Wang and Zhang *EURASIP Journal on Image and Video Processing* (2019) 2019:73
https://doi.org/10.1186/s13640-019-0474-z

EURASIP Journal on Image
and Video Processing

RESEARCH Open Access

Mutual kernelized correlation filters with elastic net constraint for visual tracking

Haijun Wang[*] and Shengyan Zhang

Abstract

In this paper, we propose a robust visual tracking method based on mutual kernelized correlation filters with elastic net constraint. First, two correlation filters are trained in a general framework jointly in a closed form, which are interrelated and interacted on each other. Second, elastic net constraint is imposed on each discriminative filter, which is able to filter some interfering features. Third, scale estimation and target re-detection scheme are adopted in our framework, which can deal with scale variation and tracking failure effectively. Extensive experiments on some challenging tracking benchmarks demonstrate that our proposed method is able to obtain a competitive tracking performance against other state-of-the-art algorithms.

Keywords: Visual tracking, Mutual kernelized correlation filters, Elastic net constraint, Convolutional neural networks

1 Introduction

Visual tracking is a fundamental task in computer vision with numerous applications, such as unmanned control systems, surveillance, assistant driving, and so on. Given the position of the tracked object in the first frame, the goal of visual tracking is to estimate the position of the tracked target in the subsequent frame precisely. Although great progress has been made in recent years [1, 2], designing a robust tracking algorithm is still a challenging problem due to negative factors such as background clutters, severe occlusion, motion blur, and illumination variation (see Fig. 1).

Generally speaking, visual tracking methods can be divided into two categories: generative methods [3–7] and discriminative methods [8–13]. Generative methods attempt to build a model to represent tracked target and find the region with the minimum reconstruction error from a great deal of candidates. For example, under the particle filter framework, Mei et al. [14] developed a tracker method based on sparse representation, called the ℓ_1 method, which reconstructs each candidate with dictionary template and trivial template. The sparse representation coefficients of each candidate can be computed by solving ℓ_1 minimization. Despite ℓ_1 method demonstrated impressive tracking performance, the tracking speed is very slow because of its huge computation load. In order to solve this problem, Bao et al. [15] proposed a fast ℓ_1 tracking method by using accelerated proximal gradient approach. Xiao et al. [16] presented a fast object tracking method by solving ℓ_2 regularized least square problem. Wang et al. [17] developed a novel and fast visual tracking method via probability continuous outlier model. Different from the general method, discriminative algorithms regard visual tracking as a binary classification problem which distinguishes the correct tracked object from the background. For example, Babenko et al. [18] trained an online discriminative classifier to separate the tracked object from the background by online multiple instance learning. Zhang et al. [19] formulated visual tracking as a binary classification via a naive

* Correspondence: whjlym@163.com
Aviation Information Technology Research and Development, Binzhou University, Binzhou 256603, China

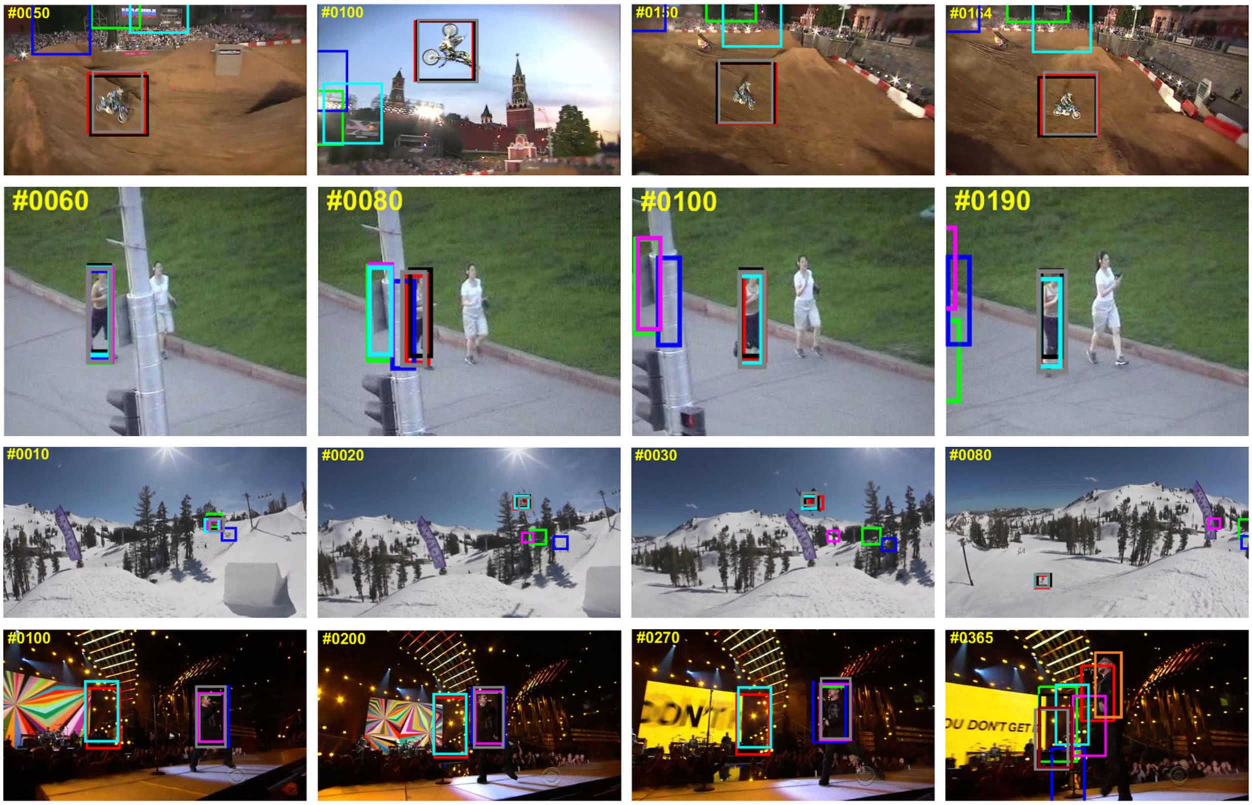


Fig. 1 Tracking results in challenging environments including background clutters (motorRolling), severe occlusion (Jogging-1), fast motion (skiing), illumination change (Singer2). The tracking results of HDT, Staple, KCF, CNN-SVM, DSST, MEEM, and our tracker are shown by red, green, blue, black, magenta, cyan, and gray rectangles, respectively

Bayes classifier with an online update scheme in the compressed domain.

In recent years, visual tracking methods based on correlation filter [20–25] have attracted great attention due to its real-time tracking speed and robust tracking performance. Under the framework of correlation filter, a discriminative classifier is trained with a great deal of dense sampling examples. These dense sampling examples are with circulant structure which allows the use of the fast Fourier transform (FFT). Bolme et al. [26] first developed a minimum output sum of squared error filter for real-time visual tracking. After that, a great deal of tracking methods based on correlation filter has been proposed to improve tracking performance. Henriques et al. [27] developed a high-speed tracker with kernelized correlation filters which can deal with multi-channel features. Danelljan et al. [28] presented a discriminative scale space tracker with a correlation filter based on a scale pyramid representation. In order to mitigate the unwanted boundary effect which appeared in traditional correlation-based trackers, Danelljan et al. [29] figured out spatially regularized discriminative correlation filters (SRDCF) for visual tracking. Recent researches have shown that features from convolutional neural networks (CNN) can improve tracking performance greatly [30–33]. Zhang et al. [34] builded a simple two-layer convolutional network to learn robust representation for visual tracking without offline training. Ma et al. [35] utilized three convolutional layers to learn robust target appearance for visual tracking. Wang et al. [36] exploited robust target appearance representation from the top layer to lower layer for object tracking. Heng et al. [37] incorporated recurrent neural network (RNN) into CNN to improve tracking performance. He et al. [38] integrated weighted convolution responses from 10 layers and achieved a very promising performance.

Although correlation filters based trackers have obtained superior tracking performance, many trackers utilized a single correlation filter and could not achieve promising tracking results. Figure 2 gives the precision plots and success plots of OPE by methods with a different number of correlation filters on OTB-2013. It is obvious that just simply merging two correlation filters is able to greatly improve tracking

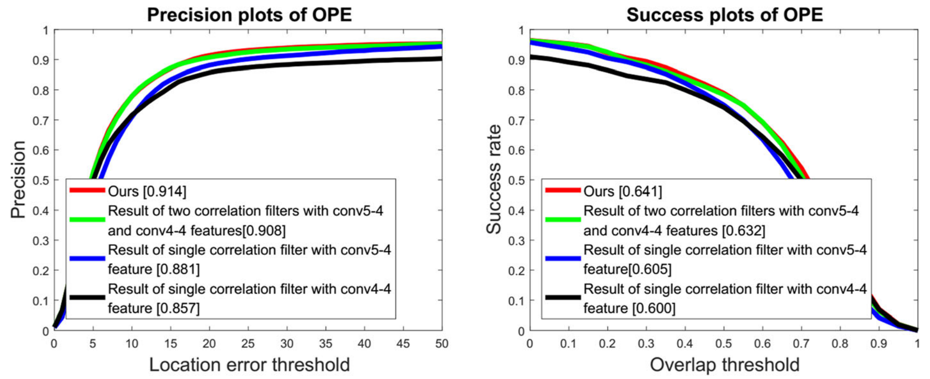


Fig. 2 Precision plots and success plots of OPE by methods with different correlation filters on OTB-2013

performance in both precision and success rate. However, there is still much room for improvement for methods using two correlation filters which are independent of each other.

Inspired by the above discussions, we develop a robust visual tracking method via mutual kernelized correlation filters using features from convolutional neural networks (MKCN_CNN), where each tracker works on its own and tries to correct the other one. At the same time, an elastic net constraint is imposed on each filter, which can eliminate some distractive features. Finally, the proposed tracking framework can be solved in a closed-form fashion. Extensive experiments demonstrate that our method can achieve promising tracking performance competing with some other state-of-the-art trackers.

The rest of this paper is organized as follows. Section 2 briefly summarizes the principle of visual tracking based on kernelized correlation filter. Section 3 introduces the proposed tracking algorithm in details. The experimental results and corresponding discussions are described in Section 4, followed by the conclusion in Section 5.

2 Visual tracking based on kernelized correlation filters

Henriques et al. [27] proposed a fast discriminative visual tracking method based on kernelized correlation filters (KCF). Given a $n \times 1$ vector $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^T$ denoting a base image, a shifted version of $\mathbf{x}$ can be defined by $\{P^u\mathbf{x} | u = 1\} = [\mathbf{x}_n, \mathbf{x}_1, \ldots, \mathbf{x}_{n-1}]^T$. Here, P is a permutation matrix. So, the full shifted signals of $\mathbf{x}$ are given by

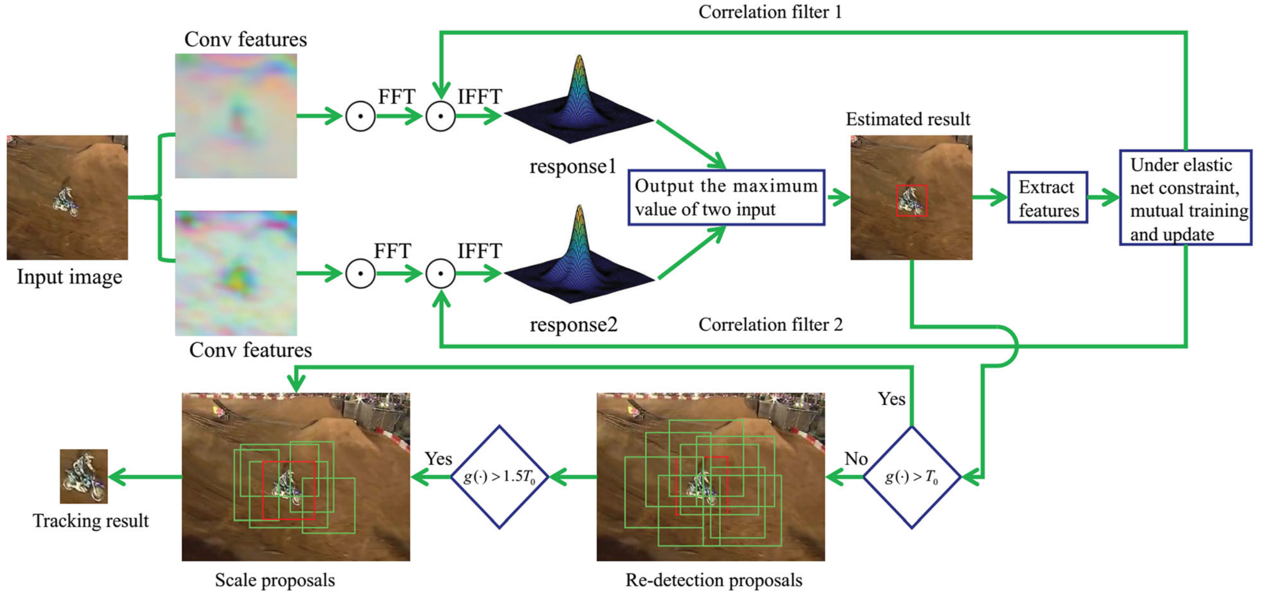


Fig. 3 The flowchart of our proposed tracking framework

Table 1 Matlab pseudo-codes of MKCF_CNN

```
The details of MKCF_CNN method
Input:
$\mathbf{x}_1$ :training features from $conv5-4$ convolution layer.
$\mathbf{x}_2$ :training features from $conv4-4$ convolution layer.
$\mathbf{x}_1'$ :testing features from $conv5-4$ convolution layer.
$\mathbf{x}_2'$ :testing features from $conv4-4$ convolution layer.
$\mathbf{y}$ :regression label.
$\lambda, \mu, \tau, \rho$ :constant parameters in equation (15) and (17).
Output:
The final response and the position of target

Function $[\alpha_1, \beta_1, \alpha_2, \beta_2]$=train$(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}, \lambda, \mu, \tau, \rho)$
    $\mathbf{k}_1 = \text{kernel\_correlation}(\mathbf{x}_1, \mathbf{x}_1);$
    $\mathbf{k}_2 = \text{kernel\_correlation}(\mathbf{x}_2, \mathbf{x}_2);$
    $\hat{\mathbf{y}} = \text{fft2}(\mathbf{y});$
    $\hat{\alpha}_1 = \frac{\hat{\mathbf{k}}_1 \circ \hat{\mathbf{y}} + 2\rho\hat{\mathbf{k}}_1 \circ \hat{\mathbf{k}}_2 \circ \alpha_2 + \mu\beta_1}{(1+2\rho)\hat{\mathbf{k}}_1 \circ \hat{\mathbf{k}}_1 + \lambda\hat{\mathbf{k}}_1 + \mu};$
    $\beta_1 = sign(\alpha_1)\max\left(0, |\alpha_1| - \frac{\tau}{2\mu}\right);$
    $\hat{\alpha}_2 = \frac{\hat{\mathbf{k}}_2 \circ \hat{\mathbf{y}} + 2\rho\hat{\mathbf{k}}_2 \circ \hat{\mathbf{k}}_1 \circ \alpha_1 + \mu\beta_2}{(1+2\rho)\hat{\mathbf{k}}_2 \circ \hat{\mathbf{k}}_2 + \lambda\hat{\mathbf{k}}_2 + \mu};$
    $\beta_2 = sign(\alpha_2)\max\left(0, |\alpha_2| - \frac{\tau}{2\mu}\right);$
End

Function response = detect$(\alpha_1, \alpha_2, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_1', \mathbf{x}_2')$
    $\mathbf{k}_1'$ =kernel_correlation$(\mathbf{x}_1, \mathbf{x}_1');$
    $\mathbf{k}_2'$ =kernel_correlation$(\mathbf{x}_2, \mathbf{x}_2');$
    $response1 = \mathcal{F}^{-1}(\mathbf{k}_1' \circ \alpha_{1,t});$
    $response2 = \mathcal{F}^{-1}(\mathbf{k}_1' \circ \alpha_{2,t});$
    If $\max(response1(:)) > \max(response2(:))$
        $response = \max(response1(:))$ ;
    Else
        $response = \max(response2(:))$
    End
End

Function $\mathbf{k} = \text{kernel\_correlation}(\mathbf{x}_1, \mathbf{x}_2, sigma)$
    c = iff2(sum(conj(ifft2($\mathbf{x}_1$)).*fft2($\mathbf{x}_1$),3));
    $d = \mathbf{x}_1(:)' * \mathbf{x}_1(:) + \mathbf{x}_2(:)' * \mathbf{x}_2(:) - 2 * c;$
    $\mathbf{k} = \exp\left(\frac{\frac{-1}{sigma^2} * abs(d)}{numel(d)}\right);$
End
```

$\{P^u\mathbf{x}|u = 1, 2, \ldots, n-1\}$. Then, the data matrix $\mathbf{X}$ is defined by all the cyclic shifted version of $\mathbf{x}$ which can be made diagonal by discrete Fourier transform (DFT).

$$\mathbf{X} = F^H \operatorname{diag}(\hat{\mathbf{x}})F \tag{1}$$

Here, F means the DFT matrix, H stands for transpose and complex-conjugate, $\hat{\mathbf{x}} = \mathscr{F}(\mathbf{x})$, which computes the DFT of vector $\mathbf{x}$. The goal of KCF is to find a discriminative correlation classifier $f(\mathbf{x})$ over the data matrix $\mathbf{X}$ for separating the target object from the surrounding environment. Given the training dataset and their corresponding labels $(\mathbf{x}_1, y_1)$, …, $(\mathbf{x}_m, y_m)$, the discriminative correlation classifier $f(\mathbf{x})$ can be obtained by the following equation,

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

where λ means the regularization parameter. $\mathbf{x}_i$ stands for the ith row element of the data matrix $\mathbf{X}$. A Gaussian function is adopted to model the label y_i. When $\mathbf{x}_i$ is the centered target, y_i is set to 1. For the other cyclic shifted version of $\mathbf{x}_i$ around the center target, their labels smoothly decay to 0. The solution $\mathbf{w}$ can be easily obtained by $\mathbf{w} = (\mathbf{X}^H\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^H\mathbf{y}$. In order to get a powerful model, kernel trick is introduced into Eq. (2). The new model is rewritten as

$$\min_{\alpha} \|\mathbf{K}\alpha - \mathbf{y}\|_2^2 + \alpha\mathbf{K}\alpha \tag{3}$$

where $\mathbf{K}$ is a $n \times n$ kernel matrix and one of its elements is $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. Matrix $\mathbf{K}$ has a circulant structure and can be diagonalized as

$$\mathbf{K} = F^H \operatorname{diag}\left(\hat{\mathbf{k}}\right) F \tag{4}$$

Here, $\mathbf{k}$ is the first row of matrix $\mathbf{K}$. The solution α in the dual space can be given by

$$\alpha = (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{y} \tag{5}$$

where $\mathbf{I}$ is an identity matrix. Just as the data matrix $\mathbf{X}$, kernel matrix $\mathbf{K}$ is also circulant. So, the solution of Eq. (3) can be efficiently computed in the frequency domain.

$$\hat{\alpha} = \frac{\hat{\mathbf{y}}}{\hat{\mathbf{K}}^{\mathbf{xx}} + \lambda} \tag{6}$$

In the next frame, a great deal of candidates, denoted as $\mathbf{x}^{'}$, are extracted at the same position as the current frame. Actually, all these candidates' $\mathbf{x}^{'}$ are obtained from the cyclic shift of the base image $\mathbf{x}$. The response of these candidates can be computed from

$$f\left(\mathbf{x}^{'}\right) = \mathscr{F}^{-1}\left(\hat{\mathbf{k}}^{'} \circ \hat{\alpha}_t\right) \tag{7}$$

Here, $\mathscr{F}^{1}$ stands for the inverse discrete Fourier transform (IDFT). $\hat{\mathbf{k}}^{'}$ means the kernel correlation of candidates $\mathbf{x}^{'}$ and base image $\mathbf{x}$ in the frequency domain. $\circ$ denotes element by element multiplication. The candidate with the largest response is chosen as the final target object in the next frame.

3 Methods

Though the KCF method has obtained promising tracking performance, only one discriminative classifier is used in this model, which makes the KCF method not able to deal with complex sciences. In order to overcome these problems, inspired by ensemble tracking methods, we proposed mutual kernelized correlation filters with elastic net constraint for visual tracking. Extensive experiments show that our method can perform better than the state-of-the-art methods. The flowchart of our proposed tracking framework is demonstrated in Fig. 3.

3.1 Problem statement

In order to find the best target object from a great deal of candidates, we introduce a linear regressor model in the proposed method.

$$\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 \tag{8}$$

Here, $\mathbf{X}$ has the same definition as KCF. $\mathbf{y}$ means regression label value of $\mathbf{X}$. $\mathbf{w}$ represents the corresponding coefficient. In order to promote the performance of Eq. (8), just as least absolute shrinkage and selection operator (LASSO) model, ℓ_1 norm is adopted to regularize the coefficients $\mathbf{w}$.

$$\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \tau\|\mathbf{w}\|_1 \tag{9}$$

where τ is a constant weight parameter. In Eq. (9), some values of $\mathbf{w}$ are set to zero which can make some occluded pixels excluded in this new model. So, the occluded pixels have less effect on the final decision of regression values. However, we find that the occluded pixels often assemble in one position together. Eq. (9) cannot group these pixels with the same features. So, in order to overcome the limitations of the LASSO model, an elastic net regularization [39] is introduced in Eq. (9).

$$\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda\|\mathbf{w}\|_2 + \tau\|\mathbf{w}\|_1 \tag{10}$$

Here, λ is a constant weight parameter. $\|\mathbf{w}\|_2$ is used to group pixels with the similar property. In order to promote the tracking performance of our method, kernel trick is exploited in Eq. (10). The candidates

Table 2 Tracking pipeline of MKCF_CNN method

```
Proposed tracking method.
Input: The $t$-th frame, position of target at the $(t-1)$-th frame.
Output: Position of target at the $t$-th frame.
Repeat:
    1: obtain two training features of $conv5-4$ convolution layer and $conv4-4$ convolution layer at the $(t-1)$-th frame, respectively;
    2: get a great deal of features $\mathbf{x}_1^{'}$ and $\mathbf{x}_2^{'}$ of candidates from $conv5-4$ convolution layer and $conv4-4$ convolution layer, respectively;
    3: compute $response1$ and $response2$ by equation (30) and (31), respectively;
    4: estimate the new position of target by finding the maximum value $response$ between $response1$ and $response2$;
    5: if $response1 < T_0$, conduct target re-detection scheme using (32) and obtain new position $(x_t, y_t)$ of target; Estimate the size of target $(w_t, h_t)$ using (33).
Until: End of video sequences.
```

are mapped to a high-dimensional feature space $\phi(\mathbf{x})$. Then, in the dual space, the solution $\mathbf{w}$ is given by a linear combination of mapped candidates.

$$\mathbf{w} = \sum_i \alpha_i \phi(\mathbf{x}_i) \tag{11}$$

Equation (10) in the dual space can be described as

$$\min_{\alpha} \|\mathbf{y}-\mathbf{K}\alpha\|_2^2 + \lambda\alpha^T\mathbf{K}\alpha + \tau\|\alpha\|_1 \tag{12}$$

where $\mathbf{K}$ represents kernel matrix. The solution of α involves square norm and ℓ_1 norm simultaneously. In order to compute α efficiently, another variable β is introduced in Eq. (12).

$$\min_{\alpha} \|\mathbf{y}-\mathbf{K}\alpha\|_2^2 + \lambda\alpha^T\mathbf{K}\alpha + \tau\|\beta\|_1 + \mu\|\alpha-\beta\|_2^2 \tag{13}$$

Here, μ is a constant weight parameter.

3.2 Mutual kernelized correlation filters

In this part, we introduce mutual kernelized correlation filters based on Eq. (13). Then, the proposed mutual kernelized correlation filters will solve this following problem

$$\begin{aligned} T(\alpha_1,\alpha_2) = \min_{\alpha_1,\alpha_2} & \|\mathbf{y}-\mathbf{K}_1\alpha_1\|_2^2 + \|\mathbf{y}-\mathbf{K}_2\alpha_2\|_2^2 + \lambda\alpha_1^T\mathbf{K}_1\alpha_1 \\ & +\lambda\alpha_2^T\mathbf{K}_2\alpha_2 + \tau\|\beta_1\|_1 + \tau\|\beta_2\|_1 + \mu\|\alpha_1-\beta_1\|_2^2 \\ & +\mu\|\alpha_2-\beta_2\|_2^2 + 2\rho\|\mathbf{K}_1\alpha_1-\mathbf{K}_2\alpha_2\|_2^2 \end{aligned} \tag{14}$$

The first two parts of Eq. (14) force each kernelized correlation filter model to have the minimum squared error with respect to the desired output regression label $\mathbf{y}$. $\lambda\alpha_1^T\mathbf{K}_1\alpha_1 + \lambda\alpha_2^T\mathbf{K}_2\alpha_2$ denote the elastic net regularization on two models respectively. $\tau\|\beta\|_1 + \tau\|\beta\|_2 + \mu\|\alpha_1-\beta_1\|_2^2 + \mu\|\alpha_2-\beta_2\|_2^2$ are introduced to ex-

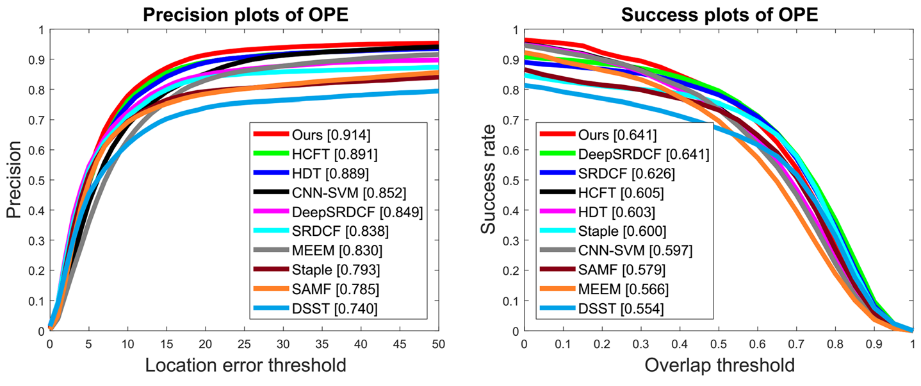


Fig. 4 Precision plots and success plots of OPE of our proposed method against other state-of-the-art methods on OTB-2013

clude the occluded pixels in the target object. $2\rho\|\mathbf{K}_1\alpha_1 - \mathbf{K}_2\alpha_2\|_2^2$ is used to weight the influence of the two kernelized correlation filter models.

It is obvious that Eq. (14) is convex with respect to α_1, α_2 if β_1, β_2 are fixed, and vice versa. So, we propose an iterative algorithm to compute the solution α_1, α_2. Thus, four subproblems with respect to α_1, α_2, β_1, β_2 are given as follows

$$\begin{aligned} T_1(\alpha_1) = \; & \min_{\alpha_1} \|\mathbf{y} - \mathbf{K}_1\alpha_1\|_2^2 + \lambda\alpha_1^T\mathbf{K}_1\alpha_1 + \tau\left\|\beta_1\right\|_1 \\ & + \mu\left\|\alpha_1 - \beta_1\right\|_2^2 + 2\rho\|\mathbf{K}_1\alpha_1 - \mathbf{K}_2\alpha_2\|_2^2 \end{aligned} \tag{15}$$

$$T_2(\beta_1) = \min_{\beta_1} \tau\|\beta\|_1 + \mu\left\|\alpha_1 - \beta_1\right\|_2^2 \tag{16}$$

$$\begin{aligned} T_3(\alpha_2) = \; & \min_{\alpha_2} \|\mathbf{y} - \mathbf{K}_2\alpha_2\|_2^2 + \lambda\alpha_2^T\mathbf{K}_2\alpha_2 \\ & + \tau\left\|\beta_2\right\|_1 + \mu\left\|\alpha_2 - \beta_2\right\|_2^2 \\ & + 2\rho\|\mathbf{K}_1\alpha_1 - \mathbf{K}_2\alpha_2\|_2^2 \end{aligned} \tag{17}$$

$$T_4(\beta_2) = \min_{\beta_2} \tau\left\|\beta_2\right\|_1 + \mu\left\|\alpha_2 - \beta_2\right\|_2^2 \tag{18}$$

Set the derivation of T_1 with respect to α_1 to be zero; Eq. (15) can be rewritten as follows:

$$\begin{aligned} \frac{\partial T_1}{\partial \alpha_1} & = -2\mathbf{K}_1(\mathbf{y} - \mathbf{K}_1\alpha_1) + 2\lambda\mathbf{K}_1\alpha_1 + 4\rho\mathbf{K}_1(\mathbf{K}_1\alpha_1 - \mathbf{K}_2\alpha_2) + 2\mu(\alpha_1 - \beta_1) \\ & = -2\mathbf{K}_1\mathbf{y} + 2\mathbf{K}_1\mathbf{K}_1\alpha_1 + 2\lambda\mathbf{K}_1\alpha_1 + 4\rho\mathbf{K}_1\mathbf{K}_1\alpha_1 - 4\rho\mathbf{K}_1\mathbf{K}_2\alpha_2 + 2\mu\alpha_1 - 2\mu\beta_1 \\ & = 0 \end{aligned} \tag{19}$$

Change the order of formula (19), we obtain

$$\begin{aligned} & \mathbf{K}_1\mathbf{K}_1\alpha_1 + \lambda\mathbf{K}_1\alpha_1 + 2\rho\mathbf{K}_1\mathbf{K}_1\alpha_1 + \mu\alpha_1 = \mathbf{K}_1\mathbf{y} + 2\rho\mathbf{K}_1\mathbf{K}_2\alpha_2 + \mu\beta_1 \\ & \Rightarrow (\mathbf{K}_1\mathbf{K}_1 + \lambda\mathbf{K}_1 + 2\rho\mathbf{K}_1\mathbf{K}_1 + \mu\mathbf{I})\alpha_1 = \mathbf{K}_1\mathbf{y} + 2\rho\mathbf{K}_1\mathbf{K}_2\alpha_2 + \mu\beta_1 \end{aligned} \tag{20}$$

Then, we obtain the solution α_1

$$\alpha_1 = (\mathbf{K}_1\mathbf{K}_1 + \lambda\mathbf{K}_1 + 2\rho\mathbf{K}_1\mathbf{K}_1 + \mu\mathbf{I})^{-1}(\mathbf{K}_1\mathbf{y} + 2\rho\mathbf{K}_1\mathbf{K}_2\alpha_2 + \mu\beta_1) \tag{21}$$

Set the derivation of T_3 with respect to α_2 to be zero; a similar solution α_2 is given as follows:

$$\alpha_2 = (\mathbf{K}_2\mathbf{K}_2 + \lambda\mathbf{K}_2 + 2\rho\mathbf{K}_2\mathbf{K}_2 + \mu\mathbf{I})^{-1}(\mathbf{K}_2\mathbf{y} + 2\rho\mathbf{K}_2\mathbf{K}_1\alpha_1 + \mu\beta_2) \tag{22}$$

It is straightforward that Eqs. (16) and (18) are least squared by ℓ_1 norm regularization. Thus, the solution β_1 and β_2 have closed form which can be easily achieved by a soft shrinkage function

$$\beta_1 = \operatorname{sign}(\alpha_1)\ \max\left(0, |\alpha_1| - \frac{\tau}{2\mu}\right) \tag{23}$$

$$\beta_2 = \ \operatorname{sign}(\alpha_2)\ \max\left(0, |\alpha_2| - \frac{\tau}{2\mu}\right) \tag{24}$$

By introducing Eqs. (4), (21) can be reformulated as follows:

$$\begin{aligned} \alpha_1 & = (\mathbf{K}_1\mathbf{K}_1 + \lambda\mathbf{K}_1 + 2\rho\mathbf{K}_1\mathbf{K}_1 + \mu\mathbf{I})^{-1}(\mathbf{K}_1\mathbf{y} + 2\rho\mathbf{K}_1\mathbf{K}_2\alpha_2 + \mu\beta_1) \\ & = \left((1+2\rho)F^H \operatorname{diag}\left(\hat{\mathbf{k}}_1 \circ \hat{\mathbf{k}}_1\right)F + \lambda F^H \operatorname{diag}\left(\hat{\mathbf{k}}_1\right)F + \mu\mathbf{I}\right)^{-1} \\ & \quad \times\left(F^H \operatorname{diag}\left(\hat{\mathbf{k}}_1\right)F\mathbf{y} + 2\rho F^H \operatorname{diag}\left(\hat{\mathbf{k}}_1 \circ \hat{\mathbf{k}}_2\right)F\alpha_2 + \mu\beta_1\right) \\ & = F^H \operatorname{diag}\left(\frac{1}{(1+2\rho)\hat{\mathbf{k}}_1 \circ \hat{\mathbf{k}}_2 + \lambda\hat{\mathbf{k}}_1 + \mu}\right) \operatorname{diag}\left(\hat{\mathbf{k}}_1\right)F\mathbf{y} \\ & \quad + 2\rho F^H \operatorname{diag}\left(\frac{1}{(1+2\rho)\hat{\mathbf{k}}_1 \circ \hat{\mathbf{k}}_2 + \lambda\hat{\mathbf{k}}_1 + \mu}\right) \operatorname{diag}\left(\hat{\mathbf{k}}_1 \circ \hat{\mathbf{k}}_2\right)F\alpha_2 \\ & \quad + \mu F^H \operatorname{diag}\left(\frac{1}{(1+2\rho)\hat{\mathbf{k}}_1 \circ \hat{\mathbf{k}}_2 + \lambda\hat{\mathbf{k}}_1 + \mu}\right)F\beta_1 \end{aligned} \tag{25}$$

Then, the DFT of α_1 is found by

$$\begin{aligned} \alpha_1 = \; & \operatorname{diag}\left(\frac{\hat{\mathbf{k}}_1}{(1+2\rho)\hat{\mathbf{k}}_1 \circ \hat{\mathbf{k}}_1 + \lambda\hat{\mathbf{k}}_1 + \mu}\right)\hat{\mathbf{y}} + 2\rho \\ & \operatorname{diag}\left(\frac{\hat{\mathbf{k}}_1 \circ \hat{\mathbf{k}}_2}{(1+2\rho)\hat{\mathbf{k}}_1 \circ \hat{\mathbf{k}}_1 + \lambda\hat{\mathbf{k}}_1 + \mu}\right)\alpha_2 + \mu \\ & \operatorname{diag}\left(\frac{1}{(1+2\rho)\hat{\mathbf{k}}_1 \circ \hat{\mathbf{k}}_1 + \lambda\hat{\mathbf{k}}_1 + \mu}\right)\beta_1 \\ = \; & \frac{\hat{\mathbf{k}}_1 \circ \hat{\mathbf{y}} + 2\rho\hat{\mathbf{k}}_1 \circ \hat{\mathbf{k}}_2 \circ \alpha_2 + \mu\beta_1}{(1+2\rho)\hat{\mathbf{k}}_1 \circ \hat{\mathbf{k}}_1 + \lambda\hat{\mathbf{k}}_1 + \mu} \end{aligned} \tag{26}$$

In the same way, the DFT of α_2 is obtained from

$$\hat{\alpha}_2 = \frac{\hat{\mathbf{k}}_2 \circ \hat{\mathbf{y}} + 2\rho\hat{\mathbf{k}}_2 \circ \hat{\mathbf{k}}_1 \circ \alpha_1 + \mu\beta_2}{(1+2\rho)\hat{\mathbf{k}}_2 \circ \hat{\mathbf{k}}_2 + \lambda\hat{\mathbf{k}}_2 + \mu} \tag{27}$$

Here, $\mathbf{k}_2$ is the first row of matrix $\mathbf{K}_2$.

3.3 Model update

To update the proposed MKCF_CNN method for robust visual tracking, an incremental scheme is adopted to update the proposed model,

$$\alpha_{1,t} = (1-\eta)\alpha_{1,t-1} + \eta\alpha_{1,t}\alpha_{2,t} = (1-\eta)\alpha_{2,t-1} + \eta\alpha_{2,t} \tag{28}$$

$$\mathbf{x}_{1,t} = (1-\eta)\mathbf{x}_{1,t-1} + \eta\mathbf{x}_{1,t}\mathbf{x}_{2,t} = (1-\eta)\mathbf{x}_{2,t-1} + \eta\mathbf{x}_{2,t} \tag{29}$$

where η is a constant parameter which controls the learning rate. The subscript t denotes the tth frame. The incremental update strategy can deal with the abrupt change in successive frame.

3.4 Target detection

For kernel correlation filter $\mathbf{K}_1$, in the tth frame sequence, a great deal of circulant candidates, denoted as $\mathbf{x}'_{1,t}$, are extracted around the base image $\mathbf{x}_{1,\ t-1}$. The base image $\mathbf{x}_{1,\ t-1}$ locates at the position of the target at the $(t-1)$th frame. The candidates $\mathbf{x}'_{1,t}$ have a circulant structure. Thus, the responses of these candidates are given by

$$\text{response1} = \mathscr{F}^{-1}\left(\hat{\mathbf{k}}_1^{'} \circ \alpha_{1,t}\right) \tag{30}$$

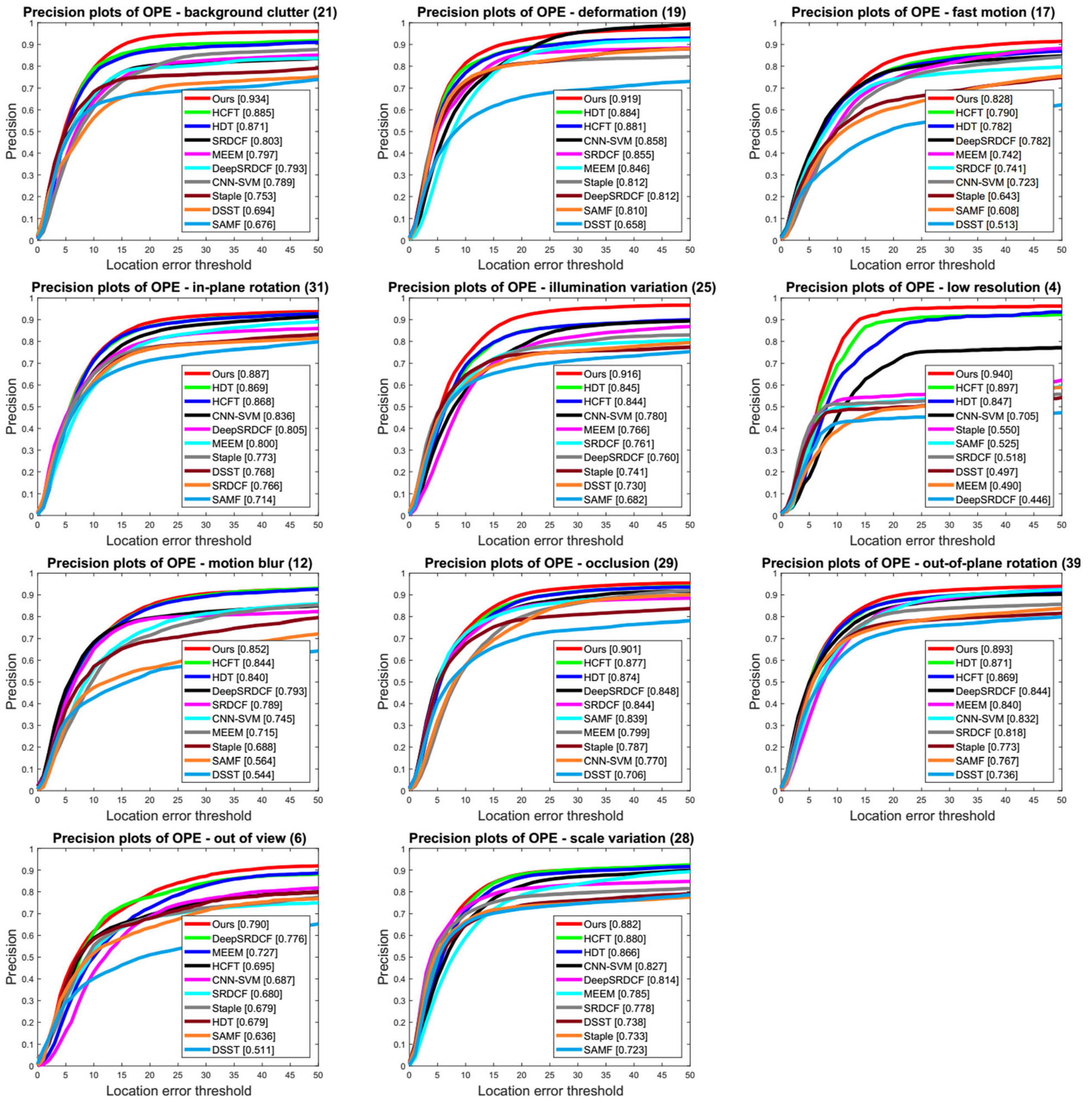


Fig. 5 Precision plots of OPE with different attributes on OTB-2013

In the same way, the responses of these candidates $\mathbf{x}'_{2,t}$ with respect to kernel correlation filter $\mathbf{K}_2$ are obtained by

$$\text{response2} = \mathscr{F}^{-1}\left(\hat{\mathbf{k}}'_2 \circ \alpha_{2,t}\right) \tag{31}$$

The maximum values of response1 and response2 are easily achieved by max(response1(:)) and max (response2(:)), respectively. if max(response1(:)) > max(response2(:)), the final response is equal to max(-response1(:)). Otherwise, the final response is equal to max(response2(:)). The best position of the target is obtained according to the final response.

3.5 Convolutional neural network (CNN) features extracted from MatConvNet

Traditional features, such as histogram of oriented gradient (HOG), SIFT, and CN, have achieved

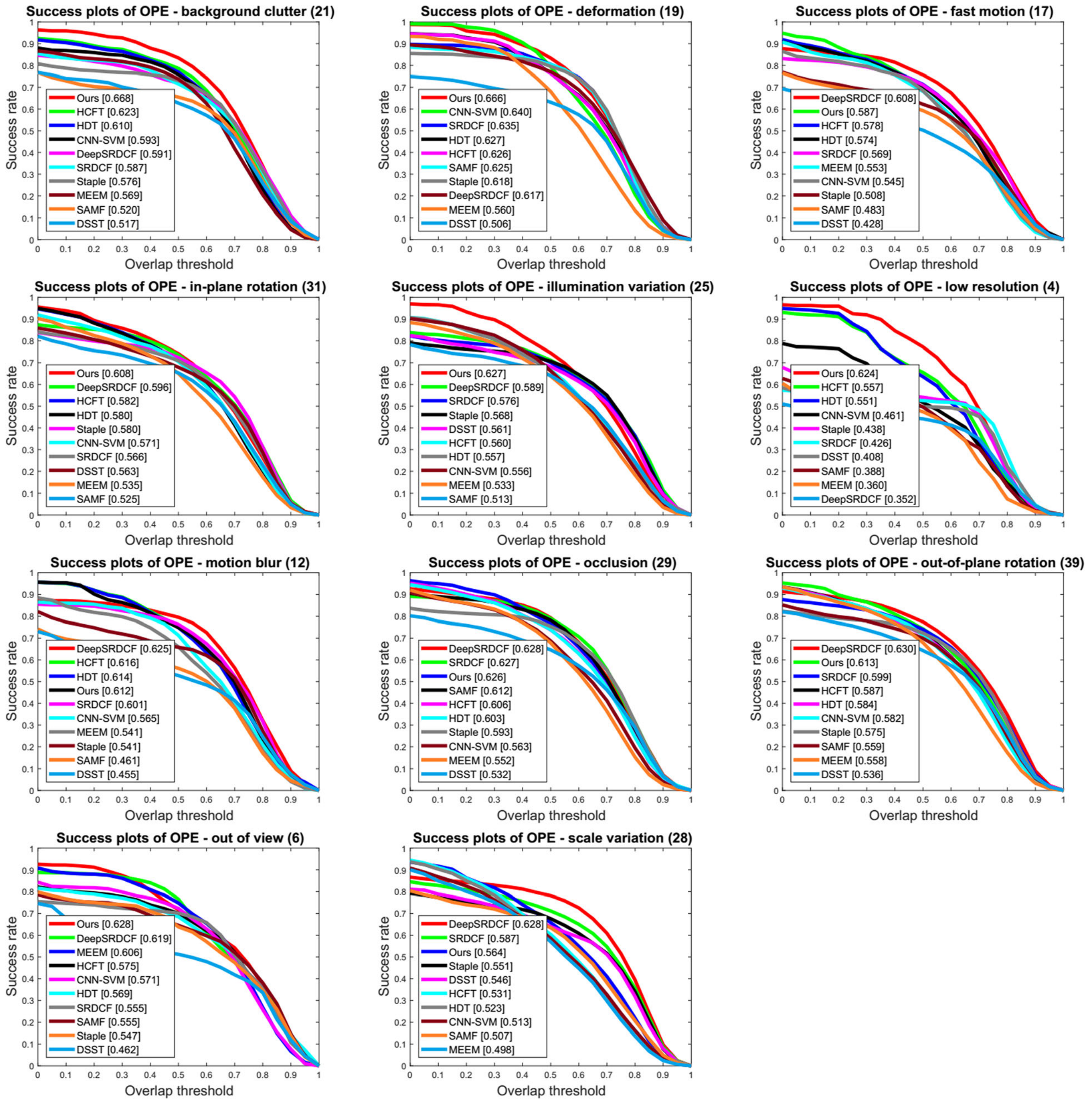


Fig. 6 Success plots of OPE with different attributes on OTB-2013

promising tracking performance in the past decade. However, these handcrafted features are out-of-date along with the rise of CNN features. In [40], the properties of CNN-based representation have gained impressive results on image recognition and object detection. In [35], three convolutional layers, conv3 – 4, conv4 – 4, conv5 – 4, utilizing VGG-19 model are introduced to the field of visual tracking and demonstrate powerful representation ability. Inspired by [41], we used the conv5 – 4 convolution layer and conv4 – 4 convolution layer of VGG-19 to model the appearance of the target. Features from conv5 – 4 convolution layer with more semantic information can discriminate the target from the dramatically changing background. Features from conv4 – 4 convolution layer with more spatial details can locate the position of target precisely.

3.6 Target recovery

We adopt the EdgeBox method [42] to re-detect the target from the failures of tracking. A great deal of object bounding box detection proposals $\mathbf{P}_d$ are generated by the EdgeBox method, and these proposals are evaluated under the framework of correlation filter to decide the final tracking position. Given the position (x_{t-1}, y_{t-1}) of the target in the $(t-1)$th frame, a set of bounding box proposals are extracted around the position of the target in the current frame. The position of each bounding box proposal p_i is set to (x_t^i, y_t^i) in the tth frame. The maximum response score of each bounding box proposal p_i is given by $r(p_i)$, which is computed by Eq. (7) using the HOG feature. If the score of tracking results in the tth frame is smaller than the threshold T_0, it can be believed that the tracker loses the target and the scheme of re-detection should be triggered. The optimal bounding box proposal in the tth frame is obtained by minimizing the following expression:

$$\begin{aligned} \arg\ \min_i r\left(p_t^i\right) + \alpha L\left(p_t^i, p_{t-1}\right) \\ s.t.\ \ r\left(p_t^i\right) > 1.5T_0 \end{aligned} \tag{32}$$

where $L(p_t^i, p_{t-1}) = \exp(-\frac{1}{2\sigma^2}\left\|(x_t^i, y_t^i) - (x_{t-1}, y_{t-1})\right\|^2)$. The formula $L(p_t^i, p_{t-1})$ is motion constraint between two successive frames. α is a constant parameter which controls the balance between the response score and the motion constraint. σ means the diagonal length of the initial target size.

3.7 Scale estimation

Scale estimation is very important for robust tracking. Motivated by [42], we use the EdgeBox method to deal with scale variation appeared in sequences. Given the size (w_{t-1}, h_{t-1}) of the target in the $(t-1)$th frame, we use the EdgeBox method to conduct on the multi-scale bounding box proposals $\mathbf{P}_s$ with the size of $sw_{t-1} \times sh_{t-1}$ in the current frame and reject the proposals whose intersection over union (IoU) is lower than 0.6 or higher than 0.9. For each accepted scale proposal, we compute the response score under the framework of correlation filter. If the maximum response score $\{r(p_i)|p_i \in \mathbf{P}_s\}$ is smaller than response obtained in Section 3.4, we keep the size of the target in the $(t-1)$th frame. Otherwise, we update the size of the target by the following equation:

$$(w_t, h_t) = \gamma\left(w_t^*, h_t^*\right) + (1-\gamma)(w_{t-1}, h_{t-1}) \tag{33}$$

where (w_t^*, h_t^*) is the size of the proposal with the maximum response score. γ is a constant parameter which controls the update rate.

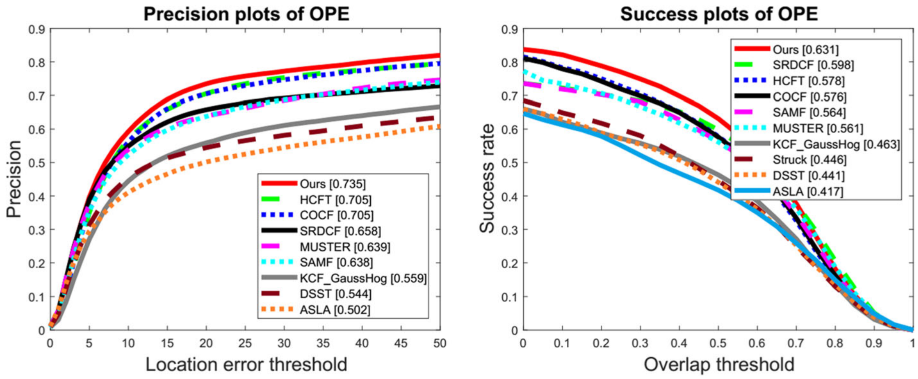


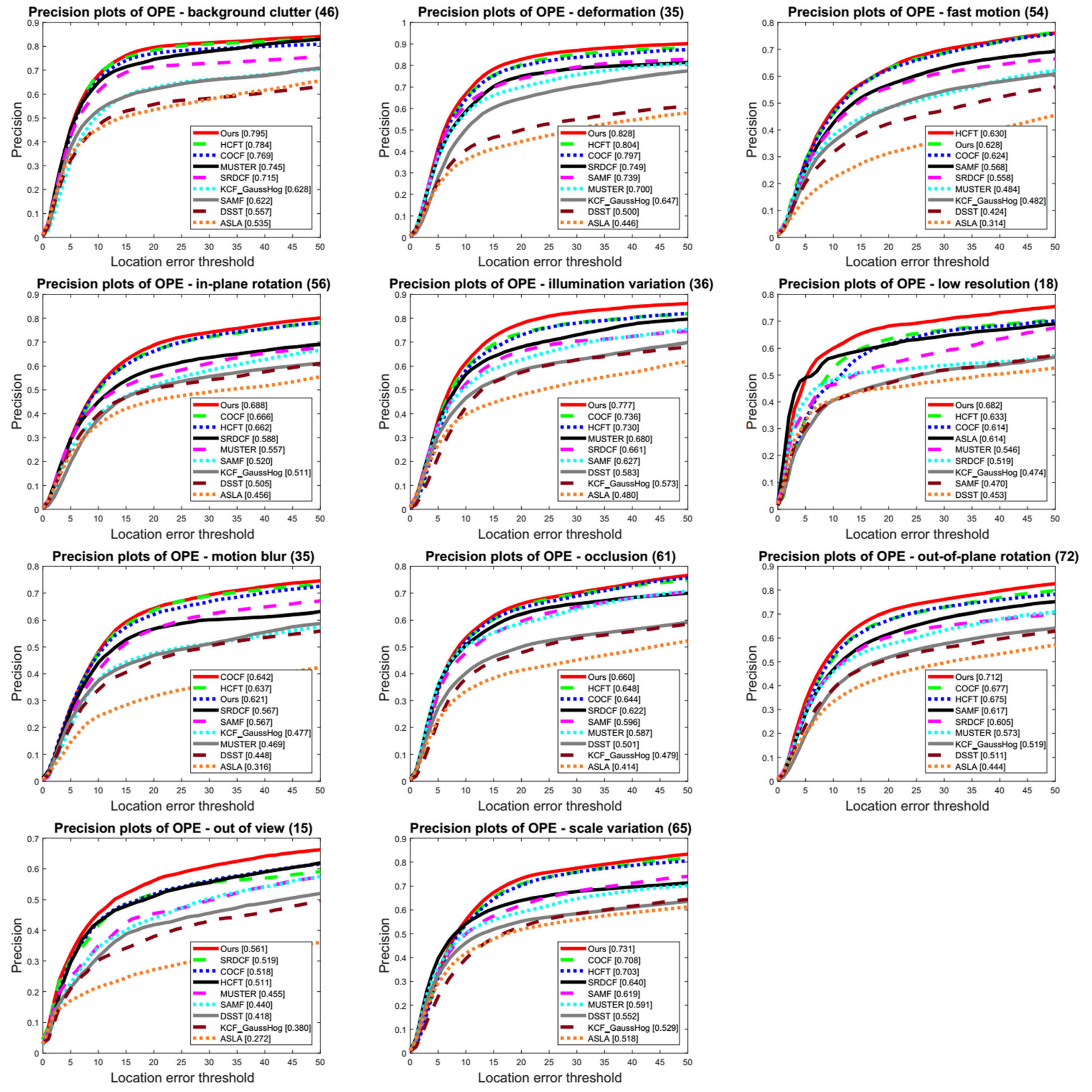
Fig. 7 Precision plots and success plots of OPE of our proposed method against other state-of-the-art methods on TColor-128

4 Results and discussion

In this section, we evaluate our proposed method on three public datasets: OTB-2013 [43], TColor-128 [44], and DTB70 [45]. Matlab pseudo-codes and tracking pipeline of our MKCF_CNN method are given in Tables 1 and 2, separately. Extensive experiments demonstrate that our method is able to achieve a very appealing performance in terms of effectiveness and robustness.

4.1 Experimental setup

The proposed MKCF_CNN method is implemented in MATLAB on a PC equipped with an Intel Xeon CPU E5-2640 v4 with 128G RAM and a single NVIDIA GeForce GTX 1080Ti. We adopt the pretrained VGGNet-19 as our feature extractor and utilize matcovnet for feature generation. We train two correlation filters utilizing outputs from the conv4 – 4 and conv5 – 4 layers. The linear kernel is adopted in this paper. The parameters λ, τ, μ,


Precision plots of OPE - background clutter (46)

Precision plots of OPE - deformation (35)

Precision plots of OPE - fast motion (54)

Precision plots of OPE - in-plane rotation (56)

Precision plots of OPE - illumination variation (36)

Precision plots of OPE - low resolution (18)

Precision plots of OPE - motion blur (35)

Precision plots of OPE - occlusion (61)

Precision plots of OPE - out-of-plane rotation (72)

Precision plots of OPE - out of view (15)

Precision plots of OPE - scale variation (65)

Fig. 8 Precision plots of OPE with different attributes on TColor-128

ρ in (14) are empirically set to 10^{-4}, 10^{-5}, 10^{-4}, and 10^{-3} separately. We set the update rate η in (28) and (29) to 0.01 and the weight parameter γ in (33) to 0.6. The tracking failure threshold T_0 is set to 0.2.

4.2 Evaluation metrics

We use two measurements, precision plots and success plots [46], to quantitatively assess the tracking results of our method. Precision plots illustrate the percentage of frames in which the center location error is within a given threshold. The threshold is set to 20 pixels. The center location error means the Euclidean distance between the tracked location and the ground truth. The success plots are the percentage of frames where the overlap rate S is larger than a fixed threshold T_1. The overlap rate S is defined as $S = \frac{\text{Aera}(B_E \cap B_G)}{\text{Aera}(B_E U B_G)}$. ∩ and ∪ are intersection and union operators, respectively. B_E denotes the estimated bounding

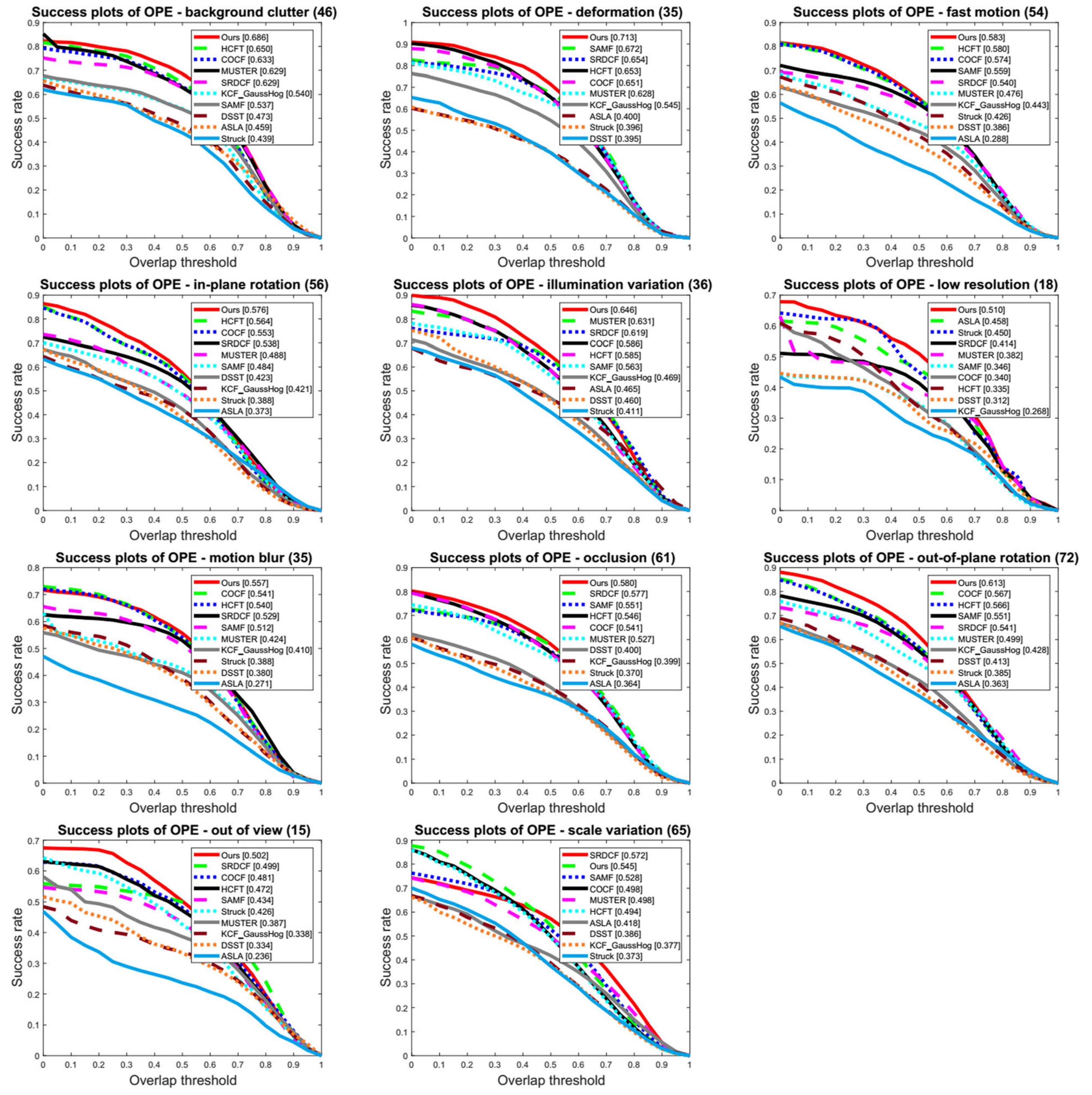


Fig. 9 Success plots of OPE with different attributes on TColor-128

Table 3 The success rates of 8 trackers with 11 challenging attributes on TColor-128 dataset. The best, second best, and third best tracking results are represented in red, blue, and green, respectively

Attribute	Ours	HCFT	COCF	MUSTER	SRDCF	KCF	SAMF	DSSF
BC	0.686	0.650	0.633	0.629	0.629	0.540	0.537	0.473
DEF	0.713	0.653	0.651	0.628	0.654	0.545	0.672	0.395
FM	0.583	0.580	0.574	0.476	0.540	0.443	0.559	0.386
IPR	0.576	0.564	0.553	0.488	0.538	0.421	0.484	0.423
IV	0.646	0.585	0.586	0.631	0.619	0.469	0.563	0.460
LR	0.510	0.335	0.340	0.382	0.414	0.268	0.346	0.312
MB	0.557	0.540	0.541	0.424	0.529	0.410	0.512	0.380
OC	0.580	0.546	0.541	0.527	0.577	0.399	0.551	0.400
OPR	0.613	0.566	0.567	0.499	0.541	0.428	0.551	0.413
OV	0.502	0.472	0.481	0.387	0.499	0.338	0.434	0.334
SV	0.545	0.494	0.498	0.498	0.572	0.377	0.528	0.386

box and B_G is the ground-truth bounding box. T_1 is set to 0.5 in this paper.

To evaluate the tracking performance of our method comprehensively, the challenging videos from OTB-2013 and TColor-128 are categorized with 11 attributes including background clutter (BC), deformation (DEF), fast motion (FM), in-plane rotation (IPR), illumination variation (IV), low resolution (LR), motion blur (MB), occlusion (OCC), out-of-plane rotation (OPR), out of view (OV), and scale variation (SV).

4.3 Comparison of tracking performance on OTB-2013

OTB-2013 benchmark dataset contains 51 sequences with 11 challenging attributes. We compare our method with 9 state-of-the-art algorithms which contain deep learning tracking methods (HCFT [35], HDT [47], CNN-SVM [48], DeepSRDCF [49]) and correlation filter tracking methods (MEEM [50], Staple [51], SAMF [52], DSST [28]). Figure 4 gives the precision plots and success plots of OPE of our proposed method against other state-of-the-state methods on OTB-2013. According to Fig. 4, our MKCF_CNN tracker outperforms most of the other trackers, demonstrating the effectiveness

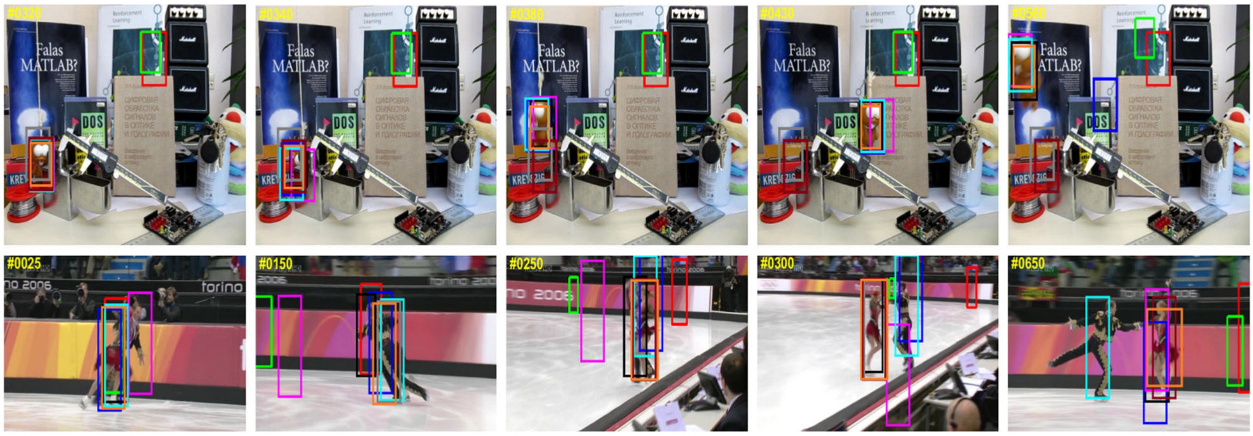


Fig. 10 Tracking results of ten trackers on sequences Lemming and skating2, in which the targets undergo occlusion. The tracking results of ASLA, IVT, CSK, SAMF, OAB, Struck, HCFT, COCF, and our tracker are shown by red, green, blue, black, magenta, cyan, gray, dark red, and orange rectangles, respectively

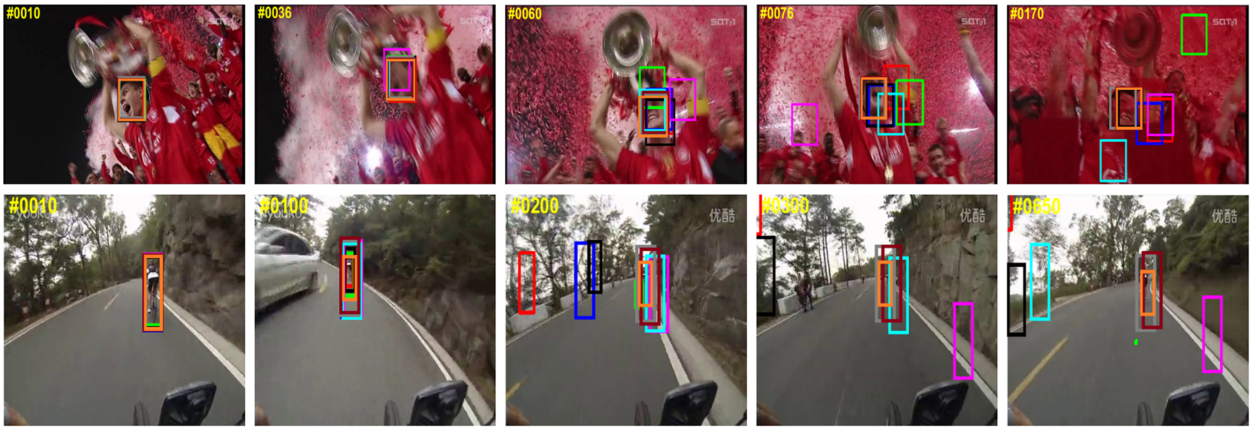


Fig. 11 Tracking results of nine trackers on sequences Soccer and Biker, in which the targets undergo fast motion. The tracking results of ASLA, IVT, CSK, SAMF, OAB, Struck, HCFT, COCF, and our tracker are shown by red, green, blue, black, magenta, cyan, gray, dark red, and orange rectangles, respectively

of MKCF_CNN. The proposed MKCF_CNN method achieves 2.3% performance gains in precision against HCFT, which is the most related tracking method with us. Meanwhile, MKCF_CNN and DeepSRDCF rank first on the success score.

In order to comprehensively assess the tracking performance of our proposed MKCF_CNN tracker, we present tracking results under OPE regarding 11 attributes in Figs. 5 and 6. We can observe that on the 51 videos with all the 11 challenging attributes, our method ranks first among the 10 evaluated trackers on precision plots. On the videos with attributes such as background clutter, deformation, in-plane rotation, illumination variation, low resolution, and out of view, MKCF_CNN ranks first among all the evaluated trackers on success plots. In the HCFT method, the outputs of the conv3 – 4, conv4 – 4, and conv5 – 4 layers are used as the deep features. In the HDT method, the outputs of six convolutional layers (10th–12th, 14th–16th) from VGGNet-19 are adopted as feature maps. However, only two layers (conv4 – 4, conv5 – 4) from VGGNet-19 are used in our proposed method, and two mutual kernelized correlation filters are trained to interact each other through all the tracking process without definite parameters as HCFT and definite initial parameters as HDT. From Figs. 5

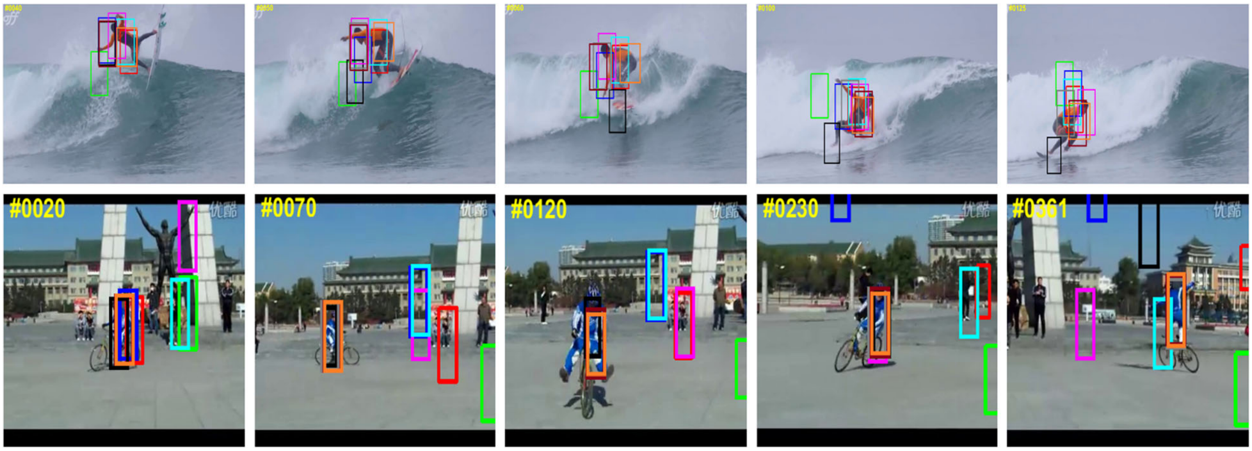


Fig. 12 Tracking results of nine trackers on sequences Surfing and Bikeshow, in which the targets undergo appearance variation. The tracking results of ASLA, IVT, CSK, SAMF, OAB, Struck, HCFT, COCF, and our tracker are shown by red, green, blue, black, magenta, cyan, gray, dark red, and orange rectangles, respectively

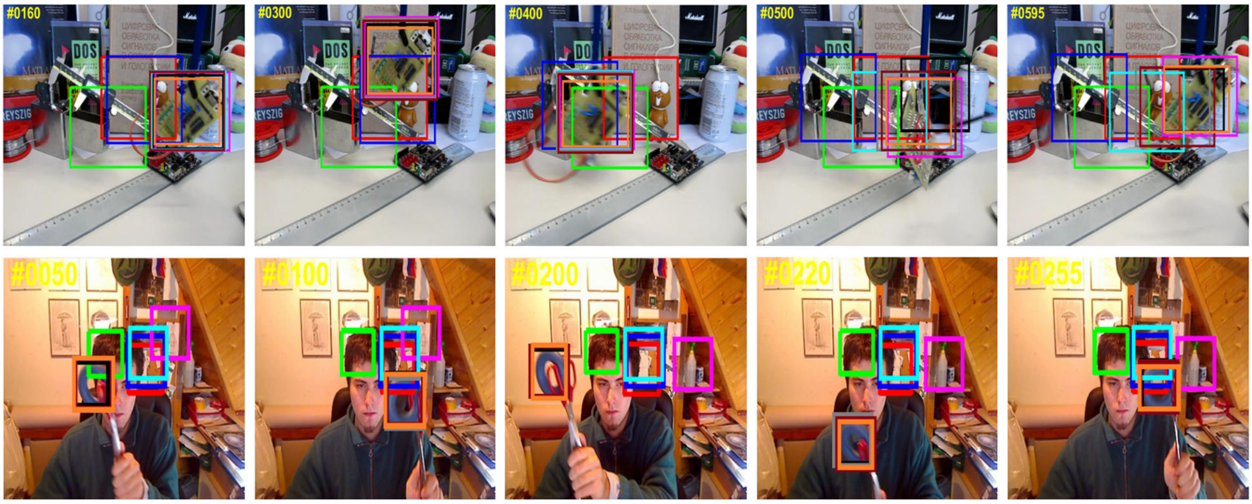


Fig. 13 Tracking results of nine trackers on sequences Board and Torus, in which the targets undergo background clutter. The tracking results of ASLA, IVT, CSK, SAMF, OAB, Struck, HCFT, COCF and our tracker are shown by red, green, blue, black, magenta, cyan, gray, dark red and orange rectangles, respectively

and 6, it is clear that our method performs better than those most relevant methods.

The tracking speed is very important for visual tracking. Correlation filter-based trackers obtained beyond real-time speed using handcrafted features. Except for DFT and inverse DFT, the computational complexity of trackers with a single correlation filter is $O(n \log n)$. n is the dimensionality of the features. Thus, the whole computational load of single correlation filter-based trackers is $O(Mn \log n)$. M is the number of base trackers. $M = 2$ in our method and $M = 3$ in HCFT. For trackers under the correlation filter framework with deep features, the computational burden mainly comes from the features extraction process. Thus, the tracking speed of our proposed method is 1.3 fps, which is a little faster than HCFT with a speed of 1.1 fps.

4.4 Comparison of tracking performance on TColor-128

The TColor-128 dataset consists of 128 challenging color videos and is designed to assess the tracking performance on color sequences. Similarly, we evaluated our proposed MKCF_CNN method with 9 state-of-the-art trackers, including HCFT [35], COCF [41], KCF_GaussianHog [27], SRDCF [29], MUSTER [53], SAMF [52], DSST [28], Struck [54], and ASLA [55]. Figure 7 shows precision plots and success plots of OPE of our proposed method against other state-of-the-art methods on TColor-128. Figures 8 and 9 present precision plots and success

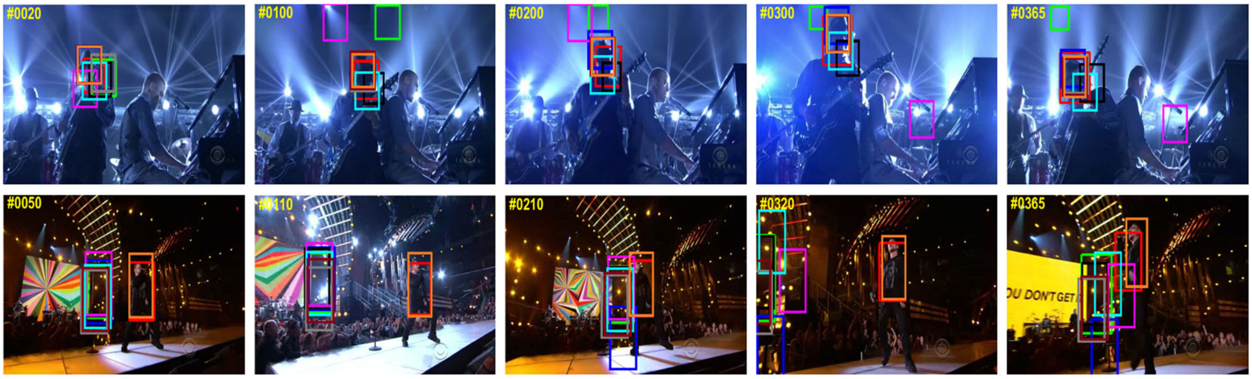


Fig. 14 Tracking results of nine trackers on sequences Shaking and Singer2, in which the targets undergo illumination change. The tracking results of ASLA, IVT, CSK, SAMF, OAB, Struck, HCFT, COCF, and our tracker are shown by red, green, blue, black, magenta, cyan, gray, dark red, and orange rectangles, respectively

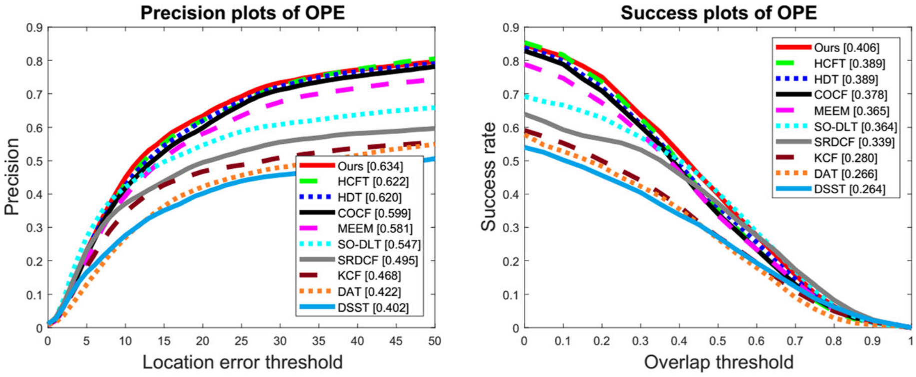


Fig. 15 Precision plots and success plots of OPE of our proposed method against other state-of-the-art methods on DTB

plots of OPE with different attributes on TColor-128, respectively. It is obvious that our method is the best one among the ten trackers on dataset TColor-128, following HCFT method. Our method obtains a precision rate of 73.5% and a success rate of 63.1%. HCFT and COCF rank second and third, respectively. Although HCFT utilizes deep features from three layers, its performance is not better than our method. COCF uses the same outputs as our method from two layers of VGGNet-19, and it performs worse than our MKCF_CNN tracker. This is because the scale estimation and re-detection scheme are able to locate the target precisely in our method. Figures 8 and 9 demonstrate the effectiveness of our method on TColor-128 with 11 challenging attributes. It can be seen that our method performs best against 9 other methods. Table 3 gives the data comparison of success rates of 8 trackers. The experimental results show that our method achieves the best performance under all challenging attributes except for scale variation.

Figure 10 shows some tracking results of two sequences with severe occlusion. In the Lemming video, the toy Lemming is severely occluded by a triangular rule when it is moving (e.g., #320, #340). It is obvious that the proposed method, SAMF, Struck, and OAB are robust to severe occlusion and can track the Lemming target steadily. In the skating2 sequence, the target woman dancer has obvious appearance variation and is totally occluded by the man dancer occasionally when they are skating (e.g., #150, #250). We can observe that the proposed method, HCFT and COCF with deep features, are able to deal with the severe occlusion and appearance variation effectively.

Figure 11 demonstrates some screenshots of two videos with fast motion. In the Soccer sequence, the player target keeps jumping and undergoes fast motion, background clutter, and occlusion when celebrating the victory (e.g., #36, #76, #170). IVT, Struck, CSK, ASLA, and OAB lose the target completely because of the challenging interference factors. The target in the Biker sequence undergoes

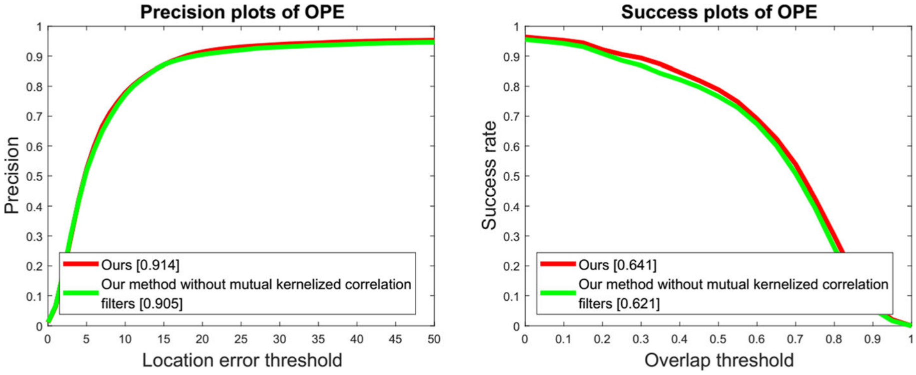


Fig. 16 Precision plots and success plots of OPE by our proposed method with mutual kernelized correlation filters and our method without mutual kernelized correlation filters on OTB-2013

fast motion and scale variation because of fast riding (e.g., #10, #100, #200). It can be easily seen that our method performs well in the entire sequence and is able to deal with motion blur and scale variation effectively.

Figure 12 illustrates some sampled tracking results of two sequences with appearance variation. The appearance of the target in the Surfing sequence changes severely when the player is going surfing (e.g., #100, #125). From the tracking results, we can see that most of the trackers are able to locate the target coarsely. However, only our method has the ability to track the target more precisely. In the Bikeshow sequence, the biker cycles in the square with severe appearance variation and scale change (e.g., #20, #120, #361). The proposed method, HCFT and COCF utilizing deep features, handle appearance change better than the other methods with handcrafted features.

Figure 13 demonstrates some tracking results of two sequences with background clutter. The target in the Board sequence moves in the complex scenes with severe background clutter (e.g., #160, #300, #400). It can be seen that our method can track the target successfully through the sequence. In the Torus sequence, the target moves in a cluttered room with slight appearance variation (e.g., #100, #200, #220). We can observe that trackers with handcrafted features can not deal with this situation and drift away to other objects.

Figure 14 shows some screenshots of tracking results in two sequences with illumination variation. In the Shaking video, a guitarist is playing on the stage with dim lights (e.g., #100, #200, #300). Although the target undergoes severe illumination variation, our method locates the target more precisely than other trackers. In the Singer2 sequence, the singer in dark clothes performing on the stage undergoes drastic illumination variation (e.g., #110, #210, #320). We can observe that HCFT and COCF with deep features move away from the target resulting in drastic illumination variation. Only our method is able to persistently track the target in the whole sequence.

4.5 Comparison of tracking performance on DTB

DTB dataset consists of 70 challenging videos captured by a camera mounted on an unmanned aerial vehicle (UAV). All of the 70 challenging sequences in the DTB dataset were manually annotated with 11 challenging attributes, including motion blur (MB), scale variation (SV), similar objects around (SOA), aspect ratio variation (ARV), background cluttered

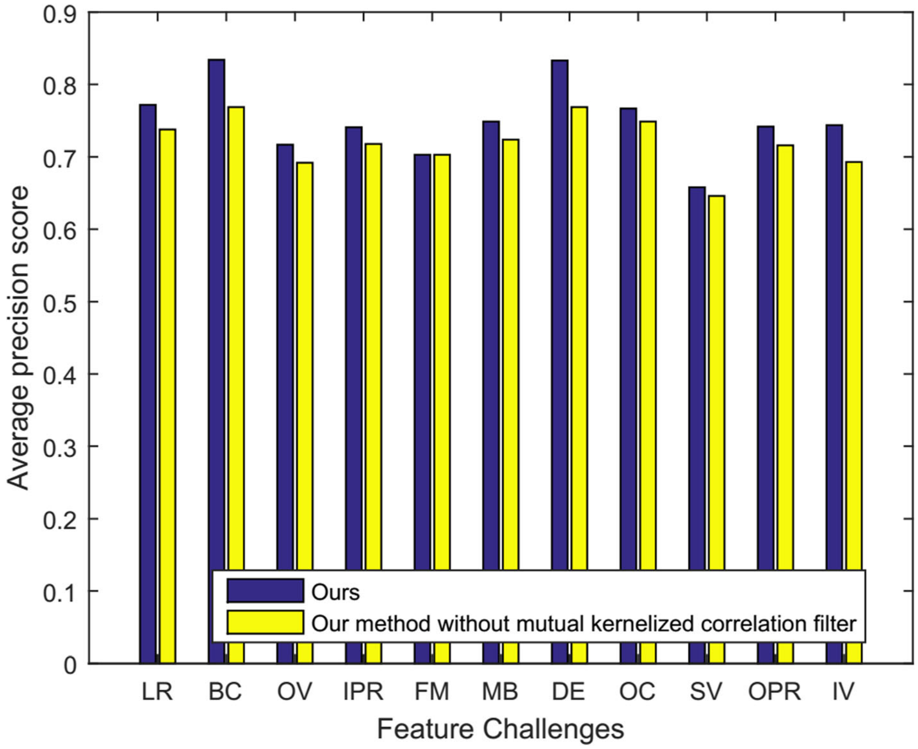


Fig. 17 Average precision score of our proposed method with mutual kernelized correlation filters and our method without mutual kernelized correlation filters in terms of 11 challenging attributes on OTB-2013

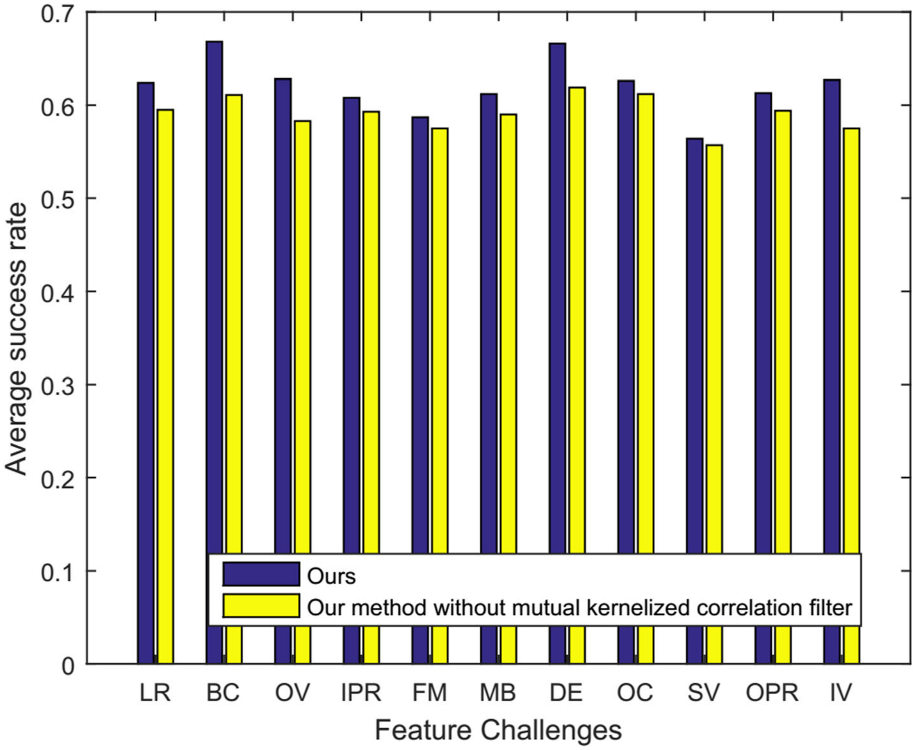


Fig. 18 Average success rate of our proposed method with mutual kernelized correlation filters and our method without mutual kernelized correlation filters in terms of 11 challenging attributes on OTB-2013

(BC), occlusion (OCC), out-of-view (OV), deformation (DEF), out-of-plane rotation (OPR), fast camera motion (FCM), and in-plane rotation (IPR). We compare our method with 9 representative trackers including HCFT [35], HDT [47], COCF [41], MEEM [50], SO-DLT [56], SRDCF [29], KCF [27], DAT [57], and DSST [28]. Figure 15 shows the overall tracking performance of OPE based on precision score and success score on DTB dataset. We can see that the proposed tracker can achieve the best tracking performance against 9 other trackers.

4.6 Ablation study

4.6.1 Effect of mutual kernelized correlation filters

In order to demonstrate the effectiveness of mutual correlation filters, we investigate the tracking performance of our proposed method with mutual correlation filters

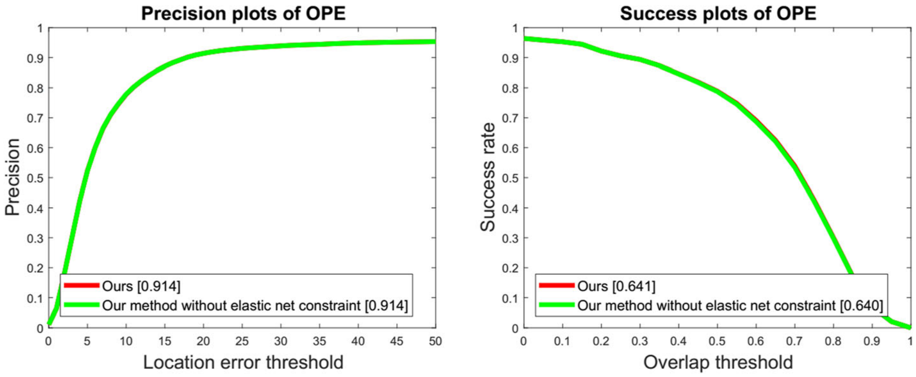


Fig. 19 Precision plots and success plots of OPE by our proposed method with elastic net constraint and our method without elastic net constraint on OTB-2013

Table 4 The success rates of our method with elastic net constraint and our method without elastic net constraint on OTB-2013. The best tracking results are represented in red. ENC denotes elastic net constraint

	LR	BC	OV	IPR	FM	MB	DE	OC	SV	OPR	IV
With ENC	0.624	0.668	0.628	0.608	0.587	0.612	0.666	0.626	0.564	0.613	0.627
Without ENC	0.624	0.668	0.628	0.606	0.587	0.612	0.666	0.624	0.562	0.612	0.626

and without mutual correlation filters on OTB-2013. Figure 16 gives the precision plots and success plots of OPE by different settings. Our method with mutual correlation filters achieves a score of 0.914 in terms of precision and the precision performance is improved by 0.9% compared with the method without mutual correlation filters. In success plots, owing to the interaction of mutual correlation filters, the tracking performance is improved by 2.0%. Figures 17 and 18 show the tracking results on OTB-2013 with 11 challenging attributes. It is obvious that our method with mutual correlation filters achieves better tracking performance in all the 11 attributes in both the average precision score and average success rate.

4.6.2 Effect of elastic net constraint

Figure 19 gives the tracking results on OTB-2013 by our method with elastic net constraint and our method without elastic net constraint in terms of precision and success rate. We can observe that the proposed method with elastic net constraint achieves slightly better than method without elastic net constraint. Table 4 demonstrates the tracking results on OTB-2013 with 11 challenging attributes. It is clear that our proposed method with elastic net constraint obtains better performance than method without elastic net constraint in terms of IPR, OC, SV, OPR, and IV.

4.6.3 Effect of scale estimation

In this section, we investigate the tracking performance with scale estimation scheme and without scale estimation scheme. Experimental results conducted on OTB-2013 are demonstrated in Figs. 20 and 21. The first picture in Fig. 20 shows the comparison of success plots of OPE on OTB-2013 and the second picture in Fig. 20 gives the success plots of OPE in terms of scale variation. Figure 21 shows the average success rate of our proposed method with scale estimation scheme and our method without scale estimation scheme in terms of 11 challenging attributes on OTB-2013. It can be seen that the scale estimation mechanism is able to improve the tracking performance greatly.

4.6.4 Effect of re-detection module

In this section, we compare the tracking performance with re-detection module and without re-detection module on OTB-2013. The first picture in Fig. 22 shows the comparison of success plots of OPE on OTB-2013 and the second picture in Fig. 22 gives the success plots of OPE in terms of occlusion. It is obvious that the re-detection module is able to recover

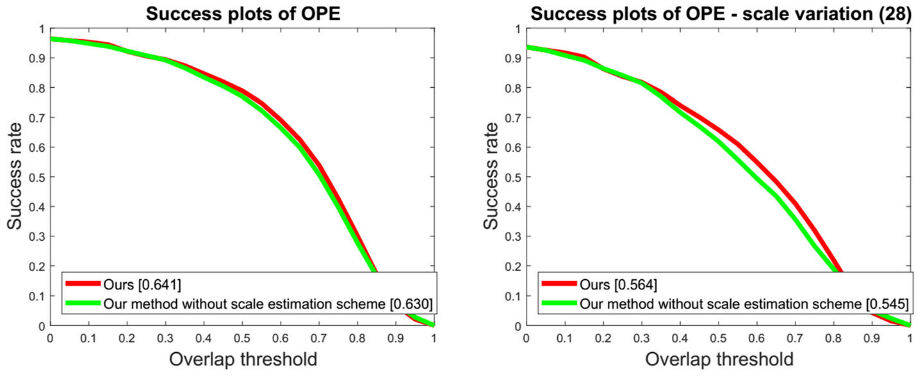


Fig. 20 The comparison of our method with scale estimation scheme and our method without scale estimation scheme on OTB-2013. The first figure demonstrates the success plots of OPE and the second one gives the success plots of OPE in terms of scale variation

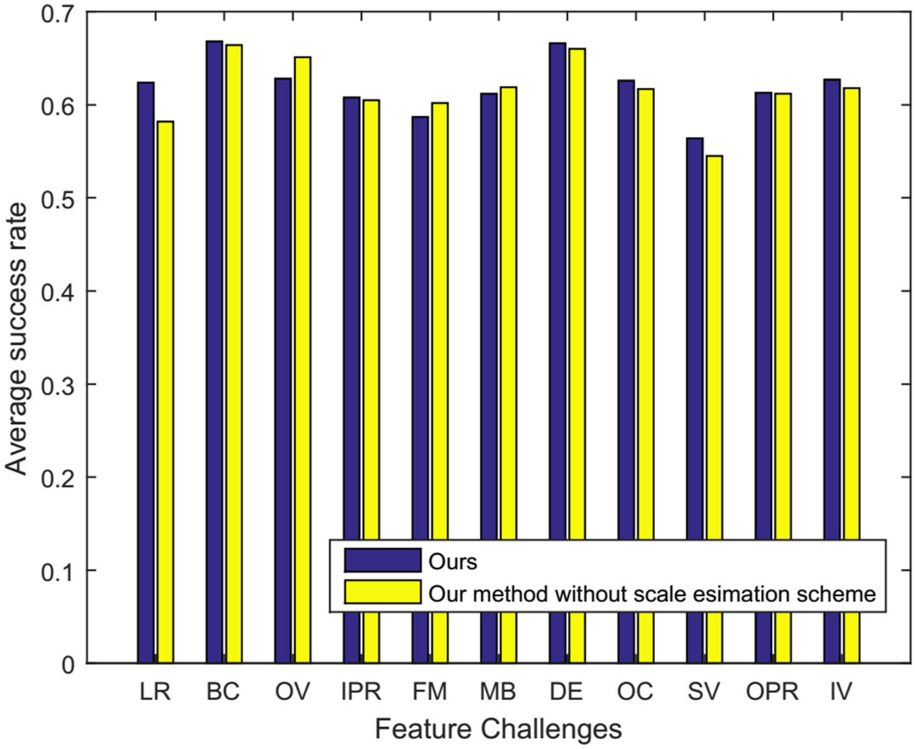


Fig. 21 Average success rate of our proposed method with scale estimation scheme and our method without scale estimation scheme in terms of 11 challenging attributes on OTB-2013

target in case of tracking failures. Table 5 gives the tracking results on OTB-2013 in terms of 11 challenging attributes. The best tracking results are shown in red. It is clear that our method with re-detection module achieves better tracking results in almost all the 11 attributes except for the LR and DE.

5 Summary and conclusion

In this paper, we propose a novel visual tracking method based on mutual kernelized correlation filters with elastic net constraint. The proposed algorithm is able to train two interactive discriminative classifiers to cope with the challenging environment and severe appearance variation. The elastic net constraint is imposed on the mutual kernelized correlation filters to group the similar

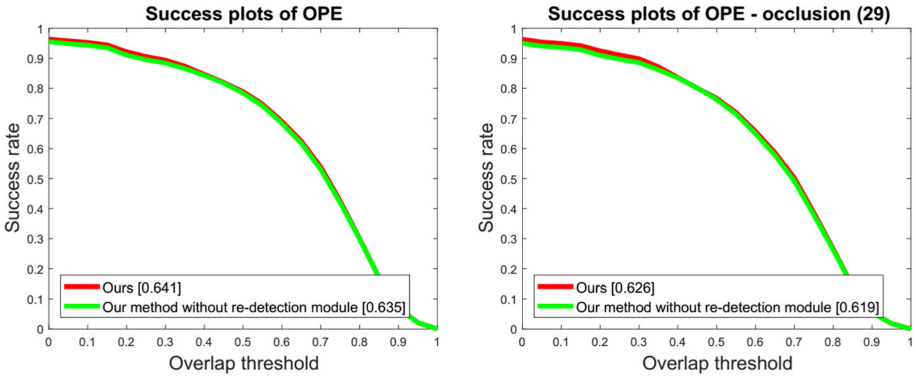
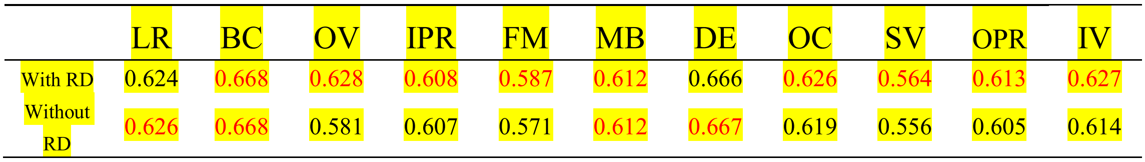


Fig. 22 Comparison of our method with re-detection module and our method without re-detection module on OTB-2013. The first picture demonstrates the success plots of OPE and the second one gives the success plots of OPE in terms of occlusion

Table 5 The success rates of our method with re-detection module and our method without re-detection module on OTB-2013. The best tracking results are represented in red. RD denotes re-detection



	LR	BC	OV	IPR	FM	MB	DE	OC	SV	OPR	IV
With RD	0.624	0.668	0.628	0.608	0.587	0.612	0.666	0.626	0.564	0.613	0.627
Without RD	0.626	0.668	0.581	0.607	0.571	0.612	0.667	0.619	0.556	0.605	0.614

features and to alleviate the impact of outliers. Scale adaption and re-detection scheme are applied in our method to promote tracking performance. Extensive experimental results demonstrate that our proposed method is able to obtain appealing tracking performance by using the interacted kernelized correlation filters with elastic net constraint. Quantitative and qualitative results show the superiority of our method in terms of effectiveness and robustness, compared with other tracking algorithms.

Abbreviations

BC: Background clutter; CN: Color name; CNN: Convolutional neural networks; DEF: Deformation; DFT: Discrete Fourier transform; ENC: Elastic net constraint; FFT: Fast Fourier transform; FM: Fast motion; HOG: Histogram of oriented gradient; IPR: In-plane rotation; IV: Illumination variation; KCF: Kernelized correlation filters; LASSO: Least absolute shrinkage and selection operator; MB: Motion blur; OCC: Occlusion; OPR: Out-of-plane rotation; OV: Low resolution; OV: Out of view; RD: Re-detection; RNN: Recurrent neural network; SIFT: Scale-invariant feature transform; SRDCF: Spatially regularized discriminative correlation filters; SV: Scale variation

Acknowledgements

The authors thank the editor and anonymous reviewers for their helpful comments and valuable suggestions.

Authors' contributions

HW proposed the study, conducted the experiments, and wrote the manuscript. SZ analyzed the data and revised the manuscript. Both authors read and approved the final manuscript.

Funding

This work was supported by A Project of Shandong Province Higher Educational Science and Technology Program under Grant No. J17KA088 and No. J16LN02, the Natural Science Foundation of Shandong Province under Grant No. ZR2015FL009 and No. ZR2019PF021, the Key Research and Development Program of Shandong Province under Grant No. 2016GGX101023, Scientific Research Fund of Binzhou University under Grant No. 2019ZD03 and Dual Service Projects of Binzhou University under Grant No. BZXYSFW201805.

Availability of data and materials

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Competing interests

The authors declare that they have no competing interests.

Received: 11 February 2019 Accepted: 11 July 2019
Published online: 23 July 2019

References

1. A. Li, M. Lin, Y. Wu, M. Yang, S. Yan, NUS-PRO: a new visual tracking challenge. IEEE Trans. Pattern Anal. Mach. Intell. **38**(2), 335–349 (2016)
2. P. Li, D. Wang, L. Wang, H. Lu, Deep visual tracking: review and experimental comparison. Pattern Recogn. **76**, 323–338 (2018)
3. S. Zhang, X. Lan, Y. Qi, C. Yuen, Robust visual tracking via basis matching, IEEE Trans. Circuits Syst. Video Technol. **27**(3), 421–430 (2017)
4. S. Zhang, H. Zhou, F. Jiang, X. Li, Robust visual tracking using structurally random projection and weighted least squares. IEEE Trans. Circuits Syst. Video Technol. **25**(11), 1749–1760 (2015)
5. D. Wang, H. Lu, M. Yang, Robust visual tracking via least soft-threshold square. IEEE Trans. Circuits Syst. Video Technol. **26**(9), 1709–1721 (2016)
6. L. Zhang, W. Wu, T. Chen, N. Strobel, D. Comaniciu, Robust object tracking using semi-supervised appearance dictionary learning. Pattern Recogn. Lett. **62**, 17–23 (2015)
7. W. Zhong, H. Lu, M. Yang, Robust object tracking via sparse collaborative appearance model. IEEE Trans. Image Process. **23**(5), 2356–2368 (2014)
8. Y. Song, C. Ma, L. Gong, J. Zhang, R. Lau, M. Yang, in *Proceedings of the IEEE International Conference on Computer Vision*. CREST: convolutional residual learning for visual tracking (2017), pp. 2555–2564
9. T. Zhang, C. Xu, M. Yang, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Multi-task correlation particle filter for robust object tracking (2017), pp. 4819–4827
10. W. Chen, K. Zhang, Q. Liu, Robust visual tracking via patch based kernel correlation filters with adaptive multiple feature ensemble. Neurocomput. **214**, 607–617 (2016)
11. K. Zhang, X. Li, H. Song, Q. Liu, Visual tracking using spatio-temporally nonlocally regularized correlation filter. Pattern Recogn. **83**, 185–195 (2018)
12. K. Zhang, Q. Liu, J. Yang, M.-H. Yang, Visual tracking via boolean map representations. Pattern Recogn. **81**, 47–160 (2018)
13. S. Yao, Z. Zhang, G. Wang, Y. Tang, L. Zhang, in *Proceedings of the European Conference on Computer Vision*. Real-time visual tracking: promoting the robustness of correlation filter learning (2016), pp. 662–678
14. M. Xue, H. Ling, in *Proceedings of the IEEE International Conference on Computer Vision*. Robust visual tracking using ℓ_1 minimization (2009), pp. 1436–1443
15. C. Bao, Y. Wu, H. Ling, H. Ji, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Real time robust ℓ_1 tracker using accelerated proximal gradient approach (2012), pp. 1830–1837
16. Z. Xiao, H. Lu, D. Wang, L2-RLS based object tracking. IEEE Trans. Circuits Syst. Video Technol. **24**(8), 1301–1308 (2014)
17. D. Wang, H. Lu, Fast and robust object tracking via probability continuous outlier model. IEEE Trans. Image Process. **24**(12), 5166–5176 (2015)
18. B. Babenko, M. Yang, S. Belongie, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Visual tracking with online multiple instance learning (2009), pp. 983–990
19. K. Zhang, L. Zhang, M. Yang, Fast compressive tracking. IEEE Trans. on Pattern Anal. Mach. Intell. **36**(10), 2002–2015 (2014)
20. K. Zhang, L. Zhang, Q. Liu, D. Zhang, M. Yang, in *Proceedings of the European Conference on Computer Vision*. Fast visual tracking via dense spatio-temporal context learning (2014), pp. 127–141
21. M. Wang, Y. Liu, Z. Huang, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Large margin object tracking with circulant feature maps (2017), pp. 4021–4029
22. H. Fan, H. Ling, in *Proceedings of the IEEE International Conference on Computer Vision*. Parallel tracking and verifying: a framework for real-time and high accuracy visual tracking (2017), pp. 5486–5494
23. F. Li, C. Tian, W. Zuo, L. Zhang, M. Yang, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Learning spatial-temporal regularized correlation filters for visual tracking (2018), pp. 4904–4913
24. W. Zuo, X. Wu, L. Lin, L. Zhang, M. Yang, Learning support correlation filters for visual tracking. IEEE Trans. on Pattern Anal. Mach. Intell. DOI: https://doi.org/10.1109/TPAMI.2018.2829180

25. M. Danelljan, G. Hager, F. Khan, M. Felsberg, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Adaptive decontamination of the training set: a unified formulation for discriminative visual tracking (2016), pp. 1430–1438
26. D. Bolme, J. Beveridge, B. Draper, Y. Lui, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Visual object tracking using adaptive correlation filters (2010), pp. 2544–2550
27. J. Henriques, R. Caseiro, P. Martins, J. Batista, High-speed tracking with kernelized correlation filters. IEEE Trans. on Pattern Anal. Mach. Intell. **37**(3), 583–596 (2015)
28. M. Danelljan, G. Hager, F. Khan, M. Felsberg, Discriminative scale space tracking. IEEE Trans. on Pattern Anal. Mach. Intell. **39**(8), 1561–1575 (2017)
29. M. Danelljan, G. Hager, F. Khan, M. Felsberg, in *Proceedings of the IEEE International Conference on Computer Vision*. Learning spatially regularized correlation filters for visual tracking (2015), pp. 4310–4318
30. L. Bertinetto, J. Valmadre, F. Henriques, A. Vedaldi, H. Philip, in *Proceedings of the European Conference on Computer Vision Workshops*. Fully-convolutional siamese networks for object tracking (2016), pp. 850–865
31. N. Hyeonseob, B. Han, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Learning multi-domain convolutional neural networks for visual tracking (2016), pp. 4293–4302
32. Z. Chi, H. Li, H. Lu, M. Yang, Dual deep network for visual tracking. IEEE Trans. Image Process. **26**(4), 2005–2015 (2017)
33. S. Zhang, Y. Qi, F. Jiang, X. Lan, P. Yuen, H. Zhou, Point-to-set distance metric learning on deep representations for visual tracking. IEEE Trans. Intell. Transp. Sys. **19**(1), 187–198 (2018)
34. K. Zhang, Q. Liu, Y. Wu, M. Yang, Robust visual tracking via convolutional networks without training. IEEE Trans. Image Process. **25**(4), 1779–1792 (2016)
35. C. Ma, J. Huang, X. Yang, M. Yang, in *Proceedings of the IEEE International Conference on Computer Vision*. Hierarchical convolutional features for visual tracking (2015), pp. 3074–3082
36. L. Wang, W. Ouyang, X. Wang, H. Lu, in *Proceedings of the IEEE International Conference on Computer Vision*. Visual tracking with fully convolutional networks (2015), pp. 3119–3127
37. F. Heng, H. Ling, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. SANet: structure-aware network for visual tracking (2017), pp. 42–49
38. 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
39. S. Yao, G. Wang, L. Zhang, Correlation filter learning toward peak strength for visual tracking. IEEE Trans. Cybern. **48**(4), 1290–1303 (2018)
40. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv:1409.1556(2015)
41. L. Zhang, P. Suganthan, Robust visual tracking via co-trained Kernelized correlation filters. Pattern Recogn. **69**, 82–93 (2017)
42. D. Huang, L. Luo, M. Wen, Z. Chen, C. Zhang, in *Proceedings of British Machine Vision Conference*. Enable scale and aspect ratio adaptability in visual tracking with detection proposals (2015), pp. 185.1–185.12
43. Y. Wu, J. Lim, M. Yang, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Online object tracking: a benchmark (2013), pp. 2411–2418
44. P. Liang, E. Blasch, H. Ling, Encoding color information for visual tracking: algorithms and benchmark. IEEE Trans. Image Process. **24**(12), 5630–5644 (2015)
45. S. Li, D. Yeung, in *AAAI Conference on Artificial Intelligence*. Visual object tracking for unmanned aerial vehicles: a benchmark and new motion models (2017), pp. 4140–4146
46. Y. Wu, J. Lim, M. Yang, Object tracking benchmark. IEEE Trans. on Pattern Anal. Mach. Intell. **37**(9), 1834–1848 (2015)
47. Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, M. Yang, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Hedged deep tracking (2016), pp. 4303–4311
48. S. Hong, T. You, S. Kwak, B. Han, in *Proceedings of the 32nd International Conference on International Conference on Machine Learning*. Online tracking by learning discriminative saliency map with convolutional neural network (2015), pp. 597–606
49. M. Danelljan, G. Hager, F. Khan, M. Felsberg, in *Proceedings of the IEEE International Conference on Computer Vision Workshop*. Convolutional features for correlation filter based visual tracking (2015), pp. 621–629
50. J. Zhang, S. Ma, S. Sclaroff, in *Proceedings of the European Conference on Computer Vision*. MEEM: robust tracking via multiple experts using entropy minimization (2014), pp. 188–203
51. B. Luca, V. Jack, G. Stuart, M. Ondrej, P. Torr, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016*. Staple: complementary learners for real-time tracking (2016), pp. 1401–1409
52. Y. Li, J. Zhu, in *Proceedings of the European Conference on Computer Vision*. A scale adaptive kernel correlation filter tracker with feature integration (2014), pp. 254–265
53. Z. Hong, Z. Chen, C. Wang, M. Xue, D. Prokhorov, D. Tao, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Multi-store tracker (MUSTer): a cognitive psychology inspired approach to object tracking (2015), pp. 749–758
54. S. Hare, A. Saffari, H.S. Philip, in *Proceedings of the IEEE International Conference on Computer Vision*. Struck: structured output tracking with kernels (2011), pp. 263–270
55. X. Jia, H. Lu, M. Yang, Visual tracking via coarse and fine structural local sparse appearance models. IEEE Trans. Image Process. **25**(10), 4555–4564 (2016)
56. N. Wang, S. Li, A. Gupta, D. Y. Yeung, Transferring rich feature hierarchies for robust visual tracking, arXiv:1501.04587(2015)
57. H. Possegger, T. Mauthner, H. Bischof, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. In defense of color-based model-free tracking (2015), pp. 2113–2120

Publisher's Note

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