Yuan *et al. EURASIP Journal on Image and Video Processing* (2020) 2020:7
https://doi.org/10.1186/s13640-020-0496-6

EURASIP Journal on Image and Video Processing

RESEARCH Open Access

A scale-adaptive object-tracking algorithm with occlusion detection

Yue Yuan[1,2], Jun Chu[1,2*], Lu Leng[1,2], Jun Miao[1,3] and Byung-Gyu Kim[4]

Abstract

The methods combining correlation filters (CFs) with the features of convolutional neural network (CNN) are good at object tracking. However, the high-level features of a typical CNN without residual structure suffer from the shortage of fine-grained information, it is easily affected by similar objects or background noise. Meanwhile, CF-based methods usually update filters at every frame even when occlusion occurs, which degrades the capability of discriminating the target from background. A novel scale-adaptive object-tracking method is proposed in this paper. Firstly, the features are extracted from different layers of ResNet to produce response maps, and then, in order to locate the target more accurately, these response maps are fused based on AdaBoost algorithm. Secondly, to prevent the filters from updating when occlusion occurs, an update strategy with occlusion detection is proposed. Finally, a scale filter is used to estimate the target scale. The experimental results demonstrate that the proposed method performs favorably compared with several mainstream methods especially in the case of occlusion and scale change.

Keywords: Scale adaption, Object tracking, Resnet, Correlation filters, Occlusion detection

1 Introduction

Video surveillance is significant for public security [1], while object tracking is the key technology of video surveillance [2, 3]. Object tracking has many practical applications in video surveillance, human-computer interaction and automatic driving [4–6]. Object tracking aims to estimate the target position in a video sequence by giving an initial position of the target. Due to the deformation, illumination variety, occlusion, and scale change, it is possible that the appearance changes significantly. Therefore, the usage of the powerful convolutional neural network (CNN) features to describe the target appearance can effectively improve the success rate and accuracy of object-tracking algorithms [7, 8].

CNN pre-trained for image classification, such as AlexNet [9] and VGG[10], are used to extract target features in most deep-learning-based trackers. Those methods have high computational complexity as they need to extract the features of positive and negative samples. While correlation filter (CF)-based trackers have shown efficient performance by solving a ridge regression problem in the Fourier frequency domain. Therefore, the combination of CNN features and efficient CFs has been exploited in object-tracking research. The multi-channel features are extracted from CNN instead of the hand-crafted features for CF-based methods, which achieves the state-of-the-art results on object tracking benchmarks [11, 12]. However, there are still some problems:

1. Target localization relies heavily on the high-level features from CNN, such as the outputs of the last layer of VGG network. The high-level features contain more semantic information but lack of detailed information of the target.
2. The weights are fixed in the fusion of response maps. Inaccurate predictions are inevitable if the filters with a large error have large weights.
3. The filters need to be updated to maintain its discriminative ability as the target appearance changes in the video sequence. Generally, CF-based trackers adopt the updating strategy in all frames, even the frames in which the target is occluded, which degrades the discriminative ability of the filters and results in the loss of tracked target.

*Correspondence: chuj@nchu.edu.cn
[1]Key Laboratory of Jiangxi Province for Image Processing and Pattern Recognition, Nanchang Hangkong University, 330063 Nanchang, China
[2]School of software, Nanchang Hangkong University, 330063 Nanchang, China
Full list of author information is available at the end of the article

4. The change of target scale commonly affects the position estimation, since the size of search area is highly correlated with the target scale.

The main contributions and originality of this paper are as follows:

1. The CNN with residual structure is used to extract features. DenseNet [13] and Inception [14] are two networks with residual structure. However, the features from DenseNet are not comparable to those of ResNet [15] in terms of success rate and accuracy of tracking. Meanwhile, the features from Inception have large number of channels, and accordingly its implementation is time-consuming. Thus, ResNet is used in this paper due to its advantages of success rate, accuracy, and efficiency. The residual structure of ResNet integrates low-level and high-level features with identical mapping [16]. The high-level features contain more fine-gained details, which are more robust to similar objects and background noise.
2. The response maps are fused based on AdaBoost algorithm. AdaBoost algorithm enlarges the weights of the filters with small error rates while reduces the weights of the filters with large error rates. Consequently, the stronger the discriminative abilities of the filters are, the greater roles they can play in the tracking process.
3. An update strategy with occlusion detection is adopted. When the target is occluded, there are many local maxima in the response map, so the number of effective local maxima (NELM) is used to detect occlusion. If the occluded target is detected, the filters stop the update to avoid the interference of background information.
4. Scale filters are used to track the scale change of the target to solve the scale variation problem.

In the remainder of this paper, we first review some related works in Section 2. Then, we propose a scale-adaptive object-tracking algorithm with occlusion detection in Section 3. The experiments and comparisons are reported in Section 4. We end the paper with a conclusion in Section 5.

2 Related works

2.1 Tracking by deep learning

Visual representation is significant in the tracking algorithm [17]. The traditional tracking-by-detection methods focus on the discriminative ability of the discriminator, for example, Zhang et al. [18] proposed a multiple experts using entropy minimization (MEEM) scheme based on support vector machine with hand-crafted features. While, most methods based on deep learning usually focus on the expression of the target feature. Wang and Yeung [19] trained a multi-layer auto-encoder to encode the appearance of the target. Li et al. [20] used face dataset to train CNN and then used the pre-trained CNN to extract face features for tracking. Nam and Han [21] trained a convolutional network to extract target features in multi-domain way and used full connection layers to classify target and background. Hong et al. [22] used the features extracted by a pre-trained CNN and learned discriminative saliency map with back propagation and then used a support vector machine as the classifier. Pu et al. [23] used back propagation to generate attention map to enhance the discriminative ability of full connection layers in [21]. Wang et al. [24] built two complementary prediction networks based on the analysis on the features of the different levels of CNN to obtain the heat map for target localization. Lu et al. [25] proposed a deconvolution network to upsample the features with low spatial resolution; then, the features of the low and high levels are fused by the sum operation to get better target representation. Song et al. [26] solved the problem of unbalanced positive and negative samples based on the generative adversarial networks [27].

The above methods usually need to compute the features of a large number of candidates, while our method only needs the features of search region. Moreover, these methods need back propagation for time-consuming online update; in contrast, our method can online update efficiently thanks to linear interpolation.

2.2 Tracking by correlation filter

CF-based methods have shown continuous performance improvements in terms of accuracy and robustness. Bolme et al. [28] proposed a minimum output sum of squared error filter. Meanwhile, peak-to-sidelobe ratio (PSR) was introduced to measure the confidence of response map. It was pointed out that PSR would decrease to about 7.0 when tracking failed. Henriques et al. [29] employed the circulant structure and the kernel method (CSK) to train filters on the basis of [28]. Henriques et al. [30] used the cyclic shift of target features and the diagonalization property of cyclic matrix in the Fourier domain to obtain closed-form solutions based on kernel correlation filter (KCF), which improved the effectiveness and efficiency of the algorithm. Danelljan et al. [31] used position filter and scale filter for discriminative scale space tracking (DSST). Li and Zhu [32] applied scale adaption with multiple features (SAMF) to estimate the target scale adaptively. Danellian et al. [33] performed spatial regularization on the discriminative CFs to alleviate the boundary effect. Li et al. [34] introduced temporal regularization to [33]. Cen and Jung [35] proposed a complex form of local orientation plane descriptor to overcome occlusion; this descriptor effectively considers the spatiotemporal relationship between the target and background in CF framework.

The above methods usually use hand-crafted features [36], [37], which lack robustness to target appearance variance. Furthermore, they update filters even when the target is occluded, which degrades the discriminative capability of filters. In our method, robust convolutional features deal with the target appearance variance. In addition, occlusion detection avoids the updating when the target is occluded. Similar to [31], we apply scale filters to track the target scale variance, and we decrease the number of the scale for efficiency.

2.3 Tracking combining deep learning and correlation filter

As the robustness of CNN features and the efficiency of CF, some algorithms combined the two methods. Danelljan et al. [38] used the feature extracted from only one layer of CNN on the basis of [33]. In order to use the multi-resolution deep feature maps, Danelljan et al. [39] applied a continuous convolution operators for visual tracking, and after that, Danelljan et al. [40] proposed an efficient convolution operators based on [39] for efficiency. Ma et al. [41] developed CFs using hierarchical convolutional features (HCF). Li et al. [42] localized the target using the deep convolution operator in a large search area firstly, and then performed a shallow convolution operator around the location given by the first step. Li et al. [43] trained background-aware filters using a set of representative background patches as negative samples to handle background clutter, and trained scale-aware CFs using a set of samples with different scales to handle scale variance. Qi et al. [44] used convolution operation to model the correlation between the apparent features of the target and background, and employed a two-layer convolution network to learn geometirc structural information for scale estimation. Qi et al. [45] applied CFs on the multiple CNN layers, and then all layer trackers were integrated to a single stronger tracker by Hedge algorithm. Wang et al. [46] proposed a discriminative CFs network (DCFNet) to learn the convolutional features and performed the correlation tracking process simultaneously. Similar to [46], Jack et al. [47] used correlation filters as one layer of the neural network and proposed an end-to-end algorithm.

In some algorithms, ResNet is also used. Zhu et al. [48] proposed a CF-based algorithm using temporal and spatial features. They used two ResNets to learn spatial and temporal features, respectively. He et al. [49] used ResNet to extract features instead of the deep learning features from VGG and hand-crafted features in [40], but the response maps are fused with fixed threshold weights. The boundary effect in correlation filters is dealt with in the algorithms based on [40], but it is not a focus of this paper.

Our method seems similar to HCF, but there are some differences as follows. In HCF, typical CNN without residual structure is used to extract features which lack fine-gained details, and the response maps are fused with fixed weights. Moreover, in HCF, the filters are updated at all frames even when the target is occluded, which definitely declines the discriminative ability of the filters. In our work, the features are extracted with the pre-trained ResNet, which are more robust to background noisy and occlusion. In addition, the response maps are fused based on AdaBoost algorithm [50], which can choose more reliable weights. Meanwhile, the filters are updated while considering occlusion detection to ensure that the filters are not disturbed by noise.

3 Methods

3.1 Procedure

Figure 1 illustrates the procedure of our method. Our method initializes the filters according to the given target position. In the subsequent frames, we first crop the search area centered at the target location in the previous frame, and then, extract the CNN features from different layers of pre-trained ResNet. Secondly, the learned linear filters convolved with the extracted features to generate the response maps of different layers. Then, multiple response maps are weighted and fused to one response map. The target position is located according to the position of the maximum value in the fused response map. After that, in the estimated target location, the histogram of oriented gradient (HOG) features in the regions with different scales are used to find the optimal target scale by scale filters. Finally, the NELM and the PSR of the fused response map are performed to decide whether to update the filter or not.

3.2 Convolutional features

The convolutional feature maps from ResNet are used to encode target appearance. With the increment of CNN layer number, the spatial resolution of feature map is gradually reduced. For object tracking, low resolution is not sufficient to accurately locate target. Thus, we ignore the features from the last convolutional layer (conv5) and full connection layers. The features from different layers have different spatial resolutions that are relatively low compared with the input image. Therefore, bilinear interpolation is used to enlarge the resolutions of the features to the same size by:

$$x_i = \sum_k \alpha_{ik} h_{ik} \tag{1}$$

where h represents the features, x represents the features enlarged by interpolation operation, and the interpolation weight depends on the position of i and k-neighbor feature

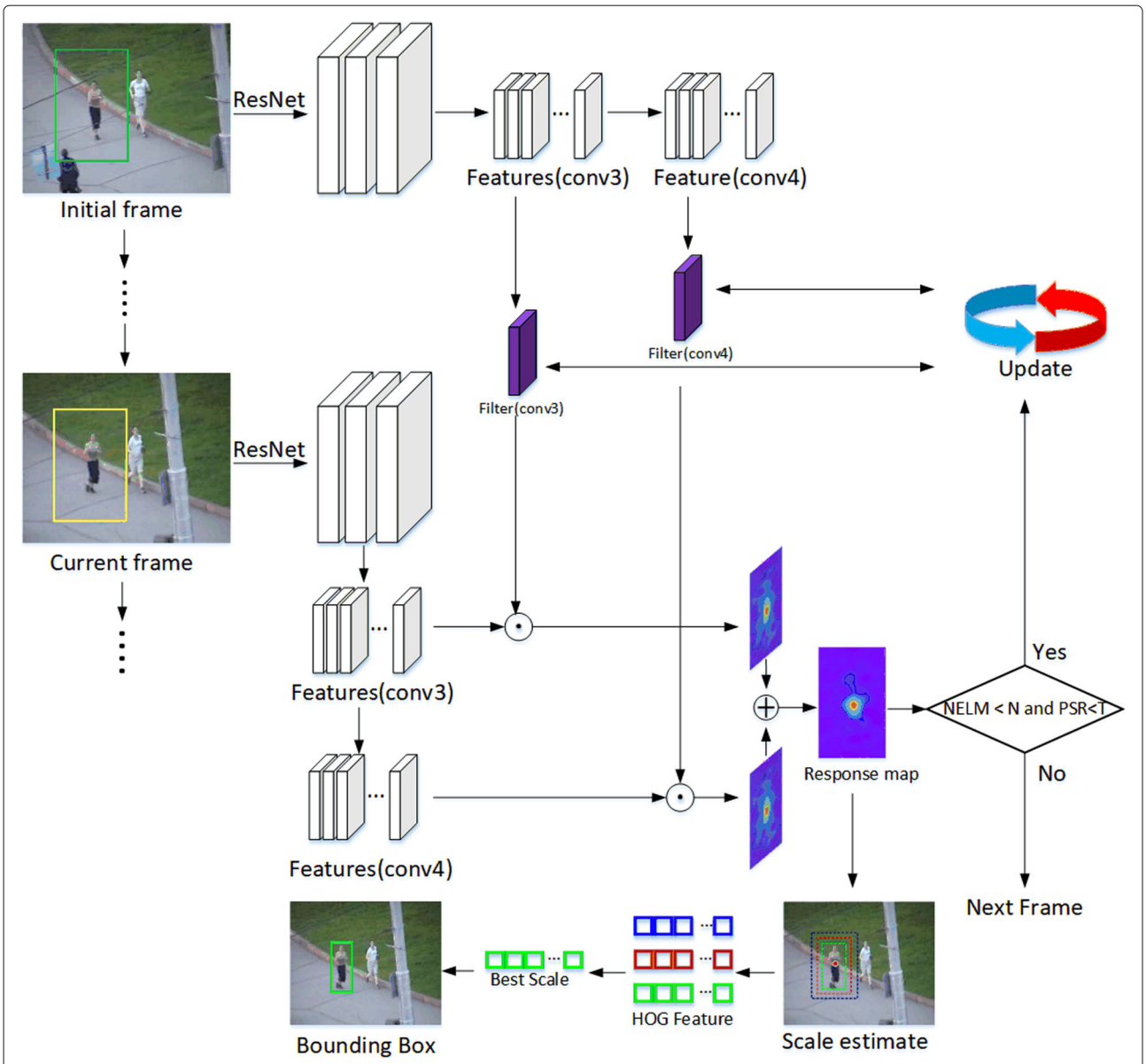


Fig. 1 Procedure of the proposed method

value. The visualization of the features from ResNet is shown in Fig. 2.

3.3 Correlation filter

Denote x_l as the feature from the conv-l layer with the size of $M \times N \times D$ after bilinear interpolation operation, where M, N, and D indicate the width, height, and the number of channels, respectively. The shifted sample $x_{(m,n)}, (m,n) \in \{0,1,\ldots,M-1\} \times \{0,1,\ldots,N-1\}$ has a Gaussian function label $y(m,n) = e^{-\frac{(m-M/2)^2+(n-N/2)^2}{2\delta^2}}$, where δ indicates the kernel width. Correlation filters w_l are obtained by minimizing the objective function:

$$\min_{w} \|w_l \star x_l(m,n) - y(m,n)\|^2 + \lambda \|w_l\|^2 \tag{2}$$

where $\star$ means circular correlation and λ indicates the regularization parameter. The optimization problem can be solved in Fourier domain and the solutions are:

$$W_l = \frac{\bar{Y} \odot X}{\sum_{d=1}^{D} X_l^d \odot \bar{X}_l^d + \lambda} \tag{3}$$

Here, X and Y are the fast Fourier transformation (FFT) $F(x)$ and $F(y)$, respectively. The over bar represents the complex conjugate. The symbol $\odot$ denotes the element-wise product. At the detection process, the features of the search patch are extracted and transformed to the Fourier

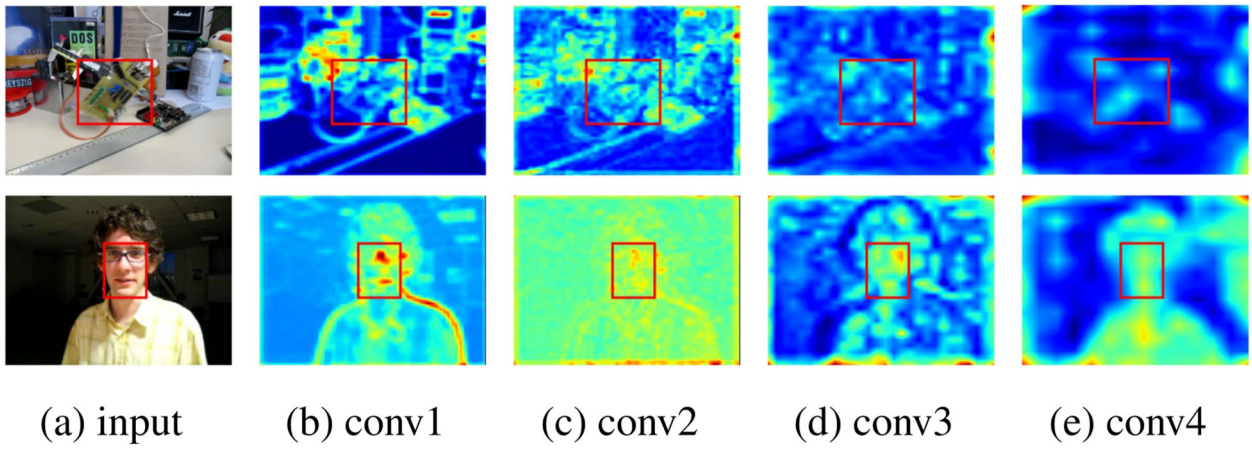


Fig. 2 Visualization of the features from ResNet. **a** Input frame. **b**–**e** Features are extracted on convolutional layers conv1, conv2, conv3, and conv4 of ResNet; their original spatial resolutions are 128 × 128, 56 × 56, 28 × 28, and 14 × 14, respectively. The red boxes indicate the tracking results by our approach

domain, the complex conjugate is $\bar{Z}$. The response map at conv-l layer can be computed by:

$$f_l = F^{-1}(\sum_{d=1}^{D} \bar{W}_l^d \odot Z_l^d) \tag{4}$$

where F^{-1} is the inverse FFT.

3.4 Response map fusion based on AdaBoost

In order to select the appropriate weights to fuse the response maps, AdaBoost algorithm is used for adaptive weight adjustment. The error rate e is computed between the normalized response maps at different layers f_l, and the desired response map g peaked at the estimated target position in $t-1$ frame is:

$$e_l^{t-1} = \text{Mean}(\frac{\text{abc}(f_l^{t-1} - g^{t-1})}{f_l^{t-1} + g^{t-1}}) \tag{5}$$

where abs represents absolute value, Mean denotes the operation of average, the weight of conv-l layer β_l is:

$$\beta_l = \log \frac{1 - e_l^{t-1}}{e_l^{t-1}} \tag{6}$$

Then, at t frame, the fused response map is:

$$f^t = \sum_{l=3,4} \beta_l f_l^t \tag{7}$$

The target position $(\hat{m}, \hat{n})$ is estimated as:

$$(\hat{m}, \hat{n}) = \arg\max_{(m,n)} f^t(m,n) \tag{8}$$

After the filters are initialized, the filters of different layers can correctly track the target in the initial frame, as the computation is performed in the initial frame. In other words, these filters have the same error rate; thus, the initial weights are both set to 0.5.

For scale estimation, we construct a feature pyramid center in the estimated target position. Let $P \times R$ denote the target size in the current frame, S be the size of the scale dimension, and a represent the scale factor. For each $n \in \left\{\lfloor -\frac{S-1}{2} \rfloor, \ldots, \lfloor \frac{S-1}{2} \rfloor\right\}$, we crop the image patch of the size $a^n P \times a^n R$ and extract the HOG features; then, the scale response map R_n is computed by:

$$R_{t+1}(n) = F^{-1}\{\sum_{k=1}^{K} \bar{H}_k^{t+1}(n) \odot I_{t+1}^k(n)\} \tag{9}$$

where

$$H_t(n) = \frac{\bar{G}(n) \odot I_t(n)}{\sum_{k=1}^{K} I_t^k(n) \odot \bar{I}_t^k(n) + \lambda_s} \tag{10}$$

where I is the FFT of HOG features, and $\bar{G}$ is the complex conjugate of Gaussian label. We can find the $\hat{n}$ corresponded maximum value as:

$$\hat{n} = \arg\max_{n} R_{t+1}(n) \tag{11}$$

Then, the best scale of target is $a^{\hat{n}} P \times a^{\hat{n}} R$.

3.5 Optimized update strategy with occlusion detection

The filters need to be updated to maintain discriminative ability as the target often undergoes appearance variance. However, when the target is occluded, the filters should avoid using background information to update, or it may cause model drift.

In minimum output sum of squared error (MOSSE) filter [28], PSR was used to describe the state of the response map to detect tracking failure. The peak means the maximum, and the side lobe is defined as the rest of the pixels, excluding an 11 × 11 window around the peak. The PSR is defined as $PSR = \frac{g^{\max} - \mu}{\sigma}$, where $g^{\max}$ is the peak value,

μ is the mean and σ is the standard deviation of the side lobe. The PSR is between 20.0 and 60.0 when the tracking is normal, while PSR drops to lower than 7.0 when the target is occluded or the tracking failed, as shown in Fig. 3. However, when the target moves rapidly or is of low resolution, the PSR stays in a low value, as shown in c and d of Fig. 3. Therefore, PSR cannot accurately reflect whether the target is occluded or not.

In this work, NELM is employed to detect occlusion. Observing the response maps, we found that the response maps have more local maxima when the target is occluded than when the target is not occluded. As shown in Fig. 4, the red dotted lines show the locations of the local maxima in the 3D response map.

Let f denote the fused response map in current frame and $f_{\max}$ be the peak of f. For each local maximum $f^i_{\text{loc}} (i \in \{1, 2, 3, \ldots, L\})$, L is the number of local maximum except $f_{\max}$, the ratio between f^i_{loc} and $f_{\max}$ is $T_i = \frac{f^i_{\text{loc}}}{f_{\max}}$. In the response map, some local maxima are possibly generated because of the background interference which needs to be avoided. The motion of the target between the initial frame and the second frame should be smooth. Therefore, in the response map obtained from the second frame of the video sequence, the local maximum except the peak (which is the target position) is taken as the threshold γ:

$$\gamma = \max(T_i) \tag{12}$$

In the response map of subsequent frame, T_i is greater than the threshold γ; then, f_i is recorded as the effective local maximum, and the number of effective local maximum is expressed as:

$$\text{NELM} = \text{Crad}\{T_i | T_i > \gamma\} \tag{13}$$

where Crad represents the number of elements in a collection. If the effective local maxima exist, i.e., NELM > 1, and the PSR is less than the given threshold, the algorithm does not update the filters. PSR is only used to evaluate the response map, similar to MOSSE, the PSR threshold is set to 7.000. If no effective local maximum exists or the PSR is greater than the given threshold, the algorithm allows updating the filters. In Fig. 3b, the PSR value is lower than the empirical value and the NELM is equal to zero, target occlusion is not detected, then the filters can be updated at this time. At t frame, the filter in (3) is represented by W_t, A_t is the molecule of W_t, and B_t is the denominator. The updating formulae are:

$$A_t = (1 - \eta_p) A_{t-1} + \eta_p * \bar{Y} \odot X_t \tag{14}$$

$$B_t = (1 - \eta_p) B_{t-1} + \eta_p * \sum_{d=1}^{D} X_t^d \odot \bar{X}_t^d \tag{15}$$

$$W_t = \frac{A_t}{B_t + \lambda} \tag{16}$$

C and D represent the molecules and denominators of the filters H_t in (10) , respectively. The updating formulae are:

$$C_t = (1 - \eta_s) C_{t-1} + \eta_s * \bar{G} \odot I_t \tag{17}$$

$$D_t = (1 - \eta_s) D_{t-1} + \eta_s * \sum_{k=1}^{K} I_t^k \odot \bar{I}_t^k \tag{18}$$

$$H_t = \frac{C_t}{D_t + \lambda} \tag{19}$$

where η_p and η_s are the learning rates for W_t and H_t, respectively.

4 Experimental

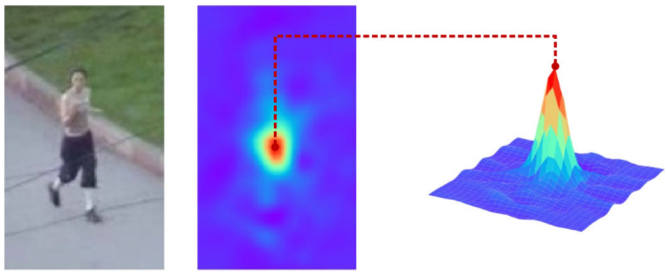
We compare the proposed method with the state-of-the-art methods on OTB and VOT [51]. Pre-trained ResNet is used to extract features. The learn rate η_p is set to 0.01, the same as [30], and η_s is set to 0.01, the same as [31]. The scale factor is set to 1.087. The number of scale dimension is set to 5. The parameters are not changed during test.

Our tracker is implemented by Python with PyTorch. The experiments are performed on Intel Core i7-6850K 3.6 GHz CPU and a NVIDIA GTX-1080Ti GPU. Our tracker runs at an average of 8 fps on GPU.

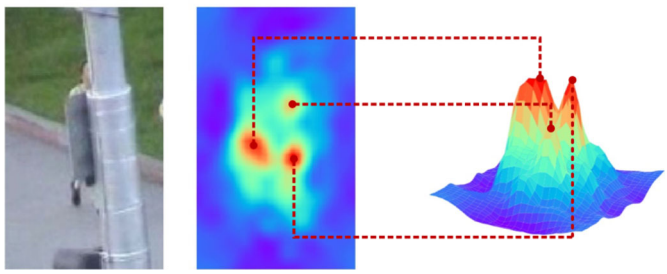
The algorithm is validated on standard tracking data sets OTB-13 and OTB-15. OTB-13 and OTB-15 contain 50 and 100 video sequences, respectively. These video

(a) PSR=37.50,NELM=0 (b) PSR=5.19,NELM=2 (c) PSR=5.42,NELM=0 (d) PSR=5.71,NELM=0

Fig. 3 "Jogging-1", "Freeman3" sequences with the PSR and NELM. The red bounding boxes indicate the tracking results by our method. **a** PSR = 37.50, NELM = 0. **b** PSR = 5.19, NELM=2. **c** PSR = 5.42, NELM = 0. **d** PSR = 5.71, NELM = 0



(a) The response map without occluded target, PSR = 45.61, NELM = 0



(b) The response map with occluded target, PSR = 5.79, NELM = 2

Fig. 4 The response maps with and without target occlusion. The red dotted lines show the locations of the local maxima in the 3D response map. Notice that, NELM does not count the global maximum, so NELM less than the number of dotted lines. **a** The response map without occluded target, PSR = 45.61, NELM = 0. **b** The response map with occluded target, PSR = 5.79, NELM = 2

sequences contain common challenges in target tracking, including illumination variance, scale variance, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, background interference, and low resolution. OTB recommends three evaluation methods, one pass evaluation (OPE), spatial robustness evaluation (SRE), and temporal robustness evaluation (TRE). OPE gives the exact location of the target in the first frame for initialization and then runs the tracker on all frames. Unlike OPE, SRE initializes the tracker by moving or scaling the target position in the first frame, including four kinds of center offset, four kinds of angle offset, and four kinds of scale variance. While, TRE runs the tracker at the part of the whole sequence. The algorithm is evaluated by calculating the precision score and success rate in three evaluation methods. Precision ε is the Euclidean distance between the center positions of the tracked target and the ground truth:

$$\varepsilon = \sqrt{(x_c - x_g)^2 + (y_c - y_g)^2} \tag{20}$$

where (x_c, y_c) and (x_g, y_g) denote the locations of the tracked target center and the real target center. Precision score is defined as the percentage of the frames whose precision values are lower than a certain threshold in the total number of frames.

The overlap rate is the ratio of the overlap area of the ground truth and the bounding box obtained by the tracking algorithm to the total area of the two boxes:

$$\text{IoU} = \frac{\text{area(Bbox)} \cap \text{area(Gbox)}}{\text{area(Bbox)} \cap \text{area(Gbox)}} \tag{21}$$

where Bbox and Gbox represent the bounding box obtained by the algorithm and the ground truth, respectively. The success score is the percentage of the number of the frames whose overlap rates are greater than a certain threshold.

5 Results and discussion

5.1 Quantitative evaluation

The proposed method is compared with seven mainstream algorithms including MEEM [18], CSK [29], KCF [30], DSST [31], SAMF [32], HCF [41], CFNet [47], and DCFNet [46]. HCF and DCFNet combine the correlation filters and CNN features. KCF, DSST, SAMF, and CSK use the correlation filters based on the hand-crafted features.

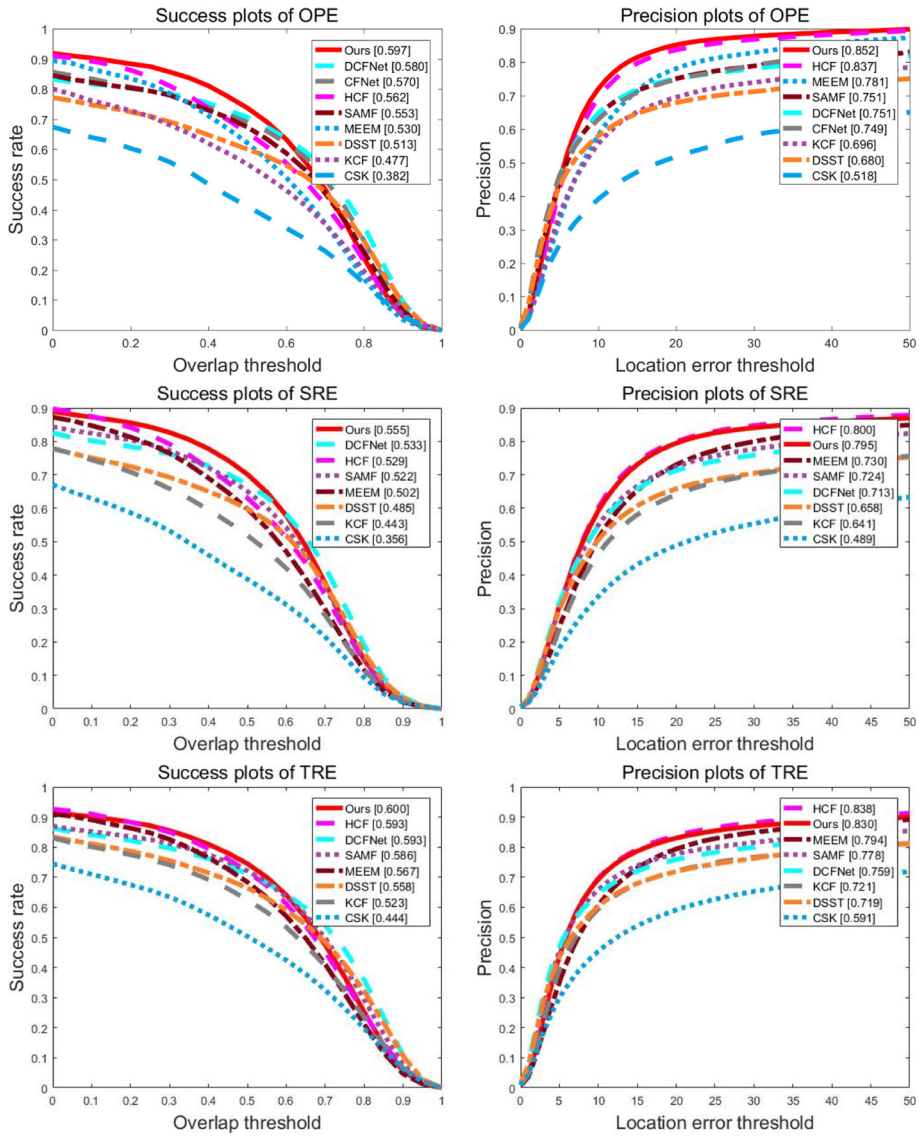


Fig. 5 Overlap success plots and Distance precision plots over 100 benchmark sequences in OPE, SRE, TRE

While the scale variance of the target is processed in SAMF and DSST.

5.1.1 Results over all OTB

The results of the algorithms are evaluated in three methods. In Fig. 5, the score in overlap success plots legend represents the area under curve (AUC), the score in distance precision legend represents the distance precision score at a threshold of 20 pixels. Our algorithm achieves the best results in OPE. In TRE and SRE, HCF uses more convolution layer features for target localization, the accuracy score of proposed algorithm is slightly lower than that of HCF. Please notice that some algorithms, including CFNet, do not supply the data for SRE and TRE.

5.1.2 Results at fixed threshold

Table 1 shows the comparison results at the distance precision threshold of 20 pixels and the overlap threshold of 0.5 on OTB-13 and OTB-15. Note that OTB-15 has more

Table 1 Results at fixed threshold

		Ours	DCFNet	HCF	SAMF	KCF
DP(%)	OTB-50	*85.7*	79.5	**89.1**	78.5	74.0
	OTB-100	**84.4**	75.1	*83.7*	75.1	69.6
OS(%)	OTB-50	*76.0*	**77.9**	74.0	73.2	62.3
	OTB-100	**73.3**	*70.7*	65.5	67.4	55.1
SPEED(FPS)	OTB-50	8.3	*41.1*	11.0	18.6	**245**
	OTB-100	8.0	*41.2*	10.4	16.9	**245**

Table 2 Results on VOT2016

	Supervised			UnSupervised
	Accuracy	Robustness	EAO	Accuray
Ours	*0.49*	*27.22*	0.22	**0.39**
DSST	**0.53**	44.81	0.18	0.33
HCF	0.44	**23.86**	*0.22*	*0.37*
KCF	0.49	38.08	0.19	0.30

challenging videos than OTB-13. DP, OS, and SPEED represent the score of distance precision, the score of overlap rate, and the speed of the algorithm, respectively. The first and second best results in each row are highlighted by bold and italics. Under the above threshold, the tracking precision and success rate of the proposed algorithm are the best on OTB-15. However, the speed of this algorithm is about 8 frames per second (fps), as the interpolation operation lower the speed of the algorithm.

5.1.3 Results on VOT2016

VOT-2016 dataset contains 60 video sequences. There are two kinds of evaluation methods for VOT, namely supervised and unsupervised evaluation methods. Supervised evaluation method provides the target position to re-initialize the algorithm for continue tracking when the tracked target is lost. In contrast, the unsupervised evaluation method does not re-initialize the algorithm. In VOT, accuracy, robustness, and expected average overlap (EAO) [52] are used to evaluate the tracking results. Accuracy refers to the average overlap rate of tracking algorithm results, robustness refers to the average number of tracking failures (when the overlap rate is 0, it can be determined as failure), and EAO is the average of the average overlap rate on a short-term sequence.

The comparison results are shown in Table 2 and the results of the best algorithm are in bold, and the results of the second best algorithm are with italics. The accuracy and robustness of the proposed algorithm rank the second in the case of supervised. The supervised evaluation re-initializes when target occlusion occurs; then, the algorithms can track the target in the video sequence after occlusion. Thus, the advantages of our method is not remarkable in supervised evaluation. Without re-initialization, the accuracy and robustness of the proposed method are the best.

The A-R plot shows the performance of tracker directly. The abscissa and the ordinate of A-R plot are Accuracy and Robustness, respectively. Since the robustness has no upper bound, the reliability of VOT is replaced by robustness and the reliability is computed by $R_s = e^{-SM}$, where M represents the mean time-between-failures, S is the number of the successful object tracking frames since the last failure. The closer the dot is to the upper right corner, the better accuracy and robustness the algorithm has. In Fig. 6, the accuracy and robustness of the proposed algorithm are remarkably good.

5.1.4 Video with occlusion

The convolution operation further degrades the frame resolution. The proposed algorithm focuses on the solution of the occlusion problem, so the experimental results

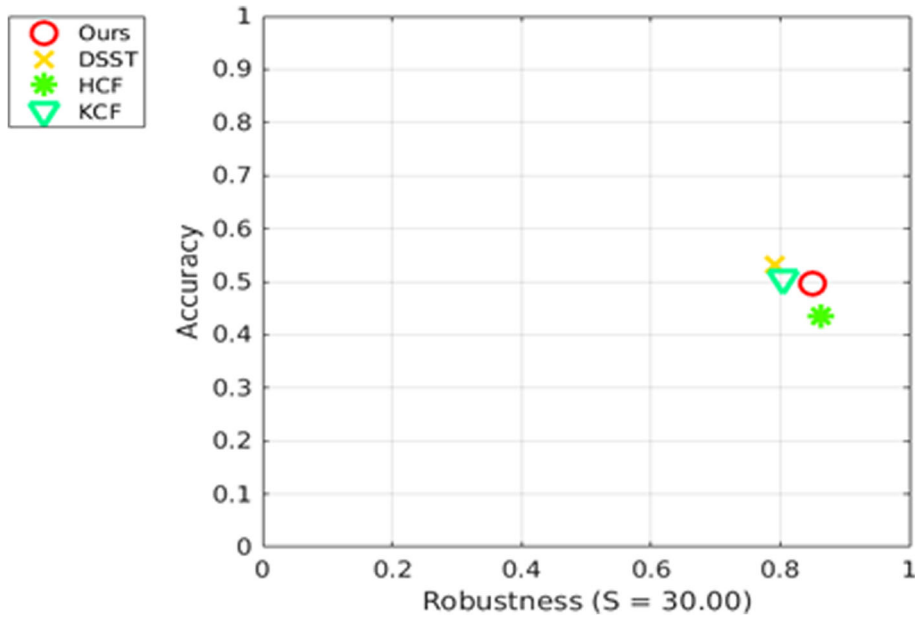


Fig. 6 A-R plot

in the OTB-15 dataset without some low-resolution sequences (Skiing and Walking) are shown in Figs. 7 and 8. The proposed algorithm achieves the best results in the video sequences with the challenges of occlusion, fast motion, deformation, illumination variance, and scale variance.

Occlusion is a great challenge for CF-based methods. The conventional filters usually need to be updated at all frames, including the frames in which the target is occluded, so it is possible that the background information is used to update the filter, and declines the discriminative ability of the filters. The standard CF-based trackers obtain the AUC scores of 0.560 (SAMF), 0.467 (KCF), and 0.464 (DSST). We use the features extracted by ResNet and a novel update strategy to improve the robustness to occlusion. In the video sequence with occlusion, the proposed method obtains the best AUC score (0.592), which is 5.1% higher than that of HCF (0.541), followed by DCFNet (0.584), as shown in the first row of Fig. 7.

5.1.5 Video with scale variation

The tracking overlap rate of our method is improved in the video sequences with target scale variation. The variance of target scale remarkably affects the position estimation, since the size of search area is highly correlated with the target scale. In the video sequences with scale variation, the standard CF-based trackers, without the consideration of scale variation, obtain the scores of 0.425 (KCF) and 0.343 (CSK), while the standard CF-based trackers considering scale variation can obtain the scores of 0.522 (SAMF) and 0.498 (DSST).

The features also can affect the scale estimation, so deep features are used in HCF without the consideration

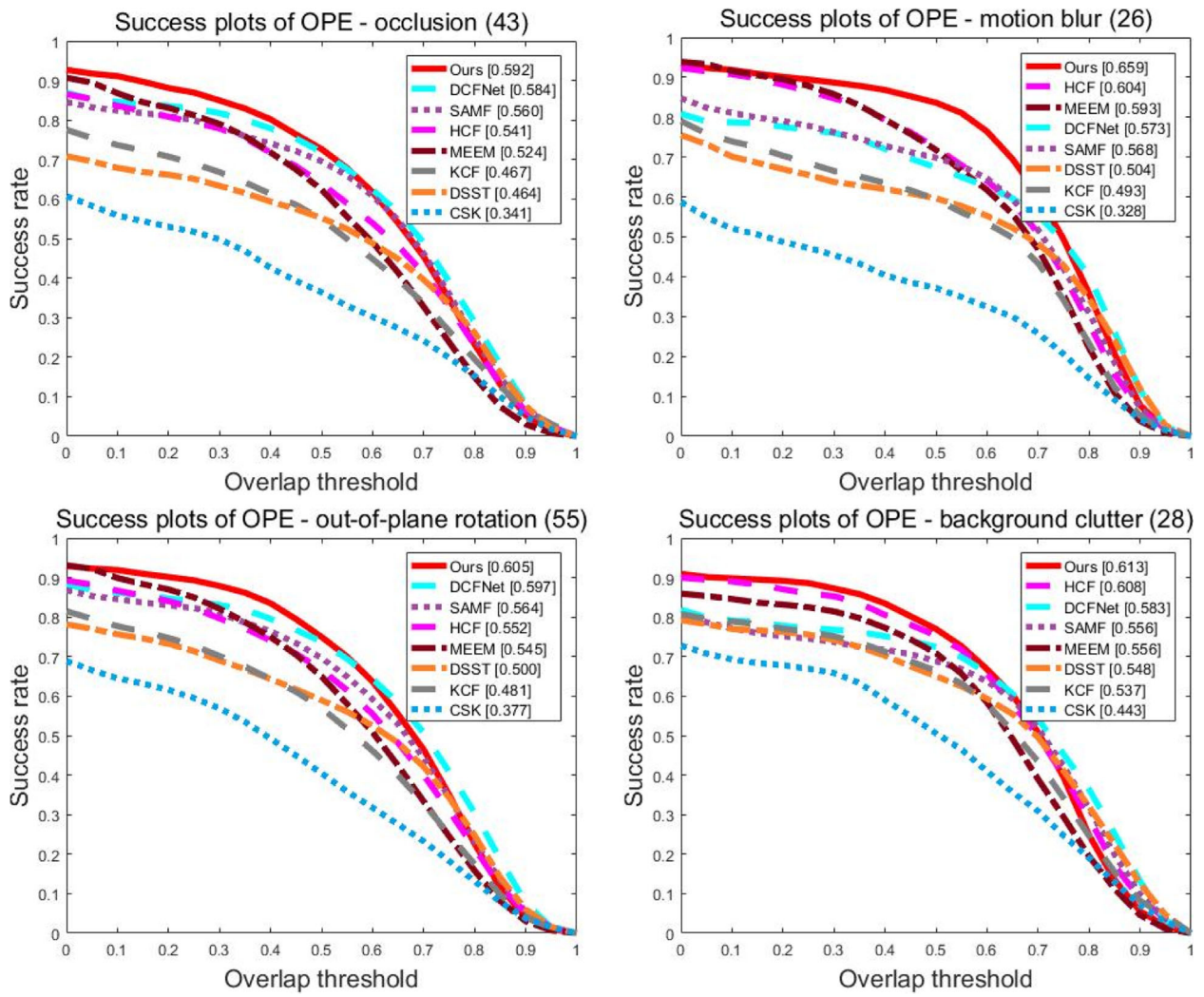


Fig. 7 Overlap success plots in sequence with out-of-plane rotation, occlusion, motion blur, and background clutter

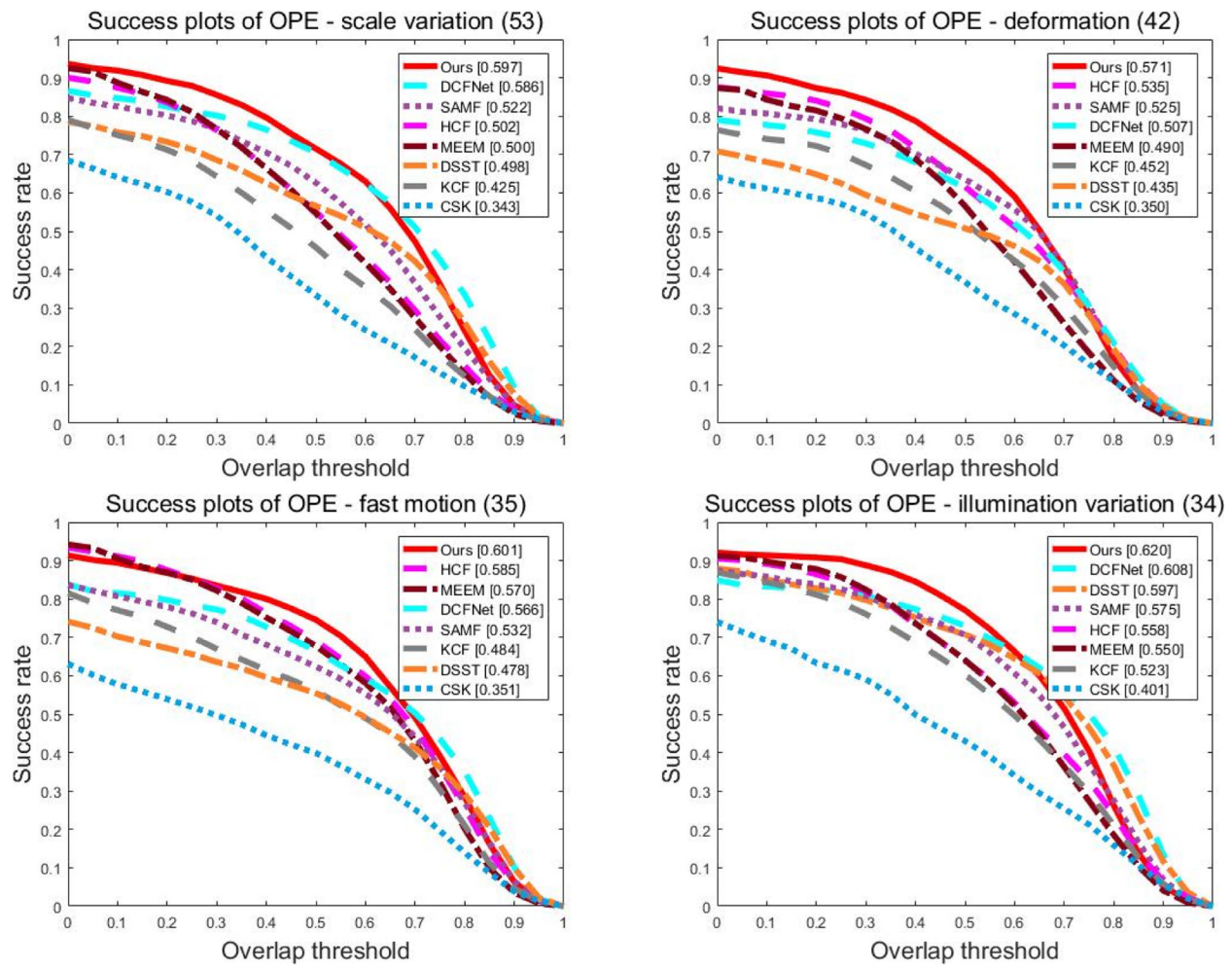


Fig. 8 Overlap success plots in sequence with deformation, fast motion, illumination variation, and scale variation

of scale variance, the AUC score of HCF is 0.502. Our method takes into account both deep features and scale variance, as shown in the first row of Fig. 8. Our method achieves the best AUC score (0.597), which is higher 9.5% than that of HCF (0.502).

5.2 Qualitative evaluation

Figure 9 shows the qualitative evaluation of the proposed method, HCF, DCFNet, KCF, and DSST on 8 video sequences including occlusion and scale variance. HCF performs well in fast moving (Skiing) while fails to track the occluded target (Girl2, Lemming). DCFNet is good at low-resolution sequences as the resolutions of the extracted features are the same as that of the input image, and it is prone to track unsuccessfully for fast moving, target deformation, and background clutter (Skiing, Human9, and Football). HOG features and kernel method are used in KCF to improve the operation efficiency, so it performs well in the cases of fast moving and background interference (Human9), but it is easy to fail when the target is occluded (Girl2, Lemming). In DSST, scale filter is employed to find the current scale (Dog1) of the target when the target scale changes. The proposed method applies the features extracted with ResNet, which are more robust to several challenges. At the same time, it is not easily disturbed by target occlusion due to the optimized update strategy. Therefore, the proposed algorithm can still track the target stably (Girl2, Lemming, Skiing, Football) in the cases of occlusion, deformed and background interference. We also use scale filters for the variance of the target scale (Human9).

5.3 Feature comparison

In order to compare the different combination strategies, the features from different layers of ResNet are combined,

Fig. 9 Qualitative evaluation of the proposed method, DCFNet, DSST, KCF, and HCF on seven challenging sequences

as shown in Table 3. The best results are in bold. On the OTB-15 dataset, the combination of the features extracted from conv3 and conv4 layers achieves the best results, which verifies the rationality of the feature selection of the proposed algorithm.

Table 3 Results of different features

ConvLayer	2	3	4	3, 4	2, 3, 4
AUC	0.523	0.565	0.567	**0.597**	0.573
Precision	0.714	0.781	0.827	**0.852**	0.804

5.4 Update strategies

NELM and PSR are two methods for occlusion detection. NELM + PSR represents an update strategy combining the two methods, and None represents an update strategy without occlusion detection. The result are shown in

Table 4 Results of different update strategies

Update strategy	NELM + PSR	NELM	PSR	None
AUC	**0.597**	0.588	0.575	0.557
Precision	**0.852**	0.583	0.827	0.788

Table 5 Results of different networks

Network	Feature size	AUC (background clutter)
VGG [10]	1×512 × 14×14	0.544
DenseNet [13]	1×384 × 14×14	0.562
ResNet [15]	1×256 × 14×14	**0.574**

Table 4 and the best result are in bold. The proposed method achieves the best results by combining the two methods, which verifies the effectiveness of the proposed update strategy.

5.5 Different networks

We compare the features extracted from different network structures, and the results are shown in Table 5. The best results are in bold. DenseNet [13] is also a network with residual structure, with fewer parameters and deeper network layers than ResNet, in the same time, its extracted features have more channels. According to the classification of OTB-15, we choose the video sequences with background clutter. What is more, we use only one feature with the same resolution from each network and we do not use any strategies. The experimental results show that the results of DensNet are slightly lower than ResNet. However, the results of ResNet and DensNet have achieved better results than VGG.

5.6 Failure cases

We show a few failure cases in Fig. 10. For the Panda sequence, the resolution is 312 × 233. When the target becomes very small, the proposed tracker fails to follow the target because it has few pixels, which can result in poor performance features. An alternative implementation using the feature from conv2 alone is able to track the target, because the conv2 features have higher resolution than the features from deeper layers. For the Biker sequence, the target suddenly moves violently beyond the search area of the proposed tracker. This sequence is still a challenge sequence for many trackers.

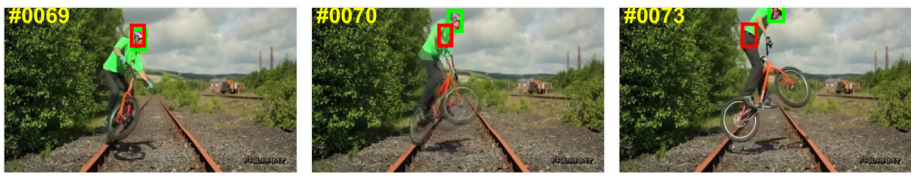
6 Conclusions

Object tracking is a very useful public safety technology. The object tracking algorithm can track specific target in the surveillance video. In addition, combined with some ReID technologies [53], object tracking algorithms can be in used across camera scenes.

A scale-adaptive object-tracking algorithm with occlusion detection has been proposed in this paper. ResNet was used to extract more robust features. In the tracking process, the response maps computed from the different layers are weighted and fused based on AdaBoost algorithm for accurate localization. The NELM and PSR of the response map were used for the optimized update strategy, which can handle the problem of target occlusion. Scale filters have been extended for scale tracking. Compared with the mainstream algorithms, the experimental results showed that the proposed method could track the target robustly and accurately even in the cases of occlusion and scale variation.

In the future, we will try to further improve the robustness of algorithm to low-resolution and the real-time performance.

(a) 'Panda' sequence



(b) 'Biker' sequence

Fig. 10 Failure cases. Red and green boxes represent tracking results and ground truth, respectively. **a** "Panda" sequence. **b** "Biker" sequence

Abbreviations
AUC: Area under curve; CF: Correlation filter; CNN: Convolutional neural network; FFT: Fast Fourier transformation; HOG: Histogram of oriented gradient; NELM: Number of effective local maxima; OPE: One pass evaluation; PSR: Peak-to-sidelobe ratio; SRE: Spatial robustness evaluation; TRE: Temporal robustness evaluation

Acknowledgements
Not applicable.

Authors' contributions
All authors took part in the discussion of the work described in this paper. All authors read and approved the final manuscript.

Funding
This work was supported in part by the National Natural Science Foundation of China under grants 61663031, 61866028, 61661036, 61763033, 61662049, 61741312, 61881340421, and 61866025; the Key Program Project of Research and Development (Jiangxi Provincial Department of Science and Technology) under grants 20171ACE50024 and 20192BBE50073; the Construction Project of Advantageous Science and Technology Innovation Team in Jiangxi Province under grant 20165BCB19007; the Application Innovation Plan (Ministry of Public Security of P. R. China) under grant 2017YYCXJXST048; and the Open Foundation of Key Laboratory of Jiangxi Province for Image Processing and Pattern Recognition under grants ET201680245, TX201604002.

Availability of data and materials
We used publicly available dataset in order to illustrate and test our methods. The OTB dataset can be found in http://cvlab.hanyang.ac.kr/trackerbenchmark/datasets.html and the VOT dataset can be found in http://www.votchallenge.net/. The code of this paper is available in https://github.com/YueYuan95/RCF4.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]Key Laboratory of Jiangxi Province for Image Processing and Pattern Recognition, Nanchang Hangkong University, 330063 Nanchang, China. [2]School of software, Nanchang Hangkong University, 330063 Nanchang, China. [3]School of Aeronautical Manufacturing Engineering, Nanchang Hangkong University, 330063 Nanchang, China. [4]Intelligent Vision Processing Lab, Department of IT Engineering, Sookmyung Women's University, Seoul, Korea.

Received: 13 June 2019 Accepted: 30 January 2020
Published online: 17 February 2020

References

1. G. Zhang, J. Yang, W. Wang, Y. H. Hu, J. Liu, Adaptive visual target tracking algorithm based on classified-patch kernel particle filter. EURASIP J. Image Video Process. **2019**(1), 20 (2019)
2. B.-G. Kim, G.-S. Hong, J.-H. Kim, Y.-J. Choi, An efficient vision-based object detection and tracking using online learning. J. Multimed. Inf. Syst. (KMMS). **4**, 285–288 (2017)
3. B.-G. Kim, D.-J. Park, Novel target segmentation and tracking based on fuzzy membership distribution for vision-based target tracking system. Image Vis. Comput. **24**, 1319–1331 (2006)
4. G.-S. Hong, S.-H. Yang, B.-G. Kim, Y.-S. Hwang, K.-K. Kwoni, Fast multi-feature pedestrian detection algorithm based on discrete wavelet transform for interactive driver assistance system. Multimed. Tools Appl. **75**, 15229–15245 (2016)
5. S. Jung, Y. Kim, E. Hwang, Real-time car tracking system based on surveillance videos. EURASIP J. Image Video Process. **2018**(1), 133 (2018)
6. E. Kermani, D. Asemani, A robust adaptive algorithm of moving object detection for video surveillance. EURASIP J. Image Video Process. **2014**(1), 27 (2014)
7. G. Bhat, J. Johnander, M. Danelljan, F. S. Khan, M. Felsberg, in *ECCV*. Unveiling the power of deep tracking, (2018), pp. 493–509. https://doi.org/10.1007/978-3-030-01216-8_30
8. P. Li, D. Wang, L. Wang, H. Lu, Deep visual tracking: Review and experimental comparison. Pattern Recogn. **76**, 323–338 (2018)
9. A. Krizhevsky, I. Sutskever, G. E. Hinton, in *NIPS*. Imagenet classification with deep convolutional neural networks, (2012), pp. 1097–1105. https://doi.org/10.1145/3065386
10. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
11. Y. Wu, J. Lim, M.-H. Yang, in *CVPR*. Online object tracking: A benchmark, (2013), pp. 2411–2418. https://doi.org/10.1109/cvpr.2013.312
12. Y. Wu, J. Lim, M.-H. Yang, Object tracking benchmark. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1834–1848 (2015)
13. G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, in *CVPR*. Densely connected convolutional networks, (2017), pp. 4700–4708. https://doi.org/10.1109/cvpr.2017.243
14. C. Szegedy, S. Ioffe, V. Vanhoucke, A. A. Alemi, in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning (AAAI Press, 2017), pp. 4278–4284. https://dl.acm.org/doi/abs/10.5555/3298023.3298188
15. K. He, X. Zhang, S. Ren, J. Sun, in *CVPR*. Deep residual learning for image recognition, (2016), pp. 770–778
16. K. He, X. Zhang, S. Ren, J. Sun, in *ECCV*. Identity mappings in deep residual networks, (2016), pp. 630–645. https://doi.org/10.1007/978-3-319-46493-0_38
17. N. Wang, J. Shi, D.-Y. Yeung, J. Jia, in *ICCV*. Understanding and diagnosing visual tracking systems, (2015), pp. 3101–3109. https://doi.org/10.1109/iccv.2015.355
18. J. Zhang, S. Ma, S. Sclaroff, in *European Conference on Computer Vision*. Meem: Robust tracking via multiple experts using entropy minimization (Springer, 2014), pp. 188–203. https://doi.org/10.1007/978-3-319-10599-4_13
19. N. Wang, D.-Y. Yeung, in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*. Learning a Deep Compact Image Representation for Visual Tracking (Curran Associates Inc., Red Hook, 2013), pp. 809–817. https://dl.acm.org/doi/10.5555/2999611.2999702. https://papers.nips.cc/paper/5192-learning-a-deep-compact-image-representation-for-visual-tracking.pdf
20. H. Li, Y. Li, F. Porikli, Deeptrack: Learning discriminative feature representations online for robust visual tracking. IEEE Trans. Image Process. **25**(4), 1834–1848 (2016)
21. H. Nam, B. Han, in *CVPR*. Learning multi-domain convolutional neural networks for visual tracking, (2016), pp. 4293–4302. https://doi.org/10.1109/cvpr.2016.465
22. S. Hong, T. You, S. Kwak, B. Han, in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. Online Tracking by Learning Discriminative Saliency Map with Convolutional Neural Network (JMLR.org, 2015), pp. 597–606. https://dl.acm.org/doi/10.5555/3045118.3045183. http://proceedings.mlr.press/v37/hong15.pdf
23. S. Pu, Y. Song, C. Ma, H. Zhang, M.-H. Yang, in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. Deep Attentive Tracking via Reciprocative Learning (Curran Associates Inc., Red Hook, 2018), pp. 1935–1945. https://dl.acm.org/doi/abs/10.5555/3326943.3327121
24. L. Wang, W. Ouyang, X. Wang, H. Lu, in *CVPR*. Visual tracking with fully convolutional networks, (2015), pp. 3119–3127. https://doi.org/10.1109/iccv.2015.357
25. X. Lu, H. Huo, T. Fang, H. Zhang, Learning deconvolutional network for object tracking. IEEE Access. **6**, 18032–18041 (2018)
26. Y. Song, C. Ma, X. Wu, L. Gong, L. Bao, W. Zuo, C. Shen, R. W. Lau, M.-H. Yang, in *CVPR*. Vital: Visual tracking via adversarial learning, (2018), pp. 8990–8999. https://doi.org/10.1109/cvpr.2018.00937
27. I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. Generative Adversarial Nets (MIT Press, Cambridge, 2014), pp. 2672–2680. https://dl.acm.org/doi/10.5555/2969033.2969125
28. D. S. Bolme, J. R. Beveridge, B. A. Draper, Y. M. Lui, in *CVPR*. Visual object tracking using adaptive correlation filters, (2010), pp. 2544–2550. https://doi.org/10.1109/cvpr.2010.5539960

29. J. F. Henriques, R. Caseiro, P. Martins, J. Batista, in *ECCV*. Exploiting the circulant structure of tracking-by-detection with kernels, (2012), pp. 702–715. https://doi.org/10.1007/978-3-642-33765-9_50
30. J. F. Henriques, R. Caseiro, P. Martins, J. Batista, High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. **37**(3), 583–596 (2015)
31. M. Danelljan, G. Häger, F. Khan, M. Felsberg, in *BMVC*. Accurate scale estimation for robust visual tracking, (2014), pp. 1–11. https://doi.org/10.5244/c.28.65
32. Y. Li, J. Zhu, in *ECCV*. A scale adaptive kernel correlation filter tracker with feature integration, (2014), pp. 254–265. https://doi.org/10.1007/978-3-319-16181-5_18
33. M. Danelljan, G. Hager, F. Shahbaz Khan, M. Felsberg, in *ICCV*. Learning spatially regularized correlation filters for visual tracking, (2015), pp. 4310–4318. https://doi.org/10.1109/iccv.2015.490
34. F. Li, C. Tian, W. Zuo, L. Zhang, M.-H. Yang, in *CVPR*. Learning spatial-temporal regularized correlation filters for visual tracking, (2018), pp. 1–11. https://doi.org/10.1109/cvpr.2018.00515
35. M. Cen, C. Jung, Complex form of local orientation plane for visual object tracking. IEEE Access. **5**, 21597–21604 (2017)
36. N. Dalal, B. Triggs, in *CVPR*. Histograms of oriented gradients for human detection, (2005), pp. 886–893. https://doi.org/10.1109/cvpr.2005.177
37. M. Danelljan, F. Shahbaz Khan, M. Felsberg, J. Van de Weijer, in *CVPR*. Adaptive color attributes for real-time visual tracking, (2014), pp. 1090–1097. https://doi.org/10.1109/cvpr.2014.143
38. M. Danelljan, G. Hager, F. Shahbaz Khan, M. Felsberg, in *ICCV*. Convolutional features for correlation filter based visual tracking, (2015), pp. 58–66. https://doi.org/10.1109/iccvw.2015.84
39. M. Danelljan, A. Robinson, F. S. Khan, M. Felsberg, in *ECCV*. Beyond correlation filters: Learning continuous convolution operators for visual tracking, (2016), pp. 472–488. https://doi.org/10.1007/978-3-319-46454-1_29
40. M. Danelljan, G. Bhat, F. S. Khan, M. Felsberg, et al., in *CVPR*. Eco: Efficient convolution operators for tracking, (2017), pp. 3–15. https://doi.org/10.1109/cvpr.2017.733
41. C. Ma, J.-B. Huang, X. Yang, M.-H. Yang, in *ICCV*. Hierarchical convolutional features for visual tracking, (2015), pp. 3074–3082. https://doi.org/10.1109/iccv.2015.352
42. D. Li, G. Wen, Y. Kuai, Collaborative convolution operators for real-time coarse-to-fine tracking. IEEE Access. **6**, 14357–14366 (2018)
43. J. Li, X. Zhou, S. Chan, S. Chen, Robust object tracking via large margin and scale-adaptive correlation filter. IEEE Access. **6**, 12642–12655 (2018)
44. X. Qi, W. Huabin, Z. Jian, T. Liang, Real-time online tracking via a convolution-based complementary model. IEEE Access. **6**, 30073–30085 (2018)
45. Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, M.-H. Yang, in *CVPR*. Hedged deep tracking, (2016), pp. 4303–4311. https://doi.org/10.1109/cvpr.2016.466
46. Q. Wang, J. Gao, J. Xing, M. Zhang, W. Hu, Dcfnet: Discriminant correlation filters network for visual tracking. arXiv preprint, 1–5 (2017). arXiv:1704.04057
47. M. Collins, R. E. Schapire, Y. Singer, Logistic regression, adaboost and bregman distances. Mach. Learn. **48**(1–3), 253–285 (2002)
48. Z. Zhu, B. Liu, Y. Rao, Q. Liu, R. Zhang, Stresnet_cf tracker: The deep spatiotemporal features learning for correlation filter based robust visual object tracking. IEEE Access. **7**, 30142–30156 (2019)
49. Z. He, Y. Fan, J. Zhuang, Y. Dong, H. Bai, in *Proceedings of the IEEE International Conference on Computer Vision*. Correlation filters with weighted convolution responses, (2017), pp. 1992–2000. https://doi.org/10.1109/iccvw.2017.233
50. M. Collins, R. E. Schapire, Y. Singer, Logistic regression, adaboost and bregman distances. Mach. Learn. **48**(1–3), 253–285 (2002)
51. M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernandez, T. Vojir, G. Hager, G. Nebehay, R. Pflugfelder, in *ICCV*. The visual object tracking vot2015 challenge results, (2015), pp. 1–23. https://doi.org/10.1109/iccvw.2015.79
52. L. Čehovin, A. Leonardis, M. Kristan, Visual object tracking performance measures revisited. IEEE Trans. Image Process. **25**(3), 1261–1274 (2016)
53. X. Ma, X. Zhu, S. Gong, X. Xie, J. Hu, K. M. Lam, Y. Zhong, Person re-identification by unsupervised video matching. Pattern Recogn. **65**(C), 197–210 (2016)

Publisher's Note