International Journal of Innovative Computing, Information and Control
Volume **14**, Number **5**, October **2018**

ICIC International ©2018 ISSN 1349-4198
pp. 1855–1875

VISUAL OBJECT TRACKING WITH SALIENCY REFINER AND ADAPTIVE UPDATING

Wei Zhang[1,2], Baosheng Kang[1], Shunli Zhang[1,*] and Mei Gao[1]

[1]School of Information Science and Technology
Northwest University
No. 1, Xuefu Road, Chang'an District, Xi'an 710127, P. R. China
{ zhangwei.personal; bskang }@163.com; *Corresponding author: slzhang@nwu.edu.cn

[2]Department of Computer Science
Baoji University of Arts and Sciences
No. 44, Baoguang Road, Weibin District, Baoji 721016, P. R. China

Received December 2017; revised April 2018

Abstract. *In this paper, we propose a novel correlation tracking approach with saliency refiner and adaptive updating. To solve the model drift problem, the visual saliency as prior information is integrated into the kernelized correlation filter (KCF) framework, which aims to rectify small inaccuracy in case of low tracking confidence. Meanwhile, a separate scale filter is exploited to estimate the scale variation. Furthermore, a simple yet effective occlusion discriminative factor is designed based on the correlation response variation, and the changes of target appearance between consecutive frames are also taken into consideration. Both of them are introduced into the model update procedure to adaptively adjust the learning rate, which can help address the occlusion problem as well as maintain the most reliable target appearance. Extensive experiments are conducted on a recent tracking benchmark OTB-2015 to verify the effectiveness of the proposed approach. Experimental results demonstrate that the proposed approach yields better performance than 12 state-of-the-art trackers while operating at an average speed of 42 frames per second (FPS).*

Keywords: Salient object detection, Correlation filter, Object tracking, Occlusion handling, Model updating

1. **Introduction.** Visual object tracking is one of the fundamental problems in computer vision. Its applications vary from video surveillance, motion recognition, and navigation to traffic control and autonomous driving. Numerous tracking-by-detection methods [1-3] have been proposed and considerable progress has been made over the past years. A typical tracking-by-detection paradigm treats tracking task as a classification problem, which detects a target object over time while updating a classifier using the collected positive and negative samples.

Despite the improved performance, visual tracking remains to be a challenging task due to factors such as deformation, heavy occlusion, fast motion, and motion blur existing in realistic scenarios. During the tracking process, the target object often undergoes significant appearance changes caused by factors mentioned above. Even if online models are trained to adapt to these changes, small errors often accumulate, and model drift tends to happen. Model drift occurs because the object model is maintained to account for appearance changes of the tracked target via online updates. Factors like heavy occlusions and misalignment of training samples can lead to bad model updates. As the

DOI: 10.24507/ijicic.14.05.1855

appearance model is updated with noisy samples, this often degrades the object model and leads to the model drift problem, which is a key challenge in online object tracking.

Recently, correlation filter based trackers (CFTs) [4-6], which follow the tracking-by-detection framework, have gained more and more attention of corresponding researchers for their high computational efficiency. However, due to the limited prior information and the significant appearance changes, many CFTs still suffer from the model drift problem. They fail to track the target object accurately when facing challenging scenarios. The changes of target appearance also increase the difficulty to update the target model efficiently.

Several ways are designed to handle the drift problem. One strategy is to use tracker ensemble, which combines the estimates of more than one base tracker, e.g., visual tracking decomposition (VTD) [7], tracking by sampling trackers (VTS) [8] and multi-expert entropy minimization (MEEM) [9]. Another strategy adopted to alleviate the risk of drifting is long-term tracking with re-detection, which can re-locate the target object when track failures occur, e.g., tracking-learning-detection (TLD) [10] and long-term correlation tracking (LCT) [11]. Training samples are essential in both methods. How to collect and manage these samples online is difficult, and how to guarantee the reliability of these samples is still a problem. For tracking-by-detection methods, the choice of training samples utilized to update the classifier is critical for robust tracking and maintaining the model's reliability. If the current tracker loses the locations of a target object in a frame because of any interruption, the classifier may be corrupted and the tracking error will propagate to subsequent frames. It is not effective to incorporate a single, fixed model to estimate the target locations in other frames, which likely tends to drift when large appearance changes occur.

In practice, the prior information presented in subsequent frames can be treated as training data and used to help alleviate the model drift problem. Human has the ability to selectively process only salient visual stimuli in complex scene [12], which might have a higher probability of being the tracked target in each frame. Visual saliency can highlight the salient object, and at the same time, suppress the background. These advantages of visual saliency might contribute to the tracking task. However, the low computational efficiency of some salient object detection methods limits their application in real-time object tracking. It can be observed that there is little change between consecutive frames due to small time interval [11]. Except for fast and irregular motion, the tracked target can be estimated from a local search window around the previous location, and the change of the context around the target is not obvious. Therefore, the saliency map can be estimated from a relatively small area consisting of the target and its surrounding context, which can then be employed as prior information to obtain a candidate proposal. Processing on this relatively small area not only suppresses the background interference from the entire image but also reduces the computation load to a large degree. Because the visual saliency emerges from several concepts such as uniqueness, rarity, local/global contrast [12], it is relatively insensitive to shape deformation, rotation, and scale variation. For small inaccuracy, it can be rectified or corrected by the saliency prior.

Motivated by the above observations, this paper intends to integrate the visual saliency as prior information into the KCF framework and proposes a novel object tracking approach based on correlation filter with saliency refiner and adaptive updating. Firstly, multi-feature kernelized correlation filter (KCF) is employed for estimating the preliminary target location in each frame. When the tracking confidence is below a certain threshold, it indicates that the current tracking result is unreliable. Then, we activate a simple yet effective saliency refiner to rectify small inaccuracy caused by drift, which is implemented by fast salient object detection [13], salient object extraction, candidate proposal evaluation and optimal location determination respectively. After the optimal location is obtained, we estimate the scale variation by constructing a scale pyramid. Furthermore, we present an occlusion discriminative factor based on the correlation response variation and consider the changes of target appearance between consecutive frames at the same time. Both of them are introduced into the model update procedure to adaptively adjust the learning rate, which can help alleviate the occlusion problem as well as maintain the most reliable target appearance. Finally, to evaluate our proposed approach, we conduct extensive experiments on 100 challenging sequences with various attributes from a publicly available benchmark dataset OTB-2015 [14]. In the experiments and analysis, we carry out comparisons to analyze different parameters and components that impact the tracking performance. The experimental results illustrate that the saliency refiner and the adaptive updating strategy play an important role in helping alleviate the model drift problem.

The main contributions of this paper can be summarized as follows.

(1) We propose a novel object tracking method by integrating the visual saliency as prior information into the KCF framework.

(2) We introduce a model update strategy, which is based on the occlusion discriminative factor and the significant appearance change, to adaptively adjust the learning rate.

(3) The proposed approach achieves better performance than several state-of-the-art trackers while running efficiently in real time.

The rest of this paper is organized as follows. Section 2 reviews some works related to ours. We present our approach in detail in Section 3 and perform the overall evaluation against several state-of-the-art trackers in Section 4. In Section 5, the conclusion is made.

2. **Related Works.**

In this section, we introduce some works that are more relevant to ours, namely tracking by correlation filter, tracking-by-detection, and tracking with saliency.

2.1. **Tracking by correlation filter.**

Correlation filters have shown great potential and superior performance on visual object tracking, and have been studied extensively in recent years. Bolme et al. [4] proposed a minimum output sum of squared error (MOSSE) filter and applied it to tracking field. Henriques et al. [5] proposed a circulant structure with kernels tracker (CSK), which formulated tracking as a kernelized least squares classification problem. Henriques et al. [6] further proposed a KCF tracker, which extended the CSK by re-interpreting the problem using the kernelized ridge regression. Danelljan et al. [15] proposed a color naming (CN) tracker, which applied the color attributes [16] to CSK. To handle the scale variation problem, Li and Zhu [17] proposed a scale adaptive with multiple feature integration (SAMF) tracker and Danelljan et al. [18] proposed a discriminative scale space tracker (DSST) respectively. Danelljan et al. [19] further investigated strategies to reduce the computational cost of the DSST without sacrificing its robustness and accuracy. Huang et al. [20] integrated a class-agnostic detection proposal method into CFTs to handle the aspect ratio change problem. Bertinetto et al. [21] introduced a pixel-wise object likelihood map of the target into DSST to rectify the estimation of the correlation filter. Despite the improved performance, conventional CFTs still suffer from the model drift problem in realistic scenarios. To address this issue, we introduce the visual saliency as prior information into KCF framework and design a simple yet effective refiner based on salient object detection. Unlike [21], which needs to update the color model every frame, the saliency refiner is performed online without training and updating process. Furthermore, we exploit an adaptive strategy based on the changes of target appearance and occlusion detection result to make the tracking process robust.

2.2. **Tracking-by-detection.** In tracking-by-detection paradigm, a discriminative model is trained to separate the target from its surrounding background. Hare et al. [1] proposed an adaptive tracking-by-detection framework (Struck), which utilized a kernelized structured output support vector machine (SVM) learned online. Babenko et al. [2] proposed an online multiple instance learning (MIL) algorithm for object tracking to alleviate the inaccuracy in labeled training examples. Kalal et al. [10] proposed a novel tracking framework (TLD) by decomposing the long-term tracking task into three components: tracking, learning, and detection. Furthermore, a new learning paradigm was explored, which utilized two types of experts called P-expert and N-expert to generate positive and negative samples to alleviate drift. Ma et al. [11] addressed the long-term tracking problem by decomposing the tracking task into translation and scale estimation process with online random fern re-detection module. Zhu et al. [22] introduced a novel long-term CUR filter for detection into the multi-scale KCF framework to reduce the model drift problem. Unlike [10] that activates the detector in every frame, we only consider the saliency refiner in case of low tracking confidence to improve the efficiency.

2.3. **Tracking with saliency.** Visual saliency is often attributed to the variations in color, gradient, boundaries, and edges of a given image. Recently, salient object detection has received a lot of attention in many computer vision applications, ranging from object detection/tracking, image/video processing to action recognition, etc. Mahadevan and Vasconcelos [23] proposed a biologically inspired discriminant object tracking method by utilizing a center-surround saliency mechanisms. Hong et al. [24] proposed an online object tracking method by learning discriminative saliency map using convolutional neural network (CNN). However, this method required a CNN pre-trained on a large number of images to represent the target. Zhang et al. [25] employed the spatial-temporal saliency detection to guide a more accurate target location and proposed a deep learning based tracking method with CNN, whereas, the fps of this CNN based tracker is only 4-5, far from real-time tracking. Wu et al. [26] incorporated the saliency information into a dynamic Kalman model and proposed a vision-based localization and tracking method for unmanned aerial vehicles. However, it cannot handle the occlusion problem very well. Zhu et al. [27] introduced the saliency map into correlation filter based tracking framework. The saliency map was taken as prior information to obtain candidate proposals and was conducive to alleviate the model drift problem caused by occlusion or distractors as well. Similar to [27], we consider the saliency information as well. However, our approach differs from [27] in several aspects: a) we carry out the saliency object detection on a relatively small area, which has some context surrounding the tracked target, instead of the entire frame, to reduce time complexity; b) the salient object detection results adopted in our approach have exact boundaries, which are facilitated to salient object extraction and make it accurate; c) the saliency refiner is carried out when the track failures occur, rather than considering it for a fixed period.

In our approach, a fast minimum barrier salient (MBS) object detection method [13] is adopted to obtain the most salient object area, which might have a higher probability to be the candidate proposal of the tracked target, and is used for further evaluation to determine the optimal location of the target. It is shown to be advantageous to integrate the MBS object detection into our approach based on correlation filter. Firstly, it is fast enough to consider the salient object detection in a relatively small context of the target in a real-time tracking task. (It was reported to run at about 80 FPS.) Secondly, there is no need to train an online classifier like previous tracking methods [10,11,28]. The MBS object detection performs easily and has few parameters. Thirdly, the critical limitation mentioned in MBS object detection seldom occurs in our case, because the salient object extracted in the video frame is surrounded by a small context of the target, and it will not connect to the image boundary in most cases.

3. The Proposed Approach.

The goal of our approach is to handle the model drift problem frequently occurring in object tracking and to improve the tracking performance of the KCF tracker.

The flowchart of our proposed approach is shown in Figure 1, which illustrates the tracking process of the *blurowl* sequence in the t-th frame. It can be observed from Figure 1 that we first estimate the preliminary location $\boldsymbol{l}_t$ by utilizing the multi-feature KCF and obtain the response map. The tracking confidence is then determined by the maximum response value R. Small confidence value indicates the likely track failure. When the confidence value is below a certain threshold, a saliency refiner is automatically activated to rectify the preliminary location estimation of the multi-feature KCF. Otherwise, it means that the location obtained by the multi-feature KCF is more accurate. After the optimal location is determined, we evaluate the scale variation by constructing a scale pyramid. Finally, we update the model adaptively based on the designed occlusion discriminative factor and the significant appearance change. Details of our approach are presented as follows. The whole algorithm is provided in the end.

3.1. Multi-feature kernelized correlation filter.

Our approach is based on the multi-feature KCF, which is responsible for estimating the preliminary location of the tracked target in each frame.

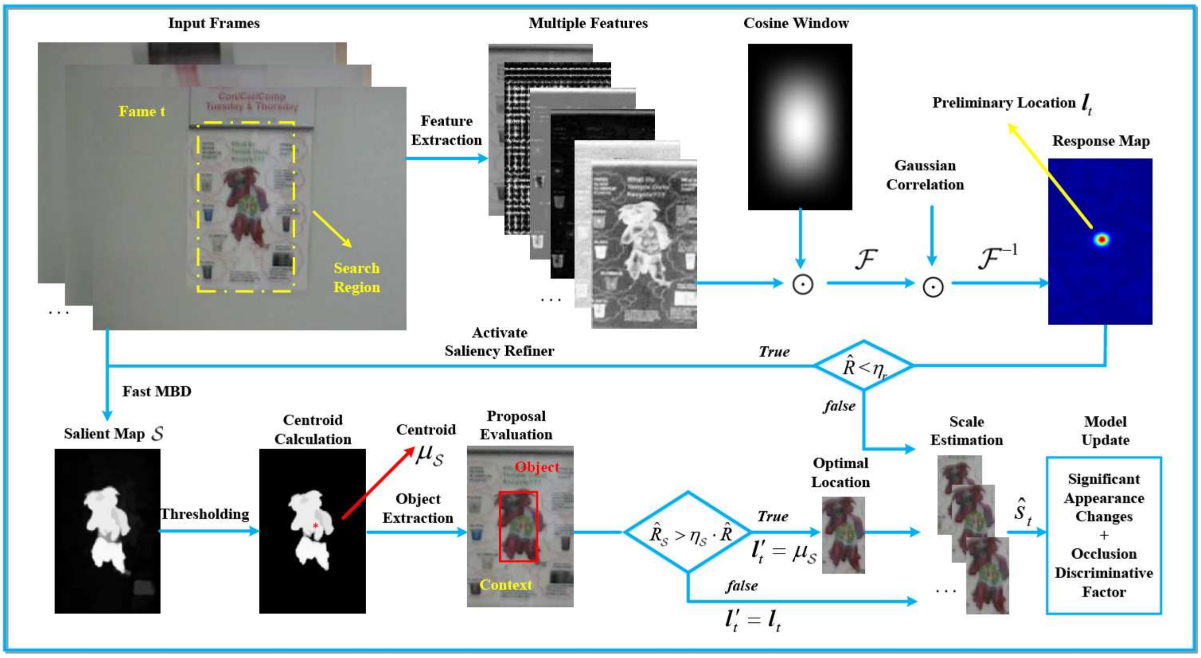


Figure 1. Flowchart of the proposed approach (Operator $\odot$ denotes the element-wise production, $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the discrete Fourier transform and its inverse transform, $\boldsymbol{l}_t$ and $\boldsymbol{l}'_t$ denote the preliminary and the newly estimated location, $\hat{R}$ and $\hat{R}_{\mathcal{S}}$ are the maximum response values obtained by the preliminary estimation and the proposal evaluation, η_r and $\eta_{\mathcal{S}}$ are the corresponding thresholds, and $\hat{s}_t$ is the estimated scale factor of the tracked target)

In KCF, the correlation filter can be efficiently solved by introducing the circulant matrix and the kernel ridge regression. In frame t, the appearance of the target object is modeled on an image patch $\boldsymbol{x}$ of size $W \times H$ centered at the target position, which is p times larger than the target size to provide some context. In order to predict the probability of the image patch $\boldsymbol{x}$ being the tracked target, a classifier $f(\boldsymbol{x}) = \boldsymbol{y}$ is trained on all circular shift versions of $\boldsymbol{x}_{w,h}$, where $(w, h) \in \{0, \ldots, W-1\} \times \{0, \ldots, H-1\}$. Each shifted sample $\boldsymbol{x}_{w,h}$ corresponds to a Gaussian function label $y_{w,h} \in [0, 1]$. The classifier is trained by solving the minimization problem of ridge regression:

$$\min_{\boldsymbol{w}} \sum_{w,h} \mathcal{L}\left(f\left(\phi\left(\boldsymbol{x}_{w,h}\right); \boldsymbol{w}\right), y_{w,h}\right) + \lambda \|\boldsymbol{w}\|^2 \tag{1}$$

where $\mathcal{L}(\cdot)$ is the loss function, $\boldsymbol{w}$ is the classifier parameter, $\phi\left(\boldsymbol{x}_{w,h}\right)$ is the mapping to nonlinear feature space, and λ ($\lambda \geq 0$) is the regularization parameter that controls overfitting. In ridge regression, the loss function $\mathcal{L}(\cdot)$ can be defined as a quadratic formulation:

$$\mathcal{L}\left(f\left(\phi(\boldsymbol{x}_{w,h}); \boldsymbol{w}\right), y_{w,h}\right) = \left\|\left\langle \phi\left(\boldsymbol{x}_{w,h}\right), \boldsymbol{w}\right\rangle - y_{w,h}\right\|^2 \tag{2}$$

where $\langle\cdot\rangle$ denotes the inner product. With the kernel method, the solution $\boldsymbol{w}$ can be represented by a linear combination of training samples: $\boldsymbol{w} = \sum_{w,h} \alpha_{w,h} \phi(\boldsymbol{x}_{w,h})$, where $\boldsymbol{\alpha} = \{\alpha_{w,h}\}$ is the dual space variable of $\boldsymbol{w}$. Then the goal of minimization problem has changed to find variable $\boldsymbol{\alpha}$ instead of $\boldsymbol{w}$. According to [3], for a unitarily invariant kernel, the filter coefficient $\boldsymbol{\alpha}$ can be derived as:

$$\mathcal{F}(\boldsymbol{\alpha}) = \frac{\mathcal{F}(\boldsymbol{y})}{\mathcal{F}(\boldsymbol{k}^{xx}) + \lambda} \tag{3}$$

where $\mathcal{F}$ denotes the discrete Fourier transform (DFT). The kernel correlation of $\boldsymbol{x}$ with itself can be obtained by $\boldsymbol{k}^{xx} = \kappa\left(\boldsymbol{x}_{w,h}, \boldsymbol{x}\right) = \left\langle \phi\left(\boldsymbol{x}_{w,h}\right), \phi(\boldsymbol{x})\right\rangle$.

After the training process, for the frame $t+1$, the detection task is carried out on an image patch $\boldsymbol{z}$ in the new frame with the search window size $W \times H$ centered at the last target location:

$$\mathcal{F}\left(\hat{\boldsymbol{y}}\right) = \mathcal{F}\left(\boldsymbol{k}^{\boldsymbol{z}\hat{\boldsymbol{x}}}\right) \odot \mathcal{F}(\boldsymbol{\alpha}) \tag{4}$$

where $\odot$ denotes the element-wise production, and $\hat{\boldsymbol{x}}$ is the learned target appearance. $\mathcal{F}\left(\hat{\boldsymbol{y}}\right)$ is the response map for the new frame image patch in Fourier domain. We obtain the maximum response value R as below:

$$\hat{R} = \max\left(\mathcal{F}^{-1}\left(\mathcal{F}\left(\hat{\boldsymbol{y}}\right)\right)\right) \tag{5}$$

where $\mathcal{F}^{-1}$ denotes the inverse DFT. Therefore, the new target location $\boldsymbol{l}_t$ is estimated by finding the corresponding position of the maximal response value $\hat{R}$.

To improve the precision, multiple features such as the raw intensity, color naming [16] and histogram of gradient (HOG) feature [29] are considered simultaneously. These three features are complementary to each other and can be simply concatenated, totally 42 channels, which can be used in both training and detecting procedure. To remove the discontinuities at the image boundaries caused by cyclic assumption, the extracted input feature channels are weighted by a cosine window.

3.2. **Fast salient object detection.** The salient object detection method used in our approach is based on the minimum barrier distance (MBD) transform [30], which is much more robust to noise, blur, and pixel value fluctuation than the widely used geodesic distance. The MBD transform can be directly applied to raw pixels without region abstraction. Consider a single channel image $\mathcal{I}$, the minimum barrier path cost function

$\mathcal{C}_{\mathcal{I}}(\tau)$ is defined as follows:

$$\mathcal{C}_{\mathcal{I}}(\tau) = \max_{i=0}^{k} \mathcal{I}(\tau(i)) - \min_{i=0}^{k} \mathcal{I}(\tau(i)) \tag{6}$$

where $\tau = \{\tau(0), \ldots, \tau(k)\}$ is the path, i.e., a sequence of adjacent pixels, and $\mathcal{I}(\tau(i))$ represents the pixel value for the i-th pixel on the path τ. The function $\mathcal{C}_{\mathcal{I}}(\tau)$ measures the distance between the highest and the lowest points along the path τ. In our approach, four-adjacent paths are considered.

Given the path cost function $\mathcal{C}_{\mathcal{I}}(\tau)$, the MBD map $\mathcal{D}(u)$ is obtained by the following minimization:

$$\mathcal{D}(u) = \min_{\tau \in \prod_{Q,u}} (\mathcal{C}_{\mathcal{I}}(\tau)) \tag{7}$$

where $\tau \in \prod_{Q,u}$ denotes the set of all paths connecting a pixel in background seed set Q with u. The computation of the MBD map is formulated as finding the shortest path for each pixel in the image $\mathcal{I}$. However, the time complexity for exact MBD transform is $O(mn \log n)$, where n is the pixel number in the image and m is the number of distinct pixel values the image contains. To reduce the computational cost and make the algorithm more efficient, a fast MBD algorithm is proposed [13], which uses the raster scanning to approximate MBD transform iteratively. We denote $\mathcal{P}(v)$ as the path currently assigned to pixel v and $e_{v,u}$ as the edge from v to u. $\mathcal{P}(v) \cdot e_{v,u}$ is a path for u that appends edge $e_{v,u}$ to $\mathcal{P}(v)$. We denote $\mathcal{P}(v) \cdot e_{v,u}$ as $\mathcal{P}_v(u)$. Thus, a new path cost function is redefined as follows:

$$\mathcal{C}_{\mathcal{I}}(\mathcal{P}_v(u)) = \max\{\mathcal{H}(v), \mathcal{I}(u)\} - \min\{\mathcal{L}(v), \mathcal{I}(u)\} \tag{8}$$

where $\mathcal{H}(v)$ and $\mathcal{L}(v)$ denote the highest and lowest pixel values on the current path $\mathcal{P}(v)$ for pixel v respectively. The new MBD cost function is computed by using two auxiliary maps $\mathcal{H}$ and $\mathcal{L}$.

For each pixel u, we visit it in a raster scan or inverse raster scan order, which is implemented alternately to update the MBD map $\mathcal{D}(u)$ via:

$$\mathcal{D}(u) \leftarrow \min \begin{cases} \mathcal{D}(u) \\ \mathcal{C}_{\mathcal{I}}(\mathcal{P}_v(u)) \end{cases} \tag{9}$$

During a pass, we use each adjacent neighbor v in the corresponding half of neighborhood of u to iteratively minimize the path cost at u.

Then the saliency map $\mathcal{B}$ is achieved by pixel-wise adding the maps that compute in each color channel respectively. Furthermore, an appearance based background cue is integrated and a series of post-processing operations are adopted to enhance the quality of saliency map $\mathcal{B}$. We denote the final saliency map after post-processing as $\mathcal{S}$.

In our approach, the salient object detection is performed on an image patch with size $W \times H$ centered at the preliminary location of the target, which has the same size as the patch used in the multi-feature KCF. The salient object detection result is illustrated in Figure 1. Unlike [27] computes the saliency map over the entire frame and considers it every ten frames, we carry out the salient object detection in a relatively small region, which has a higher probability of containing the target object, and we consider it in case of low tracking confidence to reduce computation load.

3.3. **Saliency refiner.** Because the visual saliency is relatively invariant to some appearance changes such as deformation, rotation, and scale variation, we intend to integrate it as prior information into the KCF framework to handle the drift problem. In our approach, we propose a simple yet effective saliency refiner to rectify the preliminary location estimation of the multi-feature KCF, and it is implemented by fast salient object detection (described in Section 3.2), salient object extraction, candidate proposal evaluation and optimal location determination respectively. We describe the saliency refiner in detail and show the overall tracking process.

Given the target ground truth of size $w \times h$ centered at the location $\boldsymbol{l}_1$ in the first frame, the filter coefficient $\boldsymbol{\alpha}$ is initialized on a patch $\boldsymbol{x}^1$ centered at $\boldsymbol{l}_1$ of size $W \times H$ to provide some context information, where $W = p \times w$, $H = p \times h$ and $p > 1$. When a new frame t comes, the detection procedure of multi-feature KCF is performed on a patch $\boldsymbol{z}^t$ extracted from the current frame, whose center locates at $\boldsymbol{l}_{t-1}$ and size is $W \times H$. The preliminary target location $\boldsymbol{l}_t$ can be found by the corresponding position of the maximal response value $\hat{R}$. To achieve real-time performance, we only consider the saliency refiner when track failures occur. For simplicity, we take the maximum response value $\hat{R}$ as the tracking confidence, which can reflect the tracking reliability to some extent.

When the tracking confidence is below a certain threshold η_r, the fast salient object detection is performed on a patch $\boldsymbol{z}_{\mathcal{S}}$ centered at $\boldsymbol{l}_t$ with size $W \times H$. After the saliency map $\mathcal{S}$ is obtained on image patch $\boldsymbol{z}_{\mathcal{S}}$, we generate a bounding box to extract the most salient object, which has a higher probability to be the tracked target. Before extraction, we conduct a threshold preprocessing procedure to get rid of some blurry area around the boundary of the target and within the background. For each pixel (i, j) of saliency map $\mathcal{S}$, only saliency value being higher than the threshold η_t is retained. By doing this, we get the highly confident object region and separate the target from the background accurately. It can be clearly found in Figure 1 that the interference area in the background has been removed after threshold preprocessing. To obtain the salient object's bounding box $B_{\mathcal{S}}$, we calculate the centroid $\mu_{\mathcal{S}} = (\bar{x}_{\mathcal{S}}, \bar{y}_{\mathcal{S}})$ of the saliency map $\mathcal{S}'$ after threshold preprocessing as follows:

$$\begin{cases} \bar{x}_{\mathcal{S}} = \dfrac{\sum\limits_{i\in[1,W],j\in[1,H]} \mathcal{S}'(i,j) \cdot i}{\sum\limits_{i\in[1,W],j\in[1,H]} \mathcal{S}'(i,j)} \\ \bar{y}_{\mathcal{S}} = \dfrac{\sum\limits_{i\in[1,W],j\in[1,H]} \mathcal{S}'(i,j) \cdot j}{\sum\limits_{i\in[1,W],j\in[1,H]} \mathcal{S}'(i,j)} \end{cases} \tag{10}$$

where $\mathcal{S}'(i, j)$ is the saliency value of pixel (i, j) only belonging to the object region. The salient object's bounding box $B_{\mathcal{S}}$ can be simply extracted by an area centered at the estimated centroid $\mu_{\mathcal{S}}$ with size $w \times h$, which is served as a candidate proposal. We evaluate it by utilizing its corresponding patch $\boldsymbol{z}'_{\mathcal{S}}$ with size $W \times H$, which is similar to Equation (4):

$$\mathcal{F}\left(\hat{\boldsymbol{y}}_{\mathcal{S}}\right) = \mathcal{F}\left(\boldsymbol{k}^{\boldsymbol{z}'_{\mathcal{S}}\hat{\boldsymbol{x}}}\right) \odot \mathcal{F}(\boldsymbol{\alpha}) \tag{11}$$

where $\mathcal{F}\left(\hat{\boldsymbol{y}}_{\mathcal{S}}\right)$ is the response map in Fourier domain and the maximum response value is obtained according to Equation (5) and denoted as $\hat{R}_{\mathcal{S}}$. If $\hat{R}_{\mathcal{S}} > \eta_{\mathcal{S}} \cdot \hat{R}$, it means that the candidate proposal obtained by salient object extraction has higher reliability. We update the current location $\boldsymbol{l}'_t$ to $\mu_{\mathcal{S}}$, the maximum response value $\hat{R}$ to $\hat{R}_{\mathcal{S}}$, and the response map $\mathcal{F}(\hat{\boldsymbol{y}})$ in Fourier domain to $\mathcal{F}\left(\hat{\boldsymbol{y}}_{\mathcal{S}}\right)$. In this way, the small inaccuracy caused by drift will be rectified. Otherwise, if $\hat{R}_{\mathcal{S}} \leq \eta_{\mathcal{S}} \cdot \hat{R}$, it means that the location estimated by multi-feature KCF is more precise, and we abandon the salient candidate proposal.

3.4. **Scale estimation.** After obtaining the optimal target location $\boldsymbol{l}'_t$, we construct a scale pyramid to estimate the scale variation of the target in the current frame. Similar to the DSST [18], a one-dimensional correlation filter $\hat{\boldsymbol{h}}_s$ is learned for scale variation, which is separate for location estimation. Letting $P \times Q$ denote the target size in current frame, the scale filter pool P_s is defined as below:

$$P_s = \left\{ a^j \middle| j = \left\lfloor -\frac{N_s - 1}{2} \right\rfloor, \left\lfloor -\frac{N_s - 3}{2} \right\rfloor, \dots, \left\lfloor \frac{N_s - 1}{2} \right\rfloor \right\} \tag{12}$$

where a denotes the scale factor between feature layers and N_s is the size of the scale filter. For each $s \in P_s$, we extract an image patch J_s of size $sP \times sQ$ centered around the tracked target. The optimal scale factor is obtained by searching the scale space for the one with the highest response value.

The location and scale estimation process are completely separate in our approach. If the target is under partial occlusion, large appearance changes, or motion blur, the accuracy of scale estimation will be influenced to some extent, which is easily accumulated to cause the model drift problem. Therefore, the scale factor is not introduced into the preliminary location estimation and salient object detection process. Scale estimation is performed after the optimal location is obtained. Additionally, the scale model update is carried out when the current tracking result is reliable, which will be discussed in the next section.

3.5. **Adaptive model update.**

To track the target robustly, it is necessary to update the target appearance model continuously as it changes over time. A linear interpolation strategy is performed on the filter coefficient $\boldsymbol{\alpha}$ and the target appearance model $\boldsymbol{x}$ as follows:

$$\left\{ \begin{array}{l} \mathcal{F}(\hat{\boldsymbol{\alpha}}^t) = (1-\gamma)\mathcal{F}(\hat{\boldsymbol{\alpha}}^{t-1}) + \gamma \mathcal{F}(\boldsymbol{\alpha}^t) \\ \hat{\boldsymbol{x}}^t = (1-\gamma)\hat{\boldsymbol{x}}^{t-1} + \gamma \boldsymbol{x}^t \end{array} \right. \tag{13}$$

where t is the frame index. γ is the learning rate and has a fixed value in many conventional CFTs. However, during the tracking process, the target often suffers from occlusion, out of view, deformation and motion blur, etc. Inappropriate updating also leads to the drift problem.

To solve this issue, we propose a simple yet effective occlusion discriminative factor by analyzing the variation of correlation response, and we also take the changes of target appearance between consecutive frames into consideration. Both of them are introduced into the model update procedure to adaptively adjust the learning rate.

The changes of the target appearance can be reflected by the correlation peak value of the response map in some degree. Larger fluctuation of correlation peak values of the response maps between consecutive frames indicates significant appearance change of the tracked target. We define the first indicator $g_{app}(t)$ based on the change of the target appearance to evaluate this fluctuation:

$$g_{app}(t) = \left| 1 - \frac{\hat{R}^t}{\hat{R}^{t-1}} \right| \tag{14}$$

where $\hat{R}^t$ denotes the peak value of the response map obtained in frame t. Smooth variation of the target appearance between consecutive frames indicates more reliable tracking results. Thus, the ratio is close to 1 and the value of indicator $g_{app}(t)$ approaches to 0. Otherwise, the value of $g_{app}(t)$ will increase. Larger values of $g_{app}(t)$ reflect significant changes in target appearance.

Occlusion is one of the challenging problems frequently appearing during the tracking process. When the tracked target is occluded, some noisy information will be introduced in the appearance model and the classifier through updating, and with time increasing,

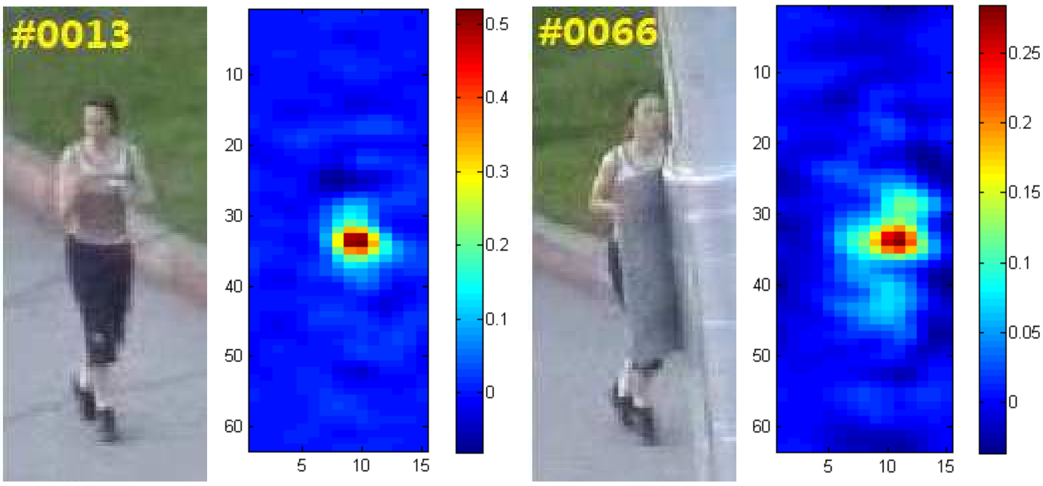


FIGURE 2. (color online) The comparison of the response maps of the *jogging1* sequence in non-occlusion and occlusion cases (# number in each subfigure represents the frame index)

eventually, lead to the drift problem. This can be alleviated by employing the occlusion detection result based on the variation of response map as the second indicator to compensate for the updating.

Figure 2 shows an illustration of the response maps of the *jogging1* sequence in non-occlusion and occlusion cases. We can find that the peak value of response map is relatively high in non-occlusion case (#13), and the response scores with higher values are distributed around the peak. However, the response map changes dramatically when the occlusion occurs (#66). The peak value of response map drops significantly, and the response scores with higher values are scattered over a larger area. According to the above analysis, we define an occlusion discriminative factor $g_{occ}(t)$ in the t-th frame as follows:

$$g_{occ}(t) = \begin{cases} \text{true}, & \text{if } \hat{R}^t < \eta_r \text{ and } G(t) > \eta_o WH \\ \text{false}, & \text{otherwise} \end{cases} \tag{15}$$

where η_r and η_o are the thresholds. The function $G(t)$ reflects the number of response scores within a specific range. It can be defined via the following equation:

$$G(t) = \left| \mathcal{F}^{-1}\left(\mathcal{F}\left(\hat{\boldsymbol{y}}^t\right)\right) > \eta_g \hat{R}^t \right| \tag{16}$$

where $|A|$ is the number of pixels within the region A and η_g is the parameter that controls the specific range of response scores.

If $g_{occ}(t)$ is true, it means that the tracked target is occluded. Therefore, the learning rate of the correlation filter can be adjusted by considering the two indicators defined above simultaneously:

$$\gamma = \begin{cases} \rho \cdot \gamma_{init}, & \text{if } g_{app}(t) > \delta_r \text{ or } g_{occ}(t) = \text{true} \\ \gamma_{init}, & \text{otherwise} \end{cases} \tag{17}$$

where γ_{init} is the initial learning rate and δ_r is the threshold. ρ is the relative ratio to reduce the learning rate if current tracking result is unreliable.

For the scale filter $\hat{\boldsymbol{h}}_s$, if $g_{occ}(t)$ is true, the scale model is not updated; otherwise, we keep the learning rate.

3.6. **The tracking algorithm.** In this section, the proposed tracking approach is summarized as follows:

```
Input: Initial location $\boldsymbol{l}_{t-1}$, a scale factor $\hat{s}_{t-1}$ of the tracked target, the target appearance
       $\hat{\boldsymbol{x}}^{t-1}$ and filter coefficient $\hat{\boldsymbol{\alpha}}^{t-1}$.
Output: The estimated location $\boldsymbol{l}'_t$ and scale factor $\hat{s}_t$ of the tracked target in frame $t$.
Repeat
    1: Crop out an image patch $\boldsymbol{z}$ at location $\boldsymbol{l}_{t-1}$ in frame $t$ and extract the CN and HOG features;
    2: Estimate the preliminary location of the target using multi-feature KCF described in Section 3.1. Compute the response map $\mathcal{F}(\hat{\boldsymbol{y}})$ in Fourier domain using Equation (4). Find the preliminary location $\boldsymbol{l}_t$ and achieve the maximum response value $\hat{R}$ according to Equation (5);
    3: If $\hat{R} < \eta_r$ then
           Activate the saliency refiner described in Section 3.3. Perform fast salient object detection on the patch $\boldsymbol{z}_{\mathcal{S}}$ at location $\boldsymbol{l}_t$ using the method described in Section 3.2 to obtain saliency map $\mathcal{S}$. The centroid $\mu_{\mathcal{S}}$ of the salient map $\mathcal{S}'$ after threshold preprocessing is calculated using Equation (10). Generate the salient object's bounding box $B_{\mathcal{S}}$ centered at $\mu_{\mathcal{S}}$ to serve as a candidate proposal. Evaluate $B_{\mathcal{S}}$ using its context patch $z'_{\mathcal{S}}$ and compute the response map $\mathcal{F}(\hat{\boldsymbol{y}}_{\mathcal{S}})$ in Fourier domain according to Equation (11). Record the corresponding maximum response value as $\hat{R}_{\mathcal{S}}$. The current position $\boldsymbol{l}'_t$ is determined by the following rules:
           If $\hat{R}_{\mathcal{S}} > \eta_{\mathcal{S}} \cdot \hat{R}$ then
               $\boldsymbol{l}'_t = \mu_{\mathcal{S}},\ \hat{R} = \hat{R}_{\mathcal{S}},\ \mathcal{F}(\hat{\boldsymbol{y}}) = \mathcal{F}(\hat{\boldsymbol{y}}_{\mathcal{S}})$;
           else
               $\boldsymbol{l}'_t = \boldsymbol{l}_t$;
           end
       else
           Turn to Step 4.
       end
    4: Construct a scale pyramid to learn a one-dimensional correlation filter $\hat{\boldsymbol{h}}_s$ around the current location $\boldsymbol{l}'_t$. Find the optimal scale $\hat{s}_t$ using method described in Section 3.4;
    5: Update the model adaptively using the method in Section 3.5. Evaluate the change of the target appearance $g_{app}(t)$ using Equation (14). Compute the occlusion discriminative factor $g_{occ}(t)$ using Equation (15) and Equation (16). Update the target appearance $x$ and the correlation filter coefficient $\alpha$ based on $g_{app}(t)$ and $g_{occ}(t)$ adaptively using Equation (13) and Equation (17). Update the scale filter $\hat{\boldsymbol{h}}_s$ if $g_{occ}(t)$ is false.
Until the end of the video sequence.
```

4. **Experiments and Analysis.**

In this section, we conduct extensive experiments with the proposed approach. We first introduce the experimental setup. Then we present the overall performance and attribute-based evaluation of the proposed approach and other state-of-the-art trackers on a recent benchmark dataset, parameters and component analysis of the proposed approach. Finally, we provide the qualitative evaluation. The overall experiments are performed in Matlab 2013b on an Intel (R) Core (TM) i7-4790 CPU (3.6 GHz) with 8 GB RAM.

4.1. **Experimental setup.**

(1) **Parameters.** The parameters used in the experiments are set as follows: In multi-feature KCF, the cell size of HOG is 4×4 and the orientation bin number of HOG is 9, the regularization parameter is $\lambda = 10^{-4}$, which are similar to those in [6]. The parameter p is 2.5 to contain some context, and it remains the same as those in salient candidate proposal evaluation. In fast MBD salient detection, all parameters are set the same as [13]. In saliency refiner, the saliency preprocessing threshold $\eta_t = 0.5$ and the tracking confidence value $\eta_r = 0.45$ are chosen according to their influence on the tracking performance, which are analyzed in Section 4.2. The parameter $\eta_S = 1.2$ is set larger than 1 to ensure that the location of the saliency candidate proposal is more accurate than the preliminary estimation. For scale estimation, the parameters are set the same as those described in [18]. The number of scales N_s is 33 with a scale factor a set to 1.02. The learning rate in scale estimation is set to 0.025. During model updating, the initial learning rate $\gamma_{init} = 0.02$. If current tracking result is unreliable, the parameter $\rho = 0.01$ is used to update the object model slowly and maintain reliable target appearance. The parameter $\delta_r = 0.4$ is empirically set for checking the significant appearance changes. For occlusion handling, the parameters $\eta_o = 0.01$ and $\eta_g = 0.7$ are empirically set to a constant. η_o is a threshold for measuring the severity of occlusion and η_g for measuring the distortion degree of the response map. All parameters are the same for all following experiments.

(2) **Datasets and evaluation methodology.** To evaluate the overall tracking performance, we conduct experiments on a recent benchmark dataset OTB-2015 [14], which consists of 100 challenge sequences. These sequences contain complex scenes with factors, e.g., partial occlusion, fast motion, deformation, scale variation, and background cluttered.

In the overall experiments, our method and the state-of-the-art trackers are compared using the evaluation methodology provided by the recent benchmark dataset [14]. One-pass evaluation (OPE) is employed and two metrics, precision and success plots, are used. The precision metric computes the percentage of frames in the sequence whose estimated target center is within some certain distance with the ground truth. The average Euclidean distance between the estimated target center and the ground truth is also defined as center location error (CLE). Smaller CLE value means a more accurate result. The success metric computes the percentage of successful frames where the bounding box overlap ratio is above a given threshold. We adopt the Pascal VOC overlap ratio (VOR), which is computed as:

$$VOR = \frac{Area\left(B_T \cap B_G\right)}{Area\left(B_T \cup B_G\right)} \tag{18}$$

where B_T represents the tracked bounding box and B_G represents the ground truth bounding box. $\cup$ and $\cap$ represent the union and intersection operators. $Area(\cdot)$ represents the region area. In the precision plots, the average distance precision is plotted over a range of thresholds, and the average precision score at 20 pixels corresponding to the OPE of each method is contained in the legend. Likewise, in the success plots, the average overlap ratio of successful frames at the thresholds varying from 0 to 1 is plotted, and the success score with the area under the curve (AUC) is reported in the legend.

4.2. **Quantitative evaluation.** We compare our approach with 12 state-of-the-art trackers including the CSK [5], KCF [6], CN [15], DSST [18], Struck [1], TLD [10], OAB [3], MIL [2], VTD [7], VTS [8], SCM [31], and TGPR [32]. Among these trackers, the CSK, KCF, CN, and DSST are CFTs. The MIL, OAB, Struck, and TLD are typical tracking-by-detection methods. The VTD, VTS, SCM, and TGPR are representative trackers using multiple classifiers.

Table 1 shows the comparison between our approach and 12 state-of-the-art trackers using average CLE and average VOR. Speed is also reported in average FPS. The best and second best results are illustrated in bold and underlined fonts respectively. We can see from Table 1 that our approach performs favorably against the existing method in terms of CLE and VOR. Our approach significantly improves the KCF tracker with a relative reduction in CLE by 20.8%. Moreover, our method achieves 0.56 in VOR, which gets a 16.7% improvement upon KCF on the dataset. The average speed of our proposed approach over 100 sequences is 42.0 FPS, which is faster than representative trackers such as TLD and Struck.

Table 1. Performance comparison of our proposed approach and 12 state-of-the-art trackers over 100 sequences on the OTB-2015 dataset

Metrics	MIL	OAB	VTS	VTD	SCM	TGPR	TLD	Struck	CN	CSK	KCF	DSST	Ours
CLE (pixel)	71.8	68.8	64.0	61.7	62.0	55.5	61.8	46.9	81.9	305	45.0	48.4	**35.6**
VOR	0.33	0.37	0.37	0.36	0.45	0.46	0.42	0.47	0.42	0.39	0.48	0.53	**0.56**
Speed (FPS)	28.0	4.99	5.70	5.70	0.36	0.64	24.1	9.84	157	**248**	243	53.1	42.0

(1) **Overall performance.** Figure 3 shows the success and precision plots of the top 10 trackers over 100 sequences on the OTB-2015 dataset. As it can be seen from the plots, our approach achieves 0.550 success score and 0.745 precision score, both of which rank in the first place among all the compared trackers. Compared to KCF, our approach improves the success and precision scores by 16% and 8% respectively.

(2) **Attribute-based performance.** The benchmark sequences are annotated with 11 different attributes, namely: scale variation, out-of-plane rotation, in-plane rotation, occlusion, deformation, fast motion, illumination variation, background clutter, motion blur, out-of-view, and low resolution. These attributes affect the performance of a tracker and are used to evaluate the tracker in different scenarios. We perform a comparison

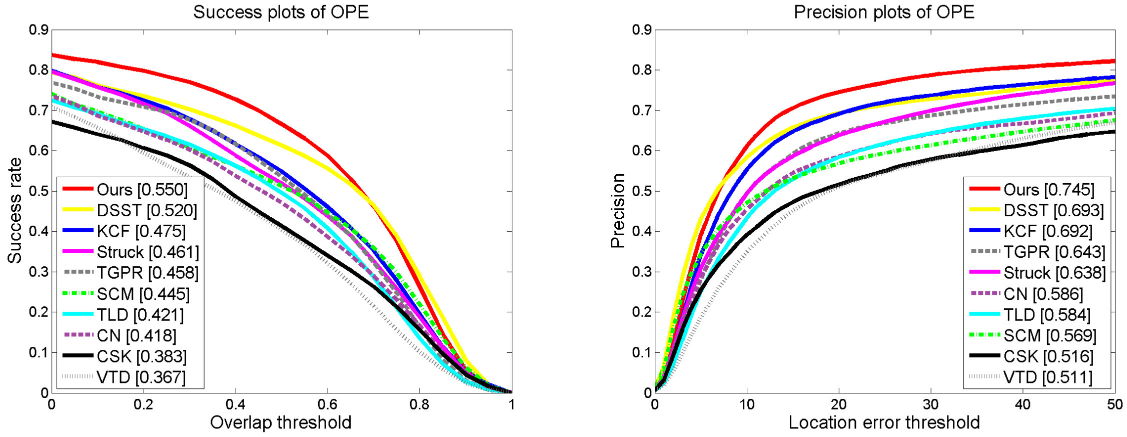


Figure 3. (color online) The precision and success plots of our approach and the other top 9 best performing state-of-the-art trackers on the OTB-2015 dataset. The plots are generated for OPE (precision score at 20 pixels) and OPE (success score with the AUC). Our approach is shown at the top of all curves. In all plots, our approach performs favorably better than other state-of-the-art trackers.

with 12 state-of-the-art trackers over 100 sequences annotated with the attributes above. Figure 4 shows the success plots of the attributes above respectively. We mainly analyze the ranked results based on the success plots, which are more accurate than precision plot, as described in [6]. Our approach ranks first on 9 out of 11 attributes with a large margin compared to other trackers, except illumination variation and low resolution. On both illumination variation and low resolution subset, DSST and SCM perform best. In detail, our method improves the success score with the AUC of 11 attributes by 24.6%, 17.8%, 13.3%, 22.6%, 18.1%, 7.20%, 11.8%, 5.43%, 17.6%, 18.1%, 23.1% respectively compared to KCF. The rank sequence is corresponding to the attribute sequence in Figure 4.

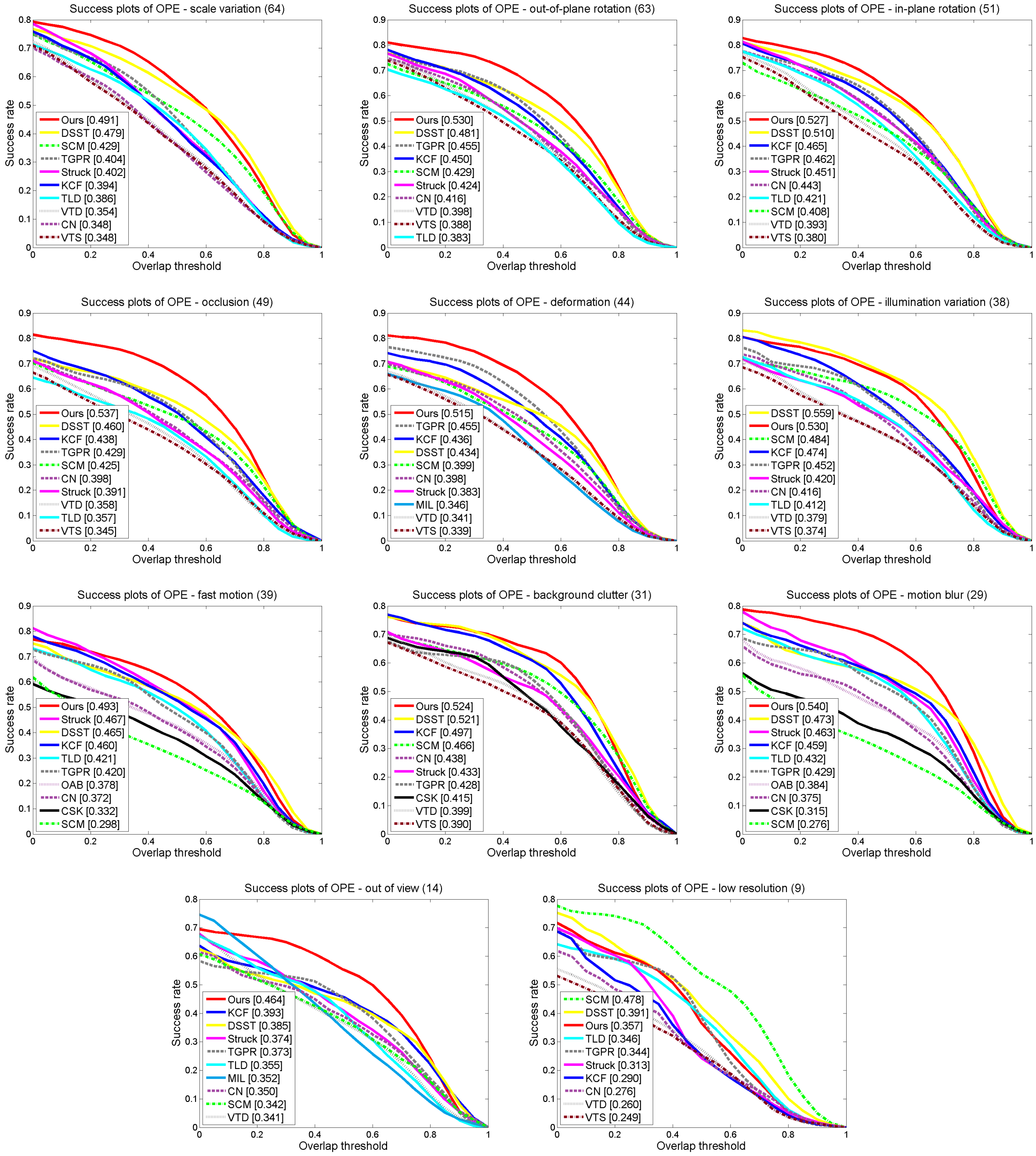
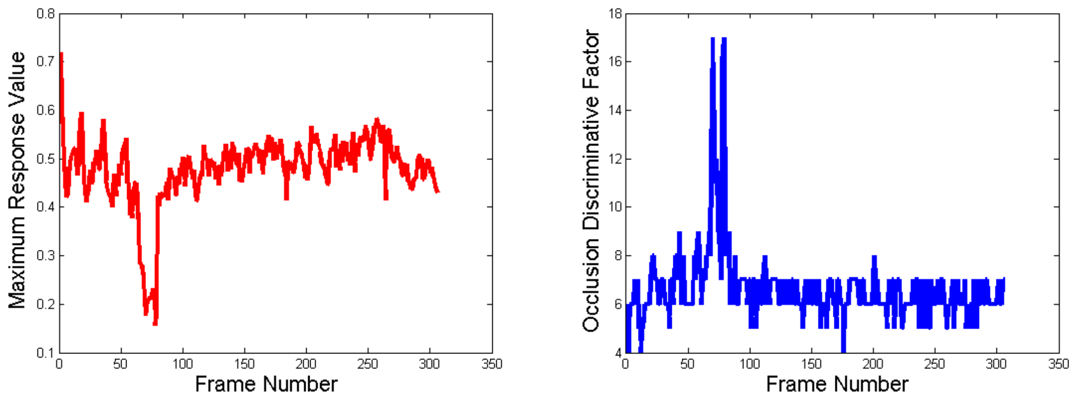


FIGURE 4. (color online) Success plots of different attributes over 100 sequences on the OTB-2015 dataset. The value appearing in the title denotes the number of videos associated with the respective attribute.

Among all the attributes, scale variation, low resolution, and occlusion subset performance are improved significantly, which shows that the scale estimation and occlusion detection scheme are effective. Deformation and out-of-view also perform well with a large improvement. The results indicate that the integration of saliency refiner is helpful for rectifying the inaccuracy caused by drift. Although our approach is not specially designed for out-of-plane rotation and motion blur, the proposed approach obtains very appealing performance on these challenging sequences.

(3) **Analysis of occlusion case.** Figure 5 shows the variation of the maximum response value and occlusion discriminative factor in *jogging1* sequence with occlusion. We can see from Figure 5(a) when the tracked target is occluded from frame 62 to 81, the maximum response value drops significantly, which can reflect the tracking confidence in some degree. From Figure 5(b), we can find that, in occlusion case, the occlusion discriminative factor is dramatically increased, which can reflect the target occlusion state to some extent. Therefore, we use maximum response value as the tracking confidence to activate the saliency refiner when its value is below a certain threshold. We further adjust the learning rate to maintain reliable target appearance when the occlusion occurs.



(a) (b)

FIGURE 5. The illustration of maximum correlation response value and occlusion discriminative factor in *jogging1* sequence

(4) **Influence of different parameters.**

Saliency Preprocessing Threshold. Figure 6 shows an illustration of the influence of different saliency preprocessing thresholds on the tracking performance. It can be found from Figure 6, when the threshold $\eta_t = 0.5$, we achieve the best performance and receive the largest precision score and success score. Smaller threshold η_t retains more interference information in the background of the saliency map, while larger η_t gets rid of some useful boundary information of the target.

Tracking Confidence Value. Table 2 gives the average CLE, average VOR and average FPS obtained under different tracking confidence values to evaluate their influences on the performance of our approach. The best result is illustrated in bold fonts. We can find from Table 2 that the best result is obtained by setting higher confidence value. However, it increases the number of saliency refinement and decreases the efficiency. Different tracking confidence values have little influence on the average VOR. We set the tracking confidence value to 0.45 to obtain better results and relatively high efficiency.

(5) **Analysis of different components.** Figure 7 shows that the saliency refiner, occlusion handling, and scale estimation can significantly improve the performance of the

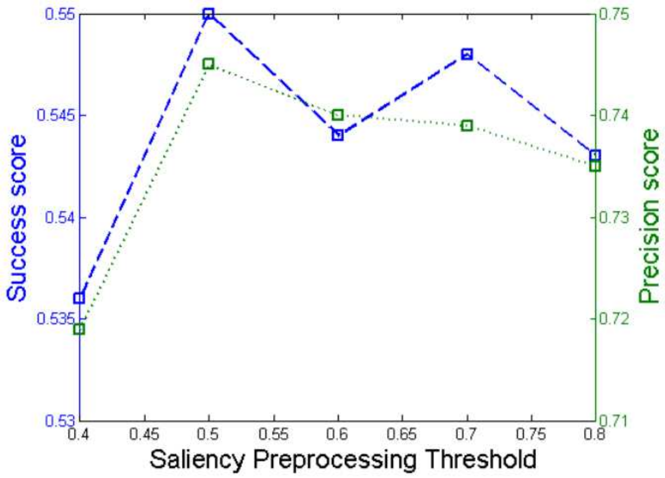


Figure 6. Illustration of the influence of different saliency preprocessing thresholds on tracking performance

Table 2. Comparisons on different tacking confidence values

Tracking Confidence Value	0.35	0.45	0.55	0.65	0.75
CLE (pixel)	36.4	35.8	36.0	35.1	**34.9**
VOR	0.55	**0.56**	0.55	**0.56**	**0.56**
Speed (FPS)	**54.4**	42.0	40.0	36.5	33.8

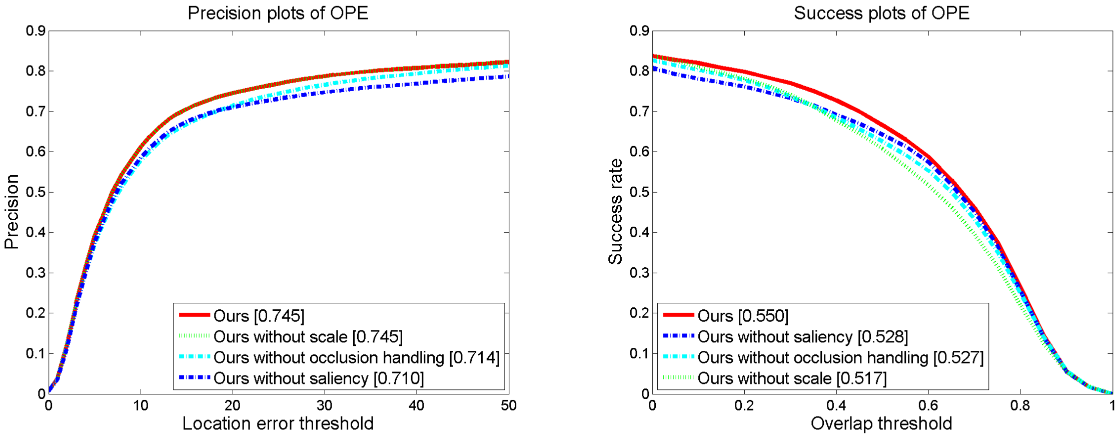
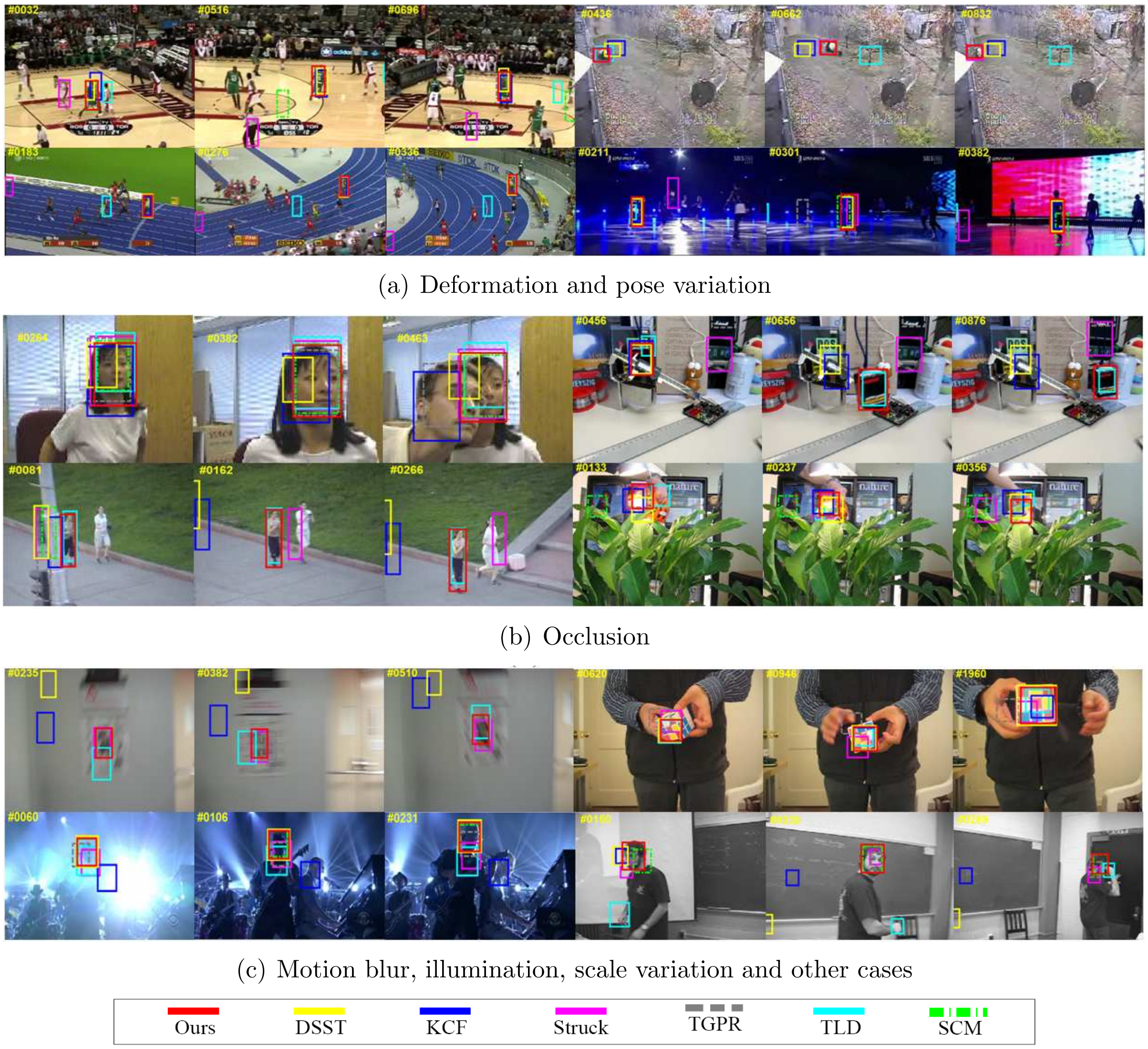


Figure 7. (color online) Comparison of different components of the proposed approach

proposed approach. In the success plot, our approach outperforms ours without saliency, ours without occlusion handling, and ours without scale on success score with the AUC by 4.2%, 4.4% and 6.4% in the overall performance respectively. Meanwhile, in the precision plot, our method also clearly outperforms the results in other cases. It is also obvious that both the saliency refiner and occlusion handling play a significant role in our work. The combination of the saliency refiner, occlusion handling, and scale estimation improves the overall performance and makes the tracking process robust.

4.3. **Qualitative evaluation.** Figure 8 shows the tracking results obtained by our approach and 6 representative trackers (DSST, KCF, Struck, TGPR, TLD, and SCM) on 12



(a) Deformation and pose variation

(b) Occlusion

(c) Motion blur, illumination, scale variation and other cases

Figure 8. (color online) Tracking results of our approach and 6 state-of-the-art trackers (# number on the left corner of each image denotes the frame index. From left to right and top to down are *basketball*, *panda*, *bolt*, *skating1*, *girl*, *box*, *jogging1*, *tiger2*, *blurowl*, *rubik*, *shaking*, and *freeman1* sequences)

challenging sequences. Figure 9 shows a frame-by-frame comparison of the CLE (in pixels) between our approach and these representative trackers on the challenging sequences. Our approach provides promising results on these sequences. We discuss the performance from three aspects as follows.

(1) **Deformation and pose variation.** As shown in Figure 8, the targets in *basketball*, *panda*, *bolt*, and *skating1* sequences undergo significant deformation and pose variation. In addition, the appearances of the targets in *basketball* and *skating1* sequences change drastically due to illumination variation, and the backgrounds are also cluttered. The target in *panda* sequence contains scale variation and occlusion, which makes the tracking task difficult. In *bolt* sequence, the DSST, KCF and our approach perform well throughout the tracking process, while other trackers exhibit drifting. In *basketball* sequence, the DSST, KCF, TGPR and our approach can track the target across the entire sequence. In *panda* sequence, the TLD, DSST, and KCF show a large deviation due to the partial

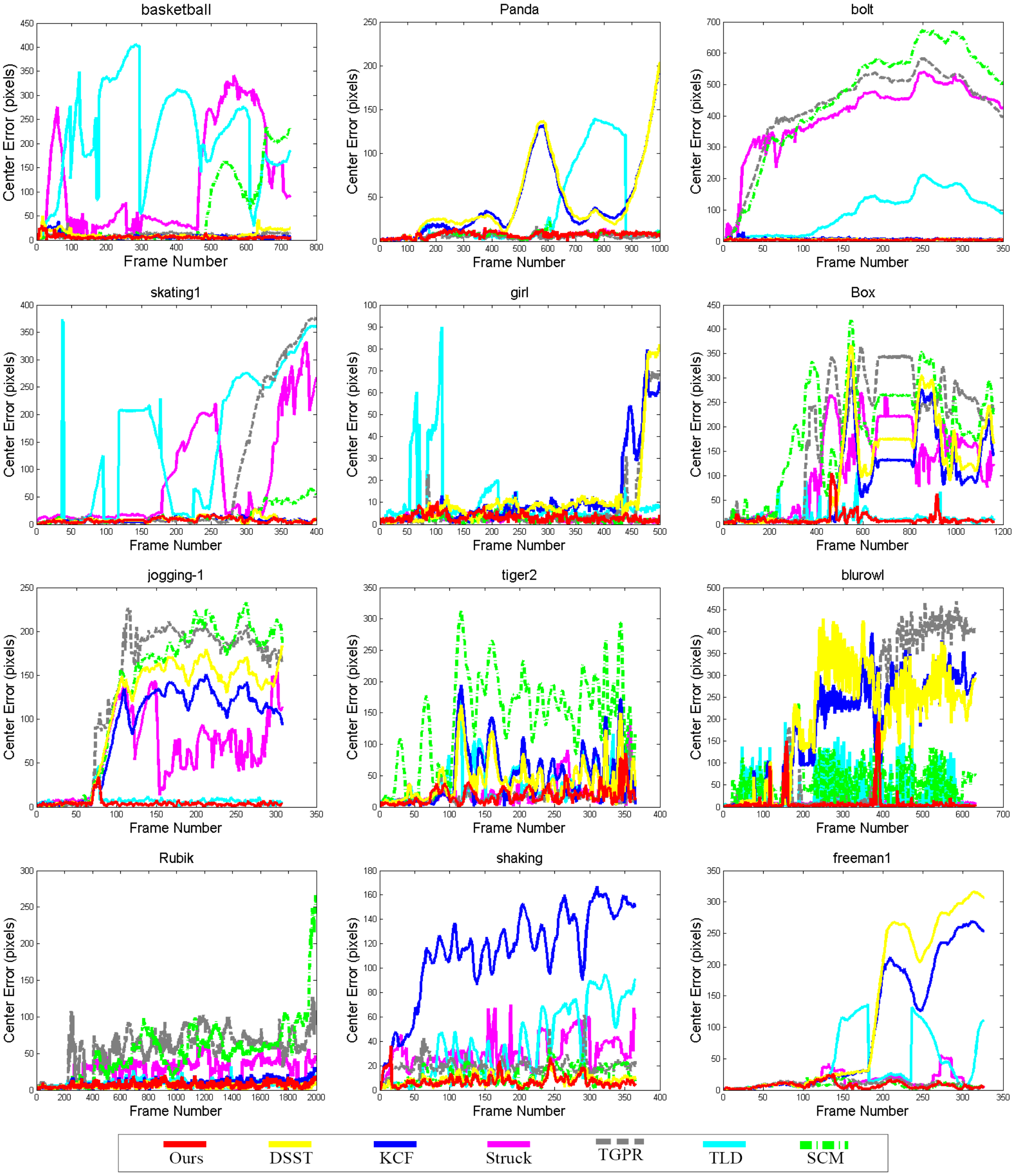


Figure 9. (color online) Center location error map (in pixels) between our approach and 6 state-of-the-art trackers

occlusion, while other trackers and our approach can track the target accurately. In *skating1* sequence, the DSST, KCF and our approach perform well throughout the whole video, and the TLD, Struck, TGPR and SCM exhibit drifting during the tracking process. Our approach works favorably well on these sequences. This is attributed to the integration of saliency information which is relatively insensitive to deformation and pose variation.

(2) **Occlusion.** The target objects in *girl* (#463), *box* (#456), *jogging1* (#81) and *tiger2* (#237, #356) sequences are partially occluded or even completely occluded. In

girl sequence, the TLD drifts to the background since #55. After a period of partial occlusion since about #430, most trackers tend to drift except our approach, SCM, TLD and Struck. In *jogging1* sequence, only TLD and our approach are able to track the target successfully after the occlusion occurs. The remaining trackers lose the target since about #80. In *box* sequence, only our approach and the TLD can track the target successfully, but the TLD lacks the ability to handle the scale variation accurately. Other trackers are drifting to the background due to the occlusion and the background clutter. In *tiger2* sequence, only the TGPR, Struck and our approach can track the target, while other trackers exhibit drifting during the tracking process. From the given tracking results, it can be found that our approach can successfully track most of the video frames, which indicates that the occlusion handling scheme in our approach is effective.

(3) **Motion blur, illumination, scale variation and other cases.** In *blurowl* sequence, the target undergoes fast motion and motion blur. Only Struck and our approach can adapt to such changes and obtain better results. The target in *rubik* sequence has scale variation and in-plane-rotation at the same time. The SCM, Struck and TGPR exhibit drifting in the whole process, and the KCF and TLD cannot adapt to scale changes. Only our approach can track and cover the target well. In *shaking* sequence, because the target suffers from significant illumination variation, all the other trackers fail to track the target except DSST and our approach. In *freeman1* sequence, the target suffers from in-plane-rotation and out-of-plane rotation. The TLD, KCF and DSST have a large drift to the background, but other trackers and our approach can track most of the frames. From Figure 8 and Figure 9, we can find that the introduction of saliency information achieves favorable tracking results. The integration of scale estimation, occlusion handling, and adaptive updating further boosts the overall performance and makes our approach robust in different situations.

5. **Conclusion.** In this paper, we propose a novel object tracking approach based on correlation filter with saliency refiner and adaptive updating to help alleviate the model drift problem frequently occurring during the tracking process. The visual saliency as prior information is relatively invariant to some appearance changes, such as deformation, rotation, and scale variation, which can serve as a saliency refiner to rectify small inaccuracy caused by drift. Furthermore, the scale variation problem is effectively solved by fusing a one-dimensional scale filter. Additionally, a model update strategy, which is based on the designed occlusion discriminative factor and the significant appearance change, is utilized by adaptively adjusting the learning rate. The experiments are conducted on 100 challenge sequences from a recent tracking benchmark dataset. The experimental results demonstrate that our proposed approach outperforms several state-of-the-art trackers while running efficiently at real-time speed.

Although the proposed tracking approach achieves better performance than several state-of-the-art trackers, it cannot handle the complete occlusion problem frequently occurring in long-term tracking tasks very well. In the future, we will investigate the re-detection component to recover the tracked target in case of tracking failures caused by long-term occlusions.

Acknowledgment. This work is supported by the National Natural Science Foundation of China (Grant Nos: 61772421, 61572400) and the Key Research Project of Baoji University of Arts and Sciences (No: ZK15032). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M. M. Cheng et al., Struck: Structured output tracking with kernels, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.38, no.10, pp.2096-2109, 2016.

[2] B. Babenko, M. 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, 2011.

[3] H. Grabner, M. Grabner and H. Bischof, Real-time tracking via on-line boosting, *Proc. of British Machine Vision Conference*, Edinburgh, UK, pp.47-56, 2006.

[4] D. S. Bolme, J. R. Beveridge, A. B. Draper et al., Visual object tracking using adaptive correlation filters, *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, USA, pp.2544-2550, 2010.

[5] J. F. Henriques, R. Caseiro, P. Martins et al., Exploiting the circulant structure of tracking-by-detection with kernels, *Lecture Notes in Computer Science, European Conference on Computer Vision*, Firenze, Italy, vol.7575, pp.702-715, 2012.

[6] J. F. Henriques, R. Caseiro, P. Martins et al., High-speed tracking with kernelized correlation filters, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.37, no.3, pp.583-596, 2015.

[7] J. Kwon and K. M. Lee, Visual tracking decomposition, *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, USA, pp.1269-1276, 2010.

[8] J. Kwon and K. M. Lee, Tracking by sampling trackers, *Proc. of IEEE International Conference on Computer Vision*, Barcelona, Spain, pp.1195-1202, 2011.

[9] J. Zhang, S. Ma and S. Sclaroff, MEEM: Robust tracking via multiple experts using entropy minimization, *Lecture Notes in Computer Science, European Conference on Computer Vision*, Zurich, vol.8694, pp.188-203, 2014.

[10] Z. Kalal, K. Mikolajczyk and J. Matas, Tracking-learning-detection, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.34, no.7, pp.1409-1422, 2012.

[11] C. Ma, X. Yang, C. Zhang et al., Long-term correlation tracking, *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, pp.5388-5396, 2015.

[12] A. Borji, D. N. Sihite and L. Itti, Salient object detection: A benchmark, *Lecture Notes in Computer Science, European Conference on Computer Vision*, Firenze, Italy, vol.7573, pp.414-429, 2012.

[13] J. Zhang, S. Sclaroff, Z. Lin et al., Minimum barrier salient object detection at 80 fps, *Proc. of IEEE International Conference on Computer Vision*, Santiago, Chile, pp.1404-1412, 2015.

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

[15] M. Danelljan, F. S. Khan, M. Felsberg et al., Adaptive color attributes for real-time visual tracking, *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, pp.1090-1097, 2014.

[16] F. S. Khan, R. M. Anwer et al., Color attributes for object detection, *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, USA, pp.3306-3313, 2012.

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

[18] M. Danelljan, G. Häger, F. Khan and M. Felsberg, Accurate scale estimation for robust visual tracking, *Proc. of the British Machine Vision Conference*, Nottingham, UK, pp.65.1-65.11, 2014.

[19] M. Danelljan, G. Häger, F. S. Khan et al., Discriminative scale space tracking, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.39, no.8, pp.1561-1575, 2017.

[20] D. Huang, L. Luo, M. Wen et al., Enable scale and aspect ratio adaptability in visual tracking with detection proposals, *Proc. of the British Machine Vision Conference*, Swansea, UK, pp.185.1-185.12, 2015.

[21] L. Bertinetto, J. Valmadre, O. Miksik et al., The importance of estimating object extent when tracking with correlation filters, *Pre-print Relating to VOT2015 Submission*, 2015.

[22] G. Zhu, J. Wang, Y. Wu et al., Collaborative correlation tracking, *Proc. of British Machine Vision Conference*, Swansea, UK, pp.184.1-184.12, 2015.

[23] V. Mahadevan and N. Vasconcelos, Biologically inspired object tracking using center-surround saliency mechanisms, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.35, no.3, pp.541-554, 2013.

[24] S. Hong, T. You, S. Kwak et al., Online tracking by learning discriminative saliency map with convolutional neural network, *International Conference on Machine Learning*, Lille, France, pp.597-606, 2015.

[25] P. Zhang, T. Zhuo, W. Huang et al., Online object tracking based on CNN with spatial-temporal saliency guided sampling, *Neurocomputing*, vol.257, pp.115-127, 2017.

[26] Y. Wu, Y. Sui and G. Wang, Vision-based real-time aerial object localization and tracking for UAV sensing system, *IEEE Access*, vol.5, pp.23969-23978, 2017.

[27] G. Zhu, J. Wang, Y. Wu et al., MC-HOG correlation tracking with saliency proposal, *Proc. of the 30th AAAI Conference on Artificial Intelligence*, Phoenix, Arizona, pp.3690-3696, 2016.

[28] J. S. Supancic and D. Ramanan, Self-paced learning for long-term tracking, *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Portland, OR, USA, pp.2379-2386, 2013.

[29] N. Dalal and B. Triggs, Histograms of oriented gradients for human detection, *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, pp.886-893, 2005.

[30] R. Strand, K. C. Ciesielski, F. Malmberg et al., The minimum barrier distance, *Computer Vision and Image Understanding*, vol.117, no.4, pp.429-437, 2013.

[31] W. Zhong, H. Lu and M. H. Yang, Robust object tracking via sparsity-based collaborative model, *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, USA, pp.1838-1845, 2012.

[32] J. Gao, H. Ling, W. Hu and J. Xing, Transfer learning based visual tracking with Gaussian processes regression, *Lecture Notes in Computer Science, European Conference on Computer Vision*, Zurich, vol.8691, pp.188-203, 2014.