The Visual Computer (2021) 37:1029–1043
https://doi.org/10.1007/s00371-020-01850-4

ORIGINAL ARTICLE

Correlation tracking with implicitly extending search region

Qiang Qian[1,2] · Xiao-Jun Wu[1] · Josef Kittler[3] · Tian-Yang Xu[1]

Published online: 25 May 2020

Abstract
Recently, the correlation filters have been successfully applied to visual tracking, but the boundary effect severely limits their tracking performance. In this paper, to overcome this problem, we propose a correlation tracking framework with the capacity implicitly to extend the search region (TESR), while inhibiting the undesirable impact of the background noise. The proposed tracking method is a two-stage detection framework. The search region of the correlation tracker is extended by considering other four search centers, in addition to the target location in the previous frame. Thus our TESR will generate five potential object loactions. Then, an SVM classifier is used to determine the correct target position. We also introduce and apply the salient object detection score to regularize the output of the SVM classifier to improve its performance. The experimental results demonstrate that TESR exhibits superior performance in comparison with the state-of-the-art trackers.

Keywords Visual tracking · Correlation filter · Boundary effect

1 Introduction

Video tracking is a hot research topic in the field of computer vision. It plays an important role in many applications, such as robotics, surveillance and human–computer interaction, to name a few [5,6,8–10,13,16–18,20,21,30,37,42,47,48]. In general, tracking target is defined in the first frame of the video in terms of an upright bounding box. A tracker identifies the location of the object in the subsequent frames. Over the last decade, video tracking has advanced significantly, but still many challenging problems remain to be solved such as fast motion, camera shaking, occlusion, and so on.

In the recent tracking algorithms, the correlation filter (CF) tracker and the two deep neural network-based tracker are the two mainstream tracking methods. The tracking algorithms based on deep networks have excellent performance because they take advantage of the large amount of data training off-line, but they often suffer in terms of tracking speed because of their slow online fine-tuning. The CF tracking algorithms have received wide attention for their impressive performance in both speed and accuracy, as well as benefiting from their very simple implementation. In contrast to neural network-based tracking method, the correlation filter method is the lightweight tracker. Extensive research and various forms of improvement of CF have been reported, but some open problems still persist. The boundary effect that is closely linked with the underlying assumption that the object signal is periodic is a prominent problem in CF. The intuitive method to solve this problem is to extend the search region of CF, but this introduces irrelevant background information which is not periodic and will lead to the decline of the tracking performance. Kiani Galoogahi et al. [20] and Danelljan et al. [10] circumvent the background noise problem associated with the search region extension by adopting other approaches. The proposed method addresses this problem, by implicitly extending the search region to overcome the boundary effect of CF by means of a multiposition tracking solution.

In this paper, we propose a correlation tracking framework, which implicitly extends the search region (TESR). The conventional CF methods take the target location in the

✉ Xiao-Jun Wu
wu_xiaojun@jiangnan.edu.cn

Qiang Qian
qianqiang_just@163.com

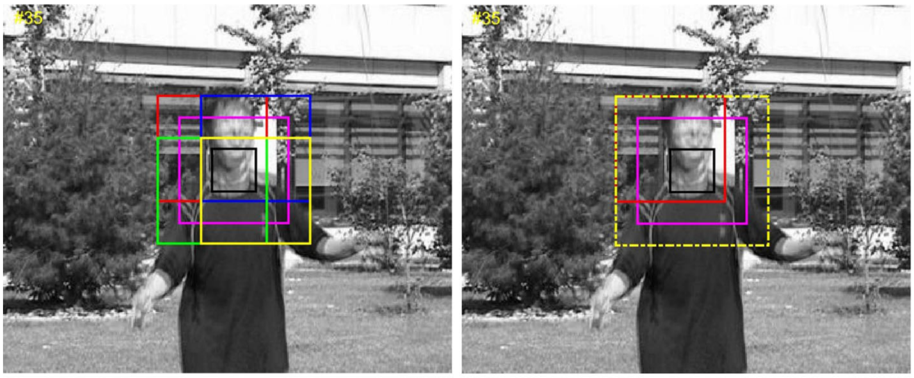
Josef Kittler
j.kittler@surrey.ac.uk

Tian-Yang Xu
tianyang_xu@163.com

[1] School of IoT Engineering, Jiangnan University, Wuxi, China

[2] School of Computer Science and Engineering, Jiangsu University of Science and Technology, Zhenjiang, China

[3] Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford, UK



(a)

(b)

Fig. 1 In **a**, the black box represents the target location in the previous frame. The conventional CF detects the object in the area defined by the magenta box. Apart from this area, we add four extra search regions around the original search center, shown in red, blue, green and yellow. In **b**, the red box represents one of the additional extra search regions the black box and the magenta box are the same as that in **a**. The yellow dash dot box is the implicit search region of TESR

previous frame as the current frame search center and detect object in a search region of bigger size than the object size, delineated by the bounding box, to find the correct location of the tracking target. The radius of the surrounding area of the bounding box is expanded by a factor as $p * r$, where r is the radius of the object. Hence, the search region radius of conventional CF methods is $(1 + p) * r$, where p is the padding scale factor. The proposed TESR adopts a multiposition tracking method. As shown in Fig. 1, in addition to the search center with the location of the tracked target in the previous frame, we consider multiple extra search center positions around the original search center. For every added position, we also search the object in the region of size: $(1 + p) * r$. The distance between the added center and the original center is set to $\alpha * r$, the total radius of our search region is $(1 + p + \alpha) * r$. We can see the search region is implicitly enlarged without introducing extra background noise.

The main contributions of this work can be summarized as follows:

– We propose a novel method to overcome the boundary effect of CF which implicitly extends the search region without compromising periodicity.
– We adopt a two-stage tracking strategy: CF detection and SVM detection. Different from the traditional multiclassifier structure, the two classifiers use independent updating periods based on the model of ASMM [1].
– To deal with the problem of drift, we apply the salient object detection score to regularize the output of the SVM classifier with promising results.

We show that TESR can obtain excellent results. The experiments demonstrate the superior performance of TESR as compared with the state-of-the-art trackers on the popular OTB [44,45] and VOT [22,23] benchmarks.

2 Related works

Video tracking has been widely studied over the past decades. There is a considerable body of related literature, where both CF trackers and deep networks trackers are reported to have achieved competitive performance. We only focus on these two main types of trackers in this paper. We refer the reader to [34,44] for further details of visual tracking.

Correlation filter The recent upsurge of interest in correlation filters started with the Bolme's seminal article [5], advocating a tracking method based on the minimum output of the sum of squared error (MOSSE). This is a high-speed tracker with 600–700 FPS. Henriques [16] realized a discriminative correlation tracker based on dense sampling by using the property of the circulant matrix. His algorithm, named KCF, was improved in 2015 by adding kernel method and multichannel features (e.g. HOG). SAMF [25] and DSST [9] improved the performance of KCF and MOSSE by endowing them with a scale estimation mechanism. To handle the occlusion, [26,27,48] incorporated a common part-based technology into CF. Both [4,36] proposed methods to handle the limitations of the assumption of isotropy of the CF response. It is worth mentioning that to reduce the boundary effect, [20] proposed a new CF objective function by introducing a specific mask matrix, and SRDCF [10] added a spatial regularization component into the CF objective function. SRDCF achieved impressive performance at the cost of very low

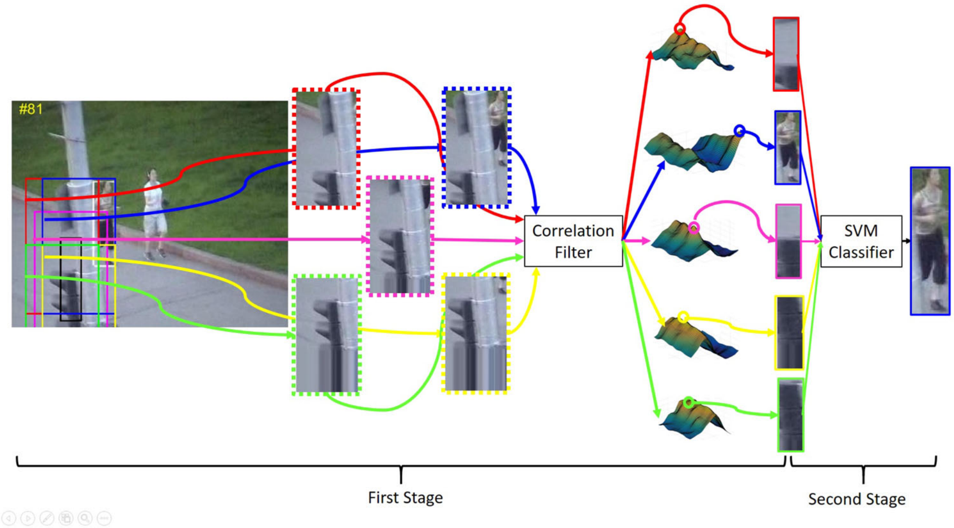


Fig. 2 A flowchart of the proposed tracking framework. In the first stage, the same CF is applied in five different search regions, i.e., red, blue, green, yellow and magenta dot boxes, to generate object candidates. The black box represents the object location in the previous frame. The white box represents the target that is predicted by our method. In the second stage, all candidates will be detected using an SVM classifier. The candidate with the highest score is treated as the tracking target

speed. The two methods have not fundamentally solved the problem of boundary effect because they both circumvent the problem of extending the search region of CF. Different from the [20] and [10], we propose a multiposition search method, which implicitly extends the search region without infecting extra background noise. The proposed method solves the problem of boundary effect elegantly. Related papers on CF also include [2,5,6,9,10,12,16–20,25,27,32,35,36,47,50]. For further details of correlation tracking, please refer to [7].

Deep neural network-based tracking In many areas of computer vision, such as image classification, object detection and segmentation, deep neural network algorithms have reported an impressive success. Recently, neural network-based methods have also been used in the tracking field. There are two main types of the deep neural network-based trackers. One is the online fine-tuning neural network method. For example, MDNet [31] pretrains a convolutional network to obtain a generic target representation and constructs a new network by the pretrained CNN by adding a new binary classification layer, which is updated online. The other is the offline training method. For example, CF2 [29] adopts a pretrained CNN to extract feature in a CF tracking framework, where the neural network is only used for feature extraction. SiameseFC [3] is a fully convolutional Siamese network trained offline on the ILSVRC15 [33] dataset for object detection in video. The two-stream networks do not update online. Related papers on deep neural network-based tracking methods also include [15,24,38–41,49]. Although deep neural network-based tracking algorithms have superior performance, their tracking speed is limited because of the slow online updating. If these methods adopt a network completely trained offline, their accuracy often decreases. Our method is a lightweight correlation filter tracker. The proposed tracker consists of the SVM classifier and the CF, we regard the two classifiers as long-term and short-term ones-respectively, rather than multiple experts for simultaneous fusion. Moreover, the proposed method is a two-stage tracking framework. The components of both stages can be replaced by arbitrary CF and discriminative classifier.

3 Proposed approach

3.1 Overview

TESR is a two-stage tracking framework: The first stage sets to detect object in five different search regions, respectively, by the same correlation filter. The CF will give a response map as its output in each search region. TESR takes the position of the peak value in the response map as the candidate of the tracking object. The second stage tests the five candidates using an SVM classifier and views the candidate with the highest score as the predicted target position. Figure 2 presents a visual representation of the overall tracking procedure.

In the CF detection stage, in addition to the tracked target location found in the previous frame, TESR takes the other multiple tracking positions around the original search center as the base search points. In fact, they are the vertices of the bounding box. As discussed in Sect. 4, when the number of added positions is set to be four, TESR exhibits the best performance. Thus TESR tracks the object five times using the same CF and attains five response maps. During this processing, because we maintain the search size of every CF constant, no extra background noise is introduced. However, on the search region is extended on a whole, the strategy effectively deals with the problem of motion blur, camera shaking, occlusion, and so on.

In the second stage, we choose the SVM as the classifier to select the correct candidate. We can also use the other classifiers such as the neural network, but compared with the neural

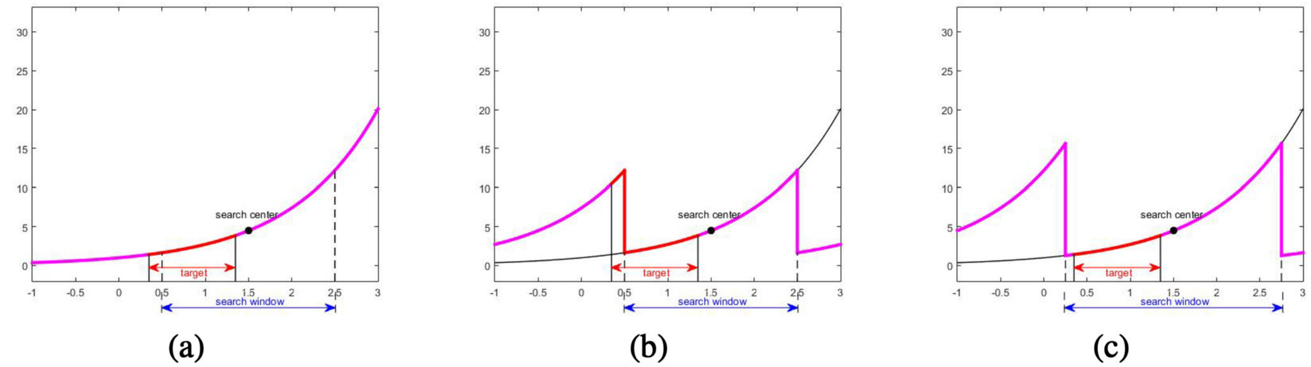


Fig. 3 In **a**, for the sake of simplicity, the 1D pink signal represents the 2D image signal. In **b**, the assumption of periodicity distorts the pink signal in correlation tracking. The smaller the search window, the more sever the distortion of the target patch signal, shown as the red curve segment, and this serves as input to the correlation filter. In **c** when the search window size is enlarged, the target patch signal is undistorted

network, SVM classifier does not need the offline training, and it is also faster than neural network, as well as free from exhibiting degradation in performance. In this stage, TESR applies the method proposed in [43,46] and regularizes the output of the SVM classifier by the salient object detection score to enhance the performance of the SVM classifier. For the five candidates generated in the first stage, we take the candidate with the highest score as the final tracking object position.

The proposed tracking framework consists of two classifiers. Both classifiers must be continuously updated. Similar to [18], we also argue that the tracker should have both the long-term and short-term memories. TESR updates CF every frame to handle any fast appearance change of the object and updates SVM slower, e.g. every 10–15 frames, which is considered as a long-term memory. The support vectors in SVM primarily capture the elementary features of the object.

3.2 CF detection stage

For simplicity, we assume that $\boldsymbol{x} \in \mathbb{R}^n$ represents the tracking target. To take advantage of the property of the circulant matrix, we generate a matrix $\boldsymbol{X}=[\boldsymbol{x}, \boldsymbol{P x}, \boldsymbol{P}^2\boldsymbol{x}, \ldots, \boldsymbol{P}^{n-1}\boldsymbol{x}]$, where $\boldsymbol{P}$ is a permutation matrix denoting the circulant shift of a vector, $\boldsymbol{P x}=[x_n, x_1, \ldots, x_{n-1}]$. The response of CF is modelled as a 2D Gaussian map in the ideal case. The peak of the response map is the object location. The goal of CF is to learn a filter $\boldsymbol{w}$, which can minimize the cost of ridge regression problem as follows:

$$\min_{\boldsymbol{w}}\{\|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2 + \lambda \|\boldsymbol{w}\|^2\} \tag{1}$$

where $\boldsymbol{y}$ is the ideal 2D Gaussian response and λ is the trade-off parameter.

We can solve Eq. 1 and its dual problem to obtain $\boldsymbol{w} = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{\alpha}$, where $\boldsymbol{\alpha} = (\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} + \lambda \boldsymbol{I})^{-1}\boldsymbol{y}$ is the solution of the dual problem of Eq. 1. Using the relationship between circulant matrix and its Fourier transform, we can compute $\boldsymbol{w}$ in the frequency domain very fast as follows:

$$\hat{\boldsymbol{w}} = \frac{\hat{\boldsymbol{x}} \odot \hat{\boldsymbol{y}}}{\hat{\boldsymbol{x}}^* \odot \hat{\boldsymbol{x}} + \lambda} \tag{2}$$

where $\hat{\boldsymbol{x}}$ is the FFT of $\boldsymbol{x}$, and $\hat{\boldsymbol{x}}^*$ is the conjugated FFT of $\boldsymbol{x}$. $\odot$ denotes the element-wise product.

In the object detection stage, we compute the response map $f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$ and take the peak point of the response as the current target location. The details of CF can be found in [17].

As mentioned in Sect. 1, although there has been much improvement in CF in recent years, the boundary effect is the main problem which is difficult to be solved. As shown in Fig. 3, for simplicity, we use the 1D signal to replace the 2D image signal. We find the signal distorted because of the assumption of the periodicity, as indicated by the pink signal shown in Fig. 3b. If the search window is not large enough, the distorted target patch signal input to the correlation filter will cause the tracker to drift. To overcome this problem directly, as shown in Fig. 3c, the target patch signal will not be influenced by the assumption of the periodicity. But the drawback of the process is also obvious. On one hand, we need to expand the search region around the target to cope with fast motion or camera shaking in video. Otherwise, the object will partly move out of the search window and CF will fail to detect it. On the other hand, as CF relies on the property of circulant matrix to speed up the detection and learning of the tracker, if the search region is extended, injecting much background noise in the process, the tracker will start to drift.

As shown in Fig. 1, TESR adds other four search positions, which can implicitly extend total search size without leading to the drift problem. TESR takes the four diagonal vertices of the bounding box (black box) as the additional correlation filter search centers. Consequently, all filters use the original search size. We can see the search region of the left top CF (red box) and the right top CF (blue box) can cover the target, and TESR can ultimately track the object position correctly.

Let us assume the size of target is $m * n$. The padding area around target of conventional CF is p times of the target size, i.e., $(pm) * (pn)$. Assume that the size of search region of conventional CF is $M * N$ such that $M = (1 + p)m$, $N = (1 + p)n$. We denote the image patch of conventional search region as $\boldsymbol{x}_0 = [x_1, x_2, \ldots, x_{M*N}]$. In CF detection stage of TESR, $\boldsymbol{x}_0$ is shifted to the neighbor of the vertexes of bounding box. We denote $\boldsymbol{x}_i = \boldsymbol{x}_0(a_i, b_i)(i = 1, \ldots, 4)$ as added search region, which is shifted from $\boldsymbol{x}_0$ by (a_i, b_i), with a_i and b_i given by

$$a_i = \begin{cases} \alpha * m & i = 1, 3 \\ -\alpha * m & i = 2, 4 \end{cases} \tag{3}$$

$$b_i = \begin{cases} \alpha * n & i = 1, 2 \\ -\alpha * n & i = 3, 4 \end{cases} \tag{4}$$

where α is a scale parameter.

For each search region, TESR will track the object using the same CF. In the first stage of TESR, the response map of each CF is computed as follows:

$$\boldsymbol{y}_i = f(\boldsymbol{x}_i) = \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i \quad \forall i = 0 : 4 \tag{5}$$

For each $\boldsymbol{y}_i$, we find the location of its peak value and regard this position as a target candidate. TESR generates five candidates in this stage. As shown in Sect. 3.3, an SVM classifier selects the correct result of our tracking framework. Each CF still tracks the object in the search region with the size $M * N$ or $(1+p)m*(1+p)n$, but the total search size of TESR is implicitly increased to be $(1+p+\alpha)m*(1+p+\alpha)n$. This approach of extending search region implicitly does not introduce any extra background noise.

To speed up our TESR, we do not use multiposition detection method described above in all frames. For the response of original CF, i.e., $\boldsymbol{y}_0$, is transformed to the probability form by a Gaussian cumulative distribution with mean of 0 and STD of 1. If the peak value of the response is less than a threshold s, TESR will track the object using multiposition detection and apply the SVM classifier to select the best candidate. If the peak value is greater than s, we accept the original CF tracking target as the TESR's result.

3.3 SVM detection stage

In the first stage, there are separate correlation filters, which generate five target candidates, respectively. We could adopt the method of multiclassifier fusion to define our tracker. But as shown in Fig. 2, some outputs of correlation filters look like the target, while some outputs are severely contaminated by noisy background pixels. If we simply adopt a multiclassifier fusion method, the tracker will easily start to drift. For this reason, we adopt an SVM to determine which candidate is the correct target position in the second detection stage. There are still several problems which we must consider: The two classifiers, i.e., CF and SVM, should adopt different feature extraction methods, and SVM classifier should be able to detect and update rapidly. Furthermore, the two classifiers should be updated with different frequencies to adapt to rapid change in the appearance of the target and focus on stable properties of the object.

The SVM classifier used in TESR is inspired by TVM [43,46], which uses a fixed-size training data $Q = \{\zeta_i = (\phi(\boldsymbol{q}_i), \omega_i, s_i)\}_1^B$, $\phi(\boldsymbol{q}_i)$ is the feature vector of an image patch $\boldsymbol{q}_i$, ω_i is a binary label and s_i is the number of support vectors representing the decision boundary. Given new data $L = \{\boldsymbol{x}_i, y_i\}_1^J$, the goal of the TVM is to learn the weight $\boldsymbol{w}$ by minimizing the objective function:

$$\min_{\boldsymbol{w},b} \left\{ \frac{1}{2} \|\boldsymbol{w}\|^2 + C \left\{ \sum_{i=1}^{B} \frac{s_i}{N_{\omega_i}} L_{\mathrm{h}}(\omega_i, \boldsymbol{q}_i; \boldsymbol{w}) + \sum_{i=1}^{J} \frac{1}{N_{y_i}} L_{\mathrm{h}}(y_i, \boldsymbol{x}_i; \boldsymbol{w}) \right\} \right\} \tag{6}$$

where C is the slack parameter, L_{h} is the hinge loss function.

By taking advantage of the twin prototypes, TVM can maintain a reasonable support vector budget and can be designed as a linear SVM classifier, so it can detect and learn to adapt the decision boundary to support the real-time requirement of visual tracking. For the feature extraction, the current mainstream CF methods adopt multichannel feature like HOG [10,17,18,20]. TESR makes use of CIELab feature as the feature representation of TVM. Thus, the two features which are used in CF and TVM are complementary to each other. HOG focuses on the gradient information, while CIELab feature focuses on the color information. Similar to MEEM, to deal with illumination change, we also adopt the nonparametric local rank transform (LRT) [18]. As TVM is viewed as a long-term memory, its updating frequency is slower than that of CF. We find that when SVM classifier updates at every 10–15 frames, TESR achieves the best performance. By managing the updating frequency of SVM classifier, we retain control over the tracking speed as it is inversely proportional to updating frame rate.

3.4 Salient feature detection

As an online learning method, the memory of TESR, i.e., the filter of CF and the support vectors of SVM, bound to be contaminated in practice. To alleviate this problem, we

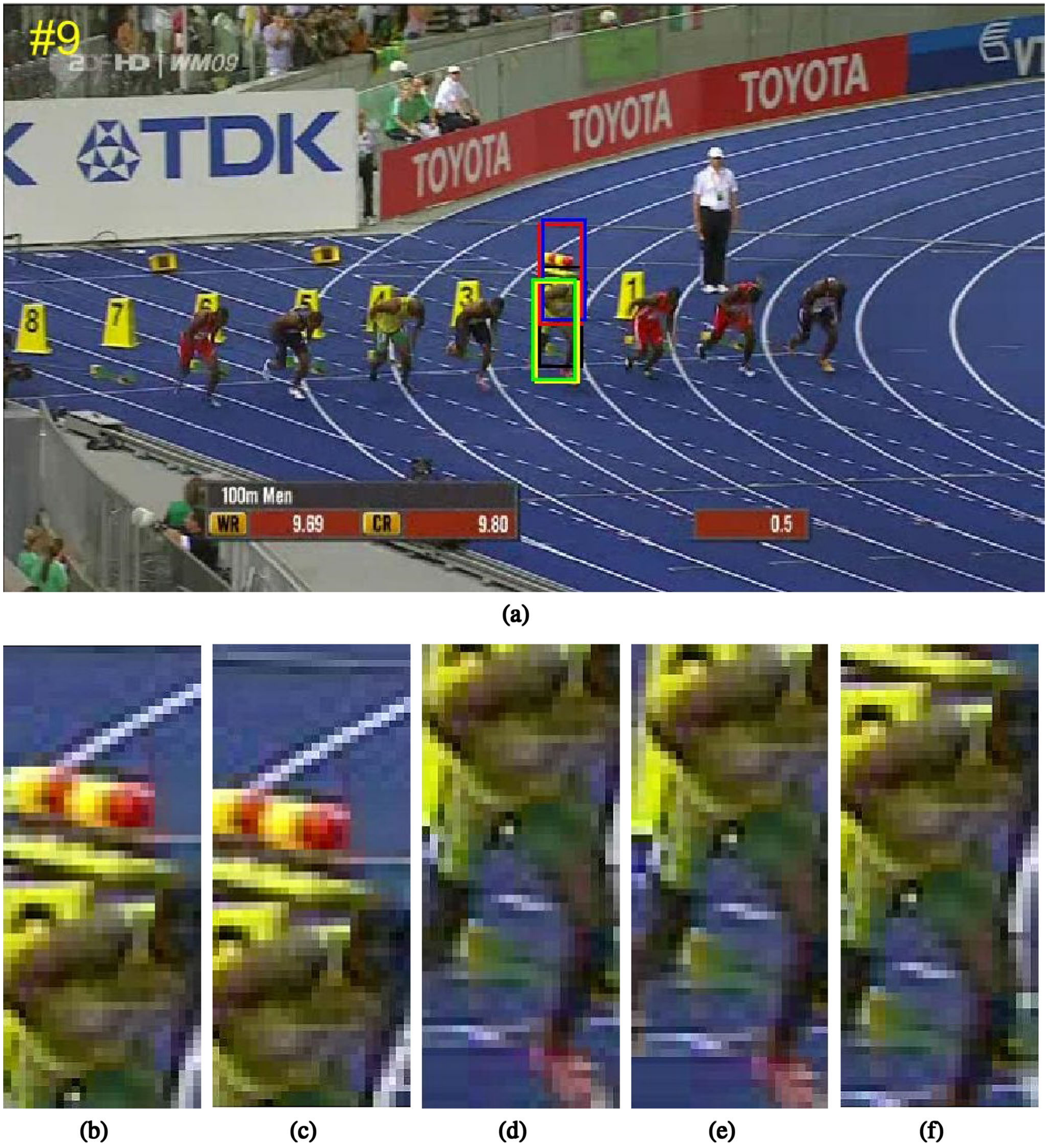


Fig. 4 In **a**, the five candidates generated in the first stage of TESR at the ninth frame of *bolt* sequence are shown as red, blue, green, yellow and black boxes. In **b**–**f**, the five candidate patches are extracted from the original image

adopt a salient feature detection regularization method. We find that the tracking target is always a salient object in general, which can be considered as a priori knowledge of visual tracking. In some cases, we can accurately determine the correct target position from several candidates even without any information extracted from the previous frames. Shown in Fig. 4d–f are more like the tracking target because they contain a salient feature manifest by the whole object. We modify the discriminative method in Sect. 3.3 based on regularizing the output of SVM classifier using the priori knowledge of the saliency of the object.

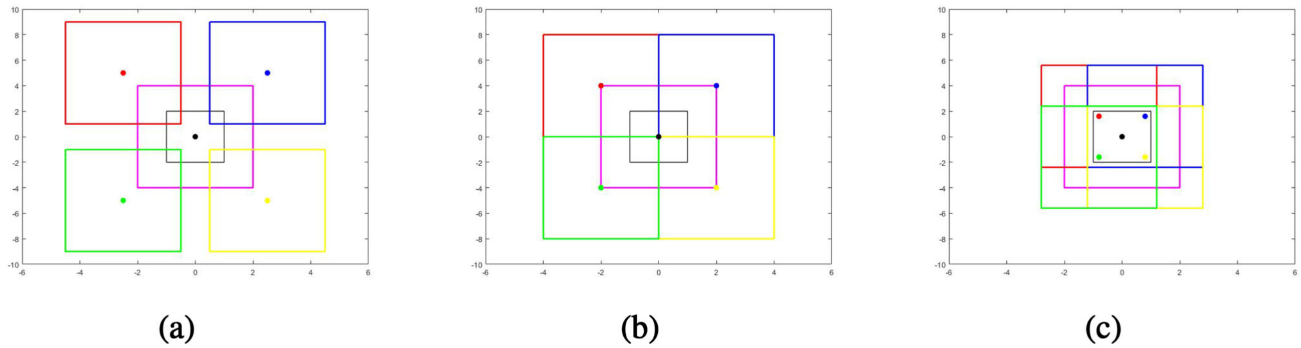


Fig. 5 The black box represents the target location in the previous frame, and the black point represents the its center. The conventional CF detects the object in the area defined by the magenta box. The additional search regions are shown with red, blue, green and yellow boxes, where the value of p is 1, and the color points represent the additional search centers. We set the α to be 2.5, 2 and 0.8 in **a**–**c**. We can see when α is greater than $1 + p$, the additional search regions will not overlap

For the salient feature detection, we use the approach proposed in [28]. We assume that the area of the bounding box is R, and the search region of CF is R_{search}. The surrounding area of the bounding box is R_{s}, i.e., the area in R_{search} besides R, $R_{\text{s}} = R_{\text{search}} - R$. We adopt the Chi-square of the color histogram as the salient feature detection score as follows:

$$\chi^2(R, R_{\text{s}}) = \frac{1}{2}\sum_i \frac{(R^i - R_{\text{s}}^i)^2}{R^i + R_{\text{s}}^i} \tag{7}$$

where R^i and R_{s}^i are the ith bins of the color histogram of R and R_{s}.

Obviously, if the color information of the target candidate area is significantly different from the surrounding area, it is more likely that it contains a salient object. Accordingly, the objective function of SVM detection stage is turned to:

$$f(\boldsymbol{x}) = \boldsymbol{w}^{\text{T}}\boldsymbol{x} + \mu\chi^2(R, R_{\text{s}}) \tag{8}$$

where $\boldsymbol{w}$ is the weight of the SVM classifier and μ is a trade-off parameter.

The five candidates generated in CF detection stage are assessed by the criterion in Eq. 8 to determine the correct tracking target.

4 Experiments

4.1 Base tracker

TESR is regarded as a two-stage detection framework based on CF. The base CF tracker used in the first stage can be any tracker based on CF. We choose two CF trackers as our base trackers: KCF [17] and STAPLE [2] because both have high tracking speed and appealing performance. The discriminative classifier used in the second stage is TVM as mentioned above. We choose it also for its highly efficient and effective performance. We call the two implementations of TESR as TESR_{KCF} and $\text{TESR}_{\text{STAPLE}}$ in the following sections.

Like KCF, our TESR_{KCF} does not adapt to scale changes, but similar to STAPLE, the tracking of $\text{TESR}_{\text{STAPLE}}$ is robust to scale change. The motivation for choosing base CF trackers with different characteristics is to allow a fair comparison between our TESR version and its existing approaches. We shall show that our two CF-based trackers TESR_{KCF} and $\text{TESR}_{\text{STAPLE}}$ are not only better than KCF and STPALE, but also deliver promising performance in comparison with the state-of-the-art trackers.

4.2 Implementation details and parameters

In all our experiments, we use MATLAB on an Intel(R) Xeon(R) 3.30 GHz CPU with 8 GB RAM. For the base tracker KCF and STAPLE, we retain the original parameter settings. In the multiposition detection in CF, we set the threshold s of response map peak value to be 0.6 and the value of α to be 0.8. Although the larger the value of α, the larger the search region, if the α is too large, our additional search regions will not overlap, which may result in the loss of the target. As shown in Fig. 5, the maximum value of α cannot be greater than $1+p$, where p is the padding scale factor. We test the value of α from 0.1 to 2 . With p equal to 1, the best value of α is 0.8. We also test the impact of additional search centers in other positions and increase the counting number of the multiposition detection (the maximum number is set to be 20). Interestingly, the performance is not sensitive to the number of search centers; the reason may be that more false candidates are generated as the number of additional search centers increases. Moreover, the speed will be slower.

For TVM used in the second detection stage, we also retain the original parameter setting in MEEM [46]. The trade-off parameter μ determines the impact of the salient feature

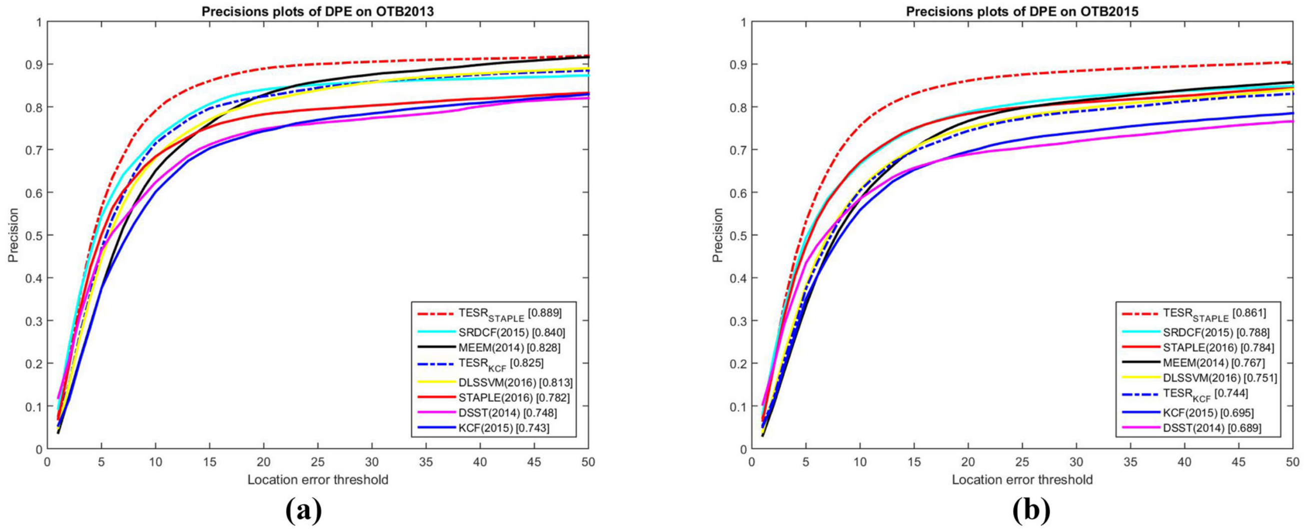


Fig. 6 Precision plots of our TESR by comparison with the state-of-the-art trackers on the OTB2013 **a** and OTB2015 **b** benchmark

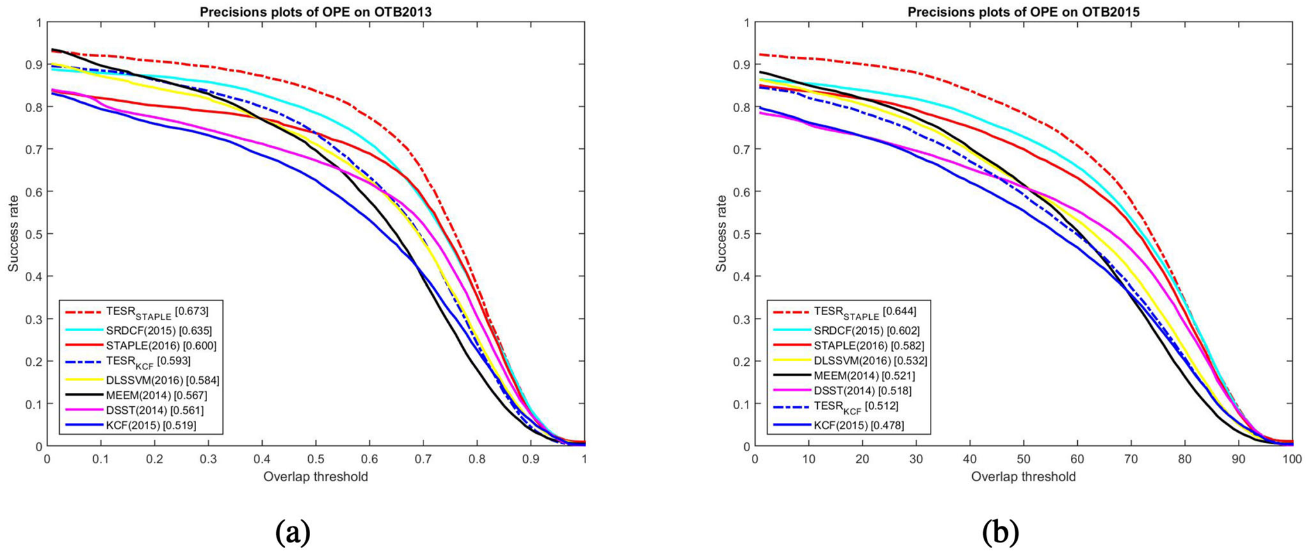


Fig. 7 Success plots of our TESR by comparison with the state-of-the-art trackers on the OTB2013 **a** and OTB2015 **b** benchmark

detection as shown in Eq. 8. We test its value from 0.01 to 10 and find it is best at 0.7. The updating frequencies of CF and SVM are one frame and nine frames for TESR$_{\mathrm{KCF}}$ and one frame and 12 frames for TESR$_{\mathrm{STAPLE}}$, respectively. We use the sliding window sampling approach around the predicted location to obtain the training samples in the updating of TVM. The sampling region has the same size for the CF original search size, i.e., $(1 + p) * r$, where r is the radius of object, p is the padding scale factor. We only extract sample patches at the updating period, i.e., every 9 frames or 12 frames. In addition to these samples, we also retain the object candidates generated in CF detection as the training samples at every frame. For all training sample patches, we calculate the overlaps of them and the predicted object location, We regard the samples with an overlap greater than 90% as positive samples and consider the samples with an overlap less than 50% as negative samples. All the other samples are discarded.

4.3 Evaluation

We run our TESR on two recent popular benchmarks, i.e., OTB [44,45] and VOT [22,23], and compare it to several state-of-the-art trackers. We use the source codes provided by the original authors and run the code ourselves on the OTB benchmark to evaluate their performance. For VOT benchmark, since VOT challenge provides the results of all

Table 1 A comparison of the DPE/OPE scores on sequences classified according 11 challenging factors: IV (illumination variation), SV (scale variation), OCC (occlusion), DEF (deformation), MB (motion blur), FM (fast motion), IPR (in-plane rotation), OPR (out-of-plane rotation), OV (out-of-view), BC (background clutters) and LR (low resolution)

	KCF	DSST	MEEM	DLSSVM	STAPLE	SRDCF	TESR_{KCF}	$\text{TESR}_{\text{STAPLE}}$
IV(38)	0.720/0.482	0.739/0.567	0.739/0.519	0.700/0.511	0.782/0.599	*0.787/0.615*	0.724/0.491	***0.868/0.657***
SV(64)	0.637/0.395	0.648/0.477	0.717/0.460	0.696/0.457	0.731/0.528	*0.750/0.570*	0.688/0.433	***0.822/0.602***
OCC(47)	0.630/0.445	0.614/0.464	0.719/0.495	0.693/0.497	0.728/0.548	*0.732/0.561*	0.667/0.463	***0.820/0.621***
DEF(42)	0.618/0.439	0.562/0.426	0.703/0.467	0.704/0.500	*0.751/0.554*	0.736/0.548	0.677/0.470	***0.824/0.618***
MB(31)	0.618/0.459	0.608/0.499	0.695/0.537	0.693/0.551	0.716/0.559	*0.780/0.617*	0.689/0.518	***0.809/0.633***
FM(42)	0.629/0.456	0.578/0.471	0.696/0.515	0.698/0.525	0.715/0.547	***0.767/0.604***	0.712/0.512	***0.767***/*0.590*
IPR(51)	0.703/0.471	0.707/0.510	0.789/0.524	0.758/0.525	*0.768/0.553*	0.742/0.546	0.748/0.509	***0.830/0.605***
OPR(62)	0.662/0.458	0.663/0.482	*0.786*/0.523	0.755/0.525	0.742/0.541	0.744/*0.556*	0.705/0.483	***0.852/0.624***
OV(14)	0.495/0.393	0.478/0.385	0.647/*0.484*	0.620/0.468	*0.664*/0.477	0.599/0.464	0.552/0.427	***0.747/0.557***
BC(32)	0.717/0.501	0.721/0.526	0.730/0.514	0.735/0.522	0.754/0.565	*0.776/0.585*	0.713/0.505	***0.851/0.630***
LR(9)	0.664/0.284	0.673/0.375	*0.842*/0.381	0.783/0.368	0.689/0.393	0.769/*0.528*	0.735/0.330	***0.891/0.748***

Numbers in parenthesis in the first column refer to the numbers of sequences with the corresponding attributes

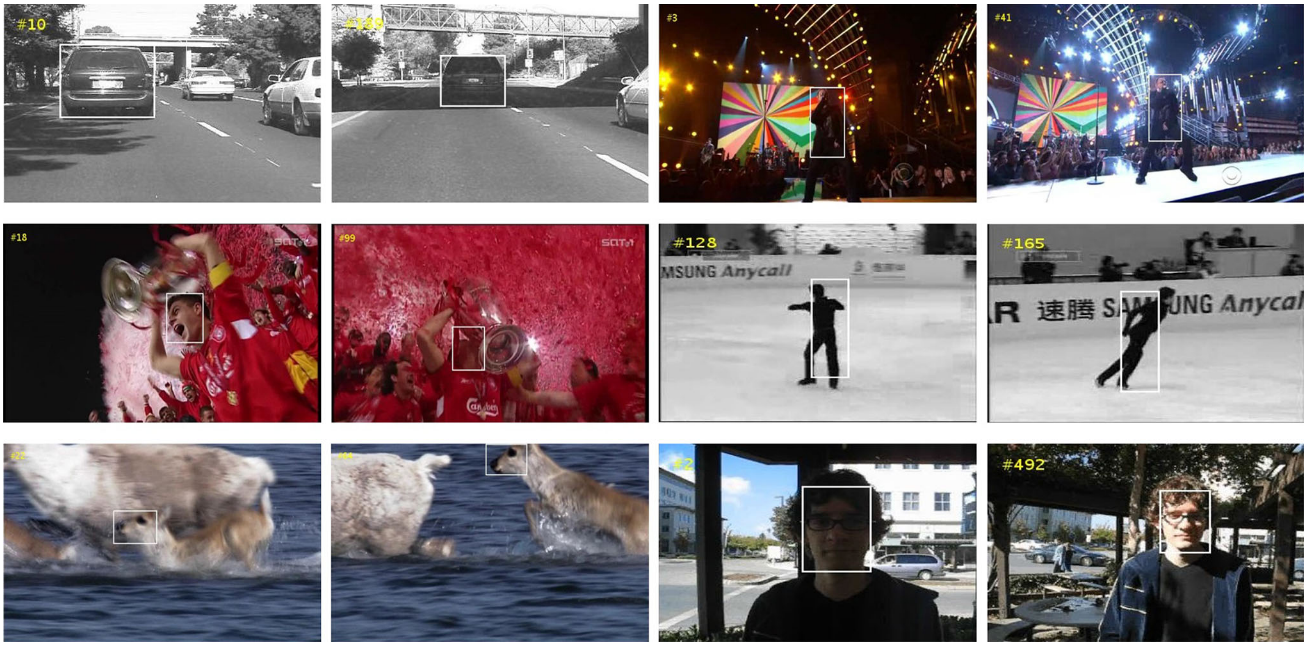


Fig. 8 Tracking results for challenging factors such as changing illumination and action

participated trackers, we use these reported results to compare with our TESR.

4.3.1 OTB

We run our TESR_{KCF} and $\text{TESR}_{\text{STAPLE}}$ on benchmark OTB2013 [44] and OTB2015 [45]. We also run the other state-of-the-art trackers including KCF [17], DSST [9], STAPLE [2], MEEM [46], SRDCF [10] and DLSSVM [32]. Two performance measures are used. The *precision evaluation* of a tracker on a sequence is expressed as the average per-frame location error between its predicted bounding box and the ground truth. The *success rate evaluation* is expressed as the average per-frame overlap between its predicted bounding box and the ground truth using the intersection-over-union (IOU) criterion $S_t = \frac{r_t \cap r_{GT}}{r_t \cup r_{GT}}$, where r_t is the predicted bounding box and r_{GT} is the ground truth.

Figure 6 shows the precision plot at the threshold set to 20. Figure 7 shows the success plot of the proportion of successful frames at the IOU thresholds varying from 0 to 1. We use the area-under-curve (AUC) to measure the representative success score. We can see that on both OTB2013

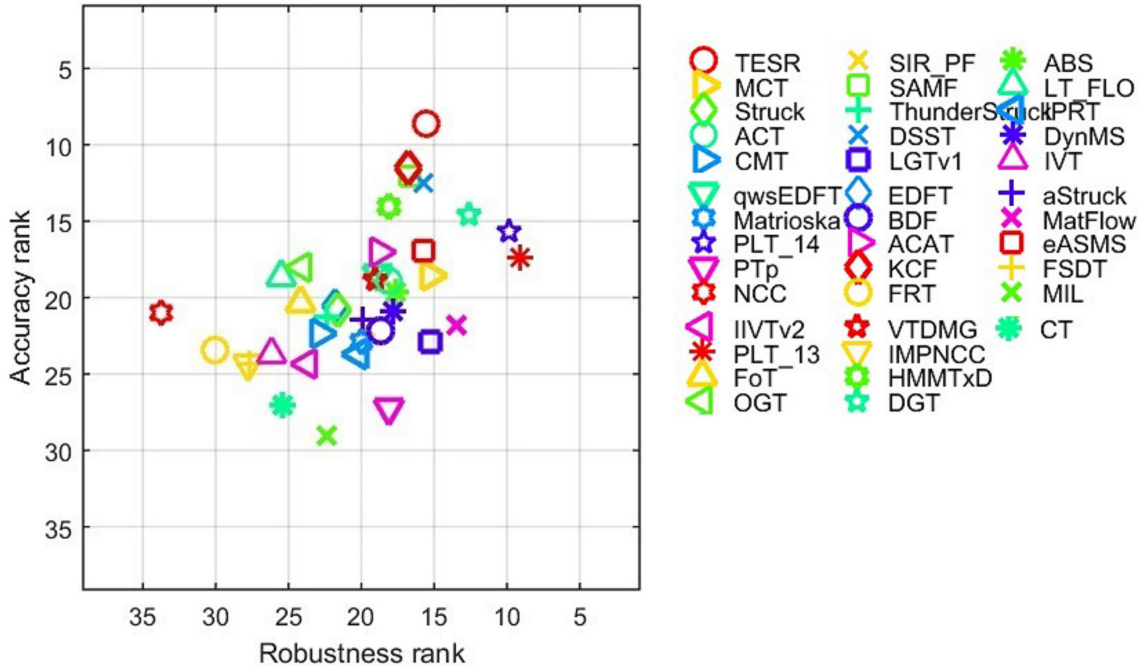


Fig. 9 *Accuracy–Robustness* (AR) rank plot on VOT2014 benchmark. Better trackers are closer to the top right corner

Table 2 The raw *accuracy* and *robustness* scores on VOT2014 benchmark (over 25 sequences) of some top performers

	KCF	SAMF	DSST	LGTv1	PLT14	PLT13	$\text{TESR}_{\text{STAPLE}}$
Accuracy	*0.62*	0.61	*0.62*	0.46	0.56	0.54	***0.66***
Robustness	1.32	1.28	1.16	0.66	*0.16*	***0.08***	1.00

Higher accuracy score and lower robustness score are better

and OTB2015, our TESR_{KCF} and $\text{TESR}_{\text{STAPLE}}$ achieve very good results in comparison with base trackers KCF and STAPLE. TESR_{KCF} is better by 11% in precision and 14% in success rate on OTB2013 and 7% in precision and 7% in success rate on OTB2015. $\text{TESR}_{\text{STAPLE}}$ shows improvement of 14% in precision and 12% in success rate on OTB2013 and 10% in precision and 11% in success rate on OTB2015. $\text{TESR}_{\text{STAPLE}}$ is the best in terms of both evaluation measures. The precision measure achieved by $\text{TESR}_{\text{STAPLE}}$ is 88.9% and 86.1% on OTB2013 and OTB2015, respectively. The success rate measure achieved by $\text{TESR}_{\text{STAPLE}}$ is 67.3% and 64.4% on OTB2013 and OTB2015, respectively.

Table 1 compares the precision and success rate scores of TESR and the state-of-the-art trackers on sequences classified according to various challenging factors. All the videos in OTB2013 are annotated with 11 different attributes, namely: illumination variation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, out-of-view, background clutter and low resolution. Some tracking results according to various factors such as light changing and action changing dramatically are also shown in Fig. 8 We can see our $\text{TESR}_{\text{STAPLE}}$ outperforms existing trackers on all attributes in precision and on ten attributes in success rate. In particular, $\text{TESR}_{\text{STAPLE}}$ achieves relatively larger performance gains, as compared with the second best tracker, when target object experiences illumination variation, deformation, occlusion, out-of-view and background clutter. According to the results for the subset of background clutter, TESR can identify the target due to the stable features of object stored in the support vectors of SVM classifier. The superior performance on the subset of illumination variation and deformation is the result of adaptation to fast appearance change by our CF store. Since we adopt a large search region, TESR can re-capture the object easily after it starts drifting due to occlusion.

4.3.2 VOT

We also run our $\text{TESR}_{\text{STAPLE}}$ on VOT2014 [23] and VOT2016 [22]. On VOT benchmark, there are two highly interpretable weakly correlated performance measures to analyze tracking behavior in reset-based experiments: *accuracy* (A) and *robustness* (R). Unlike OTB, VOT-related methodology resets the tracker after it drifts off the target. The *accuracy* (A) is the average overlap between the predicted bounding boxes and ground truth during successful

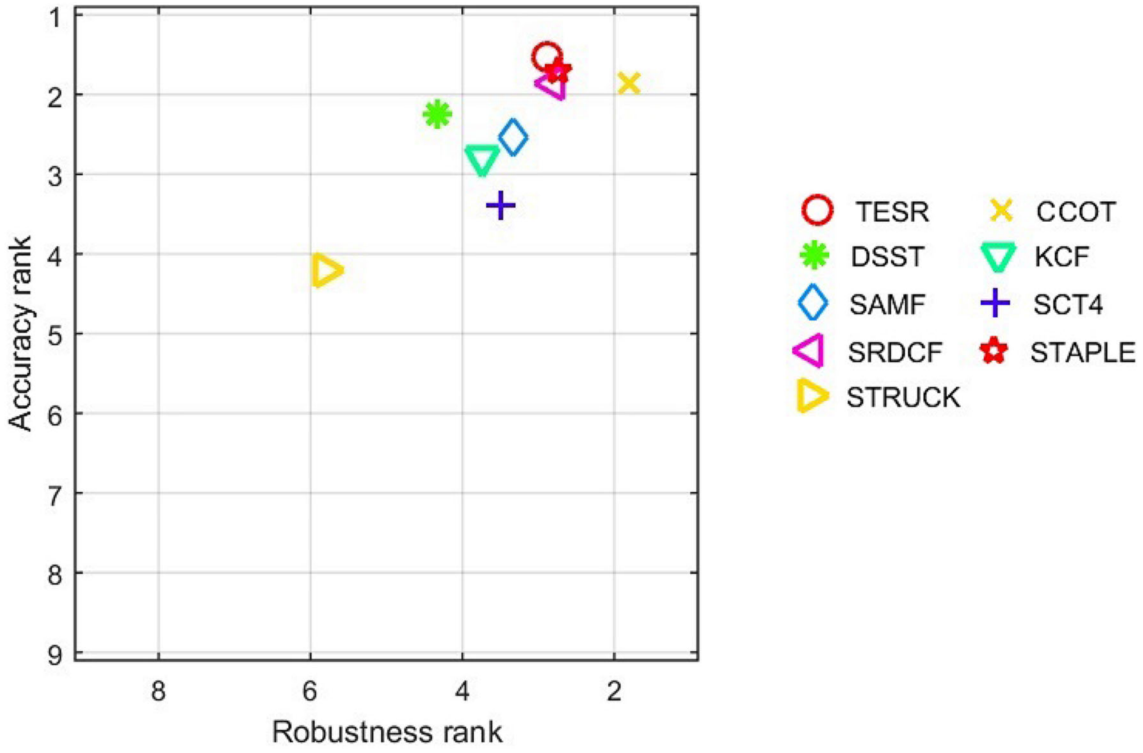


Fig. 10 *Accuracy–Robustness* (AR) rank plot on VOT2016 benchmark. Better trackers are closer to the top right corner. Note CCOT is the champion of VOT2016 challenge which is shown as yellow cross

Table 3 The raw *accuracy* and *robustness* scores on VOT2016 benchmark (over 60 sequences) of some top performers

	STRUCK	STAPLE	SRDCF	SCT4	SAMF	KCF	DSST	CCOT	$\text{TESR}_{\text{STAPLE}}$
Accuracy	0.42	***0.54***	0.52	0.45	0.48	0.48	0.48	0.52	***0.54***
Robustness	3.37	*1.35*	1.50	1.95	2.10	2.03	2.52	***0.85***	1.50

Higher accuracy score and lower robustness score are better

tracking periods. The *robustness* (R) measures the number of the failures of the tracker.

To generate Fig. 9 and Table 2, we use the most recent version of the VOT toolkit, which can be downloaded from the VOT challenges Web site. As shown in Fig. 9, for all 38 participated trackers of VOT2014 challenge, our $\text{TESR}_{\text{STAPLE}}$ gets the best accuracy ranking. Table 2 presents the raw accuracy score for some trackers of VOT2014 challenge. We can see our $\text{TESR}_{\text{STAPLE}}$ achieves a 6% improvement in accuracy in comparison with the second best tracker: KCF [17] and DSST [9].

The experimental results of 70 trackers that participated in the VOT16 challenge are publicly available. For simplicity, we compare $\text{TESR}_{\text{STAPLE}}$ to some state-of-the-art trackers. We only include the trackers: CCOT [11], DSST [9], KCF [17], SAMF [25], SCT4 [8], SRDCF [10], STAPLE [2], STRUCK [14]. Note CCOT is the champion tracker of VOT2016 challenge. As shown in Fig. 10 and Table 3, we can see $\text{TESR}_{\text{STAPLE}}$ also significantly outperforms other trackers in accuracy measure. We achieve the better ranking than CCOT in accuracy ranking which is shown in Fig. 10.

It is worth recalling that the robustness performance defined for the benchmark is the number of the failures on tracking periods. The VOT toolkit resets the tracker once the IOU between the predicted bounding box and the ground truth reduces to zero. This measure method is not favorable to our $\text{TESR}_{\text{STAPLE}}$. As TESR implicitly extends the search region, it can capture the target again after it completely misses the object. As shown in Fig. 11, in sequence *couple*, *jogging1* and *birds1*, $\text{TESR}_{\text{STAPLE}}$ captures the target again in 107th, 81th and 32th frames, respectively, after its IOU completely drops to zero. Then it continues tracking well in the following frames. Note that this is the advantage of our method. We do not need redetection technology. Nevertheless, our TESR is still a short-term tracker. Unfortunately, the robustness measure of VOT benchmark views these frames as failures, and this increases the value of robustness score.

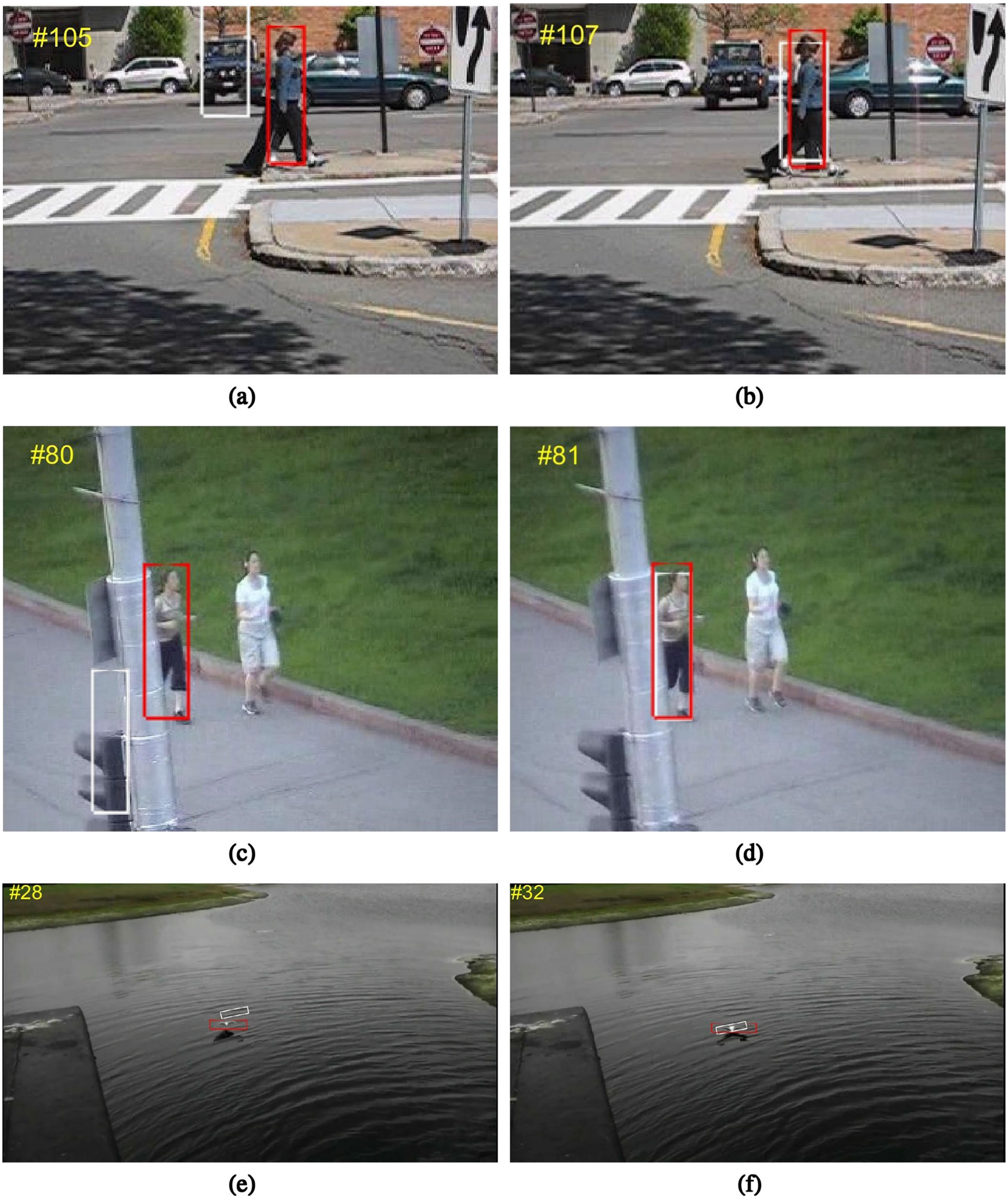


Fig. 11 **a**, **b**, sequence *couple*, **c**, **d**, sequence *jogging1*, **e**, **f**, sequence *birds1*. The red box represents the ground truth, and the white box represents the predicted location of TESR

Table 4 Comparing the impact of different components of TESR on OTB2013/2015 with the precision/success rate scores/fps

	TESR$_{\text{STAPLE}}$	TESR$_{\text{NonSD}}$	TESR$_{\text{NonSVM}}$	TESR$_{\text{1F}}$
OTB2013	0.889/0.671/14.2	0.875/0.659/19.3	0.814/0.627/16.9	0.833/0.636/9.2
OTB2015	0.861/0.644/14.0	0.851/0.638/18.9	0.815/0.609/16.6	0.785/0.593/8.7

As a result TESR$_{\text{STAPLE}}$ has not achieved a very high ranking in robustness measure on VOT2014 and VOT2016.

4.4 Impact of different components of TESR

To verify the effectiveness of different innovations of TESR, we conducted the ablation study by suppressing different parts of TESR$_{\text{STAPLE}}$ technology, by performing experiments to measure the impact on OTB datasets, as shown in Table 4. TESR is a fast tracker. In particular, if we remove the significance feature detection module, the TESR will reach a speed of 19fps with only a slight performance degradation.

For the impact of the saliency detection, we removed the salient feature detection module, and denoted the resulting tracker as TESR$_{\text{NonSD}}$. The experimental results obtained on OTB2013 and OTB2015 are presented as follows: The precision is 87.5% and 85.1%, and the success rate is 65.9% and 63.8%, respectively. This shows that the salient feature detection improves the performance.

For the impact of the SVM detection, we deleted it and maximized over the five peak values to judge which candidate is most likely the object. The tracker is denoted as TESR$_{\text{NonSVM}}$. The results obtained on OTB2013 and OTB2015 are given as follows: The precision is 81.4% and 81.5%, and the success score is 62.7% and 60.9%. This shows clearly that our two-stage strategy improves the performance. The reason of the decrease in the performance is that peak value of the response of CF does not always correspond to the target position.

Our TESR is not the simple combination of two classifiers, i.e CF and SVM. Firstly, our CF detection stage implicitly extends the search region in comparison with the conventional CF trackers. Secondly, different from the traditional multiclassifier structures, the two classifiers use independent update periods. The correlation filter is viewed as a long-term memory, adapting to the rapid appearance change, while the SVM classifier is viewed as a long-term memory, adapting to stable property of the object. These complementary concepts are of paramount importance to our tracking framework. Note, if SVM classifier updates every frame, denoted as TESR$_{\text{1F}}$, the precision on OTB2013 and OTB2015 would decrease to 83.3% and 78.5%, the success would also decrease to 63.6% and 59.3% respectively.

5 Conclusion

In this paper, we propose a correlation tracking framework employing implicitly extending search region (TESR) to deal with the problem of occlusion, motion blur and camera shaking in visual tracking. Our method is a two-stage detection solution. In the first stage, we decrease the boundary effect with a very unique way. In the second stage, we use an SVM classifier to choose the best candidate generated in the first stage. The results of experiments demonstrate that our method exhibits superior performance in comparison with the current state-of-the-art trackers on benchmark OTB and VOT.

Funding This study was funded by the 111 Project of Chinese Ministry of Education under Grant B12018 and the National Natural Science Foundation of China under Grant 61373055; 61672265.

Compliance with ethical standards

Conflict of interest All authors declare that they have no conflict of interest.

References

1. Atkinson, R.C., Shiffrin, R.M.: Human memory: a proposed system and its control processes. Psychol. Learn. Motiv. **2**, 89–195 (1968)
2. Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., Torr, P.H.: Staple: complementary learners for real-time tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1401–1409 (2016)
3. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional Siamese networks for object tracking. In: European Conference on Computer Vision, pp. 850–865. Springer (2016)
4. Bibi, A., Mueller, M., Ghanem, B.: Target response adaptation for correlation filter tracking. In: European Conference on Computer Vision, pp. 419–433. Springer (2016)
5. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2544–2550. IEEE (2010)
6. Bolme, D.S., Draper, B.A., Beveridge, J.R.: Average of synthetic exact filters. In: IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009, pp. 2105–2112. IEEE (2009)
7. Chen, Z., Hong, Z., Tao, D.: An experimental survey on correlation filter-based tracking. arXiv:1509.05520 (2015)
8. Choi, J., Jin Chang, H., Jeong, J., Demiris, Y., Young Choi, J.: Visual tracking using attention-modulated disintegration and integration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4321–4330 (2016)

9. Danelljan, M., Häger, G., Khan, F., Felsberg, M.: Accurate scale estimation for robust visual tracking. In: British Machine Vision Conference, Nottingham, September 1–5, 2014. BMVA Press (2014)
10. Danelljan, M., Hager, G., Shahbaz Khan, F., Felsberg, M.: Learning spatially regularized correlation filters for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4310–4318 (2015)
11. Danelljan, M., Robinson, A., Khan, F.S., Felsberg, M.: Beyond correlation filters: learning continuous convolution operators for visual tracking. In: European Conference on Computer Vision, pp. 472–488. Springer (2016)
12. Danelljan, M., Shahbaz Khan, F., Felsberg, M., Van de Weijer, J.: Adaptive color attributes for real-time visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1090–1097 (2014)
13. Guan, H., Cheng, B.: How do deep convolutional features affect tracking performance: an experimental study. Vis. Comput. **34**(12), 1701–1711 (2018)
14. Hare, S., Golodetz, S., Saffari, A., Vineet, V., Cheng, M.M., Hicks, S.L., Torr, P.H.: Struck: structured output tracking with kernels. IEEE Trans. Pattern Anal. Mach. Intell. **38**(10), 2096–2109 (2016)
15. He, A., Luo, C., Tian, X., Zeng, W.: A twofold siamese network for real-time object tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4834–4843 (2018)
16. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. In: European Conference on Computer Vision, pp. 702–715. Springer (2012)
17. Henriques, J.F., Rui, C., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. **37**(3), 583–596 (2015)
18. Hong, Z., Chen, Z., Wang, C., Mei, X., Prokhorov, D., Tao, D.: Multi-store tracker (muster): a cognitive psychology inspired approach to object tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 749–758 (2015)
19. Kiani Galoogahi, H., Fagg, A., Lucey, S.: Learning background-aware correlation filters for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1135–1143 (2017)
20. Kiani Galoogahi, H., Sim, T., Lucey, S.: Correlation filters with limited boundaries. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4630–4638 (2015)
21. Kim, H.U., Lee, D.Y., Sim, J.Y., Kim, C.S.: Sowp: spatially ordered and weighted patch descriptor for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3011–3019 (2015)
22. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Čehovin, L., Vojȓ, T., Häger, G., Lukežič, A., Fernndez, G.: The visual object tracking vot2016 challenge results. In: European Conference on Computer Vision (2016)
23. Kristan, M., Pflugfelder, R., Leonardis, A., Matas, J., Porikli, F., Čehovin, L., Nebehay, G., Fernandez, G., Vojȓ, T., Gatt, A.: The visual object tracking vot2014 challenge results. In: European Conference on Computer Vision (2014)
24. Li, B., Yan, J., Wu, W., Zhu, Z., Hu, X.: High performance visual tracking with siamese region proposal network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8971–8980 (2018)
25. Li, Y., Zhu, J.: A scale adaptive kernel correlation filter tracker with feature integration. ECCV Workshops **2**, 254–265 (2014)
26. Li, Y., Zhu, J., Hoi, S.C.: Reliable patch trackers: robust visual tracking by exploiting reliable patches. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 353–361 (2015)
27. Liu, T., Wang, G., Yang, Q.: Real-time part-based visual tracking via adaptive correlation filters. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4902–4912 (2015)
28. Liu, T., Yuan, Z., Sun, J., Wang, J., Zheng, N., Tang, X., Shum, H.Y.: Learning to detect a salient object. IEEE Trans. Pattern Anal. Mach. Intell. **33**(2), 353–367 (2011)
29. Ma, C., Huang, J.B., Yang, X., Yang, M.H.: Hierarchical convolutional features for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3074–3082 (2015)
30. Mbelwa, J.T., Zhao, Q., Lu, Y., Liu, H., Wang, F., Mbise, M.: Objectness-based smoothing stochastic sampling and coherence approximate nearest neighbor for visual tracking. Vis. Comput. **35**(3), 371–384 (2019)
31. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4293–4302 (2016)
32. Ning, J., Yang, J., Jiang, S., Zhang, L., Yang, M.H.: Object tracking via dual linear structured SVM and explicit feature map. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4266–4274 (2016)
33. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015)
34. Smeulders, A.W., Chu, D.M., Cucchiara, R., Calderara, S., Dehghan, A., Shah, M.: Visual tracking: an experimental survey. IEEE Trans. Pattern Anal. Mach. Intell. **36**(7), 1442–1468 (2014)
35. Song, Y., Ma, C., Gong, L., Zhang, J., Lau, R.W., Yang, M.H.: Crest: convolutional residual learning for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2555–2564 (2017)
36. Sui, Y., Zhang, Z., Wang, G., Tang, Y., Zhang, L.: Real-time visual tracking: promoting the robustness of correlation filter learning. In: European Conference on Computer Vision, pp. 662–678. Springer (2016)
37. Wang, N., Shi, J., Yeung, D.Y., Jia, J.: Understanding and diagnosing visual tracking systems. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3101–3109 (2015)
38. Wang, Q., Teng, Z., Xing, J., Gao, J., Hu, W., Maybank, S.: Learning attentions: residual attentional siamese network for high performance online visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4854–4863 (2018)
39. Wang, Q., Zhang, L., Bertinetto, L., Hu, W., Torr, P.H.: Fast online object tracking and segmentation: A unifying approach. arXiv:1812.05050 (2018)
40. Wang, Q., Zhang, M., Xing, J., Gao, J., Hu, W., Maybank, S.: Do not lose the details: reinforced representation learning for high performance visual tracking. In: 27th International Joint Conference on Artificial Intelligence (2018)
41. Wang, X., Jabri, A., Efros, A.A.: Learning correspondence from the cycle-consistency of time. arXiv:1903.07593 (2019)
42. Wang, Y., Hu, S., Wu, S.: Object tracking based on Huber loss function. Vis. Comput. **35**(11), 1641–1654 (2019)
43. Wang, Z., Vucetic, S.: Online training on a budget of support vector machines using twin prototypes. Stat. Anal. Data Min. ASA Data Sci. J. **3**(3), 149–169 (2010)
44. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: a benchmark. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2411–2418 (2013)

45. Wu, Y., Lim, J., Yang, M.H.: Object tracking benchmark. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1834–1848 (2015)
46. Zhang, J., Ma, S., Sclaroff, S.: Meem: robust tracking via multiple experts using entropy minimization. In: European Conference on Computer Vision, pp. 188–203. Springer (2014)
47. Zhang, K., Zhang, L., Liu, Q., Zhang, D., Yang, M.H.: Fast visual tracking via dense spatio-temporal context learning. In: European Conference on Computer Vision, pp. 127–141. Springer (2014)
48. Zhang, T., Bibi, A., Ghanem, B.: In defense of sparse tracking: circulant sparse tracker. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3880–3888 (2016)
49. Zhu, Z., Wang, Q., Li, B., Wu, W., Yan, J., Hu, W.: Distractor-aware siamese networks for visual object tracking. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 101–117 (2018)
50. Zuo, W., Wu, X., Lin, L., Zhang, L., Yang, M.H.: Learning support correlation filters for visual tracking. arXiv:1601.06032 (2016)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Qiang Qian received the B.S. degree in communication engineering from Anhui University, Hefei, China, in 2001 and M.S. degree in pattern recognition and intelligent system from Jiangsu University of Science and Technology, Zhenjiang, China, in 2013. He joined School of Computer Science and Engineering, Jiangsu University of Science and Technology in 2001 and is currently pursuing the Ph.D. degree in control science and engineering in Jiangnan University, Wuxi, China. His research interest includes visual object tracking, deep learning and computer vision.

Xiao-Jun Wu received his B.S. degree in mathematics from Nanjing Normal University, Nanjing, China, in 1991 and M.S. degree in 1996 and Ph.D. degree in pattern recognition and intelligent system in 2002, both from Nanjing University of Science and Technology, Nanjing, China, respectively. From 1996 to 2006, he taught in the School of Electronics and Information, Jiangsu University of Science and Technology, where he was an exceptionally promoted professor. He joined School of Information Engineering (now renamed as School of IoT Engineering), Jiangnan University, in 2006, where he is a distinguished professor of level 2. He was a fellow of United Nations University, International Institute for Software Technology (UNU/IIST) from 1999 to 2000. He was a visiting postdoctoral researcher in the Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey, UK, from 2003 to 2004, under the supervision of Professor Josef Kittler. His current research interests are pattern recognition, computer vision, fuzzy systems, neural networks and intelligent systems. He is an associate editor of International Journal of Computer Mathematics, review editor of Frontiers in Neurorobotics and editor of Journal of Algorithm and Computational Technology, respectively.

Josef Kittler has been a Research Assistant in the Engineering Department of Cambridge University (1973–1975), SERC Research Fellow at the University of Southampton (1975–1977), Royal Society European Research Fellow, Ecole Nationale Superieure des Telecommunications, Paris (1977–1978), IBM Research Fellow, Balliol College, Oxford (1978–1980), Principal Research Associate, SERC Rutherford Appleton Laboratory (1980–1984) and Principal Scientific Officer, SERC Rutherford Appleton Laboratory (1985). He also worked as the SERC Coordinator for Pattern Analysis (1982) and was Rutherford Research Fellow in Oxford University, Dept. Engineering Science (1985). He joined the Department of Electrical Engineering of Surrey University in 1986 as a reader in Information Technology and became professor of Machine Intelligence in 1991 and gained the title Distinguished Professor in 2004. In computer vision, his major contributions include robust statistical methods for shape analysis and detection, motion estimation and segmentation and image segmentation by thresholding and edge detection.

Tian-Yang Xu received the B.S. degree in electronic science and engineering from Nanjing University, Nanjing, China, in 2011. He is currently pursuing the Ph.D. degree in Control Science and Engineering in Jiangnan University, Wuxi, China. His research interest includes visual object tracking, deep learning and manifold learning.