

Journal of Electronic Imaging

JElectronicImaging.org

Correlation filter-based visual tracking via holistic and reliable local parts

Chunbao Li
Bo Yang

Correlation filter-based visual tracking via holistic and reliable local parts

Chunbao Li^a and Bo Yang^{a,b,*}

^aUniversity of Electronic Science and Technology of China, School of Computer Science and Engineering, Chengdu, China

^bSichuan Electronic Information Industry Technology Research Institute Co., Ltd., Chengdu, China

Abstract. Visual tracking is a challenging task in computer vision due to various appearance changes of the target. Although correlation filter-based trackers have achieved competitive results, they may easily lead to tracking failure because of the high sensitivity of correlation filter to occlusion. Part-based correlation filter trackers can deal with partial occlusion to some extent, but they may easily drift to the background in the case of fast motion or heavy occlusion. To better solve the above-mentioned problems, a kernelized correlation filter-based tracker that processes both holistic and reliable local parts is proposed. For local parts, reliable parts are identified by peak-to-sidelobe ratio. When all parts are unreliable, we propose to apply a sliding window on each part to generate patches, among which a reliable patch is identified, and the part is replaced by the reliable patch. In holistic level, holistic tracking is performed with the rough position voted by reliable local parts, and then the holistic tracking result is used to provide feedback for parts to update its scale and filter. Moreover, we propose to reset unreliable parts when the holistic tracking result is reliable. The experimental results illustrate that the proposed tracker outperforms those of several state-of-the-art trackers. © The Authors. Published by SPIE under a Creative Commons Attribution 4.0 Unported License. Distribution or reproduction of this work in whole or in part requires full attribution of the original publication, including its DOI. [DOI: [10.1117/1.JEI.28.1.013039](https://doi.org/10.1117/1.JEI.28.1.013039)]

Keywords: visual tracking; correlation filter; reliable local parts; unreliable parts reset.

Paper 180652 received Jul. 21, 2018; accepted for publication Jan. 22, 2019; published online Feb. 18, 2019.

1 Introduction

Visual object tracking has attracted much attention in computer vision and robotics communities, which enjoys a wide range of applications such as traffic control, medical imaging, surveillance, and auto-control systems. Given the initial state (e.g., position and extent) of a target object in the first image, the goal of tracking is to estimate the states of the target in the subsequent frames.^{1,2} Despite having achieved considerable progress over the past decade, effective modeling of the appearance of tracked objects remains a challenging problem due to visual appearance changes,^{3,4} such as illumination variation (IV), partial and heavy occlusion, background clutters (BC), motion blur (MB), deformation, and low resolution (LR). As a result, it remains a hot area of research to design a robust visual tracker.

To handle the above-mentioned visual appearance changes problem, many visual trackers have been proposed, which can be categorized into two classes according to the appearance modeling methods, i.e., generative methods and discriminative methods. Generative methods^{5,6} mainly concentrate on how to minimize the distance between candidates and the tracked target, whereas discriminative methods^{7,8} pose a visual tracking problem as a binary classification one in order to separate the target object from the background. Generative methods are based on templates or subspace models, such as mean shift or sparse coding-based visual trackers.^{9,10} These trackers incrementally learn visual representations for the foreground object region information while ignoring the influence of the background.¹⁰ For discriminative methods, correlation filter-based visual trackers

have become increasingly popular. The correlation filter can be trained quickly based on the property of the circulant matrix in the Fourier domain, so these trackers generally achieve high performance with low-computational load.^{11,12} In particular, Henriques et al.¹¹ proposed a visual tracking algorithm with kernelized correlation filters (KCF) by combining multichannel features with kernel trick, and experimental results show that the tracking performance was improved significantly. Not surprisingly, many recent visual trackers^{13,14} are developed based on correlation filter.

Nevertheless, due to the high sensitivity of correlation filter to occlusion, many correlation filter-based trackers may easily drift to the background and lead to tracking failure.^{15,16} In order to solve this occlusion problem, part-based correlation filter visual trackers have been proposed,^{12,15,17} which have been shown to have improved performance. However, there are still some deficiencies in the state-of-the-art part-based correlation filter trackers. First, these trackers can deal with partial occlusion and slight deformation to some extent, but they may easily drift to the background and fail to track the right target in subsequent frames when the target is undergoing heavy occlusion or severe deformation. Second, most of these trackers cannot deal with MB caused by shaking of the lens or fast motion (FM) well.

In this paper, we propose a KCF-based visual tracker via both holistic and reliable local parts (HR), abbreviated as KCF-HR tracker thereafter, which could handle the above-mentioned heavy occlusion, FM, and other challenging factors. The confidence metric of peak-to-sidelobe ratio (PSR) is used to measure how reliably a part can be tracked, and the estimated part is reliable when the PSR of it is greater than a given threshold. In the KCF-HR tracker, both holistic classifier and local reliable parts classifiers are used. The tracking results of all reliable parts are employed to vote

*Address all correspondence to Bo Yang, E-mail: yangbo@uestc.edu.cn

a rough position of the target, and then the target position and scale in the current frame are obtained by holistic tracking with the rough position. After that, the holistic tracking result is used to provide feedback for each part to update its scale and filter. In the above process, it can be noted that if a part is occluded or unreliable, it may lead to tracking failure if a fixed weight or high weight is assigned to the part. We propose to assign proper weight to each part by using the PSR, which is used to measure the signal peak intensity in response map, and then reliable parts can be identified as well. To deal with the situation where all parts are unreliable, we propose to apply sliding window on each part to generate several patches and all patches are tracked to find a reliable patch for each part, and then the part will be replaced by the reliable patch. When the holistic tracking result is reliable and the overlap between an unreliable part and the holistic target is less than a given threshold, the unreliable part is reset in the proposed tracker.

The main contributions of this work are summarized as follows. (1) We design a KCF-based collaborative tracker via holistic and reliable local parts. (2) A resetting unreliable parts method is proposed to ensure their reliability. (3) The sliding window method is applied to handle the case where the target moves outside the tracking window. (4) Experimental results on the OOTB² and OTB-100⁴ datasets show that the KCF-HR tracker could effectively deal with FM, MB, and occlusion.

The rest of the paper is organized as follows. In Sec. 2, the related work is briefly reviewed. In Sec. 3, the detailed description of the KCF-HR tracker is presented. In Sec. 4, experimental results and comparison with other state-of-the-art trackers are presented and analyzed. Finally, we conclude the paper in Sec. 5.

2 Related Work

As one of the most challenging problems in computer vision, visual tracking has attracted a lot of attention and a number of visual trackers^{12,18–22} have been proposed over the decade. In this section, we briefly review related works, with the main focus on the correlation filter-based trackers.

Correlation filter-based visual trackers have achieved promising results in recent years. For example, Bolme et al.²² first proposed a new type of correlation filter that is robust to several kinds of appearance variations by minimizing the output sum of squared errors. Henriques et al.¹¹ proposed the circulant matrices and multiple channels feature-based KCF tracker to further accelerate the development of correlation filtering. Sui et al.¹⁴ tracked the target by imposing an elastic net constraint on the correlation filter learning to learn a more discriminative filter. Despite having achieved considerable progress in accuracy and robustness, these global-based correlation filter trackers cannot perform well when the target is undergoing occlusion or deformation.

In order to deal with the above-mentioned problem, many part-based visual trackers have been proposed. For example, Liu et al.¹² divided the global target into multiple parts and modeled the object appearance by combining all adaptive weighted part classifiers. Li et al.¹⁵ estimated the target position and scale by identifying and exploiting reliable patches that can be tracked effectively through the whole tracking process. Xu et al.²³ proposed an efficient scale calculation method by dividing the target into four patches

and computing the scale factor with patch-based KCF trackers. Huang et al.²⁴ proposed to represent the target by a part space with two online learned probabilities to capture the structure of the target. Ding et al.¹⁶ proposed a quadrangle Gaussian training label matrix to incorporate the location and size estimation problem into one filtering operation and the location and size of the target is estimated by a weighted Bayesian inference framework. The main idea of these trackers is to get the final position and scale of the target by combining weighted responses of all parts. These trackers can deal with partial occlusion and slight deformation to some extent, but they may easily fail to track the right target in subsequent frames when the target is undergoing heavy occlusion or severe deformation.

To better solve these problems, coupled-layer-based trackers are proposed. For example, Chen et al.¹⁷ proposed an enhanced structural correlation filter (ESC)-based visual tracker. Zhao et al.²⁵ developed a tracker by combining the proposed discriminative global model and generative local model into a Bayesian inference framework for visual tracking. Zhang and Liu²⁶ proposed a coupled-layer-based tracker based on the work in Ref. 15 by using global tracking result in local layer to improve the performance of local patches. Akin et al.²⁷ proposed a deformable part-based correlation filter (DPCF) tracker that depends on coupled interactions between a global filter and several part filters, which is similar to our tracker. The difference between our method and the DPCF tracker is that the KCF-HR tracker estimates the scale of the target by using a discriminative filter on multiple resolutions of the searching area and resets unreliable parts by using PSR threshold of global tracking result while the DPCF tracker estimates the scale of the target by using the average distance between two reliable parts in successive frames and reinitializes the tracking system when the scaling occurs.

In recent years, many deep learning-based trackers have been proposed. For example, Qi et al.²⁸ proposed a hedged deep tracking (HDT) framework, which uses an adaptive decision learning algorithm to hedge several weak CNN trackers into a stronger one. Bertinetto et al.²⁹ proposed a fully-convolutional Siamese network (SiamFC)-based tracker and treat object tracking as a similarity learning problem. Valmadre et al.³⁰ improved the SiamFC tracker with redesigning the CNN network architecture in which the correlation filter is interpreted as a differentiable CNN layer. Furthermore, Danelljan et al.³¹ proposed an efficient convolution operator tracking scheme to counter the issues of computational complexity and overfitting for discriminative correlation filter-based trackers. The performance of the deep learning-based trackers has been greatly improved, but its tracking speed still needs to be improved.

3 KCF-HR Tracker

In this section, we present the KCF-HR tracker that consists of local part level and holistic level in detail. The overview of the tracker is shown in Fig. 1. First, the ground truth bounding box is divided into four parts in the initial frame. Then, in local part level, the part-based KCF tracking algorithm is used to track each part, and rough position of the target is voted by weighted reliable parts. In holistic level, holistic-based KCF tracking is implemented with the rough position. Finally, according to the reliable holistic tracking result,

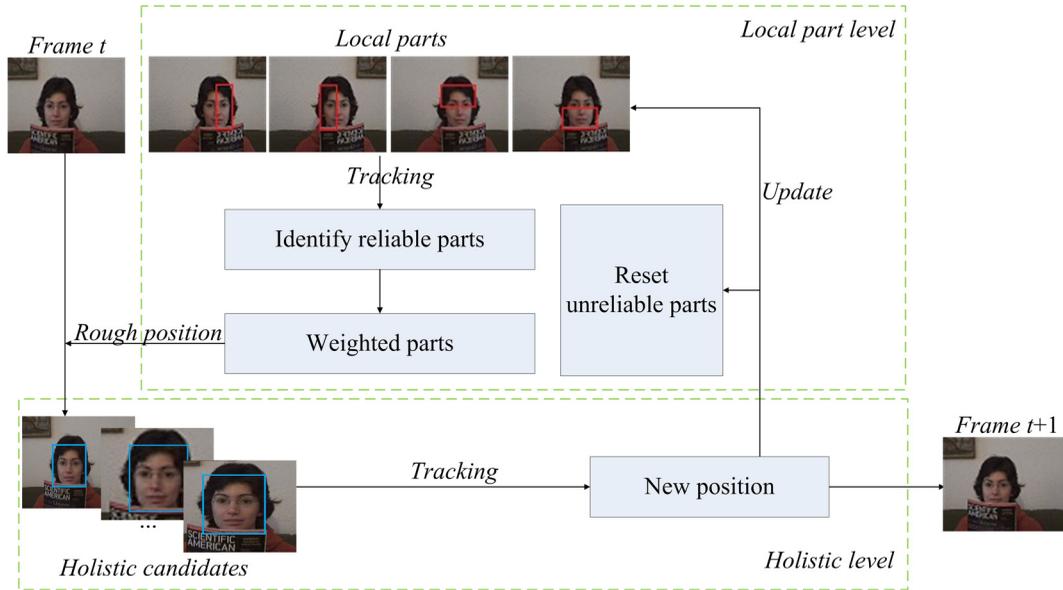


Fig. 1 The overall tracking flowchart of the KCF-HR tracker.

unreliable local parts are reset and holistic correlation filter and reliable parts correlation filters are updated.

3.1 KCF Tracker

In this section, we briefly reviewed the main idea of the KCF tracker¹¹ on which our method is built. As a discriminative method, KCF trains a classifier by taking advantages of the cyclic property and appropriate padding with a large number of densely sampled on a single image patch \mathbf{x} of size $W \times H$ centered around the target. Given a set of training samples and labels, the goal of training is to find the function $f(\mathbf{z}) = \mathbf{w}^T \mathbf{z}$ that minimizes the squared error over all the circular shifted image samples \mathbf{x}_i and their regression targets y_i ,

$$\min_{\mathbf{w}} \sum_i |f(\mathbf{x}_i) - y_i|^2 + \lambda \|\mathbf{w}\|^2, \quad (1)$$

where $i \in \{0, 1, \dots, W-1\} \times \{0, 1, \dots, H-1\}$, and λ is the regularization parameter used to control overfitting.

Mapping the inputs \mathbf{x}_i of the linear ridge regression to a nonlinear feature space $\phi(\mathbf{x})$ with kernel trick that is defined by the kernel $\kappa(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$ gets the nonlinear ridge regression that can be resolved as $f(\mathbf{z}) = \mathbf{w}^T \mathbf{z} = \sum_{i=1}^n \alpha_i \kappa(\mathbf{z}, \mathbf{x}_i)$, where $\kappa(\mathbf{z}, \mathbf{x}_i)$ denotes the dot-products of \mathbf{x}_i and \mathbf{x} , and the coefficient α can be expressed as

$$\alpha = \mathcal{F}^{-1} \left\{ \frac{\mathcal{F}(y)}{\mathcal{F}[\kappa(\mathbf{x}, \mathbf{x})] + \lambda} \right\}, \quad (2)$$

where \mathcal{F} and \mathcal{F}^{-1} denote the Fourier transform and its inverse, respectively; the vector α contains all the $\alpha(i)$ coefficients. In the KCF tracker, the model consists of the transformed classifier coefficients $\mathcal{F}(\alpha)$ and the target appearance $\hat{\mathbf{x}}$ that is learned over time.

In the tracking stage, the size of interesting image patch \mathbf{z} is cropped as the same with \mathbf{x} , and the confidence score of new patch \mathbf{z} are calculated as

$$y(\mathbf{z}) = \mathcal{F}^{-1} \{ \mathcal{F}[\kappa(\mathbf{z}, \mathbf{x})] \odot \mathcal{F}(\alpha) \}, \quad (3)$$

where \odot is the elementwise product. And the target position is detected by finding the coordinate at which $y(\mathbf{z})$ has the maximum value.

3.2 Local Part Level

In the initial frame, target image \mathbf{x} is divided into four parts, and the spatial layouts of these parts are shown in Fig. 2, where the size of each part is half of the target size. Obviously, when the target is partially occluded, the remaining visible parts can still provide reliable cues for tracking.

Then, the KCF tracker for each part is carried out by searching for the image part in each subsequent frame with appearance most similar to the part \mathbf{x}_i . To predict the tracking quality and estimate the reliability of a patch, we adopt the PSR as a confidence metric, which is widely used in signal processing to measure the signal peak strength in a correlation filter response map.¹⁵ Given a correlation filter response map $y(\mathbf{x}_i)$ of an image part \mathbf{x}_i , the PSR is calculated as

$$\text{PSR}(\mathbf{x}_i) = \frac{\max[y(\mathbf{x}_i)] - \mu_{\Phi}[y(\mathbf{x}_i)]}{\sigma_{\Phi}[y(\mathbf{x}_i)]}, \quad (4)$$

where Φ is the sidelobe area around the peak that is 15% of the response map area. μ_{Φ} and σ_{Φ} are the mean value and standard deviation of $y(\mathbf{x}_i)$ excluding the area Φ , respectively. From Eq. (4), we can observe that the PSR becomes

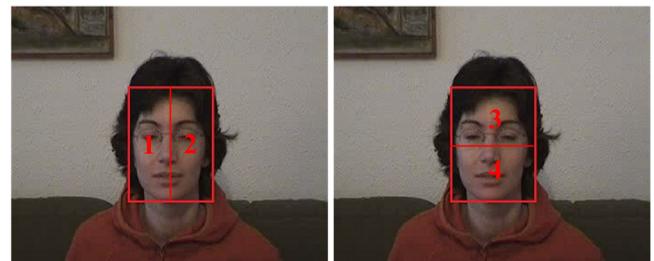


Fig. 2 The ground truth bounding box is divided into four parts.

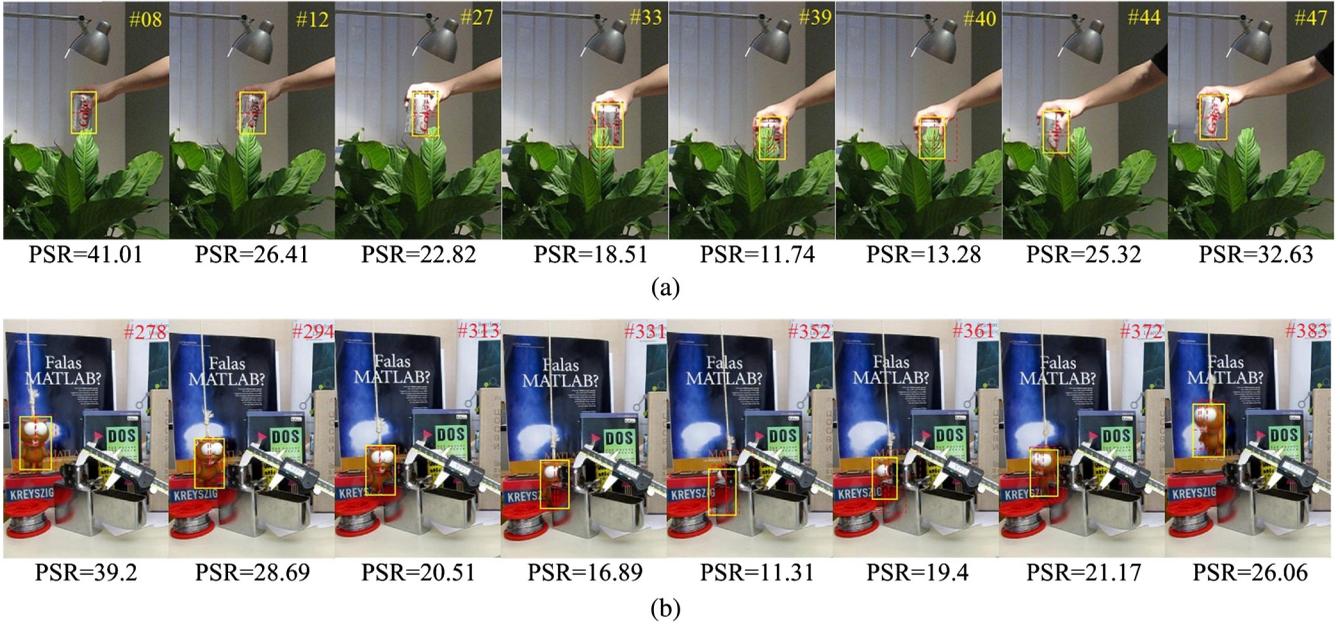


Fig. 3 The holistic tracking results and its corresponding PSR are shown in (a) and (b). For example, in (b), the PSR drops to 20.51 at frame 313 when the target is partial occluded, the PSR drops to 11.31 at frame 352 when the target is severe occluded, and the PSR of frame 372 increases to 21.17 when the target is redetected.

large when the response peak value is strong. As shown in Fig. 3, a high PSR value indicates that the target object is tracked accurately, the PSR value is lower than 20 indicates that the target object is under severe occlusion, MB, or other appearance variations. Therefore, the tracked part is considered to be unreliable if its PSR is less than the threshold, which is set to 20 in this paper.

The rough target position can be computed by Hough voting scheme³² with the normalized weight w_i of each reliable part. For each part \mathbf{x}_i in the current frame, the normalized weight w_i can be computed as

$$w_i = \frac{\text{PSR}(\mathbf{x}_i)}{\sum_1^n \text{PSR}(\mathbf{x}_i)}, \quad (5)$$

where n is the number of reliable parts. Then, the rough target position pos_r can be defined as

$$\text{pos}_r = \sum_{i=1}^n w_i \text{pos}_i, \quad (6)$$

where pos_i is obtained from the tracking result of each reliable part \mathbf{x}_i .

When the target moves outside the tracking window due to FM, reappearance after occlusion or drifting away, all parts will be not reliable, a sliding window method is proposed to generate several patches for each part, and then each patch is tracked with the KCF tracker to find a reliable patch. The layout of generated patches by sliding window method is shown in Fig. 4. The size of each window is equal to its corresponding part, and the step is half width or height of its corresponding part. For example, a holistic target image with size of $W \times H$ and position (p_l, p_c) , the size and position of part 2 is $W/2 \times H$ and $(p_l, p_c + W/4)$,

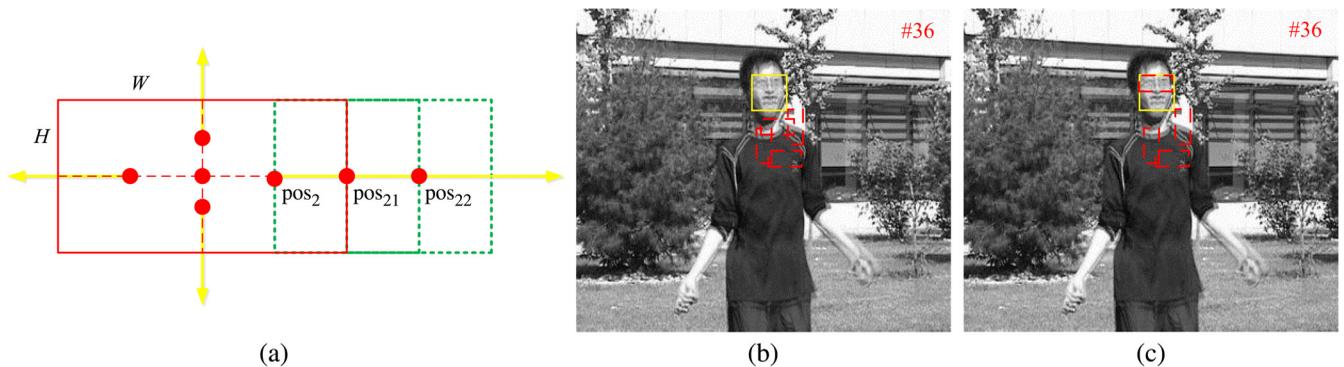


Fig. 4 The layout of generated patches by sliding window method is shown in (a). The window size and step of four parts are $\{W/2 \times H, W/2 \times H, W \times H/2, W \times H/2\}$ and $\{W/4, W/4, H/4, H/4\}$, respectively, e.g., positions of part 2 and generated patches are pos_2 , pos_{21} , and pos_{22} , respectively. The direction of sliding window of each part is indicated by the arrow. (b) and (c) are four local parts and four local parts that are replaced by generated reliable patches, respectively.

respectively. The generated m patches share the same size with its corresponding part 2, and positions of all patches are $\{(p_l, p_c + W/2), (p_l, p_c + W \times 3/4), \dots, [p_l, p_c + W \times (m + 1)/4]\}$. Then, the local KCF tracker is performed on all generated patches of part 2. If a generated patch is reliable, part 2 will be replaced by the generated reliable patch. In the same way, we can get new reliable local patches. Finally, the holistic-based KCF tracker is employed on the new position voted by Hough voting schema, and all replaced parts will be restored in the case of holistic tracking result is unreliable. If all local parts are not replaced by generated patch, holistic tracking will be performed at the position of the previous frame, and the tracking result with the maximum PSR is the final holistic result in the current frame.

3.3 Holistic Level

According to the size s_g of holistic target image in the last frame and the rough position voted by reliable parts, the holistic-based KCF tracker is carried out. Following Li and Zhu,³³ the holistic tracking is applied on multiple resolutions of the searching area to estimate changes in the target size. The scaling pool is defined as $s_p = \{t_1, t_2, \dots, t_k\}$, k samples $\{t_i s_g | t_i \in s_p\}$ are extracted that centered at the previous target location. The bilinear-interpolation strategy is employed to resize all samples into the fixed size s_g . The sample with the maximum PSR is used as the holistic tracking result.

If the holistic tracking result is reliable, a reset unreliable parts method is employed on local part level. The overlap of i 'th local part with holistic target image is given as

$$op_i = \frac{B_i \cap B_g}{B_i}, \quad (7)$$

where B_i is the bounding box of the i 'th part, B_g is the bounding box of the holistic target, and op_i indicates the ratio of the intersection of the i 'th part and holistic target image to the area of i 'th part. Each unreliable part of $op < 0.5$ will be reset when the holistic tracking result is reliable. Figure 5 is an example of resetting unreliable parts.

3.4 Adaptive Model Updating

During tracking, it is important to adaptively update parts and holistic correlation filters, because the target object's appearance may undergo significant changes such as MB,



Fig. 5 An example of resetting unreliable parts. The PSR values of parts 1, 2, 3, and 4 are 7.67, 15.95, 39.06, and 12.17, respectively. Unreliable parts 1, 2, and 4 with low overlap are shown in (a) and the holistic tracking result is reliable with $PSR = 22.74$. So, parts 1, 2, and 4 should be reset. (b) New parts after reset.

BC, partial or severe occlusion, deformation, and rotation. So, we update parts and holistic correlation filters properly to further improve the robustness and decrease the risk of drifting when the target appearance suffers variations.

For holistic level, using the current observations $\hat{\mathbf{x}}$ and the estimated coefficients $\hat{\alpha}$ in frame t , the correlation filter is updated as

$$\hat{\alpha}^t = \begin{cases} (1 - \gamma)\hat{\alpha}^{t-1} + \gamma\hat{\alpha} & \text{if } PSR \geq \vartheta \\ \hat{\alpha}^{t-1} & \text{otherwise} \end{cases}, \quad (8)$$

$$\hat{\mathbf{x}}^t = \begin{cases} (1 - \gamma)\hat{\mathbf{x}}^{t-1} + \gamma\hat{\mathbf{x}} & \text{if } PSR \geq \vartheta \\ \hat{\mathbf{x}}^{t-1} & \text{otherwise} \end{cases}, \quad (9)$$

where $\hat{\alpha}^t$ and $\hat{\alpha}^{t-1}$ are the coefficient estimated in frame t and $t - 1$, $\hat{\mathbf{x}}^t$ and $\hat{\mathbf{x}}^{t-1}$ denote the tracked target appearance. γ is the learning rate and ϑ is a given PSR threshold, which is used to determine whether the tracking model needs to be updated. In order to reduce the impact of the surrounding background, the update is stopped when the PSR is smaller than ϑ .

Moreover, all reliable parts are updated in each frame to guarantee the robustness and accuracy of all parts, and the previous filters are used for the unreliable parts. The scales of all reliable parts need to be updated according to the scale change of the holistic target image, when holistic tracking result is reliable. Using the current observations $\hat{\mathbf{x}}_i$ and the estimated coefficients $\hat{\alpha}_i$ of part i in frame t , each part can be updated as

$$\hat{\alpha}_i^t = \begin{cases} (1 - \gamma)\hat{\alpha}_i^{t-1} + \gamma\hat{\alpha}_i & \text{if } PSR \geq \vartheta \\ \hat{\alpha}_i^{t-1} & \text{otherwise} \end{cases}, \quad (10)$$

$$\hat{\mathbf{x}}_i^t = \begin{cases} (1 - \gamma)\hat{\mathbf{x}}_i^{t-1} + \gamma\hat{\mathbf{x}}_i & \text{if } PSR \geq \vartheta \\ \hat{\mathbf{x}}_i^{t-1} & \text{otherwise} \end{cases}, \quad (11)$$

where i denotes the i 'th reliable part, γ is the learning rate. Finally, the overall KCF-HR tracking algorithm is summarized into Algorithm 1.

4 Experiments

In this section, experiments are performed on two frequently used public benchmark datasets OOTB^{2,11,17} and OTB-100.^{27,29,30} For better evaluation and analysis of the strength and weakness of tracking algorithms, both of these datasets are classified into 11 attributes,^{2,4} including BC, deformation (DEF), FM, IV, LR, MB, occlusion (OCC), in-plane rotation (IPR), out-of-plane rotation (OPR), out-of-view (OV), and scale variation (SV). One sequence may be annotated with many attributes, and some attributes occur more frequently than others, for example, IPR and OPR.⁴ The ground truth of these benchmark datasets gives the position and target size of the target in each frame.

4.1 Experimental Configuration

4.1.1 Implementation details

The KCF-HR tracker is implemented in MATLAB on a regular PC with Intel i5-2450M CPU (2.50 GHz) and 4 GB memory. In the proposed tracker, most of the parameters are the same as the KCF tracker, the details are as follows.

Algorithm 1 The proposed KCF-HR tracker.

Input: Previous target position p_{t-1} ; holistic model $\hat{\alpha}^{t-1}$ and $\hat{\mathbf{x}}^{t-1}$; local parts model $\hat{\alpha}_i^{t-1}$ and $\hat{\mathbf{x}}_i^{t-1}$; frame F_t

Output: Estimated target position p_t ; updated holistic model $\hat{\alpha}^t$ and $\hat{\mathbf{x}}^t$; local parts model $\hat{\alpha}_i^t$ and $\hat{\mathbf{x}}_i^t$.

- 1: Divide the target into four parts, as shown in Fig. 2;
- 2: **for** all parts **do**
- 3: Perform KCF tracking at the last position using Eq. (3);
- 4: Calculate and identify reliable parts with Eq. (4);
- 5: **end for**
- 6: **if** number of reliable parts is 0 **then**
- 7: Generate patches using sliding window method and track each patch with KCF using Eq. (3);
- 8: Calculate and identify reliable parts with Eq. (4);
- 9: **end if**
- 10: Compute each reliable part' weight w_i^t with Eq. (5) and vote the rough position with Eq. (6);
- 11: Perform holistic KCF tracking on multiple resolutions of the target to get target position p_t and PSR_h ;
- 12: **if** $\text{PSR}_h > \vartheta$ **then**
- 13: Update holistic model according to Eqs. (8) and (9);
- 14: **for** all parts **do**
- 15: Calculate each unreliable part overlap with Eq. (7) and reset the part where the overlap < 0.5 ;
- 16: Update local part model according to Eqs. (10) and (11);
- 17: **end for**
- 18: **end if**

The Gaussian kernel parameter σ is set to 0.5 and the learning rate γ is set to 0.01. Eleven-channel color naming and 31-channel HoG features are used in our experiments; the orientation bin number and the cell size of HoG are 9 and 4×4 , respectively. Particularly, the cell size 2×2 rather than 4×4 for each part is used when the size of a part is below 40×40 pixels. Typically, the searching window used to train the discriminative correlation filter should be larger than the given target. Therefore, the sizes of searching window of the holistic target and four local parts are set to 2.5 and 2 times that of the target, respectively. For the holistic level, the scaling pool $s_p = \{0.985, 0.99, 0.995, 1, 1.005, 1.01, 1.015\}$. The number m of shifted patch is set to 4. The threshold ϑ is used to determine whether the part is reliable or the model is updated is set to 20. All the above-mentioned parameters are fixed throughout the experiments.

4.1.2 Evaluation methodology

The precision and success rates are two widely used evaluation metrics for quantitative analysis.²⁻⁴ The precision plot refers to the center location error, which is defined as the average Euclidean distance between the center locations of the tracked targets and the manually labeled ground truths. The success plot is defined as the bounding box overlap, and the overlap score can be computed with $S = \frac{|B_t \cap B_g|}{|B_t \cup B_g|}$, where B_t denotes the bounding box of the tracked result, B_g denotes the bounding box of ground truth, $|\cdot|$ is the number of pixels of the regions, and \cap and \cup represent the intersection and union of two regions. In this paper, the results of one-pass evaluation (OPE) are shown. OPE means running the tracking algorithm throughout a test sequence with initialization from the ground truth position in the first frame and reporting the average precision and success rate.² Moreover, the performance of a tracking algorithm may become much better or worse when it is initialized with different initialization or at a different start frame. So the robustness of the proposed tracker is evaluated in two aspects including temporal robustness evaluation (TRE) and spatial robustness evaluation (SRE). TRE and SRE are implemented by perturbing the initialization temporally (i.e., start at different frames) and spatially (i.e., start by different bounding boxes), respectively.²

4.2 Quantitative Comparisons**4.2.1 Comparison with correlation filter-based trackers**

The KCF-HR tracker has been quantitatively compared with seven representative and competitive correlation filter-based trackers including DPCF,²⁷ KCF,¹¹ RPT,¹⁵ ESC,¹⁷ SAMF,³³ Staple,³⁴ and DSST.³⁵ Among them, RPT is part-based tracker; DPCF and ESC are holistic-part based trackers; Staple and SAMF utilize complementary features such as HoG feature and color feature; SAMF and DSST pay more attention to the estimation of the target scale. All the above correlation filter-based trackers were proposed in recent years and outperformed other correlation filter-based trackers. Figure 6 shows the overall performance of these trackers using the OPE plots on the OOTB dataset, and the values in square brackets indicate the precision with a threshold of 20 pixels in precision plot and the area under curve (AUC) value in success plot. From Fig. 6, we can observe that the KCF-HR tracker achieves success score of 0.610 and precision score of 0.824 and outperforms other competitive correlation filter-based trackers in both measures. The KCF-HR tracker exhibits improvements in the success and precision scores by 2.3%/1.0% and 2.3%/3.1%, respectively, compared to the SAMF and staple trackers. Compared to the ESC and DPCF trackers, the performance gain is 5.1%/1.4% in terms of success scores. Compared to the original KCF tracker, the KCF-HR tracker exhibits improvements in the success and precision scores by 7.6% and 11.3%, respectively. The reason why the performance of the KCF-HR tracker is significantly improved lies in the effective holistic and reliable local parts KCF-based tracking schema, which is tracked mainly by identifying reliable parts and resetting unreliable parts.

Moreover, robustness is another important metric for evaluating the performance of visual trackers. So, we compare

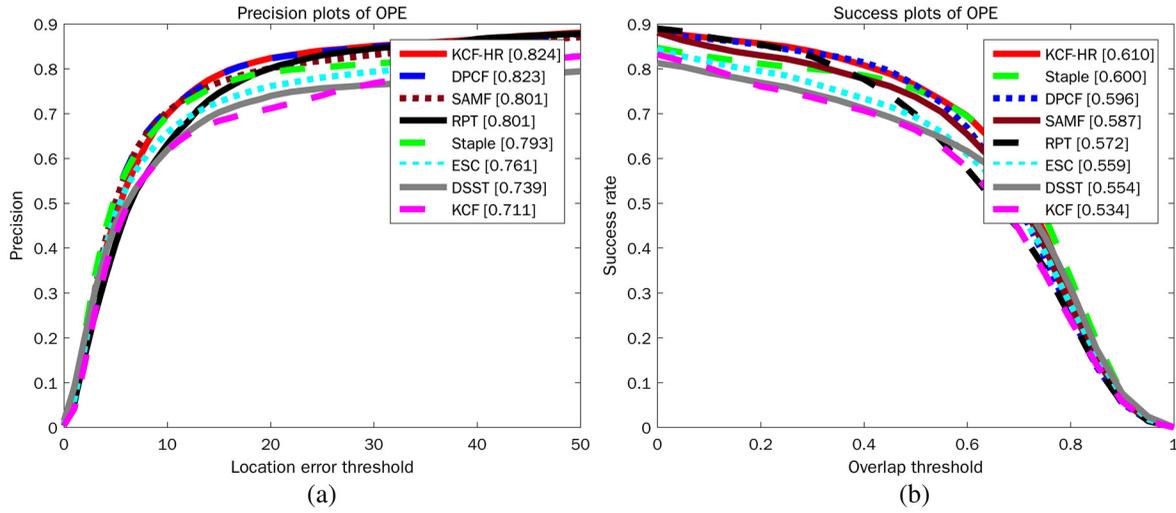


Fig. 6 The (a) precision and (b) success plots using the OPE for the KCF-HR tracker and seven other correlation filter-based trackers on the OOTB dataset.

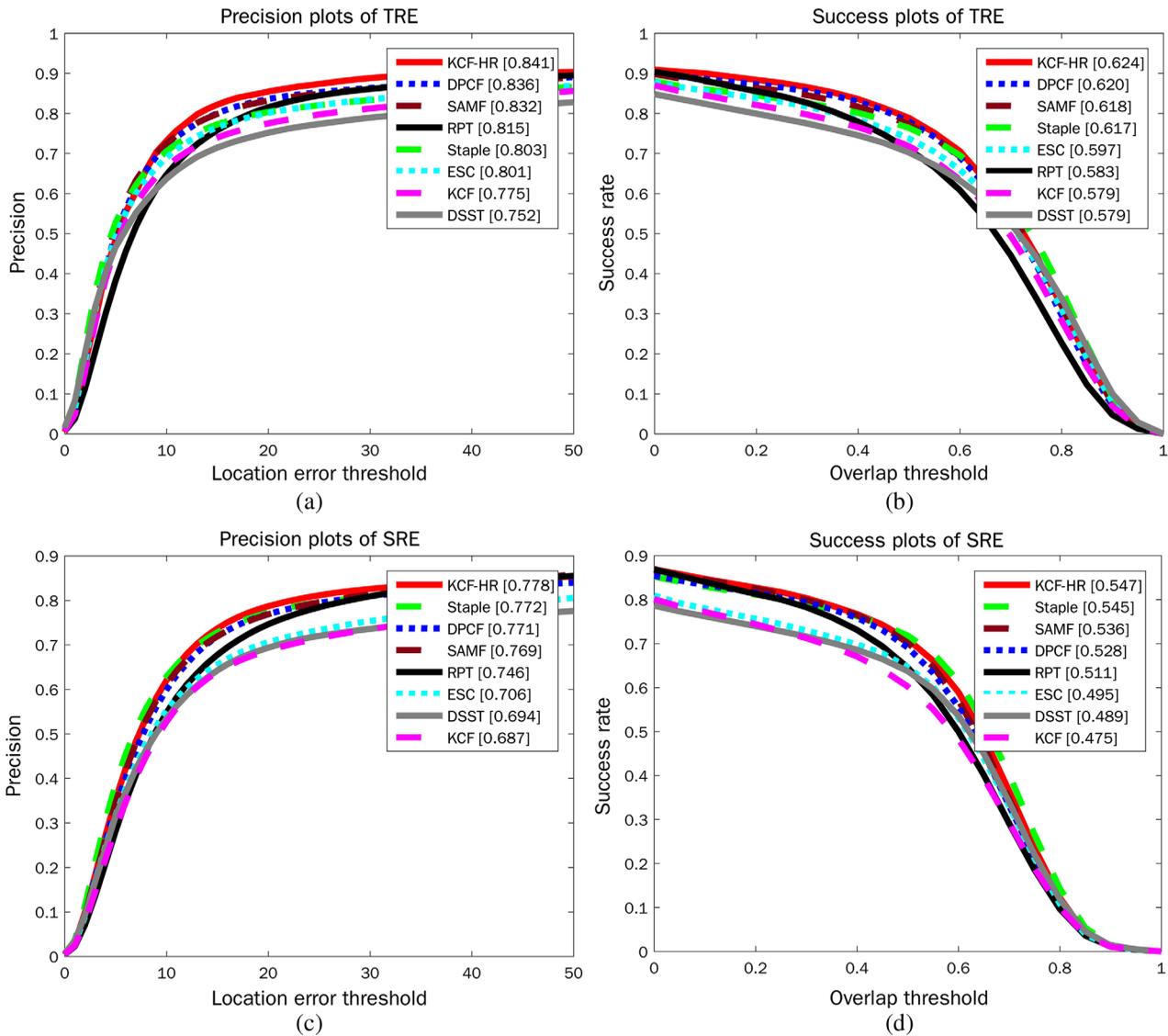


Fig. 7 The precision and success plots of the KCF-HR tracker and seven other correlation filter-based trackers on the OOTB dataset using (a) and (b) the TRE and (c) and (d) the SRE.

the KCF-HR tracker with the above-mentioned seven correlation filter-based trackers on TRE and SRE, and the experimental results on the OOTB dataset are shown in Fig. 7. In the precision plots for TRE and SRE, our KCF-HR tracker performs favorably compared to other trackers with scores of 0.841 and 0.778, respectively. Similarly, in the success plots for TRE and SRE, the proposed tracker takes the top place with scores of 0.624 and 0.547, respectively, which are better than those of the SAMF and DPCF tracker. However, due to the interference of more background information during initialization and tracking failure caused by model updating, the average SRE results drop significantly, as shown in Figs. 7(c) and 7(d).

4.2.2 Comparison with state-of-the-art trackers

We compared the KCF-HR tracker with the other 16 trackers on the OOTB and OTB-100 datasets. These state-of-the-art trackers can be broadly categorized into three classes:

(a) correlation filter-based trackers including the above-mentioned trackers, CSK¹⁸ and CN;³⁶ (b) single or multiple online classifiers-based trackers, such as MIL,³⁷ TLD,³⁸ SCM,³⁹ and Struck;⁴⁰ (c) deep convolutional neural networks (CNNs)-based trackers, including SiamFC,²⁹ CFNet,³⁰ and HDT.²⁸ Figure 8 shows the experimental results of the top 10 trackers with OPE. The KCF-HR tracker achieves success scores of 0.610 and 0.572 and precision scores of 0.824 and 0.791 for the two datasets. Evidently, the KCF-HR tracker outperforms other state-of-the-art trackers except the HDT tracker in terms of precision scores and provides comparable performance to the CNNs-based trackers in terms of success scores. As shown in Table 1, the KCF-HR tracker operates at an average speed of 6.5 frames per second (fps) on the OTB-100 dataset, which is significantly faster than the HDT tracker (1.3 fps) and slightly faster than the CFNet tracker (5.8 fps). Furthermore, compared to the KCF tracker, the KCF-HR tracker exhibits improvements in the success

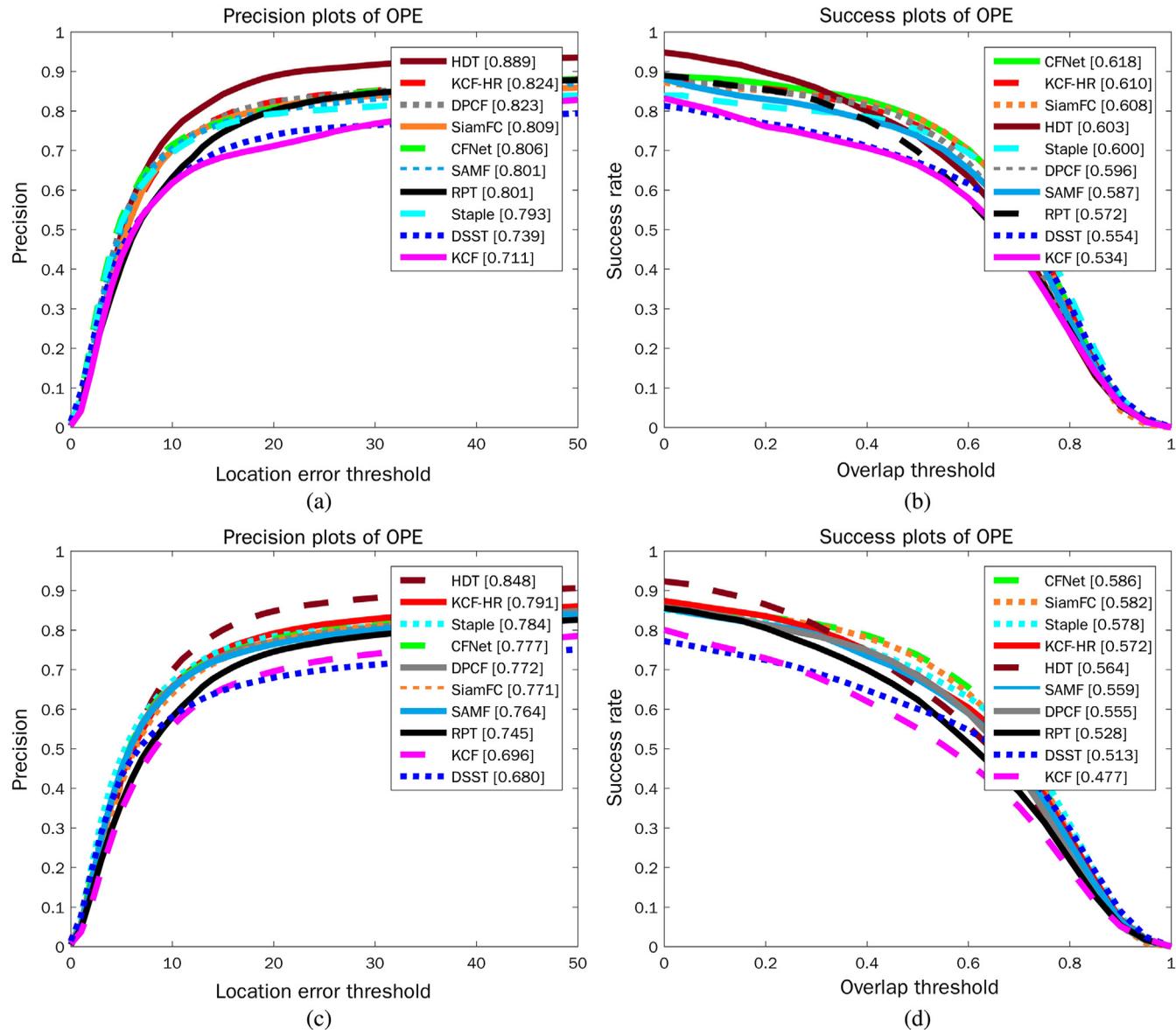


Fig. 8 The precision and success plots using the OPE for the KCF-HR tracker and the other top nine best performing trackers on [(a) and (b)] the OOTB and [(c) and (d)] OTB-100 datasets.

Table 1 The running speed of the KCF-HR tracker and the other top nine best performing state-of-the-art trackers on the OTB-100 dataset. The first, second, and third best results are highlighted as bold italics, bold, and italics, respectively, in each column.

	KCF-HR	HDT	SiamFC	CFNet	Staple	DPCF	RPT	SAMF	KCF	DSST
Speed (fps)	6.5	1.3	9.2	5.8	22.5	7.9	1.8	9.6	<i>82.5</i>	<i>25.2</i>

and precision scores by 9.5% and 9.5% on the OTB-100 dataset.

4.2.3 Speed analysis

We compared the tracking speed of our KCF-HR tracker and the other top nine best performing state-of-the-art trackers on the OTB-100 dataset. All evaluated trackers run on a PC without the hardware acceleration of GPU computation. The tracking speeds of other trackers are shown in Table 1. We can observe that the KCF-HR tracker runs at 6.5 fps using the nonoptimized single-thread MATLAB code, which is more than three times that of the reliable patches-based RPT tracker. Compared to the global-local correlation filters-based DPCF tracker, the proposed KCF-HR tracker has achieved encouraging performance, but its tracking speed still need to be improved. Note that the major computational cost of the KCF-HR tracker is the tracking of each part and the estimation of the target scale, which can be easily extended to a parallel implementation to optimize its efficiency.

4.2.4 Attribute-based performance

To evaluate the performance of the KCF-HR tracker under various appearance changes, we analyzed the performance of the tracker and the other top nine best performing state-of-the-art trackers on the OTB-100 dataset. Each sequence in the OTB-100 dataset is annotated with at least one of 11 attributes. Tables 2 and 3 show the precision scores at

the center location error threshold = 20 pixels and the success scores of AUC, respectively, regarding these challenging attributes. As shown in these tables, the KCF-HR tracker performs better than other competing trackers (except HDT) in most scenarios in terms of precision scores, and the KCF-HR tracker also provides comparable performance to the CFNet tracker in terms of success scores, which can be attributed to the use of two-level (holistic level and local part level) schema. Compared to the global-local-based DPCF tracker, the KCF-HR tracker performs better than DPCF in most scenarios including FM, MB, IPR, and OCC. In particular, the KCF-HR tracker outperforms other state-of-the-art trackers in the case of MB or OCC, which can be attributed to the effective resetting unreliable parts method and model updating schema.

4.2.5 Component analysis

To better understand the contribution of each component of our KCF-HR tracker, we performed the experiments by removing one of the related tracker modules from the tracker. First, we build a tracker KCF-HR-part by dividing the target image into four nonoverlapping equal-sized parts and keeping the other components unchanged. Second, the KCF-HR-without-shift tracker is implemented without generating patches by sliding window method. Third, in the KCF-HR-without-reset tracker, unreliable parts are not reset during tracking. The experimental results of these trackers on the OTB-100 dataset are shown in Fig. 9. The success score

Table 2 The precision scores of the KCF-HR tracker and the other top nine best performing state-of-the-art trackers at the center location error threshold = 20 pixels on the OTB-100 dataset. The column headers indicate the attributes and its number of image sequences. The first, second, and third best results are highlighted as bold italics, bold, and italics, respectively, in each column.

Tracker	BC-31	DEF-44	FM-39	IV-38	LR-9	MB-29	OCC-49	IPR-51	OPR-39	OV-14	SV-64	All
KCF-HR	0.738	<i>0.737</i>	<i>0.746</i>	0.767	0.699	<i>0.748</i>	<i>0.785</i>	<i>0.775</i>	<i>0.766</i>	0.657	<i>0.748</i>	<i>0.791</i>
HDT	<i>0.844</i>	<i>0.821</i>	<i>0.817</i>	0.720	<i>0.887</i>	<i>0.789</i>	<i>0.774</i>	<i>0.844</i>	<i>0.805</i>	0.663	<i>0.808</i>	<i>0.848</i>
SiamFC	0.690	0.690	0.743	0.736	<i>0.900</i>	0.705	0.722	0.742	0.756	<i>0.669</i>	0.735	0.771
CFNet	0.731	0.669	<i>0.757</i>	0.757	<i>0.850</i>	0.745	0.713	<i>0.803</i>	<i>0.761</i>	0.650	<i>0.744</i>	0.777
Staple	0.749	<i>0.751</i>	0.710	<i>0.782</i>	0.695	0.699	0.728	0.768	0.738	<i>0.668</i>	0.727	<i>0.784</i>
DPCF	<i>0.782</i>	0.729	0.707	<i>0.808</i>	0.711	<i>0.753</i>	<i>0.740</i>	0.737	0.754	0.610	0.724	0.772
RPT	<i>0.793</i>	0.670	<i>0.746</i>	<i>0.774</i>	0.712	0.721	0.640	0.762	0.705	0.598	0.714	0.745
SAMF	0.715	0.679	0.702	0.741	0.766	0.683	0.734	0.745	0.747	<i>0.673</i>	0.722	0.764
DSST	0.704	0.542	0.552	0.721	0.649	0.567	0.597	0.691	0.644	0.481	0.638	0.680
KCF	0.713	0.617	0.621	0.719	0.671	0.601	0.630	0.701	0.677	0.501	0.633	0.696

Table 3 The success plot's AUC scores of the KCF-HR tracker and the other top nine best performing state-of-the-art trackers. The column headers indicate the attributes of the OTB-100 dataset and its number of image sequences. The first, second, and third best results are highlighted as bold italics, bold, and italics, respectively, in each column.

Tracker	BC-31	DEF-44	FM-39	IV-38	LR-9	MB-29	OCC-49	IPR-51	OPR-39	OV-14	SV-64	All
KCF-HR	0.556	<i>0.529</i>	0.570	0.560	0.383	0.588	0.544	<i>0.554</i>	0.548	0.497	<i>0.518</i>	0.572
HDT	0.578	0.543	<i>0.568</i>	0.535	0.401	<i>0.574</i>	0.528	0.551	0.533	0.472	0.486	0.564
SiamFC	0.523	0.506	<i>0.568</i>	0.568	0.618	0.550	0.543	0.557	0.558	0.506	0.552	0.582
CFNet	0.543	0.492	0.583	<i>0.574</i>	0.582	0.584	0.536	0.590	0.558	0.480	0.552	0.586
Staple	0.581	0.550	0.541	0.593	0.399	0.540	<i>0.542</i>	0.548	0.533	0.476	0.520	<i>0.578</i>
DPCF	<i>0.575</i>	0.523	0.533	0.585	0.409	0.573	0.544	0.525	<i>0.546</i>	<i>0.482</i>	0.517	0.555
RPT	0.571	0.480	0.545	0.526	0.362	0.507	0.469	0.528	0.502	0.464	0.482	0.528
SAMF	0.538	0.500	0.535	0.545	<i>0.426</i>	0.537	0.540	0.533	0.536	0.506	0.502	0.559
DSST	0.523	0.420	0.447	0.558	0.370	0.469	0.453	0.502	0.470	0.386	0.468	0.513
KCF	0.498	0.436	0.459	0.479	0.290	0.459	0.443	0.469	0.453	0.393	0.394	0.477

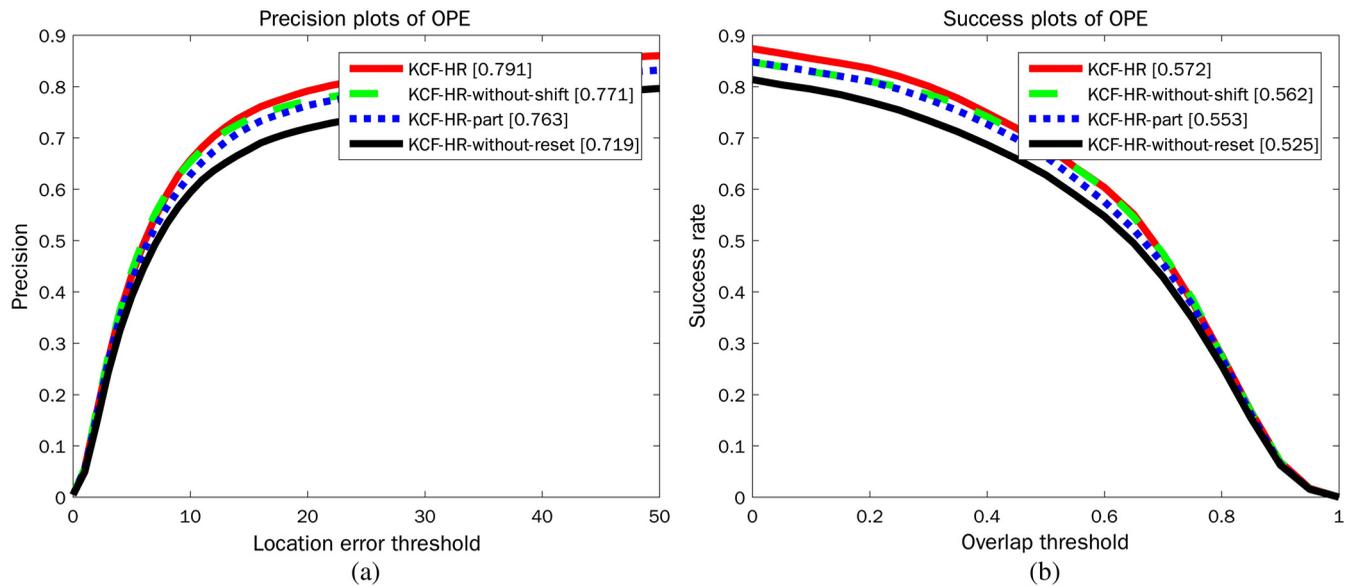


Fig. 9 The (a) precision and (b) success plots for the KCF-HR tracker with different components on the OTB-100 dataset.

and precision score of the KCF-HR-part tracker decreased by 1.9% and 2.8%, respectively, compared with the KCF-HR tracker. This is because too small unreliable parts might be reset frequently, which causes tracking failure in the case of BC. In addition, we notice that the KCF-HR-without-reset tracker shows a significant drop in the precision and success plots while the KCF-HR-without-shift tracker experiences only a slight decline. The above analysis means that among the three merged components, resetting unreliable parts contributes the most to the accurate tracking yet the sliding window method contributes the least. The main reason might be that the sliding window method only helps to accurately track the target when the target moves outside the tracking window while proper resetting

unreliable parts can increase the number and reliability of reliable parts and improve the overall tracking accuracy. Ultimately, the interaction of three components further enhances the performance of the KCF-HR tracker in challenging scenarios.

4.3 Qualitative Comparisons

In this section, the KCF-HR tracker is qualitatively compared with nine state-of-the-art trackers and the tracking results of eight representative sequences with all 11 attributes are shown in Fig. 10. It can be observed that the proposed KCF-HR tracker achieves favorable results compared with the state-of-the-art trackers on these sequences.

4.3.1 Occlusion

Figure 10 shows some sampled results of image sequences including Girl2, Human3, Tiger2, and Box where the target objects undergo partial and heavy occlusion. In the Girl2

sequence, a girl is fully occluded by another man at frame 106 and the girl reappears at frame 129. When the girl is fully occluded and reappears, only the DPCF tracker and our KCF-HR tracker are able to track the girl. When the

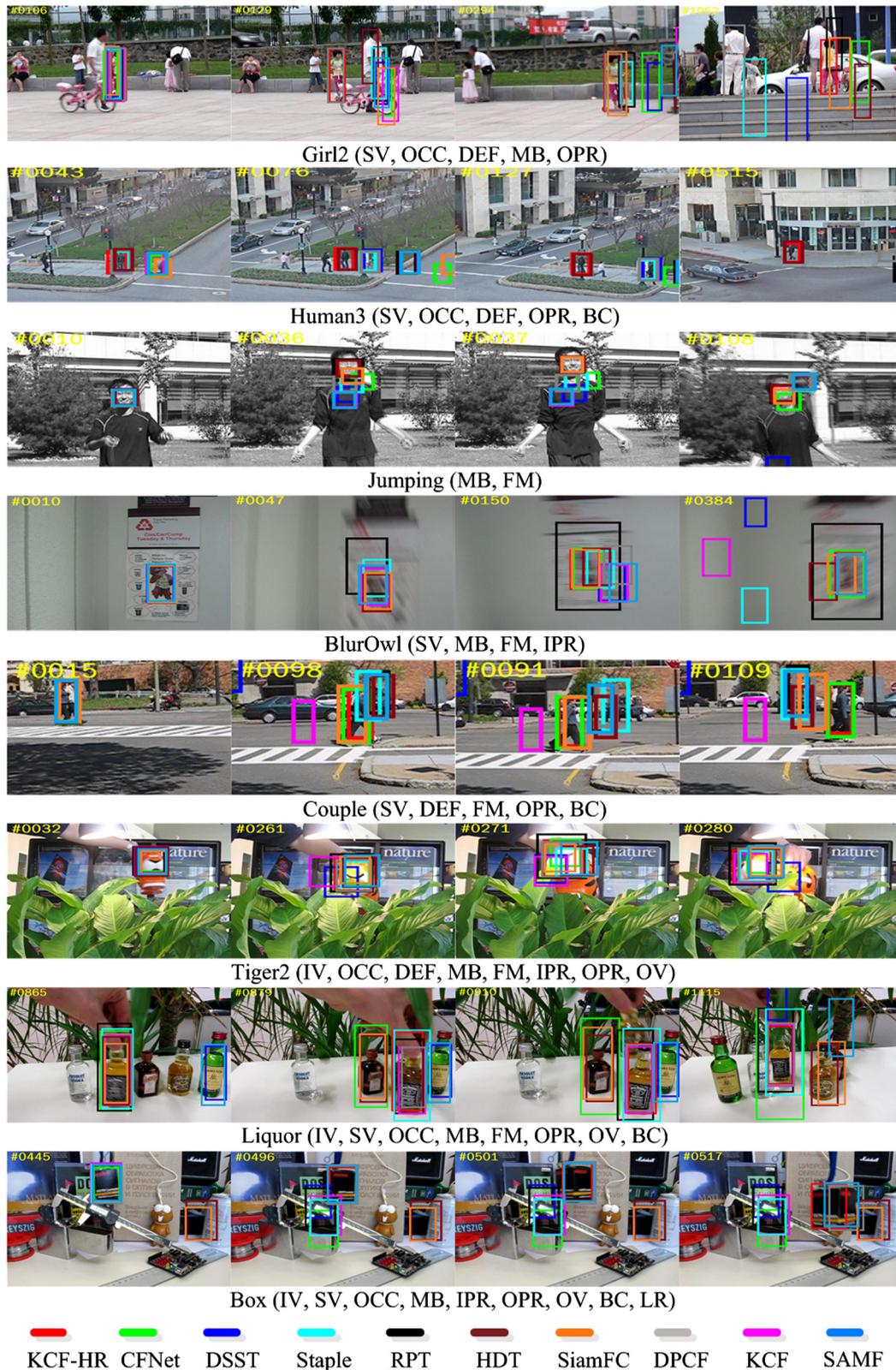


Fig. 10 A visualization of the tracking results of top 10 trackers on eight challenging sequences. The main challenges of each sequence are also listed.

girl turned around at frame 1052, the DPCF tracker also fails to track her. Only our KCF-HR tracker tracks the target girl stably throughout the sequence. In the Human3 sequence, the target man undergoes heavy occlusion, SV, and BC (e.g., #0043, #0076, and #0127), which makes it challenging to accurately track the target. Most of the trackers, except the HDT and KCF-HR trackers, cannot track the target accurately. Our tracker performs favorably because it combines the holistic-part schema and resetting unreliable parts method to find more reliable parts.

4.3.2 Motion blur

Another challenge for a tracker is to handle MB caused by FM of the target or camera. In the sequence of BlurOwl, the target owl is blurred due to its rapid movement with SV and IPR at frames 47, 150, and 384, as shown in Fig. 10. All trackers except the deep learning-based trackers (CFNet, SiamFC, and HDT) and our KCF-HR tracker failed to track the target accurately. In Jumping, the man undergoes MB several times caused by rapid movements up and down (e.g., #0036, #0037, and #0108). Only HDT, RPT, and our KCF-HR trackers track the target accurately throughout the sequence. The reason why our tracker handles MB and FM well can be attributed to the proposed sliding window method and resetting unreliable parts method.

4.3.3 Deformation

In Fig. 10, Tiger2 is a typical challenging sequence where the target object is undergoing severe deformation and other challenges such as FM, occlusion, and in- and out-of plane rotation. In the initial, all trackers track the target object successfully (e.g., #0032). When the target is continuously deformed at frames 261, 271, and 280, the trackers including DSST, KCF, and SAMF are suffering drifts or tracking failure and trackers including RPT, Staple, and CFNet fail to estimate the scale of the target. In the Couple sequence, the target woman appears in the screen with rapid appearance changes due to FM, shape deformation, and shaking of the lens. In addition, the background of the sequences is complex and changes rapidly, which further increases the difficulty of accurate tracking. All trackers perform well in the initial few frames, e.g., #0015, whereas only CFNet and our KCF-HR tracker perform well in the whole sequence.

4.3.4 Background clutter

In the sequence of Liquor, the target bottle is surrounded by several similar bottles and the background is cluttered, e.g., #0910 and #1115. All trackers except KCF, DPCF, and KCF-HR are failed to track the target bottle accurately. In the sequence of Box, the target box is moving in a cluttered background, as shown in Fig. 10. At frame 445, all trackers except HDT and SiamFC track the box successfully. When the box is partially occluded and reappears at frame 496, only the SAMF and our KCF-HR trackers perform well. When the target box is rotated at frames 501 and 517, only the KCF-HR tracker is still on the target. The KCF-HR tracker shows high robustness on sequences including Human3 and Couple that undergo BC and other challenging appearance variations. The KCF-HR tracker handles BC and other challenging appearance variations well because it employs the holistic and reliable-parts-based schema and the

adaptive updating schema, which can reduce the risk of drifting and eliminate most of the effects of the background and appearance variations.

5 Conclusion

In this paper, we propose a KCF-based visual tracker via holistic and reliable local parts. The proposed KCF-HR tracker consists of local part level and holistic level. In the local part level, the target object is divided into four overlapping parts and each of these parts is tracked with the KCF tracker. A sliding window method is employed on each part to generate several patches when all local parts are unreliable. All generated patches of a part are tracked to find reliable patch, and the part is replaced by the generated reliable patch. Then the rough position of the target is voted by weighted reliable parts. In the holistic level, the KCF tracker on multiple resolutions of the target image is performed on the rough position. A reset unreliable parts method is employed and correlation filters of reliable holistic and parts are adaptively updated. Experimental results on frequently used public benchmark datasets OOTB and OTB-100 show that the KCF-HR tracker outperforms several state-of-the-art trackers and can effectively deal with FM and MB and ease the drifting problem caused by rotation and occlusion.

Acknowledgments

This work was supported by Sichuan Science and Technology Program (Project Number: 2019YJ0164).

References

1. A. Yilmaz, O. Javed, and M. Shah, "Object tracking: a survey," *ACM Comput. Surv.* **38**(4), 1–45 (2006).
2. Y. Wu, J. Lim, and M. H. Yang, "Online object tracking: a benchmark," in *IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 2411–2418 (2013).
3. X. Li et al., "A survey of appearance models in visual object tracking," *ACM Trans. Intell. Syst. Technol.* **4**(4), 48 (2013).
4. Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.* **37**(9), 1834–1848 (2015).
5. C. Xie et al., "Multi-scale patch-based sparse appearance model for robust object tracking," *Mach. Vision Appl.* **25**(7), 1859–1876 (2014).
6. C. Gao et al., "Robust visual tracking using exemplar-based detectors," *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(2), 300–312 (2017).
7. X. Li et al., "A multi-view model for visual tracking via correlation filters," *Knowl.-Based Syst.* **113**, 88–99 (2016).
8. L. Zhang and P. N. Suganthan, "Robust visual tracking via co-trained kernelized correlation filters," *Pattern Recognit.* **69**, 82–93 (2017).
9. L. Wang et al., "Forward-backward mean-shift for visual tracking with local-background-weighted histogram," *IEEE Trans. Intell. Transp. Syst.* **14**(3), 1480–1489 (2013).
10. T. Zhang et al., "Robust visual tracking via consistent low-rank sparse learning," *Int. J. Comput. Vis.* **111**(2), 171–190 (2015).
11. J. F. Henriques et al., "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.* **37**(3), 583–596 (2015).
12. T. Liu, G. Wang, and Q. Yang, "Real-time part-based visual tracking via adaptive correlation filters," in *IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 4902–4912 (2015).
13. B. Zhang et al., "Output constraint transfer for kernelized correlation filter in tracking," *IEEE Trans. Syst. Man Cybern. Syst.* **47**(4), 693–703 (2017).
14. Y. Sui, G. Wang, and L. Zhang, "Correlation filter learning toward peak strength for visual tracking," *IEEE Trans. Cybern.* **48**(4), 1290–1303 (2018).
15. Y. Li, J. Zhu, and S. C. H. Hoi, "Reliable patch trackers: robust visual tracking by exploiting reliable patches," in *IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 353–361 (2015).
16. G. Ding et al., "Real-time scalable visual tracking via quadrangle kernelized correlation filters," *IEEE Trans. Intell. Transp. Syst.* **19**(1), 140–150 (2018).
17. K. Chen, W. Tao, and S. Han, "Visual object tracking via enhanced structural correlation filter," *Inf. Sci.* **394**, 232–245 (2017).
18. J. F. Henriques et al., "Exploiting the circulant structure of tracking-by-detection with kernels," in *Eur. Conf. Comput. Vision (ECCV)*, pp. 702–715 (2012).

19. C. Qian, T. P. Breckon, and H. Li, "Robust visual tracking via speedup multiple kernel ridge regression," *J. Electron. Imaging* **24**(5), 053016 (2015).
20. F. Li, S. Zhang, and X. Qiao, "Scene-aware adaptive updating for visual tracking via correlation filters," *Sensors* **17**(11), 2626 (2017).
21. F. Liu et al., "Robust visual tracking revisited: from correlation filter to template matching," *IEEE Trans. Image Process.* **27**(6), 2777–2790 (2018).
22. D. S. Bolme et al., "Visual object tracking using adaptive correlation filters," in *IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 2544–2550 (2010).
23. Y. Xu et al., "Patch-based scale calculation for real-time visual tracking," *IEEE Signal Process. Lett.* **23**(1), 40–44 (2016).
24. L. Huang et al., "Visual tracking by sampling in part space," *IEEE Trans. Image Process.* **26**(12), 5800–5810 (2017).
25. L. Zhao et al., "Combined discriminative global and generative local models for visual tracking," *J. Electron. Imaging* **25**(2), 023005 (2016).
26. H. Zhang and G. Liu, "Coupled-layer based visual tracking via adaptive kernelized correlation filters," *Vis. Comput.* **34**(1), 41–54 (2018).
27. O. Akin et al., "Deformable part-based tracking by coupled global and local correlation filters," *J. Vis. Commun. Image Represent.* **38**, 763–774 (2016).
28. Y. Qi et al., "Hedged deep tracking," in *IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 4303–4311 (2016).
29. L. Bertinetto et al., "Fully-convolutional Siamese networks for object tracking," in *Eur. Conf. Comput. Vision (ECCV)*, pp. 850–865 (2016).
30. J. Valmadre et al., "End-to-end representation learning for correlation filter based tracking," in *IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 5000–5008 (2017).
31. M. Danelljan et al., "ECO: efficient convolution operators for tracking," in *IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 6931–6939 (2017).
32. M. Godec, P. M. Roth, and H. Bischof, "Hough-based tracking of non-rigid objects," *Comput. Vis. Image Understanding* **117**(10), 1245–1256 (2013).
33. Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Eur. Conf. Comput. Vision (ECCV)*, pp. 254–265 (2014).
34. L. Bertinetto et al., "Staple: complementary learners for real-time tracking," in *IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 1401–1409 (2016).
35. M. Danelljan, G. Häger, and F. S. Khan, "Accurate scale estimation for robust visual tracking," in *British Machine Vision Conf. (BMVC)*, pp. 65.1–65.11 (2014).
36. M. Danelljan et al., "Adaptive color attributes for real-time visual tracking," in *IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 1090–1097 (2014).
37. B. Babenko, M. H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(8), 1619–1632 (2011).
38. Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(7), 1409–1422 (2012).
39. W. Zhong, H. Lu, and M. H. Yang, "Robust object tracking via sparsity-based collaborative model," in *IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 1838–1845 (2012).
40. S. Hare et al., "Struck: structured output tracking with kernels," *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(10), 2096–2109 (2016).

Chunbao Li is a PhD candidate in the School of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, China. He received his MS degree from the School of Automation Engineering from UESTC. His research interests include visual tracking and machine learning.

Bo Yang has been a full professor since 2008. He received his PhD from the National University of Singapore in 2002. His research interests include machine learning and data mining, cloud computing. He has published over 60 papers in information sciences, etc. He serves as a program chair of IEEE DASC 2009, CSC 2011, ICCCS 2019; program vice-chair of IEEE HPCC 2010; general chair of IWKDEWL'10. He has been a senior member of IEEE since 2013.