable tracking result; (iii) after a coarse localization, we add a dense search additionally to achieve an accurate re-detection.

III. Our Proposed Approach

In order to handle severe occlusion and significant appearance changes in long-term tracking tasks, our approach is composed of tracking-by-detection and re-detection modules. The general framework of our method is depicted in Fig. 2.

A. Tracking by Detection

To track the object by detection, we make use of the DCF model [4], [21] and the color histogram model [23], [46] to generate the response maps, respectively. After that, we linearly integrate the two response maps and identify the object location with the maximal response value.

1) Correlation Filter Response Generation: A typical correlation tracker [4], [21] learns a discriminative classifier and locates the target by searching for the maximum value of correlation response map. The tracker based on correlation filter is trained using an image patch $\mathbf{x}$ of size $m \times n$. The training image is centered around the target. All the circular shifts of the patch $\mathbf{x}_{m,n}(m,n) \in \{0,1,\ldots,M-1\} \times \{0,1,\ldots,N-1\}$ are generated as training samples with Gaussian function label $y(m,n)$ in terms of the shifted distance. Since a large number of negative samples are employed by shifting the image patch $\mathbf{x}_{m,n}$, the discriminative power of the classifier is greatly enhanced. The classifier $\mathbf{w}$ is trained by minimizing the regression error:

$$\min_{\mathbf{w}} \sum_{m,n} \|\phi(\mathbf{x}_{m,n}) \cdot \mathbf{w} - y(m,n)\|^2 + \lambda \|\mathbf{w}\|^2, \qquad (1)$$

where ϕ represents the mapping to a Hilbert space and λ is a regularization parameter ($\lambda \geq 0$). By employing a kernel $k(\mathbf{x}, \mathbf{x}^{'}) = \langle \phi(\mathbf{x}), \phi(\mathbf{x}^{'}) \rangle$, the solution can be expressed as $\mathbf{w} = \sum_{m,n} \alpha(m,n) k(\mathbf{x}_{m,n}, \mathbf{x})$, where α is the dual variable of $\mathbf{w}$ and it is defined by

$$\hat{\alpha}^* = \frac{\hat{\mathbf{y}}}{\hat{\mathbf{k}}^{\mathbf{xx}} + \lambda}, \tag{2}$$

where $\mathbf{k}^{\mathbf{xx}}$ denotes the vector whose i-th element is $k(\mathbf{x}_i, \mathbf{x})$, the hat symbol denotes the Discrete Fourier Transform (DFT) of a vector (*e*.g., $\hat{\alpha} = \mathcal{F}(\alpha)$) and $\hat{\alpha}^*$ is the complex-conjugate of $\hat{\alpha}$. The simplest linear kernel function is applied to our algorithm, $k(\mathbf{x}, \mathbf{x}^{'}) = g(\mathbf{x}, \mathbf{x}^{'})$ for some function g, and $\mathbf{k}^{\mathbf{xx}^{'}} = g(\mathcal{F}^{-1}(\hat{\mathbf{x}} \odot \hat{\mathbf{x}}^{'*}))$.

In the tracking process, a patch $\mathbf{z}$ with the same size of $\mathbf{x}$ is cropped out in the new frame. The response map of $\mathbf{z}$ is calculated as follows,

$$f_h = \mathcal{F}^{-1}((\hat{\mathbf{k}}^{\mathbf{xz}})^* \odot \hat{\alpha}), \tag{3}$$

where $\odot$ is the element-wise product and $\mathbf{x}$ is the learned target appearance. To avoid the boundary effects during learning, we apply Hann window to the signals [4]. The online update is made as follows,

$$\begin{aligned} \hat{\mathbf{x}}_t &= (1 - \eta_h)\hat{\mathbf{x}}_{t-1} + \eta_h \hat{\mathbf{x}}_t^{'}, \\ \hat{\alpha}_t &= (1 - \eta_h)\hat{\alpha}_{t-1} + \eta_h \hat{\alpha}_t^{'}, \end{aligned} \tag{4}$$

where η_h is the learning rate of HOG-based correlation filter and t is the index of the current frame. To avoid the contamination of the trained filter, η_h in our framework is set adaptively, which will be illustrated latter (Section III-D). We train two models based on correlation filters from one single frame. One is used for translation while the other is used for scale estimation [38].

2) Color Histogram Response Generation: To handle deformation and color changes in long-term tracking, we adopt a color histogram model [23], [46] in the detector. For a single image, the color histogram model should be learnt from a set of rectangular patches $\mathbf{x}$ with their corresponding labels y, including the correct position as a positive sample. Similar to correlation filter, the histogram weight vector $\mathbf{m}$ can be trained by minimizing the regression error:

$$\min_{\mathbf{m}} \sum_{\mathbf{x}} \left\| \sum_{u \in \mathcal{R}} \mathbf{m}^T \varphi_{\mathbf{x}}(u) - y \right\|^2 + \lambda \|\mathbf{m}\|^2, \tag{5}$$

where $\mathbf{m}$ is the histogram weight vector and $\varphi_{\mathbf{x}}(u)$ represents the feature pixels of patch $\mathbf{x}$ in finite region $\mathcal{R}$. Although Eq. (5) has a similar expression form with Eq. (1), the histogram weight vector $\mathbf{m}$ cannot be learnt with circular shifts. For an M-channel feature transform φ, the computation cost will be unbearable with the increasing number of feature channels.

Since the histogram score can be considered an average vote, a linear regression can be applied to each pixel independently over object ($\mathcal{O}$) and its surrounding ($\mathcal{S}$) regions [23]. Therefore, instead of taking Eq. (5), per-image loss based on color histogram can be computed as follows,

$$\min_{\mathbf{m}} \frac{1}{|\mathcal{O}|} \sum_{u \in \mathcal{O}} \left| \mathbf{m}^T \varphi(u) - 1 \right|^2 + \frac{1}{|\mathcal{S}|} \sum_{u \in \mathcal{S}} \left| \mathbf{m}^T \varphi(u) \right|^2. \tag{6}$$

The solution of the ridge regression problem above is:

$$m^j = \frac{p^j(\mathcal{O})}{p^j(\mathcal{O}) + p^j(\mathcal{S}) + \lambda}. \tag{7}$$

For each dimension $j = 1, \ldots, M$, $p^j(\mathcal{R})$ is the j-th element of the vector $\mathbf{p}$. It represents the proportion of pixels in the region $\mathcal{R}$ for which feature j is non-zero. After obtaining the histogram weight vector, for a given image patch $\mathbf{z}$, we can compute the color per-pixel score map through $\mathbf{m}$: $f_c = \mathbf{m}^T \varphi_{u \in \mathbf{z}}(u)$, and obtain the dense histogram response map using an integral image [23]. The online update for color histogram model is as follows,

$$\begin{aligned} \mathbf{p}_t(\mathcal{O}) &= (1 - \eta_c)\mathbf{p}_{t-1}(\mathcal{O}) + \eta_c \mathbf{p}_t^{'}(\mathcal{O}), \\ \mathbf{p}_t(\mathcal{S}) &= (1 - \eta_c)\mathbf{p}_{t-1}(\mathcal{S}) + \eta_c \mathbf{p}_t^{'}(\mathcal{S}), \end{aligned} \tag{8}$$

where η_c is the learning rate of color histogram model which is also combined with our adaptive update strategy, $\mathbf{p}_t(\mathcal{R})$ is the vector of $p_t^j(\mathcal{R})$ for $j = 1, \ldots, M$.

3) Target Localization: Both HOG-based correlation filter model and color histogram model are capable of detecting the object, and the combination of them in a dense translation search enables greater robustness [23]. For each single frame, after computing the correlation filter response map and color histogram response map, the final response map is a linear combination of them:

$$f^{(i)} = \zeta \cdot f_h^{(i)} + (1 - \zeta) \cdot f_c^{(i)}, \tag{9}$$

where $f_h^{(i)}$ is the response map of i-th frame calculated by HOG-based correlation filter, $f_c^{(i)}$ is the response map calculated by color histogram and ζ is the weighting factor. The position of the target in a coming frame is detected by searching for the location with the maximal value of $f^{(i)}$.

B. Reliability Estimation

In the re-detection module, we first discuss how to utilize the responses of HOG and color features to estimate the reliability of the tracking results. Then, we propose a loose "unreliability check" and a strict "reliability check" to generate the final tracking result in the current frame.

For HOG-based correlation filter response map, the peak-to-sidelobe ratio (PSR) can be computed to quantify the sharpness of the correlation peak [21]. If the PSR value is low, the correlation between the current frame and previous frames is low. We define the PSR of correlation filter response map as the score of HOG features:

$$S_h^{(i)} = \frac{max(f_h^{(i)}) - \mu_i}{\sigma_i}, \tag{10}$$

where $f_h^{(i)}$ is the i-th response map of HOG-based correlation filter, μ_i and σ_i are the mean and standard deviation of the response, respectively.

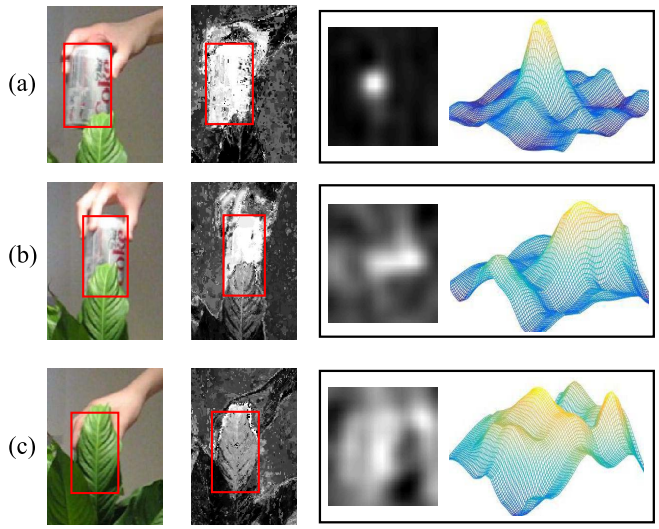

Fig. 3. From left to right: input image, per-pixel score map based on color histogram, HOG-based correlation response map (with a 3D visualization). The red boxes denote the ground truth positions of the target. In (b), when object is partially occluded, the color "area" percentage compared to the ground truth target is lower as well as the PSR of the correlation response map. In (c), when object is under full occlusion, the color score and HOG score decrease significantly.

For a certain correlation filter, the PSR value fluctuates in a certain extent and is much lower when unreliable tracking result occurs. However, it is not suitable to pre-define a constant threshold to judge the reliability of the current tracking. Due to the uncertainty of tracking difficulty, the HOG scores of the response maps may fluctuate at different values. For example, when the PSR is lower than a certain threshold, it means a failure in video $\mathcal{A}$, while it may indicate a success in another video $\mathcal{B}$ due to much more challenging factors contained in $\mathcal{B}$. So we consider the average score of the video to estimate the reliability of the tracking results. We calculate each frame's HOG score and combine them into an ensemble $\mathcal{C}_h = \{S_h^{(2)}, S_h^{(3)}, \ldots, S_h^{(i)}\}$. We define the M_h as the mean of the ensemble $\mathcal{C}_h$. Moreover, we do not accept every S_h into the ensemble through defining a small coefficient o_h for M_h. If $S_h^{(i)} < o_h \cdot M_h$, we consider the result of i-th frame performs poorly and discard the corresponding HOG score. By considering the average score of the ensemble, the reliability criterion of the HOG score is changed adaptively frame-by-frame. A significant low S_h which satisfies $S_h^{(i)} < o_h \cdot M_h$ usually means a certain extent occlusion or deformation.

For color information, we calculate per-pixel score map based on color histogram in the first frame and sum all the pixels inside the target area to obtain a color "area" (Fig. 3). As there exists no occlusion for the object in the first frame, the per-pixel score map is pure and take it as the comparison criterion. Then for each single frame, we take the same step. We define the percentage of color "area" as color score, which is simple yet effective in practice:

$$S_c^{(i)} = \frac{\sum_u \mathbf{m}^T \varphi_i(u)}{\sum_u \mathbf{m}^T \varphi_1(u)}. \tag{11}$$

In Eq. (11), $S_c^{(i)}$ is the color score of i-th frame and denominator denotes the color "area" of the groundtruth frame. However, in some videos, there is more background in the bounding box and color score S_c differs from video to video. Similar to the HOG score, we calculate each frame's color score and put them into an ensemble $\mathcal{C}_c = \{1, S_c^{(2)}, S_c^{(3)}, \ldots, S_c^{(i)}\}$. We define M_c as the mean of the ensemble $\mathcal{C}_c$. By defining a small coefficient o_c for M_c, we discard the significant low color score which satisfies $S_c^{(i)} < o_c \cdot M_c$.

We check the reliability of the tracking result in each frame. A tracking result is regarded as unreliable, if the targets HOG score $S_h < o_h \cdot M_h$ or the targets color score $S_c < o_c \cdot M_c$ ("Unreliability Check" in Fig. 2). On the other hand, a tracking result is likely to be very reliable if it satisfies $S_h > \tau_h \cdot M_h$ and $S_c > \tau_c \cdot M_c$ ("Reliability Check" in Fig. 2), where τ is a high threshold compared with o. We launch the re-detection process once the initial tracking result is found to be unreliable. If the output of the re-detection is reliable, we substitute it to the original unreliable detection target. Otherwise, we keep the original one but reduce the learning rate for the model update. An illustration of our strategy is shown in Fig. 4.

C. Re-Detection Module

In this section, we introduce how to re-detect the object in our algorithm. We first coarsely locate the target with efficiency based on a sparse coding scheme. Then, an accurate localization approach is applied to refine the target location by re-utilizing tracking-by-detection model. Finally, the reliability estimation method described in Section III-B helps switch the suitable result between tracking-by-detection and re-detection.

To obtain reliable searching areas from a large amount of ROI candidates, a simple sparse coding based method is adopted to find a coarse location efficiently. In the tracking process, we collect a template set $\mathcal{D}$ contains N_p positive templates $\mathcal{D}_+$ near the object (*e.g.,* within a radius of a few pixels) and N_n negative templates $\mathcal{D}_-$ far away from the object. If the current tracking-by-detection model results in a bad solution (satisfies "unreliability check" criterion), then we draw N candidates around the tracked result in the previous frame using particle filter. For each candidate x, it is represented by template set $\mathcal{D} = [\mathcal{D}_+ \ \mathcal{D}_-]$ with coefficients $\alpha = [\alpha_+ \ \alpha_-]$ which is obtained by Eq. (12).

$$\min_{\alpha} \|x - \mathcal{D}\alpha\|_2^2 + \lambda \|\alpha\|_1. \tag{12}$$

A candidate with smaller reconstruction error using positive template set $\mathcal{D}_+$ is more likely to be a target and vice versa. Thus, by computing the reconstruction error of each candidate using template $\mathcal{D}$, we can predict the reliability R_i of the i-th candidate roughly:

$$R_i = \|x_i - \mathcal{D}_-\alpha_-\|_2^2 - \|x_i - \mathcal{D}_+\alpha_+\|_2^2, \tag{13}$$

where $\|x_i - \mathcal{D}_-\alpha_-\|_2^2$ is the reconstruction error using negative template set $\mathcal{D}_-$ and α_- is the corresponding coefficient vector, and $\|x_i - \mathcal{D}_+\alpha_+\|_2^2$ is computed in a similar way. For the i-th candidate, higher R_i means its higher possibility of a target. Although this method is not robust enough to re-detect

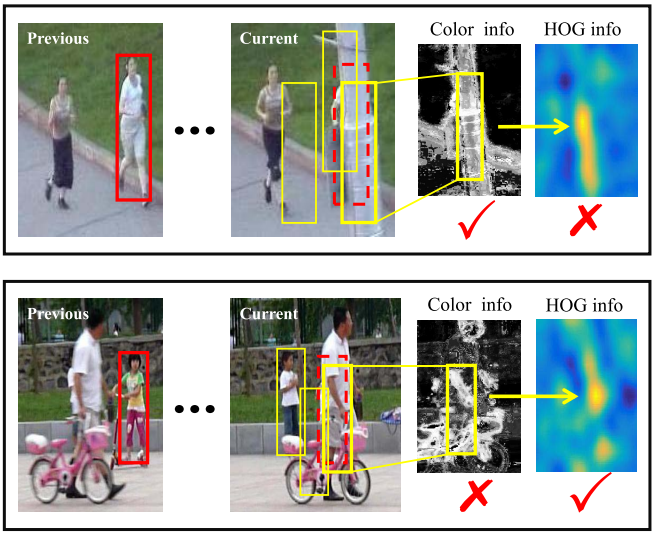

Fig. 4. In the video *Jogging* (top), when the target (shown in red ground truth box) is under occlusion, target re-detection (shown in yellow boxes) tries to locate the true object, while some of them perform poorly in either HOG or color information, others perform badly in both HOG and color information. Video *Girl2* (bottom) performs in a similar way. Our combination of HOG and color features in "reliability check" ensures the reliability of re-detection.

the target, it provides holistic information of the target and we can discard many useless candidates for efficiency. In our algorithm, we discard 90% of the candidates through reconstruction error, which predicts the location roughly and reduces much computational cost in accurate location process. The rest candidates will be exploited by particle filter for accurate target localization as discussed in the following.

Candidate selection through particle filter can be regarded as a discrete sampling process while the response map computed by correlation filter and color histogram model searches location densely. Thus, for the selected candidates, a bit larger region (ROI) is cropped to search for an accurate location. Finally, we compute the combined response map using Eq. (9) for the selected candidates and the final confidence score is defined as follows,

$$C_i = max(f^{(i)}) \cdot cos\left(\frac{\gamma}{W_t + H_t}||L_c^{(i)} - L_t||\right), \quad (14)$$

where $f^{(i)}$ is the i-th candidate's combined response map, the second term is the distance score based on Euclidean distance, W_t and H_t are the width and height of the target, respectively, $L_c^{(i)}$ and L_t are the locations of i-th candidate and the target in the last frame, the γ is a pre-defined distance penalization parameter. We assume that a target with larger size has a larger motion range, so the distance penalization is closely related to the scale of the target.

The confidence score C_i in Eq. (14) considers multiple target features as well as distance penalization, which can help choose the best re-detected candidate. Similar to the tracking-by-detection part, the position of the re-detected target is determined by searching the maximal value of the best candidate's response map. Finally, if the re-detected target satisfies the "reliability check" criterion discussed in Section III-B, we substitute it to the original detection result. Otherwise, we keep the original detection result and update the model adaptively.

D. Adaptive Template Update

With the reliability criterion, our tracking model is updated using the reliable results. When the tracked target is regarded as unreliable, the learning rate of color histogram model η_c is set to zero to avoid the update of the sudden color changes introduced by occlusion, illumination change or out-of-view. Considering the correlation filter learns both target and background information, we take full advantage of it by a designed power function [47].

$$\eta_c = \begin{cases} P & \text{if satisfy "reliability check",} \\ 0 & \text{otherwise,} \end{cases} \quad (15)$$

$$\eta_h = \begin{cases} Q & \text{if satisfy "reliability check",} \\ \nu(S_h/M_h)^{\beta} Q & \text{otherwise,} \end{cases} \quad (16)$$

where P and Q are constants, β is the power exponent of the power function, $\nu \in [0, 1]$ is a penalization coefficient which restricts the maximum value of the learning rate. The designed power function maintains reliable samples and penalizes samples with low scores severely.

As for positive and negative templates in coarse localization process, When the current result satisfies the reliability criterion, both of them are updated through particle filter to adapt appearance changes of both target and background. The outline of our method is summarized in Algorithm 1.

IV. Experiments

A. Experiment Settings

In our experiment, the regularization parameter λ is set to 10^{-3}. Following Staple [23], the weighting factor ζ in Eq. (9) is set to 0.3. In template set $\mathcal{D}$ defined in Eq. (12), N_p and N_n are set to 50 and 200, respectively. The positive templates, negative templates and particles are all normalized to the same size (20×20). For efficiency, only 50 particles are used when re-detection model works and discard 90% of them after coarse localization. In distance score defined in Eq. (14), γ is set to $\pi/9$. As for learning rates, P and Q in Eq. (15) and Eq. (16) are set to 0.01 and 0.02, respectively. In Eq. (16), penalization coefficient ν is set to 0.8 and power exponent β is set to 3.

As for the low threshold o_h in "unreliability check", a higher o_h will result in more frequent recall of the re-detection model which will lead to low efficiency, while a lower o_h means more ignorance to the potential occlusion which may lead to low accuracy. As for the high threshold τ_h in "reliability check", a higher τ_h means the re-detected true target may be ignored due to a too strict criterion while a lower τ_h means the wrong candidate may corrupt the detector due to a too loose criterion. The corresponding thresholds of the color based confidence score perform in a similar way. To make trade-off between accuracy and efficiency, "unreliability check" thresholds are $o_h = 0.6$, $o_c = 0.7$ and "reliability check" thresholds are $\tau_h = 0.7$, $\tau_c = 0.8$.

Algorithm 1: Proposed Tracking Algorithm

Input: Image I_0, previous target state $\mathbf{x}_0$,
Output: Estimated object state $\mathbf{x}_t = (x_t, y_t)$, tracking-by-detection model, template set.

for $t = 2, 3, \ldots, n$ **do**
  //Tracking-by-detection;
  Identify the searching window in frame t;
  Compute the correlation response map and color response map;
  Estimate new state $\mathbf{x}_t$ using Eq. (9) ;
  //Re-detection;
  Compute $S_h^{(t)}$ and $S_c^{(t)}$ of the current object;
  //Unreliability check;
  if $(S_h^{(t)} < o_h \cdot M_h)$ *OR* $(S_c^{(t)} < o_c \cdot M_c)$ **then**
    Draw candidates in the previous target position through particle filter;
    Discard useless particles using Eq. (13);
    Compute candidates' confidences using Eq. (14) ;
    Choose the best candidate c_i and its state is $\mathbf{x}_i$;
    Compute $S_h^{(i)}$, $S_c^{(i)}$ and distance score d_i of c_i;
    //Reliability check;
    if $(S_h^{(i)} \cdot d_i > \tau_h \cdot M_h)$ *AND* $(S_c^{(i)} \cdot d_i > \tau_c \cdot M_c)$ **then**
      //Replace original target ;
      $\mathbf{x}_t = \mathbf{x}_i$;
    end
  end
  if $(S_h^{(t)} > \tau_h \cdot M_h)$ *AND* $(S_c^{(t)} > \tau_c \cdot M_c)$ **then**
    Update detector using Eq. (4) and Eq. (8) ;
    Update positive and negative templates;
  else
    Reduce η_h and η_c using Eq. (15) and Eq. (16) ;
  end
end

We use the same setting of parameters for all sequences on OTB-2015 [48] and Temple-Color [49]. Our implementation runs at about 45 frames per second (FPS) on a computer with an Intel I7-4790K 4.00GHz CPU, 16GB RAM. Our source code is available at: https://github.com/594422814/Reliable-Re-detection-for-Long-term-Tracking.git.

We evaluate the proposed algorithm on OTB-2015 [48] and Temple-Color [49] benchmarks with comparisons with 12 recent state-of-the-art trackers: TLD [29], Struck [3], KCF [4], DSST [38], SAMF [10], MEEM [30], LCT [18], HCF [42], SRDCF [22], SCT4 [50], Staple [23] and SiamFc [51]. Besides, we also test our method on the VOT-2015 dataset [52].

OTB-2015 contains 100 videos and Temple-Color consists of 128 videos. All the tracking methods are evaluated by three metrics: distance precision (DP) at a threshold of 20 pixels, overlap precision (OP) at an overlap threshold 0.5 and center location error (CLE). For better performance measure, we use overlap success plots over OTB-2015 and Temple-Color datasets using one-pass evaluation (OPE) proposed in [53].

TABLE I

THE PERFORMANCE OF OUR ALGORITHM WITH DIFFERENT SETTINGS MEASURED USING MEAN DISTANCE PRECISION (DP) (%) AT A THRESHOLD OF 20 PIXELS AND OVERLAP PRECISION (OP) (%) AT AN OVERLAP THRESHOLD 0.5 ON THE OTB-2015 DATASET

	DP (%)	OP (%)
Baseline tracker (Staple [23])	78.3	70.2
Adaptive update without re-detection	79.0	71.8
No feature for re-detection	36.8	31.1
Only Color features for re-detection	79.3	72.8
Only HOG features for re-detection	80.0	72.5
HOG + Color features for re-detection	**83.2**	**76.1**

TABLE II

THE PERFORMANCE OF OUR ALGORITHM USING DIFFERENT NUMBER OF ROI FOR RE-DETECTION. ZERO REPRESENTS ONLY ADAPTIVE UPDATE WITHOUT RE-DETECTION

number of ROI	0	1	3	5	10	20
OP (OTB-2015)	71.8	74.9	75.4	76.1	76.1	77.2
AUC (OTB-2015)	60.4	62.6	63.2	63.7	63.7	64.1
Mean speed (FPS)	67.6	46.8	45.3	44.6	40.6	35.2

B. Evaluation on Re-Detection

To justify the effectiveness of our re-detection framework, we have studied different settings: only color features and only HOG features in the "unreliability and reliability check" in our framework as well as a tracker only updating detector adaptively without re-detection module. We show the results of different versions in Table I. Only a detection model with adaptive update can be seen as an enhanced version of Staple [23], which obtains a good performance but fails to re-detect the target. No feature for re-detection means our target reliability estimation is not applied, which just runs the detector in multiple searching areas and selects the best among them. It performs the worst because the tracker tends to drift to similar objects and background clutters. Our "unreliability and reliability check" criterion can not only enhance the efficiency but also control the quality of re-detection. Only using color or HOG information to estimate the reliability of the target is not robust enough and the re-detected objects may contaminate the detector further. Significant promotion can be achieved by combining both HOG and color features for re-detection.

Table II shows the influence of the number of ROI in re-detection process. The involvement of multiple searching areas in re-detection means the trained detector is re-utilized multiple times, which will reduce the efficiency. However, it is interesting that even only single ROI is adopted in re-detection still achieves an obvious improvement compared to no re-detection, which is attributed to the effectiveness of sparse representation based method for ROI selection and our reliability estimation for output controlling. To obtain high efficiency, we choose 5 searching areas although more ROIs in re-detection provide further improvement.

C. Integration Into Different DCF Trackers

To cope with the limited size of ROI in DCF based trackers, an intuitive idea is to enlarge the searching area. In Table III, the ROIs of KCF [4], SAMF [10] and Staple [23] are set to

TABLE III

THE TRACKING ACCURACY OF DIFFERENT DCF BASED TRACKERS WITH DIFFERENT SETTINGS OF ROI SIZE. WE TAKE THE EVALUATION METRIC OF OVERLAP PRECISION (OP) (%) AT AN OVERLAP THRESHOLD 0.5 AND REPORT RESULTS ON THE OTB-2015 DATASET [48]. THE PERFORMANCES OF THE STANDARD ALGORITHMS ARE HIGHLIGHTED BY UNDERLINE

ROI size (× target size)	2×	2.5×	3×	3.5×
KCF [4]	49.1	55.3	55.0	51.1
SAMF [10]	61.5	65.3	65.6	62.1
Staple [23]	70.2	67.0	62.1	61.5

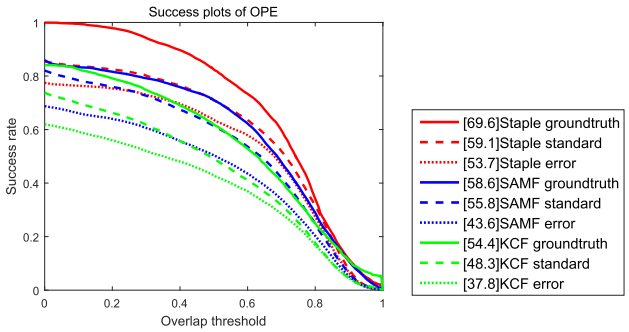

Fig. 5. Overlap success plots of different DCF based trackers when the locations of ROI are obtained from the groundtruth, the standard method (target position in the previous frame) and standard method with small perturbation error. For the Staple tracker, the above three settings are denoted by Staple groundtruth, Staple standard, and Staple error, respectively. Other trackers with those settings are denoted in a similar way. The legend illustrates the area-under-curve (AUC) score for each tracker.

different sizes. From the results, we can observe that ROI of a suitable size performs the best and a too large ROI will reduce the discriminative power of the tracker and results in worse performance. In other words, it is useless to simply enlarge the search area on the image plane. Besides, another experiment is conducted to study the impact of the ROI position (Fig. 5). In this experiment, the positions of ROIs are obtained from the groundtruth, the standard method (target position in the previous frame) and standard method with minor random perturbation (5 pixels in our experiment), respectively. As shown in Fig. 5, impressive performance is achieved by using the optimal ROI obtained from groundtruth while by perturbing its location with a minor drift error, the performance drops significantly, which illustrates the importance of the ROI position accuracy. These observations justify the practicability of improving the performance of DCF based trackers through selecting the best tracking result from multiple searching areas.

As discussed above, many DCF based trackers have much room for improvement by selecting a better ROI. We validate our re-detection framework by integrating it into 8 different trackers: MOSSE [21], CSK [36], KCF [4], SAMF [10], Staple [23], LCT [18], HCF [42] and improved HCF with scale estimation (denoted as "HCFs"). Although HOG and color features are used for reliability estimation and adaptive update, we do not integrate these features to the baseline trackers and just make full use of them for re-detection. From Table IV and Fig. 6, we can observe that after incorporating

TABLE IV

A COMPARISON OF DIFFERENT TRACKERS AND THEIR COMBINATION WITH OUR RE-DETECTION FRAMEWORK (LABELLED WITH "+") ON THE OTB-2015 DATASET

	MOSSE	MOSSE+	↑	CSK	CSK+	↑
DP	41.6	51.8	**10.2**	54.3	60.4	**6.1**
OP	33.4	42.5	**9.1**	45.1	50.5	**5.4**
	KCF	KCF+	↑	SAMF	SAMF+	↑
DP	69.6	74.6	**5.0**	72.9	75.9	**3.0**
OP	55.3	60.2	**4.9**	65.3	68.6	**3.3**
	Staple	Staple+	↑	HCF	HCF+	↑
DP	78.1	83.2	**4.9**	83.7	85.2	**1.5**
OP	70.2	76.1	**5.9**	65.4	68.5	**3.1**

our re-detection framework, the basic DCF based tracker MOSSE gains a significant improvement (10.2% in DP and 9.1% in OP) and state-of-the-art tracker Staple still obtains an obvious improvement (4.9% in DP and 5.9% in OP). Although DCF with deep features (*e.g.,* HCF) shows better robustness in occlusion and deformation compared to the hand-crafted features based trackers, it still suffers from restricted search range, imperfect ROI and corrupted training samples, and our framework improves HCF further (1.5% in DP and 3.1% in OP). Specially, HCF combined with scale estimation and our re-detection framework has shown state-of-the-art performance (67.0% AUC in OTB-2015).

LCT is one of the latest long-term tracking algorithms equipped with re-detection. However, it is more effective by simply re-utilizing its baseline detector for re-detection using our framework than its additional re-detector (Fig. 6).

D. Evaluation on OTB-2015

After self-evaluation, a comparison with other state-of-the-art trackers on OTB-2015 [48] and Temple-Color [49] datasets is shown in Table V. In the OTB-2015 dataset, HCF [42] achieves the best result in mean DP and CLE while our algorithm performs the best in mean OP as well as area-under-curve (AUC) of overlap precision plot (Fig. 7).

Currently, several deep tracking methods [54]–[57] have been developed to utilize features learned from CNN. HCF [42] adopts the VGG-Net-19 [41] trained on ImageNet for feature extraction and SiamFc [51] detects object through a pre-trained fully-convolutional siamese network. However, the feature learning process using CNN is a bit time-consuming. Deep features usually capture more semantic information with less spatial resolutions. Hand-crafted features are typically used to capture low-level information (*e*.g., color, shape, texture) which is more suitable for predicting the state of the target in re-detection. Our algorithm just extracts hand-crafted appearance features frame-by-frame and decreases the impact of corrupted samples through a re-detection scheme with low time cost.

E. Evaluation on Temple-Color

In Temple-Color dataset [49], our algorithm outperforms the state-of-the-art tracking algorithms in various evaluation criterion (Table V and Fig. 7). Among the compared tracking

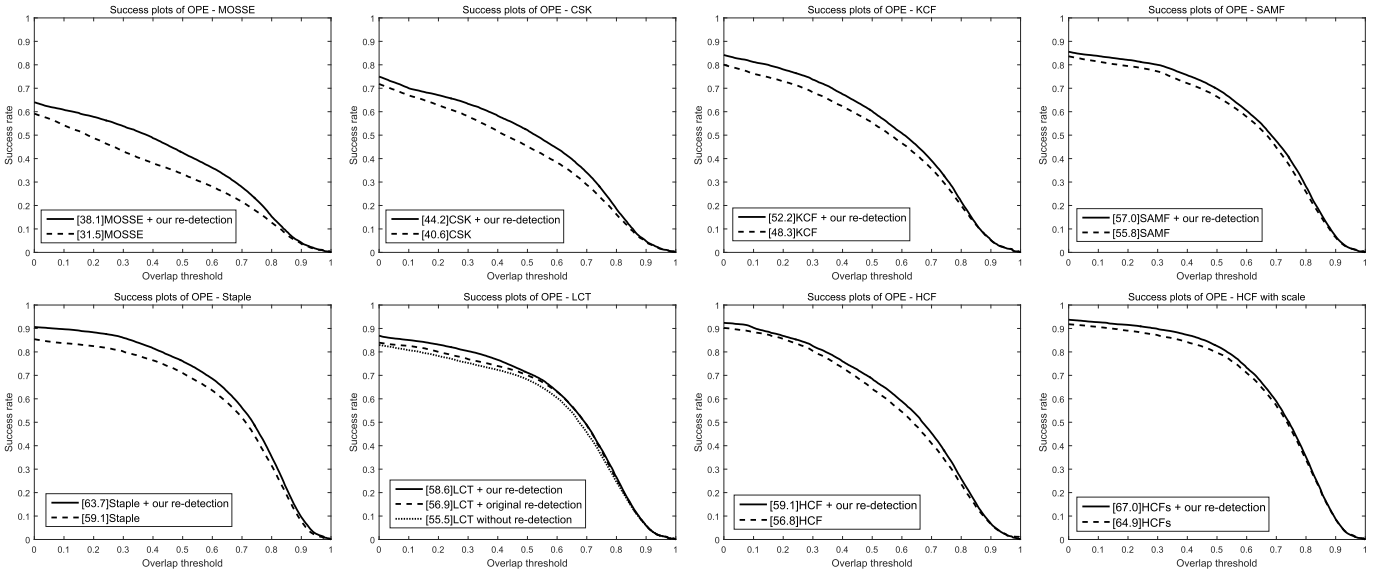

Fig. 6. Overlap success plots of DCF based trackers and their enhanced versions using our re-detection framework over OTB-2015 benchmark. The legend illustrates the area-under-curve (AUC) score for each tracker.

TABLE V

A COMPARISON OF OUR APPROACH, USING MEAN DISTANCE PRECISION (DP) (%) AT A THRESHOLD OF 20 PIXELS, OVERLAP PRECISION (OP) (%) AT AN OVERLAP THRESHOLD 0.5 AS WELL AS CENTER LOCATION ERROR (CLE)(PIXELS), WITH THE STATE-OF-THE-ART TRACKERS ON THE OTB-2015 AND TEMPLE-COLOR DATASETS. THE FIRST AND SECOND HIGHEST VALUES ARE HIGHLIGHTED BY BOLD AND UNDERLINE

	TLD [29]	Struck [3]	KCF [4]	DSST [38]	SAMF [10]	MEEM [30]	LCT [18]	HCF [42]	SRDCF [22]	SCT4 [50]	Staple [23]	SiamFc [51]	Ours
OTB-2015 OP	49.9	51.2	55.3	61.6	65.3	62.5	70.1	65.4	72.6	63.0	70.2	73.0	**76.1**
OTB-2015 DP	59.3	63.5	69.6	68.9	72.9	76.8	76.2	**83.7**	78.9	76.4	78.3	77.0	83.2
Mean CLE	61.0	47.6	44.7	48.8	36.5	30.1	66.8	**22.4**	38.9	37.6	30.9	32.9	23.2
Temple-Color OP	38.4	44.0	45.9	47.0	56.8	60.6	55.2	56.9	61.2	55.7	62.5	63.5	**67.2**
Temple-Color DP	45.8	52.2	54.9	53.4	61.8	69.9	59.7	68.9	66.4	62.7	67.1	69.2	**72.0**
Mean CLE	179.8	75.8	78.3	79.9	55.3	43.2	68.8	42.7	61.6	55.6	55.8	54.1	**42.2**
Mean speed (FPS)	23	10	**245**	32	24	21	28	6	8	44	70	55	45

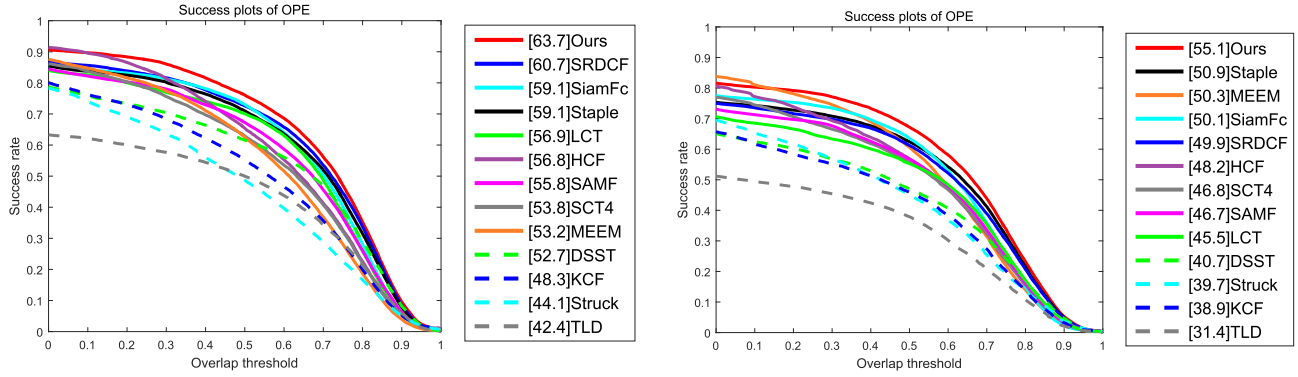

Fig. 7. Overlap success plots over OTB-2015 benchmark (left) with 100 videos and Temple-Color benchmark (right) with 128 color videos. The legend illustrates the area-under-curve (AUC) score for each tracker. Our algorithm improves the baseline (Staple [23]) obviously and maintains high speed.

methods, Staple [23], SiamFc [51] and SRDCF [22] provide the best results with AUC scores of 50.9%, 50.1% and 49.9%, respectively. Our algorithm achieves the best result with an AUC score of 55.1%. One reason of our excellent performance is the utilization of color features in both detection and re-detection models, which enhances the robustness in color videos greatly. Another reason is about 30% of the videos in OTB-2015 dataset are gray while all the 128 videos in Temple-Color benchmark are color with much more challenging factors.

F. Evaluation on VOT-2015

The VOT-2015 [52] benchmark provides an evaluation kit and the dataset consists of 60 challenging sequences which

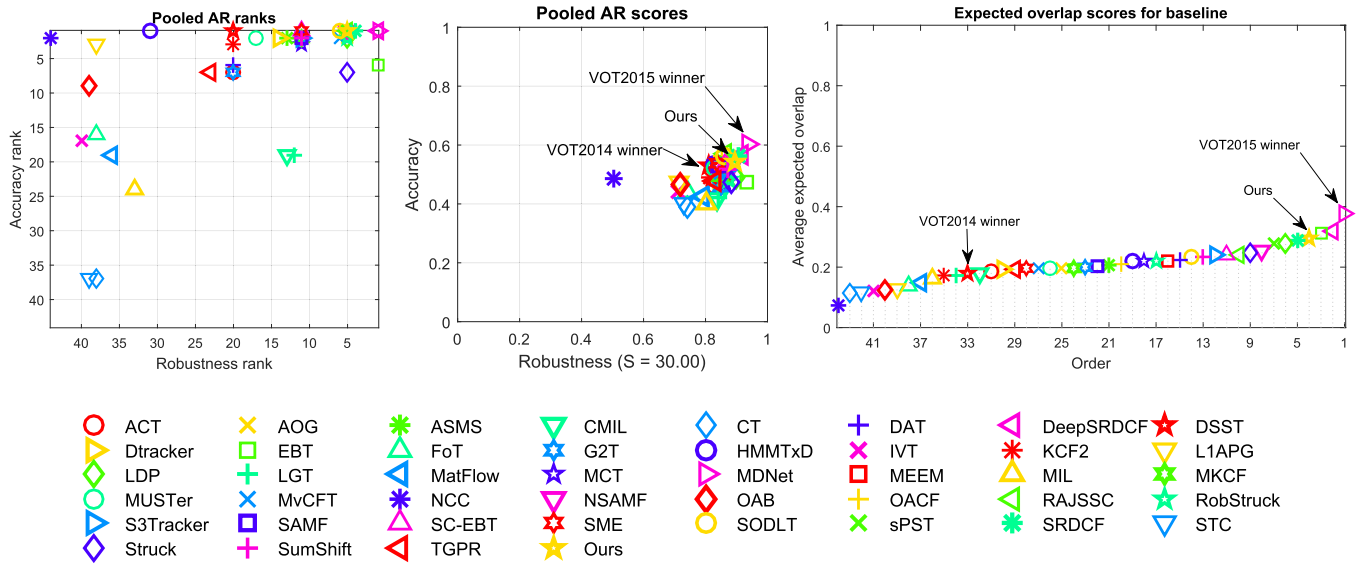

Fig. 8. The AR rank plots and AR score plots generated by sequence pooling (left) and expected average overlap graph (right) with trackers ranked from right to left. Our proposed method performs favorably against the state-of-the-art trackers even without the re-detection module, which can be attributed to the effective combination of different features and adaptive update strategy.

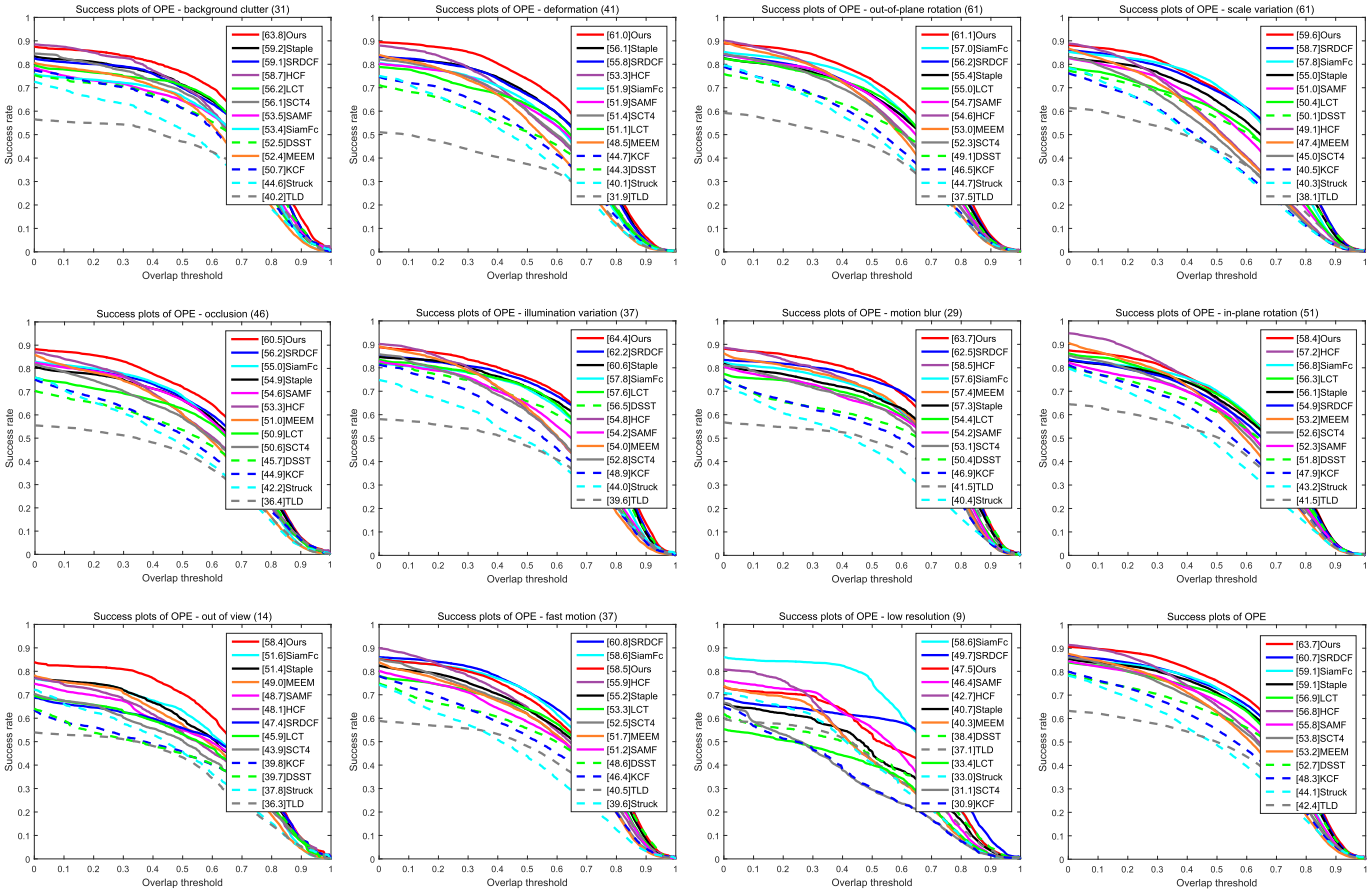

Fig. 9. Attribute-based evaluation on OTB-2015 benchmark. Success plots are shown eleven challenging factors. The title text indicate the name of the attribute and the number of videos associated with it. The legend illustrates the area-under-curve (AUC) score for each tracker. Our algorithm performs favorably against the state-of-the-art trackers, especially in out-of-view, background clutter, occlusion and deformation. We also put the overall performance here (the last one) for comparison convenience facing a single challenge and their combination.

are manually selected. The tracking performance in the VOT-2015 is evaluated on two independent metrics: accuracy and robustness. The tracker will be re-initialized to the correct position to continue tracking when tracking failure occurs. The accuracy is measured through the average of the overlap ratios. The robustness is measured in terms of the number of failures. In VOT-2015, a new measure, *i.e.*, the expected average overlap (EAO), is proposed for ranking trackers,

Fig. 10. Qualitative evaluation of our proposed algorithm, SiamFc [51], SRDCF [22], Staple [23], LCT [18] and MEEM [30] on 14 challenging videos. From left to right and top to down are *Bolt*2, *CarScale*, *DragonBaby*, *Girl*2, *Human*9, *Shaking*, *Singer*2, *Skating*1, *Airport-ce*, *Carchasing-ce*1, *Surf-ce*3, *Spiderman-ce*, *Kite-ce*1 and *Railwaystation-ce*, respectively.

which combines the raw values of per-frame accuracies and failures in a principled manner. VOT is a benchmark for short-term tracking which does not allow successful re-detection after target is lost. For fair comparison, we tested our method in VOT-2015 without re-detection module.

From the pooled AR ranks shown in Fig. 8, we can observe that our tracker is among the top performers in terms of both accuracy and robustness. The toppest two trackers, MDNet [58] and DeepSRDCF [43] obtain high robustness as well as accuracy by utilizing CNN features.

It should be noted that VOT challenge is not devoted to long-term tracking tasks with heavy occlusion and out-of-view. However, the results reveal that our method performs comparably to other state-of-the-art trackers.

G. Evaluation in Attributed Videos

We further analyze some common challenging factors in long-term tracking. All the 100 videos in OTB-2015 [48] are annotated with 11 different attributes, namely: background

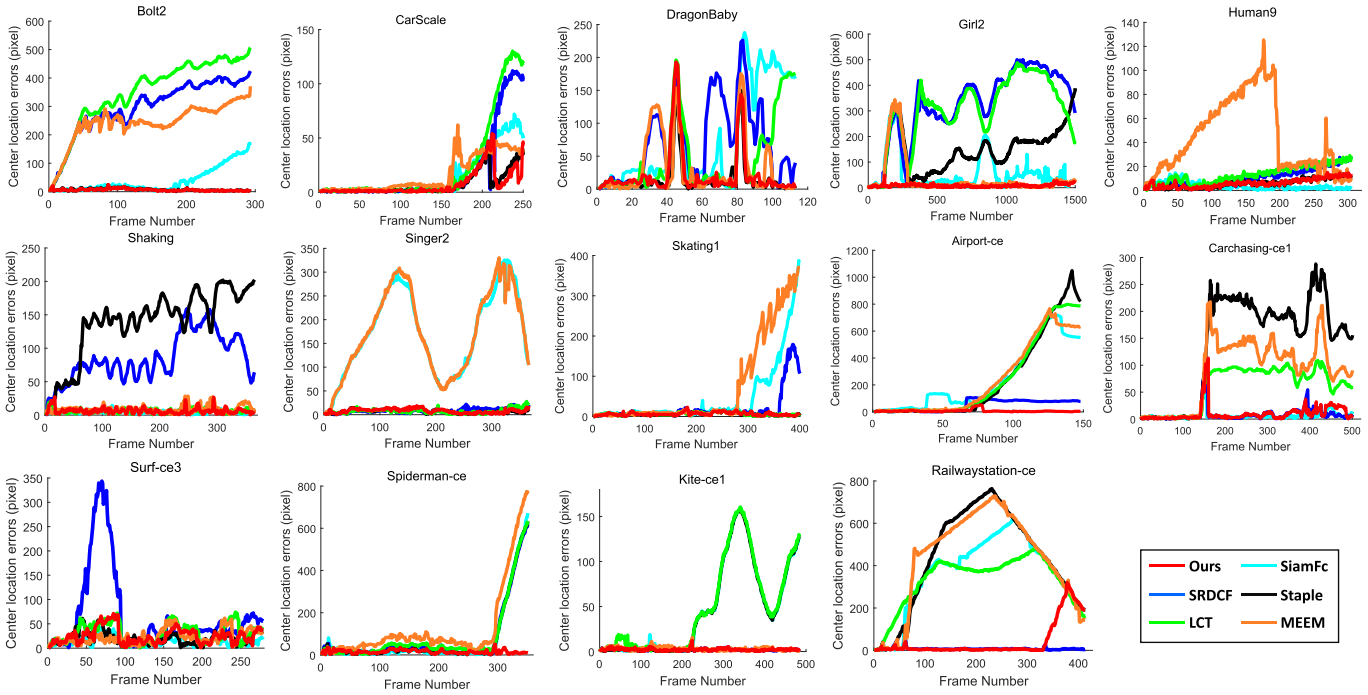

Fig. 11. Frame-by-frame comparison of center location errors (in pixel) on sixteen challenging videos. Our proposed algorithm is able to track targets accurately and stably. Especially in the video *Girl*2, *Airport-ce*, *Spiderman-ce* and *Railwaystation-ce*, almost all the trackers fail to track the target due to heavy occlusion or out-of-view while ours re-detects the target after a short-time failure.

clutter, deformation, out-of-plane rotation, scale variation, occlusion, illumination variation, motion blur, in-plane rotation, out-of-view, fast motion and low resolution.

From Fig. 9 we can observe that our tracker is adept at handling occlusion, out-of-view and deformation. In case of out-of-view, our proposed method provides a gain of 6.8% compared to the second best tracker SiamFc [51] and 7.0% compared to the baseline tracker Staple [23]. In case of occlusion, ours provides a gain of 4.3% compared to the second best tracker SRDCF [22] and 5.6% compared to the baseline tracker Staple. Many tracking-by-detection approaches fail to handle occlusion and deformation effectively due to the lack of a scheme to avoid poor updates. Some trackers have re-detection ability to some degree [18], [29], [30]. However, our proposed method outperforms them due to the combination of multiple features as well as the incorporation of the reliability estimation to search a reliable re-detected target to correct itself.

H. Qualitative Evaluation

Fig. 10 shows some comparisons of our algorithm and other five state-of-the-art trackers: SiamFc [51], SRDCF [22], Staple [23], LCT [18] and MEEM [30] on 14 challenging sequences, of which the top 8 videos are from OTB-2015 and the rest are from Temple-Color.

The SiamFC tracker performs well in handling occlusion and deformation due to the robustness of the pre-trained network. But it is less effective to distinguish target from similar objects or background clutters (*Singer*2 and *Bolt*2). The SRDCF tracker is based on CF with a novel boundary effects penalization method, which performs well in fast motion and scale variation but performs poorly in rotation (*Shaking* and *Surf-ce*3) and out-of-view (*DragonBaby*). Staple achieves high robustness in deformation and color changes due to the robust representation of color features while fails to handle heavy occlusions (*Girl*2, *Spiderman-ce* and *Railwaystation-ce*). LCT is a long-term tracker which depends on a random fern classifier to re-detect the object. However, it is not effective enough to handle heavy occlusions and fast motion (*Bolt*2). MEEM is able to handle rotation and deformation to some degree by re-utilizing reliable results using entropy minimization while fails to handle scale changes (*Carscale*) and heavy occlusion.

Finally, in Fig. 11, we compare the center location errors (Euclidean distance between groundtruth and the predicted target position) frame-by-frame on the 14 challenging videos mentioned above, which demonstrates the re-detection capability of our method. In the video *Girl*2, at about the 110-th frame, the girl (target) is fully occluded by a pedestrian (Fig. 10) and almost all the methods produce large center location errors (Fig. 11) while our method still accurately re-detects the target after a short-time failure. Similarly, at about the 70-th frame in the video *Airport-ce*, 300-th frame in *Spiderman-ce* and 150-th frame in *Carchasing-ce*1, full occlusion or out-of-view occurs to the targets and maintains for a short period (from 5 to 20 frames), and most trackers fail to locate the target when facing these situations and result in enormous center location errors. Compared to our baseline Staple, the re-detection framework improves its robustness obviously, especially in the challenging scenes with heavy occlusion and out-of-view.

There exist a few cases where our tracker fails to re-detect the object (Fig. 12). In the video *Face-ce*, the occlusions have

Fig. 12. Failure cases of our approach (*MotorRolling*, *Face-ce* and *Bike-ce2*). Our tracking results are shown in yellow and the groundtruth boxes in red.

similar hand-crafted features with the object we aim to detect which mislead our re-detector. Although our tracker tries to identify a more reliable re-detected target than the detected one, while in case of occlusions have similar features with the target, our proposed method still struggles in re-detection. In the video *MotorRolling* and *Bike-ce2*, targets undergo heavy deformation as well as fast motion, our approach fails to handle them.

V. Conclusion

In this paper, a real-time long-term tracking framework is proposed. Our approach is equipped with both tracking-by-detection and re-detection modules to handle heavy occlusion, out-of-view and significant appearance changes in long-term tracking. Through a re-detection scheme combined of HOG and color features, our tracker updates detection model adaptively to alleviate drift and false accumulation effectively. The "unreliability and reliability check" criterion enables our tracker to switch the suitable solution between detection and re-detection models. Our generic re-detection framework can be integrated into many DCF based trackers to realize consistent performance improvement. The performance of our tracker is evaluated extensively on the OTB-2015, TempleColor and VOT-2015 datasets. Our algorithm performs favorably against state-of-the-art methods especially in color videos. Given its accuracy, robustness and efficiency, our method works as a promising alternative for long-term tracking tasks where there exist much more challenging factors.

References

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, p. 13, 2006.
[2] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 864–877.
[3] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 263–270.
[4] 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.
[5] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, Aug. 2011.
[6] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1838–1845.
[7] T. Liu, G. Wang, and Q. Yang, "Real-time part-based visual tracking via adaptive correlation filters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4902–4912.
[8] S. Liu, T. Zhang, X. Cao, and C. Xu, "Structural correlation filter for robust visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4312–4320.
[9] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1269–1276.
[10] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2014, pp. 254–265.
[11] E. Kodirov, T. Xiang, Z. Fu, and S. Gong, "Person re-identification by unsupervised ℓ_1 graph learning," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 178–195.
[12] W.-S. Zheng, S. Gong, and T. Xiang, "Towards open-world person re-identification by one-shot group-based verification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 3, pp. 591–606, Mar. 2016.
[13] X. Wang *et al.*, "Greedy batch-based minimum-cost flows for tracking multiple objects," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4765–4776, Oct. 2017.
[14] A. Maksai, X. Wang, F. Fleuret, and P. Fua. (2016). "Globally consistent multi-people tracking using motion patterns." [Online]. Available: https://arxiv.org/abs/1612.00604
[15] A. Maksai, X. Wang, and P. Fua, "What players do with the ball: A physically constrained interaction modeling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 972–981.
[16] J. S. Supancic and D. Ramanan, "Self-paced learning for long-term tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2379–2386.
[17] Y. Hua, K. Alahari, and C. Schmid, "Occlusion and motion reasoning for long-term tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 172–187.
[18] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Long-term correlation tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5388–5396.
[19] K. Lebeda, S. Hadfield, J. Matas, and R. Bowden, "Long-term tracking through failure cases," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2013, pp. 153–160.
[20] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "Multi-store tracker (MUSTer): A cognitive psychology inspired approach to object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 749–758.
[21] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2544–2550.
[22] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4310–4318.
[23] 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.*, Jun. 2016, pp. 1401–1409.
[24] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, vol. 1, no. 1, pp. 886–893.
[25] 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.
[26] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time object tracking via online discriminative feature selection," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 4664–4677, Dec. 2013.
[27] S. Avidan, "Support vector tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 8, pp. 1064–1072, Aug. 2004.
[28] G. Zhu, F. Porikli, and H. Li, "Beyond local search: Tracking objects everywhere with instance-specific proposals," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 943–951.
[29] 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.
[30] J. Zhang, S. Ma, and S. Sclaroff, "Meem: Robust tracking via multiple experts using entropy minimization," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 188–203.
[31] Y. Wu, B. Shen, and H. Ling, "Visual tracking via online nonnegative matrix factorization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 3, pp. 374–383, Mar. 2014.
[32] T. Zhang, S. Liu, N. Ahuja, M.-H. Yang, and B. Ghanem, "Robust visual tracking via consistent low-rank sparse learning," *Int. J. Comput. Vis.*, vol. 111, no. 2, pp. 171–190, 2015.

[33] X. Mei, H. Ling, Y. Wu, E. P. Blasch, and L. Bai, "Efficient minimum error bounded particle resampling L1 tracker with occlusion detection," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2661–2675, Jul. 2013.

[34] T. Zhang, C. Xu, and M.-H. Yang, "Multi-task correlation particle filter for robust object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, p. 3.

[35] T. Zhang, A. Bibi, and B. Ghanem, "In defense of sparse tracking: Circulant sparse tracker," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3880–3888.

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

[37] K. Zhang, L. Zhang, M.-H. Yang, and D. Zhang, "Fast visual tracking via dense spatio-temporal context learning," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 127–141.

[38] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 65.1–65.11.

[39] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1430–1438.

[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[41] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

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

[43] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2015, pp. 621–629.

[44] X. Wang, E. Türetken, F. Fleuret, and P. Fua, "Tracking interacting objects using intertwined flows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2312–2326, Nov. 2016.

[45] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[46] H. Possegger, T. Mauthner, and H. Bischof, "In defense of color-based model-free tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 2113–2120.

[47] N. Wang, W. Zhou, Q. Tian, R. Hong, M. Wang, and H. Li, "Multi-cue correlation filters for robust visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018.

[48] 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.

[49] P. Liang, E. Blasch, and H. Ling, "Encoding color information for visual tracking: Algorithms and benchmark," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5630–5644, Dec. 2015.

[50] J. Choi, H. J. Chang, J. Jeong, Y. Demiris, and J. Y. Choi, "Visual tracking using attention-modulated disintegration and integration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4321–4330.

[51] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 850–865.

[52] M. Kristan *et al.*, "The visual object tracking VOT2015 challenge results," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2015, pp. 564–586.

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

[54] R. Tao, E. Gavves, and A. W. M. Smeulders, "Siamese instance search for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1420–1429.

[55] Y. Qi *et al.*, "Hedged deep tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4303–4311.

[56] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3119–3127.

[57] H. Li, Y. Li, and F. Porikli, "DeepTrack: Learning discriminative feature representations online for robust visual tracking," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1834–1848, Apr. 2016.

[58] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4293–4302.

Ning Wang received the B.E. degree in communication engineering from Tianjin University, in 2016. He is currently pursuing the M.Sc. degree with the Department of Electronic Engineer and Information Science, University of Science and Technology of China. His research interest is computer vision and his current research work is focused on video object tracking.

Wengang Zhou received the B.E. degree in electronic information engineering from Wuhan University, China, in 2006, and the Ph.D. degree in electronic engineering and information science from the University of Science and Technology of China (USTC), China, in 2011. From 2011 to 2013, he was a Post-Doctoral Researcher with the Computer Science Department, The University of Texas at San Antonio. He is currently an Associate Professor with the EEIS Department, USTC.

His research interests include multimedia information retrieval and computer vision.

Houqiang Li received the B.S., M.Eng., and Ph.D. degrees from the University of Science and Technology of China (USTC), Hefei, China, in 1992, 1997, and 2000, respectively, all in electronic engineering. He is currently a Professor with the Department of Electronic Engineering and Information Science, USTC. He has authored or co-authored over 100 papers in journals and at conferences. His research interests include multimedia search, image/video analysis, and video coding and communication.

Dr. Li was a recipient of the Best Paper Award for visual communications and image processing in 2012, the Best Paper Award at the International Conference on Internet Multimedia Computing and Service in 2012, the Best Paper Award at the International Conference on Mobile and Ubiquitous Multimedia from the ACM in 2011, and was a Senior Author of the Best Student Paper of the 5th International Mobile Multimedia Communications Conference in 2009. He served as an Associate Editor of IEEE TRANSACTIONS ON CIRCUIT AND SYSTEMS FOR VIDEO TECHNOLOGY from 2010 to 2013 and has been on the Editorial Board of *Journal of Multimedia* since 2009. He has served on technical/program committees, organizing committees, and as a Program Co-Chair or Track/Session Chair for over 10 international conferences.