International Journal of Computer Vision
https://doi.org/10.1007/s11263-019-01156-6

Exploiting the Anisotropy of Correlation Filter Learning for Visual Tracking

Yao Sui[1] · Ziming Zhang[2] · Guanghui Wang[3] · Yafei Tang[4] · Li Zhang[5]

Received: 30 April 2017 / Accepted: 29 January 2019

Abstract
Correlation filtering based tracking model has received significant attention and achieved great success in terms of both tracking accuracy and computational complexity. However, due to the limitation of the loss function, current correlation filtering paradigm could not reliably respond to the abrupt appearance changes of the target object. This study focuses on improving the robustness of the correlation filter learning. An anisotropy of the filter response is observed and analyzed for the correlation filtering based tracking model, through which the overfitting issue of previous methods is alleviated. Three sparsity related loss functions are proposed to exploit the anisotropy, leading to three implementations of visual trackers, correspondingly resulting in improved overall tracking performance. A large number of experiments are conducted and these experimental results demonstrate that the proposed approach greatly improves the robustness of the learned correlation filter. The proposed trackers performs comparably against state-of-the-art tracking methods on four latest standard tracking benchmark datasets.

Keywords Object tracking · Anisotropy · Correlation filtering · Loss function · Sparsity · Robustness · Sensitivity

1 Introduction

Visual tracking, as a fundamental problem in computer vision, concentrates on estimating the motion states of the object target in successive frames, given an initial motion state in the first frame. A popular approach to report the estimated motion states is to mark the object target in each frame by a bounding box. Various applications rely on visual tracking in practice, such as robotics, visual surveillance, human computer interaction, and unmanned control system. For decades, impressive achievements have been made in visual tracking. There are, however, still many challenges to conquer for a robust visual tracker, such as heavy occlusion, illumination change, non-rigid deformation, in-plane/out-of-plane rotation, background clutter, and scale variation.

Recently, a significant interest is attracted in correlation filtering based tracking model. Under this paradigm, over the previously estimated target regions, a correlation filter is

Communicated by Xiaoou Tang.

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants 61132007 and 61573351, in part by the Kansas NASA EPSCoR Program under Grant KNEP-PDG-10-2017-KU, and in part by the joint fund of Civil Aviation Research by the National Natural Science Foundation of China (NSFC) and Civil Aviation Administration under Grant U1533132.

✉ Yao Sui
suiyao@gmail.com

Ziming Zhang
zzhang@merl.com

Guanghui Wang
ghwang@ku.edu

Yafei Tang
tangyf24@chinaunicom.cn

Li Zhang
chinazhangli@tsinghua.edu.cn

[1] Harvard Medical School, Harvard University, Boston, MA 02115, USA

[2] Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA 02139, USA

[3] Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, KS 66045, USA

[4] China Unicom Research Institute, Beijing 100032, China

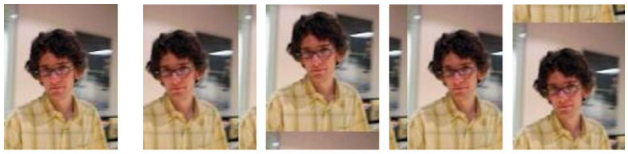
[5] Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

Published online: 12 February 2019

efficiently learned online in the Fourier frequency domain, and the object target is localized according to the magnitude of the filter response (i.e., correlation) over a large number of target candidates. The major strength of this paradigm is its high computational efficiency, because the object target and the candidate regions can be represented in frequency domain through fast Fourier transform (FFT). As a result, the computational complexity yields $\mathcal{O}(n \log n)$ for a region of $\sqrt{n} \times \sqrt{n}$ pixels. For this reason, many visual trackers (Bolme et al. 2010; Henriques et al. 2012, 2015; Danelljan et al. 2014a, b, 2015; Li and Zhu 2014; Zhang et al. 2014b; Liu et al. 2015) have been proposed within the correlation filtering paradigm in recent years.

Specifically, in the correlation filter learning, the training set comprises the samples from the regions centered at the previously estimated target regions, and the ground truth labels for each pixel within the training region are predefined and treated as the expected filter responses. The goal of the training is to make the filter have its strongest response at the center of the training region, i.e., the peak of the filter response is located at the center of the previously estimated target regions. Note that the expected filter responses employed in previous methods are always assigned to be of Gaussian shaped, which is considered as a continuous version of an impulse signal. In the testing (i.e., tracking) phase, the learned filter applies a correlation over candidate regions, and the candidate region with the strongest filter response is determined as the target.

Note that, from a signal processing perspective, the Gaussian shaped response used in previous methods is *isotropic*.[1] It indicates that all the regions that deviate the same distance away from the center of the target are assigned with the same ground truth labels (filter response values). However, from a regression point of view (Zhang et al. 2015c; Hare et al. 2011),[2] it has been demonstrated that the anisotropic response values (i.e., non-Gaussian shaped ground truth labels for the training samples) can significantly improve the tracking performance, e.g., the ground truth labels (expected response values) are set to the overlap rates between the training samples and the target. Figure 1 illustrates a popular approach to the training samples generation, which is adopted by previous correlation filtering based trackers (Henriques et al. 2012, 2015). It can be seen from Fig. 1b that the regions of interest are discontinuous. Under the learning framework employed in previous methods, the four regions with significant difference between each other are assigned with the same ground truth label due to the isotropic (i.e., Gaussian shaped) response setting. Such an isotropy brings challenges to the correlation filter learning, easily leading to an overfitting.



(a) **(b)**

Fig. 1 Illustration of the cyclic shift. **a** A base image. **b** Cyclic shifts of the base image by ± 15 pixels in horizontal and vertical directions, respectively

On the other hand, from a loss function perspective, the correlation filters in the previous methods are always learned under the squared loss (i.e., ℓ_2-loss). The choice for the squared loss is limited by the Parseval's identity, through which the learning problem can be exactly transferred into Fourier frequency domain where the correlation filtering can be conducted very efficiently. In addition, the squared loss can result in a closed-form solution for the correlation filter learning, which is a guarantee for the high computational efficiency of the visual trackers. Nevertheless, during tracking, the appearance of the object target may change significantly and abruptly in successive frames in various complicated situations, e.g., in the cases of severe occlusion and non-rigid deformation. To this end, a robust loss function is required to reliably deal with the appearance changes, and avoid the overfitting. Stochastically, the squared loss allows the filter response to fit the ground truth labels with small errors in the learning, i.e., the errors yield a Gaussian distribution with a small variance. However, in the presence of significant and drastic appearance changes, the errors might be extremely large in some feature dimensions, i.e., the errors follow a heavy-tailed distribution, e.g., Laplace distribution. The inappropriate assumption on the errors is the essential source of the instability of the squared loss.

Furthermore, under the correlation filtering framework (Henriques et al. 2012, 2015), it is difficult to efficiently incorporate scale adaptation for the target across frames. During the correlation filtering, due to the dense sampling for the regions of the target candidates, only a correlation filter with a fixed size is allowed to fast detect the object target. As a result, the target is localized with lower accuracy due to lack of scale adaptation. Two approaches are usually adopted to deal with scale variation over frames within the correlation filtering framework. One is to use a detector similar to the

[1] The Gaussian shaped response is not necessarily isotropic because the covariance matrix determines the shape of a Gaussian. It is isotropic only if the covariance matrix is diagonal and all the diagonal elements have equal values. In previous methods, only the isotropic Gaussian response is employed since it is considered as the continuous version of an impulse signal in the image space. For the sake of simplicity, the Gaussian shaped response refers to the isotropic Gaussian case in this work hereafter.

[2] The exact equivalence between regression and correlation filtering under the circulant structure assumption is proved in Henriques et al. (2015).

one used for translation on a scale pyramid to estimate the scale of the target (Danelljan et al. 2014a), and the other is to approximate the scale using a multi-resolution strategy (Li and Zhu 2014). Both approaches slow down the running speed of visual trackers significantly. It is thus critical to make a trade-off between the tracking accuracy and the running speed.

Inspired by the success from previous methods, an *anisotropy* of the filter response is exploited in this work by means of an adaptive learning approach via robust loss functions, including ℓ_1-, $\ell_1\ell_2$-, and $\ell_{2,1}$-loss functions. Since large errors are allowed, especially in the case that significant changes occur in the target appearance, the proposed loss functions greatly increase the robustness of the correlation filter learning. As a result, three visual trackers are correspondingly proposed in this study. A multi-resolution strategy (Li and Zhu 2014) is employed to incorporate with the robust loss functions, which promotes the accuracy of the scale estimation while keeping the efficiency of the correlation filtering. In addition, it is also demonstrated from an experimental approach how the loss functions essentially influence the tracking performance. We observe that the shake magnitude of the peak values of the filter responses in successive frames is consistent with the tracking performance. This observation might inspire further investigations on the correlation filter learning in the future work.

The contributions of this work are summarized as the following threefold.

- Three robust loss functions are leveraged to improve the plain correlation filtering tracking framework by exploiting the anisotropy structure of the correlation filter learning. This approach leads to an anisotropic filter response, instead of a isotropic Gaussian shaped response used by previous methods, during the correlation filter learning from a signal processing perspective. The robustness of the correlation filter learning is significantly promoted.
- A novel formulation of the correlation filter learning is proposed, where an additive error term is adopted to compensate the anisotropy of the filter response. A fast algorithm is developed to optimize the proposed formulation in frequency domain by taking the advantages of fast Fourier transform.
- Extensive experimental demonstrations are conducted to verify the effectiveness of the proposed approach. Four latest standard tracking benchmarks are employed, two types of evaluation protocols are referred to, five evaluation criteria are leveraged for these experimental evaluations, and about 70 state-of-the-art visual trackers are used as the competing counterparts. It is shown that the proposed trackers perform competitively against these state-of-the-art tracking methods.

The proposed algorithms are evaluated by extensive experiments on four popular benchmarks, the OTB 2015 (Wu et al. 2015), the VOT 2015, 2016, and 2017 benchmarks (Kristan et al. 2016), and they perform competitively against their competing counterparts. A preliminary result of this work has been reported in Sui et al. (2016b), while the current study is a substantial extension of Sui et al. (2016b) in both methodology and experiments.

The remainder of this paper is organized as follows. The related work is presented in Sect. 2; Sect. 3 depicts the proposed approach; in Sect. 4, the experimental results are reported and analyzed; and this work is concluded in Sect. 5.

2 Related Work

Correlation filtering based tracking model (Bolme et al. 2010) has achieved state-of-the-art tracking performance in recent years in terms of both tracking accuracy and running speed. Henriques et al. (2012) bridged the ridge regression and correlation filtering, leading to a discriminative tracking model. To further speed up the tracking, the circulant structure is exploited in their work, through which the training samples can be considered as fully cyclic shifts of a base image (i.e., the target region). Essentially, as illustrated in Fig. 1, the fully cyclic shifts are adopted to approximate the dense sampling around the target region, resulting in the fact that the sample matrix has circulant structure. Subsequently, Henriques et al. (2015) generalized their work to kernelized feature space, and theoretically proved that the circulant structure exists in several kernelized feature spaces, such as linear, polynomial, and Gaussian kernel spaces. By fast Fourier transform, the correlation filtering can be exactly transferred to the frequency domain under the Parseval's identity, yielding a $n \times \log(n)$ computational complexity for a base image of size $\sqrt{n} \times \sqrt{n}$ pixels. Meanwhile, because the Gaussian shaped ground truth labels are leveraged, a closed-form solution to the correlation filter learning can be obtained. For these reasons, the correlation filtering based tracking models can run at a high speed in various tracking applications.

The major problem of the correlation filtering based paradigm (Bolme et al. 2010; Henriques et al. 2012, 2015) is that the correlation filter is unable to change its size during tracking. The underlying assumption is that the scale of the target object is invariant during tracking. However, this assumption cannot be held in most tracking scenarios, i.e., the scale of the target object always varies in successive frames due to the motions of the target and the camera, viewpoint change, *etc*. Two popular approaches are employed to handle the scale variation problem: one utilizing the multi-resolution method (Li and Zhu 2014), and the other using a detector based method (Danelljan et al. 2014a). However,

no matter which approach mentioned above is adopted, the running speed will slow down significantly, because additional computational cost is required to estimate the current scale. As a result, a trade-off needs to be made between the tracking accuracy and running speed, when designing a correlation filtering based visual tracker.

With the success in the correlation filtering based tracking methods, many visual trackers are proposed in recent years. Danelljan et al. (2014b) exploited the color attributes with the correlation filtering model to develop a real-time visual tracker. Liu et al. (2015) proposed a part-based correlation filtering method to improve the tracking performance. Ma et al. (2015b) decomposed the tracking into translation and scale estimations and built a long-term correlation filtering based tracking algorithm. Tang and Feng (2015) leveraged the invariance-discriminative power spectrums of various features to design a multi-kernel correlation filter for visual tracking. Danelljan et al. (2014a) employed a detector similar to the one used for translation on a scale pyramid to estimate the scale of the target, leading to improved tracking performance. Li and Zhu (2014) developed a scale estimation method within the correlation filtering by using multiple kernels. Liu et al. (2016) designed a structural correlation filter learning approach. Sui et al. (2018b) proposed a response peak strengthened correlation filter learning approach. Danelljan et al. (2015) imposed a spatial regularization on the correlation filter learning. Bibi et al. (2016) exploited a response adaptation of the target object within the correlation filtering tracking framework.

Beside the correlation filtering based approaches , extensive tracking methods were proposed and achieved state-of-the-art tracking performance, such as structural learning (Hare et al. 2011; Kalal et al. 2012; Zhang et al. 2015d), sparse and low-rank learning (Mei and Ling 2011; Zhang et al. 2012b; Sui et al. 2015a, c, 2018c; Sui and Zhang 2016), subspace learning (Kwon and Lee 2010; Wang et al. 2013; Sui and Zhang 2015; Sui et al. 2015b, 2016a, 2018a; Zhang et al. 2015d), multi-task learning (Zhang et al. 2012a, 2015b), and deep learning (Ma et al. 2015a; Wang et al. 2015a, 2016b; Qi et al. 2016b; Nam and Han 2016b). Readers are recommended to refer to Yilmaz et al. (2006) and Smeulders et al. (2014) for a thorough review of visual tracking.

3 Proposed Approach

3.1 Problem Analysis and Statement

The correlation filtering based visual tracking framework learns a correlation filter over the possible target region, where the training samples are fully cyclic shifts of the base image and the responses are set manually to the values of Gaussian shaped. The fully cyclic shifts, which approximate the dense sampling around the target location, make the computational efficiency significantly improved in the frequency domain. The Gaussian shape manually set to the response is regarded as a continuous version of a pulse signal on the target location. Clearly, the two choices make the correlation filter learning computationally efficient. However, they may also increase the risk of overfitting.

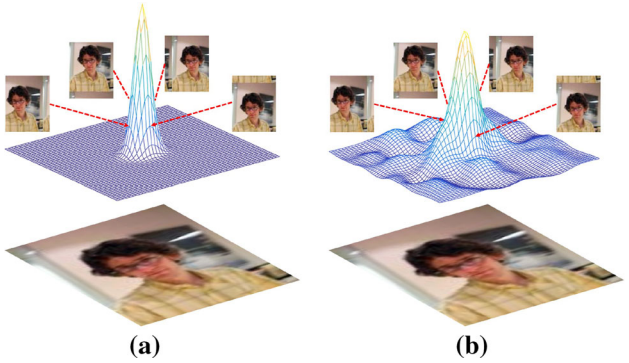


Fig. 2 Illustration of responses over a frame. **a** Gaussian shaped response. **b** Similarity based response

3.1.1 Analysis 1: Correlation Filtering

The cyclic shifts used in previous methods, as illustrated in Fig. 1, bring the discontinuity to the training samples, leading to significant differences between these shifted images. Meanwhile, imposing the Gaussian shaped response on these shifted images indicates that the samples deviating away the same distance from the target are enforced to have the same filter response, as illustrated in Fig. 2a. This may assign heavy load to the correlation filter to fit these significantly different samples by the same response value, easily leading to overfitting. For this reason, to make the correlation filter learning more accurate and robust, instead of using the isotropic (i.e., Gaussian shaped) response, an anisotropic response is desired and promising, as illustrated in Fig. 2b, where the training samples are configured to fit different response values. Various methods can be used to configure such response values in the correlation filter learning, e.g., using the similarity between candidate regions and the target region.

3.1.2 Analysis 2: Regression

The correlation filtering over the fully cyclic shifts of a base image has been demonstrated to be exactly equivalent to a ridge regression over these shifts (Henriques et al. 2015). A squared loss function (i.e., ℓ_2-loss) is employed to train a regressor. Statistically, the underlying assumption is that the fitting errors yield a Gaussian distribution with a zero mean and a small variance. It indicates that the fitting errors are

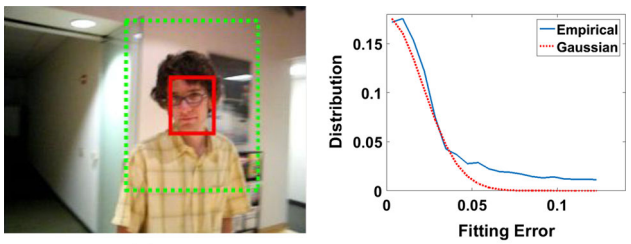


Fig. 3 Illustration of data fitting over a frame. **a** A frame image. **b** Distribution of data fitting errors

small and dense for all the training samples. However, the fitting errors may be extremely large in certain situations in visual tracking, e.g., in the presence of occlusions. Especially, the fully cyclic shifts of a base image, used as the training samples in the regressor learning, may violate the assumption of the Gaussian fitting error. Figure 3a shows a frame image, where the target is highlighted by a red (solid) box and the base image is marked by a green (dashed) box. We train a correlation filter over the frames prior to this frame, and then the fitting errors are computed from the data fitting and its objective Gaussian response in this frame. Figure 3b shows the distribution of the fitting errors. For the convenience of comparison, the corresponding Gaussian distribution is also plotted. It is evident that the fitting errors follow a heavy-tailed distribution in this frame, rather than a Gaussian. It indicates that a loss function yielding a heavy-tailed distribution is promising and potential to improve the robustness of the correlation filter learning.

3.1.3 Problem Statement

From the above analysis, we are facing such a problem to learn a robust correlation filter for visual tracking:

- from a correlation filtering perspective, an objective response with anisotropy is required in the correlation filter learning, instead of an isotropic (Gaussian shaped) response, to alleviate the risk of overfitting;
- from a regression perspective, a loss function yielding a heavy-tailed distribution is required in the regressor learning, rather than squared loss function, to enlarge the similarity (or dissimilarity) between the target and the candidate regions.

The typical correlation filtering based tracking model focuses on solving the following ridge regression problem

$$\min_{\mathbf{w}} \sum_i \left(f\left(\mathbf{x}_i\right) - y_i\right)^2 + \lambda \left\|\mathbf{w}\right\|_2^2, \tag{1}$$

where a regression function $f\left(\mathbf{x}_i\right) = \mathbf{w}^T \varphi\left(\mathbf{x}_i\right)$ is trained with a feature-space projector $\varphi\left(\cdot\right)$; the ground truth labels (i.e., objective values) y_i are specified to be of Gaussian shaped; and $\lambda > 0$ is a weight parameter balancing between the first and the second terms. The training samples $\{\mathbf{x}_i\}$ comprises the fully cyclic shifts of the base image centering at the latest target region. In the tracking (testing step), a region in the new frame, which centers at the same location of the target in the last frame, is considered as the local search region for the target to be tracked. The area of the search region is determined by the maximum translations of the target between two consecutive frames. Each pixel within the search region can be regarded as the center location of a candidate having the same size of the target. Thus, every candidate is evaluated by the learned regressor and the candidate with the largest regression value is determined as the target. Note that, according to the regression model trained from Eq. (1), the regression values can be considered as the likelihood of the candidates to be the target. From a correlation filtering approach, the regression on the candidates within the local search region can be implemented using a correlation with a filter kernel. The filter response indicates how strong a candidate is correlated to the previously estimated targets.

The proposed approach aims at improving the robustness of the correlation filter learning. As analyzed above, an anisotropy of the filter response is exploited for visual tracking from a signal processing perspective, and the robust loss functions are leveraged from an overfitting point of view to handle the significant changes in the target appearance. To this end, the regression in this work is generally defined as

$$\min_{\mathbf{w}} \sum_i \ell\left(f\left(\mathbf{x}_i\right) - y_i\right) + \lambda \left\|\mathbf{w}\right\|_2^2, \tag{2}$$

where $\ell\left(\cdot\right)$ denotes a loss function, and the regularizer $\left\|\mathbf{w}\right\|_2^2$ makes the regression stable.

The problem in Eq. (2) implicitly addresses the anisotropy of the objective values via the loss function $\ell\left(\cdot\right)$. To emphasize the anisotropy, Eq. (2) can be exactly rewritten as

$$\min_{\mathbf{w},\mathbf{e}} \sum_i \ell\left(e_i\right) + \lambda \left\|\mathbf{w}\right\|_2^2, \quad s.t.\ e_i = y_i - f\left(\mathbf{x}_i\right), \tag{3}$$

where e_i denotes the difference between the ground truth labels y_i and the regression values $f\left(\mathbf{x}_i\right)$, and y_i is of Gaussian shaped. In previous correlation filtering based methods, e_i is expected to be dense and small (tends to zeros) by incorporating with a ℓ_2-loss function. Correspondingly, the regression values $f\left(\mathbf{x}_i\right)$ is of Gaussian shaped (isotropic). In the proposed approach, e_i is allowed to be sparse and arbitrarily large under the robust loss functions. As a result, the regression values $\mathbf{f}\left(\mathbf{x}_i\right) = y_i - e_i$ is anisotropic for y_i of Gaussian shaped and e_i with arbitrary values.

In this work, an adaptive approach is adopted, which utilizes the sparsity based loss functions to adaptively fit the Gaussian shaped ground truth labels. Similar to the previous work (Henriques et al. 2012, 2015), the proposed approach is formulated from the regression point of view and solved within the correlation filtering paradigm. To improve the robustness of the proposed model against significant changes in the target appearance, the sparsity based loss function (Wright et al. 2010) is encouraged, which can tolerate large errors to fit the ground truth labels. In this work, three loss functions, ℓ_1-, $\ell_1\ell_2$- and $\ell_{2,1}$-loss, are utilized to respectively exploit the sparsity, elastic net, and group sparsity structures of the loss values.

3.2 Optimization Algorithm

The problem presented in Eq. (3) is NP-hard with respect to both $\mathbf{w}$ and $\mathbf{e}$ (Wright et al. 2010) since the sparsity constraints on the data fitting term (the first term) are involved. However, it is convex with respect to either $\mathbf{w}$ or $\mathbf{e}$. To this end, an iterative algorithm can be derived to approximate the solution by alternately optimizing one variable when fixing another. First, the Lagrange formulation of Eq. (3) is found by

$$\min_{\mathbf{w},\mathbf{e}} \sum_i \left(f\left(\mathbf{x}_i\right) + e_i - y_i\right)^2 + \lambda \left\|\mathbf{w}\right\|_2^2 + \tau \sum_i \ell\left(e_i\right), \tag{4}$$

where $\tau > 0$ is a weight parameter. Note that Eq. (4) can be split into two subproblems with respect to $\mathbf{w}$ and $\mathbf{e}$ respectively:

$$\min_{\mathbf{w}} \left\|\mathbf{f}\left(\mathbf{X}\right) + \mathbf{e} - \mathbf{y}\right\|_2^2 + \lambda \left\|\mathbf{w}\right\|_2^2 \tag{5}$$

$$\min_{\mathbf{e}} \left\|\mathbf{f}\left(\mathbf{X}\right) + \mathbf{e} - \mathbf{y}\right\|_2^2 + \tau \ell\left(\mathbf{e}\right), \tag{6}$$

where $\mathbf{X}$ is the sample matrix, and each row of the matrix $\mathbf{X}$ denotes a training sample. It can be seen that the above two subproblems are convex and have globally optimal solutions. As a result, Eq. (4) can be solved by optimizing the two subproblems alternately through an iterative way until the objective functions converge.

The dual space is utilized to solve Eq. (5), where the dual conjugate of $\mathbf{w}$, denoted by $\boldsymbol{\alpha}$, is introduced, such that $\mathbf{w} = \sum_i \alpha_i \varphi\left(\mathbf{x}_i\right)$. The subproblem with $\boldsymbol{\alpha}$ is thus written as

$$\min_{\boldsymbol{\alpha}} \left\|\mathbf{K}\boldsymbol{\alpha} + \mathbf{e} - \mathbf{y}\right\|_2^2 + \lambda \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha} \tag{7}$$

where $\mathbf{K}$ denotes the kernel matrix, of which the element $k_{ij} = \varphi^T\left(\mathbf{x}_i\right)\varphi\left(\mathbf{x}_j\right)$. Note that since the sample matrix $\mathbf{X}$ denotes all the training samples that are generated from the fully cyclic shifts of the latest target region, as demonstrated in Henriques et al. (2015), some kernels, such as linear, polynomial, and Gaussian, can lead to a circulant kernel matrix.[3] With such a circulant structure, the kernel matrix $\mathbf{K}$ can be diagonalized as

$$\mathbf{K} = \mathbf{D} diag\left(\hat{\mathbf{k}}_1\right) \mathbf{D}^H, \tag{8}$$

where $\mathbf{D}$ denotes the discrete Fourier transform (DFT) matrix, $\mathbf{D}^H$ denotes the Hermitian transpose of $\mathbf{D}$, and the hat $\hat{}$ stands for the DFT and hereafter. Note that the above diagonalization for $\mathbf{K}$ significantly improves the computational efficiency.

Return back to the problem with respect to $\boldsymbol{\alpha}$ shown in Eq. (7). It is evident that Eq. (7) is squared. It indicates that, by combining with the results shown in Eq. (8), it has a closed-form solution

$$\hat{\boldsymbol{\alpha}} = \frac{\hat{\mathbf{y}} - \hat{\mathbf{e}}}{\hat{\mathbf{k}}_1 + \lambda}, \tag{9}$$

where $\mathbf{k}_1$ denotes the first row of the kernel matrix $\mathbf{K}$, the fraction means element-wise division.

Next, for the subproblem with respect to $\mathbf{e}$, as shown in Eq. (6), three algorithms are derived, corresponding to the three loss functions adopted in Eq. (4).

(1) *ℓ_1-loss.* ℓ_1-loss can be implemented by a standard sparsity constraint that is imposed on $\mathbf{e}$ in this case. $\mathbf{e}$ has a globally optimal solution that can be obtained using the shrinkage thresholding algorithm (Beck and Teboulle 2009) from

$$\mathbf{e} = \sigma\left(\frac{1}{2}\tau, \mathcal{F}^{-1}\left(\hat{\mathbf{y}} - \hat{\boldsymbol{\alpha}} \odot \hat{\mathbf{k}}_1\right)\right), \tag{10}$$

where the operator $\mathcal{F}^{-1}\left(\cdot\right)$ stands for the inverse Fourier transform, the operator $\odot$ denotes element-wise multiplication, and the function σ is a shrinkage operator, defined as

$$\sigma\left(\varepsilon, x\right) = sign\left(x\right) \max\left(0, \left|x\right| - \varepsilon\right). \tag{11}$$

(2) *$\ell_1\ell_2$-loss.* In this case, $\mathbf{e}$ is constrained by an elastic net regularization. By completing the square, Eq. (6) can be solved in a similar way to ℓ_1-loss. $\mathbf{e}$ has a globally optimal solution and can be obtained from

$$\mathbf{e} = \sigma\left(\frac{\tau}{4+2\tau}, \frac{2}{2+\tau}\mathcal{F}^{-1}\left(\hat{\mathbf{y}} - \hat{\boldsymbol{\alpha}} \odot \hat{\mathbf{k}}_1\right)\right). \tag{12}$$

The coefficients of the ℓ_1- and ℓ_2-regularization terms in the elastic net constraint are set to be equal in the experiments.

(3) *$\ell_{2,1}$-loss.* The variables are considered to be two-dimensional (i.e., matrix variables) in this case. To this end,

[3] The rows of the kernel matrix $\mathbf{K}$ are actually obtained from the fully cyclic shifts of the vector $\mathbf{k}_1$.

the group sparsity of $\mathbf{e}$ is exploited under the $\ell_{2,1}$-loss. Note that $\mathbf{e}$ has a globally optimal solution and can be obtained using the accelerated proximal gradient method (Bach et al. 2011) from

$$\mathbf{e}_j = \begin{cases} \left(1 - \frac{1}{\tau \|\mathbf{q}_j\|_2}\right) \mathbf{q}_j, & \frac{1}{\tau} < \|\mathbf{q}_j\|_2 \\ \mathbf{0}, & otherwise, \end{cases} \tag{13}$$

where $\mathbf{e}_j$ denotes the jth column of the matrix $\mathbf{e}$, and $\mathbf{q} = \mathcal{F}^{-1}\left(\hat{\mathbf{y}} - \hat{\boldsymbol{\alpha}} \odot \hat{\mathbf{k}}_1\right)$. In addition, considering the symmetry of the matrix $\mathbf{e}$, the jth row of $\mathbf{e}$ is also zeroed for all $j \in \{k | \mathbf{e}_k = \mathbf{0}\}$.

In each iteration, the computational cost comes from the fast Fourier and the inverse fast Fourier transforms on $\mathbf{e}$, yielding a complexity of $\mathcal{O}(n \log n)$. In this work, the empirical results show that the algorithm converges within about a dozen of iterations.

In addition, to utilize the temporal information and avoid that the correlation filter learned in successive frames changes abruptly, the base image $\mathbf{x}_t$ and the correlation filter $\boldsymbol{\alpha}_t$ in the tth frame are updated in the frequency domain in an incremental manner, respectively:

$$\begin{aligned} \hat{\mathbf{x}}_t &= (1 - \eta)\, \hat{\mathbf{x}}_{t-1} + \eta \hat{\mathbf{x}}, \\ \hat{\boldsymbol{\alpha}}_t &= (1 - \eta)\, \hat{\boldsymbol{\alpha}}_{t-1} + \eta \hat{\boldsymbol{\alpha}}, \end{aligned} \tag{14}$$

where $\eta \in (0, 1)$ controls the update rate.

3.3 Target Localization

In each frame, a large number of target candidates are generated from the fully cyclic shifts of the latest target region (a base image), in the same way as the dense sampling method does. Considering the scale variation of the target over frames, the candidates with different scales are required. We use a number of base images with different scales to generate these candidates. Specifically, we employ a scale pool $\mathcal{S}\{s_1, s_2, \ldots, s_m\}$ containing m scales. Correspondingly, m sets of target candidates are generated from the m base images $\mathbf{x}_{1:m}$, where the candidates in the jth set yield the scale s_j.

Given a target candidate $\mathbf{x}'_s$ with a scale s, the regression value of this candidate is computed in the frequency domain from

$$\hat{\mathbf{f}}\left(\mathbf{x}'_s\right) = \hat{\mathbf{k}}'_s \odot \hat{\boldsymbol{\alpha}}, \tag{15}$$

where $\hat{\mathbf{k}}'_s = \varphi^T(\mathbf{x})\, \varphi\left(\mathbf{x}'_s\right)$ denotes the kernel correlation of the latest target region $\mathbf{x}$ and the candidate region $\mathbf{x}'_s$. The candidate with the largest regression value (filter response) f over all scales $s \in \mathcal{S}$ is determined as the current target. Meanwhile, the current scale of the target is estimated by

$$s^* = \arg \max_{s \in \mathcal{S}, \mathbf{x}'_s \in \mathcal{X}} \hat{\mathbf{f}}\left(\mathbf{x}'_s\right) \tag{16}$$

over all the candidates $\mathcal{X}$. Note that the above operation in Eq. (15) is in fact a spatial correlation filtering over $\mathbf{k}'_s$ using the filter $\boldsymbol{\alpha}$ in the frequency domain. The frequency representation can significantly speed up the correlation.

3.4 Explanation on the Loss Functions

In this work, the sparsity based loss functions are leveraged to promote the robustness of the correlation filter learning. The anisotropy of the response relies on the regularization implemented by the sparsity related norms. It is controlled by the weight parameter of the regularization. When the weight drops to zero, the response will be exactly isotropic. It indicates that the trackers using isotropic response like the KCF tracker are a special case of the proposed tracking model, i.e., the case where the regularization is eliminated by a zero weight. In general cases, we impose the regularization on the Gaussian shaped (isotropic) response, under which the loss, denoted by $\mathbf{e}$, can be arbitrary shapes, leading to an anisotropic response.

Under the ℓ_1-loss, a standard sparsity constraint on the errors $\mathbf{e}$ are induced for the correlation filter learning. It indicates that the errors $\mathbf{e}$ is allowed to be arbitrarily large but sparse. As a result, the learned filter $\boldsymbol{\alpha}$ may ignore the significant changes in the target appearance, e.g., in the presence of occlusion. Under the $\ell_1\ell_2$-loss, an additional ℓ_2-loss is appended to the ℓ_1-loss, leading to an elastic net constraint on the errors $\mathbf{e}$. Note that because the ℓ_2-loss always results in dense and small errors, it is very effective to the globally uniform appearance changes, e.g., those caused by illumination variation. For this reason, the $\ell_1\ell_2$-loss can effectively handle both the abrupt (through large but sparse errors compensation) and the slow (through small and dense errors compensation) appearance changes. Under the $\ell_{2,1}$-loss, the relationship of the errors between the candidates is exploited, through which the appearance changes in the local image patches can be well handled. Note that the $\ell_{2,1}$-loss imposes a structured sparsity on the filter response. The $\ell_{2,1}$-norm leads to a column-wise sparsity on the response. In this work, we use a symmetric zero operation following the $\ell_{2,1}$-norm to ensure the symmetric property of the response. Note that, although the response is symmetric, it is still anisotropic because the zeros can occur in any column. For this reason, the $\ell_{2,1}$-loss is effective to the complex appearance changes, e.g., in the scene where both illumination variations and occlusions occur simultaneously.

As addressed above, large errors can be tolerated under the three loss functions during the correlation filter learn-

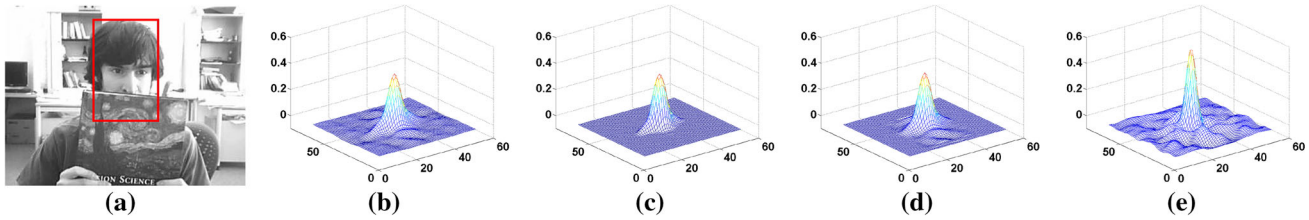


Fig. 4 The anisotropy of the filter response exploited in the frame shown in (**a**) with respect to the ℓ_1-loss (**b**), the $\ell_1\ell_2$-loss (**c**), the $\ell_{2,1}$-loss (**d**), and the ℓ_2-loss (**e**)

Algorithm 1: Tracking algorithm

Input: the initial motion state of the target.
Output: the target $\mathbf{y}_t$ in the tth frame.

Initialize the base image $\mathbf{x}_1$ in the first frame.
Learn the correlation filter $\boldsymbol{\alpha}_1$ over $\mathbf{x}_1$ using Algorithm 2.
for *each frame* $t = 2 : N$ **do**
  Generate target candidate regions $\mathbf{x}_s$ at scale $s \in \mathcal{S}$ according to the location of $\mathbf{y}_{t-1}$.
  Compute the filter responses over the candidate regions by using Eq. (15).
  Localize the tth target $\mathbf{y}_t$ by the candidate with the maximum filter response over all scales.
  Learn a new correlation filter $\boldsymbol{\alpha}$ over the target region $\mathbf{y}_t$ using Algorithm 2.
  Update the base image and the filter by using Eq. (14).
end

Algorithm 2: Optimization algorithm

Input: training samples $\{\mathbf{x}_i\}$.
Output: the correlation filter $\boldsymbol{\alpha}$ and residual error $\mathbf{e}$.

Generate a isotropic (Gaussian shaped) filter response map $\mathbf{y}$.
Construct the kernel matrix $\mathbf{K}$ from Eq. (8).
Initialize the error $\mathbf{e} = \mathbf{0}$.
Let $max\_iter = 1000$.
for *iter=1:max_iter* **do**
  Transform the error $\mathbf{e}$ to the Fourier domain.
  Compute the correlation filter $\boldsymbol{\alpha}$ from Eq. (9) in the Fourier domain.
  Compute the residual $\hat{\mathbf{y}} - \hat{\boldsymbol{\alpha}} \odot \hat{\mathbf{k}}_1$ and transform it back to the image domain.
  Compute the error $\mathbf{e}$ from Eq. (10), (12), or (13) for the respective loss functions.
  if *converged* **then**
    break.
  end
end

ing, leading to improved robustness. From Eq. (3), it can be found that the difference e_i between the filter response $f(\mathbf{x}_i)$ and the Gaussian shaped response y_i can be large in some dimensions but sparse overall, leading to an anisotropic ground truth labels (expected filter response) $y_i - e_i$. Such an anisotropy essentially facilitates tracking. The anisotropic filter response adaptively learned under the three loss functions in a representative frame are illustrated in Fig. 4. It can be seen that the three loss functions lead to relatively larger filter responses in the horizontal direction. It suggests that the three proposed loss functions punish the regions vertically with the occlusion along that direction, since the distractive object (the book) moves vertically. In contrast, the squared loss is unable to reveal such a structure in the response map.

3.5 Implementation Details

The training samples $\mathbf{X}$ are collected in each frame by cyclically shifting the base image centering at the latest target region with a spatially expanded size of 1.5 times of the target. To mitigate the discontinuity from the cyclic shifts, a cosine window is applied to the base image. Referring to the KCF tracker, histogram of orientation gradient (HOG) feature is extracted from the base image to represent the samples. The cell size is set to 4, and 9 orientations are employed for the HOG feature computation. A Gaussian kernel with a variation of 0.5 is employed to transform the samples to a non-linear high-dimensional feature space. In the tracking phase, the above operations are also applied to the target candidates in each frame. The candidates are generated from the

fully cyclic shifts of a base image in the current frame, which centers at the target region in the last frame. The isotropic filter response map $\mathbf{y}$, i.e., the regression labels in Eq. (3), are set as a Gaussian shaped map with the covariance matrix $diag\left\{\sigma^2, \sigma^2\right\}$, where $\sigma = \frac{\rho}{s_{cell}}\sqrt{h \times w}$, ρ denotes the spatial bandwidth factor that is proportional to the target size, s_{cell} denotes the cell size of the HOG features, and h and w denotes the height and the width of the target region. In this work, the spatial bandwidth factor ρ is set to 0.1 according to the suggestions of Henriques et al. (2015) and the cell size s_{cell} is set to 4 as presented above. As recommended in Henriques et al. (2015), the parameter λ in Eq. (4) is set to 10^{-4}, and the parameter η in Eq. (14) is set to 0.02. Another parameter τ in Eq. (4) is set to be equal to λ in the experiments. We employ 7 scale coefficients to build the scale pool, i.e., $\mathcal{S} = \{0.95, 0.97, 0.99, 1, 1.01, 1.03, 1.05\}$. For a clear overview of the proposed approach, the tracking algorithm is summarized in Algorithm 1. The proposed correlation filter learning method is depicted in Algorithm 2. The converge criterion for this iterative algorithm is set as the one that the difference of the values of Eq. (4) in the dual space (i.e., $\mathbf{w} = \sum_i \alpha_i \varphi(\mathbf{x}_i)$) between two consecutive iterations are less than a very small number, e.g., 10^{-5}.

4 Experiments

Three trackers are implemented, corresponding to the ℓ_1-, $\ell_1\ell_2$-, and $\ell_{2,1}$-loss functions, denoted by Ours_{S} (sparsity), Ours_{EN} (elastic net), and Ours_{GS} (group sparsity), respectively, for the experimental evaluations.

4.1 Benchmark Data and Baseline Trackers

The three proposed trackers are evaluated on two sets of benchmark databases: Object Tracking Benchmark (OTB) (Wu et al. 2013, 2015) and Visual Object Tracking (VOT) (Kristan et al. 2016). These two sets have many differences in various aspects, such as video data, ground truth labeling, evaluation protocol, and evaluation criterion. The former adopts an one pass evaluation (OPE) protocol, whereas the later employs an reset based evaluation strategy as well as an OPE method.

In this work, we use three most recent VOT benchmarks for performance evaluations, the VOT 2015, 2016, and 2017 challenge benchmarks. Every VOT benchmark has 60 video sequences. The VOT 2015 and 2016 benchmarks have exactly the same video sequences but different ground truth labels. We therefore include the evaluations on both VOT 2015 and 2016 benchmarks. The OTB database contains two benchmarks, the OTB 2013 and the OTB 2015. The former, containing 50 video sequences, is a subsets of the latter one that contains 100 video sequences. For this reason, we adopt the larger benchmark, i.e., the OTB 100 benchmark, to evaluate our trackers.

We compare the three proposed trackers to most state-of-the-art visual trackers. On each of the three VOT benchmarks, we employ the top 20 trackers as the baselines. The source codes and the pre-computed tracking results of these baseline trackers are obtained publicly from the VOT website. On the OTB 2015 benchmark, we employ 10 state-of-the-art correlation filtering based trackers [PSCF (Sui et al. 2018b), RCF (Sui et al. 2016b), KCF_AT (Bibi et al. 2016), SRDCF (Danelljan et al. 2015), HCFT (Ma et al. 2015a), SAMF (Li and Zhu 2014), DSST (Danelljan et al. 2014a), KCF (Henriques et al. 2015), CN (Danelljan et al. 2014b), and CSK (Henriques et al. 2012)] as the baseline methods. The source codes of these 10 baseline trackers are publicly provided by the respect authors.

Two criteria for tracking performance evaluations are used on the OTB 2015 benchmark: the precision plot and the success rate plot. We report the precision score at a threshold 20, as most commonly used in the literature, and the area under curve (AUC) score for the success rate plot. On the VOT benchmarks, we adopt several more criteria for evaluations, including expected average overlap (EAO), accuracy–robustness score, and OPE based AUC score. The reset based protocol is employed for the evaluations on the VOT benchmarks. A tracking failure is detected if the overlap between the tracking and the ground truth boxes comes to 0. The tracker is recovered in the case of failure by initializing it by the ground truth after 5 frames. The failure times is reflected by the robustness criterion.

4.2 Evaluations on VOT Benchmarks

On the VOT benchmarks, both the reset based and the OPE protocols are used for the performance evaluations. We employ the accuracy–robustness and the expected average overlap (EAO) for the reset based evaluations, and the area under curve (AUC) of the success rate plot for the OPE strategy. The accuracy is revealed by averaging the overlap rates on all frames, where the overlap rate is defined as $\frac{A_t \bigcap A_g}{A_t \bigcup A_g}$ with A_t and A_g the areas of the tracking and the ground truth boxes, respectively. The robustness is measured by counting the failure times where a failure is encountered when the overlap rate is 0. The robustness score is computed from $\exp\left(-\frac{Sf}{N}\right)$ where f times failures are detected on a N frames length video sequence, and S is a coefficient, which is set to $S = 100$ for all the experiments in this work. The EAO is a key criterion for the VOT evaluation, which indicates the synthetic performance of a visual tracker. Readers can reach its definition and detailed description in the VOT paper (Kristan et al. 2016) and the VOT website. In this work, we evaluate the EAO score between the frame length 108 and 371 fol-

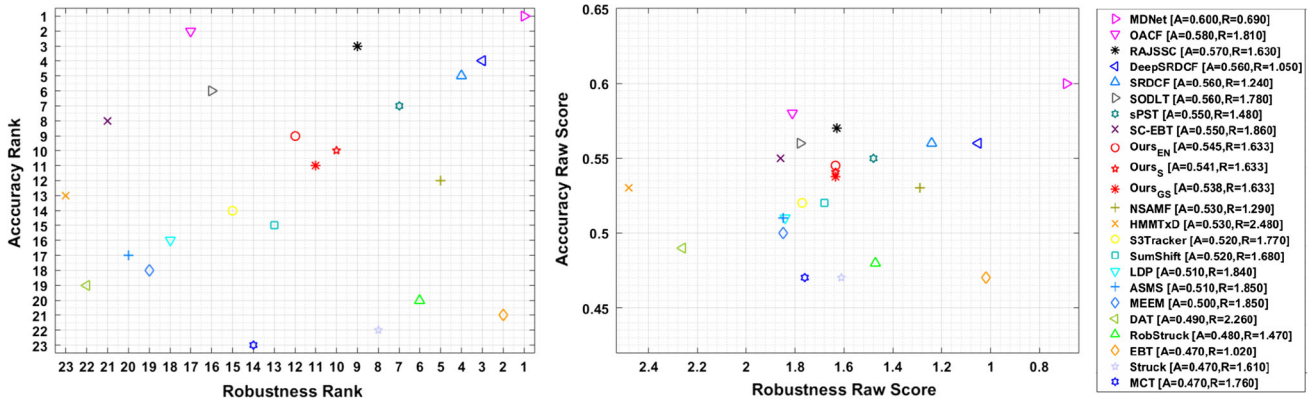


Fig. 5 Tracking performance of the proposed trackers on the VOT 2015 benchmark against the top 20 trackers. (left) Accuracy–robustness raw score plot. (right) Accuracy–robustness rank plot. In the legend of the plots, the accuracy and the robustness raw scores are shown after the names of the trackers

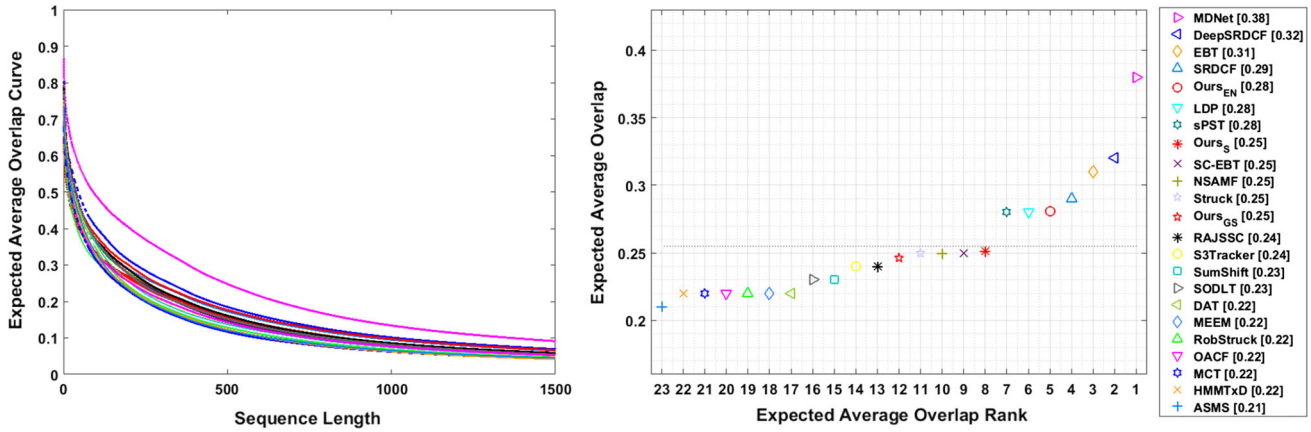


Fig. 6 Tracking performance of the proposed trackers on the VOT 2015 benchmark against the top 20 trackers. (left) EAO curve plot. (right) EAO score plot. In the legend of the plots, the EAO scores are shown after the names of the trackers

lowing the recommendation by the VOT committee. For the OPE protocol, we run the trackers without any resets and then report their AUC scores. The larger the AUC score, the better the tracker performs.

The evaluation results on the VOT 2015 benchmark are shown in Figs. 5 and 6. We compare the three proposed trackers to the top 20 trackers from the VOT 2015 challenge. The accuracy–robustness ranks and raw scores are shown in Fig. 5. It can be seen that the three proposed trackers perform competitively against the 20 state-of-the-art trackers. They rank 9, 10, and 11 in terms of accuracy, and 10, 11, and 12 in terms of robustness. The EAO curve and the EAO scores are shown in Fig. 6. The EAO curves of the 23 trackers are also plotted in Fig. 6. The three proposed trackers rank 5, 8, and 12 in terms of the EAO score. It can be seen that all the three proposed trackers perform over the average performance (above the horizontal dashed line in the right plot) among the 23 trackers in terms of EAO.

The tracking results on the VOT 2016 benchmark are shown in Figs. 7, 8 and 9. The top 20 trackers from the VOT 2016 challenge are employed as the baseline trackers. It can be seen from the results that the three proposed tracker performs competitively against the 20 state-of-the-art trackers. As shown in Fig. 7, the three proposed trackers rank 8, 10, 12 in terms of robustness score, and 11, 14, 15 in terms of accuracy score. They rank 14, 16, and 19 among all the 23 trackers in terms of the EAO score, as shown in Fig. 8. The EAO curves of the 23 trackers are also plotted in Fig. 8. Under the OPE protocol, the three proposed trackers rank 13, 15, and 16 among the 23 trackers in terms of the AUC score, as shown in Fig. 9. The success rate plot for the OPE of the 23 trackers are also shown in Fig. 9.

Figures 10, 11 and 12 show the evaluation results on the VOT 2017 benchmark. The top 20 trackers from the VOT 2017 challenge are adopted as the baseline trackers on this benchmark. We can see that the three proposed trackers obtain comparable performance on the VOT 2017 benchmark against the 20 state-of-the-art trackers. As shown in Fig. 10, the three proposed trackers rank 2, 5, and 7 in terms of accuracy score, and 15, 16, and 18 in terms of robustness

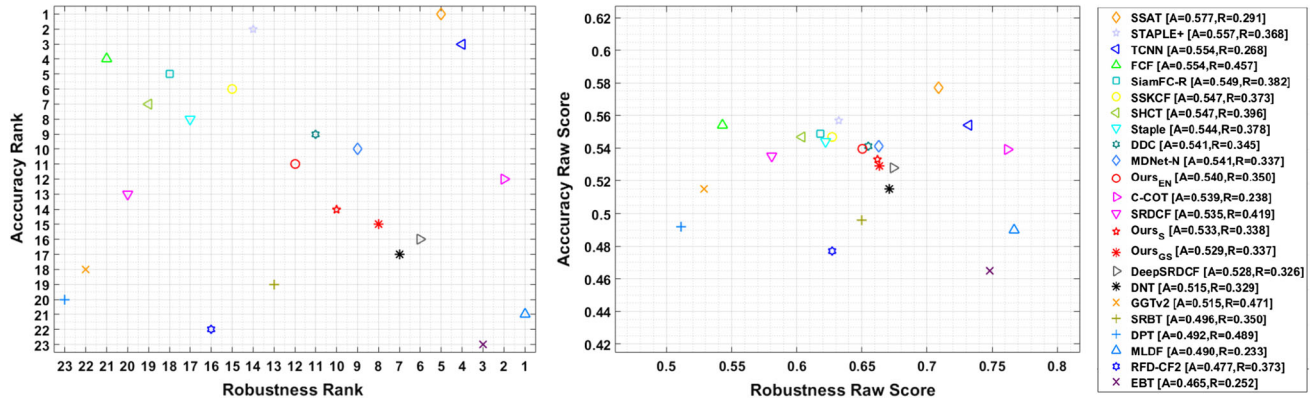


Fig. 7 Tracking performance of the proposed trackers on the VOT 2016 benchmark against the top 20 trackers. (left) Accuracy–robustness raw score plot. (right) Accuracy–robustness rank plot. In the legend of the plots, the accuracy and the robustness raw scores are shown after the names of the trackers

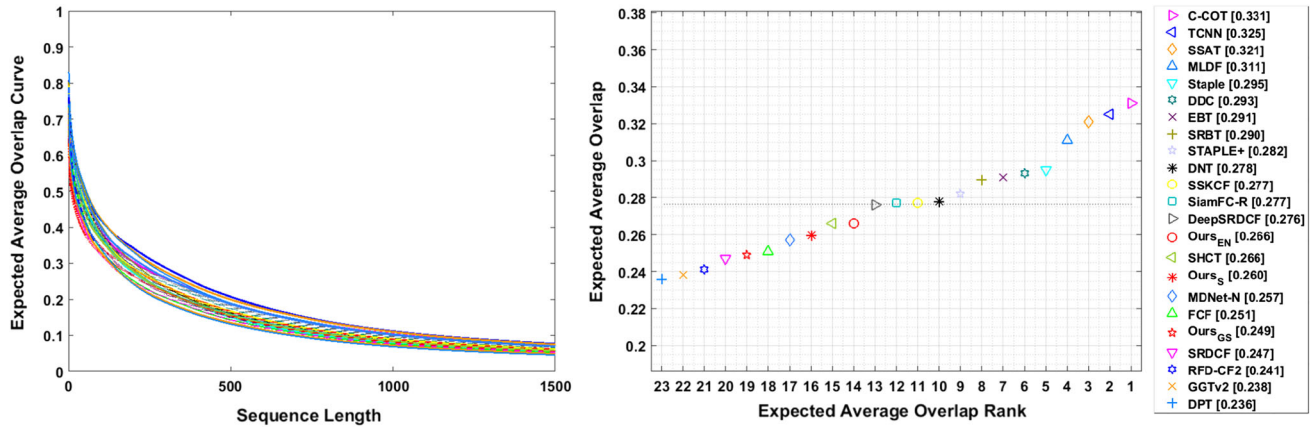


Fig. 8 Tracking performance of the proposed trackers on the VOT 2016 benchmark against the top 20 trackers. (left) EAO curve plot. (right) EAO score plot. In the legend of the plots, the EAO scores are shown after the names of the trackers

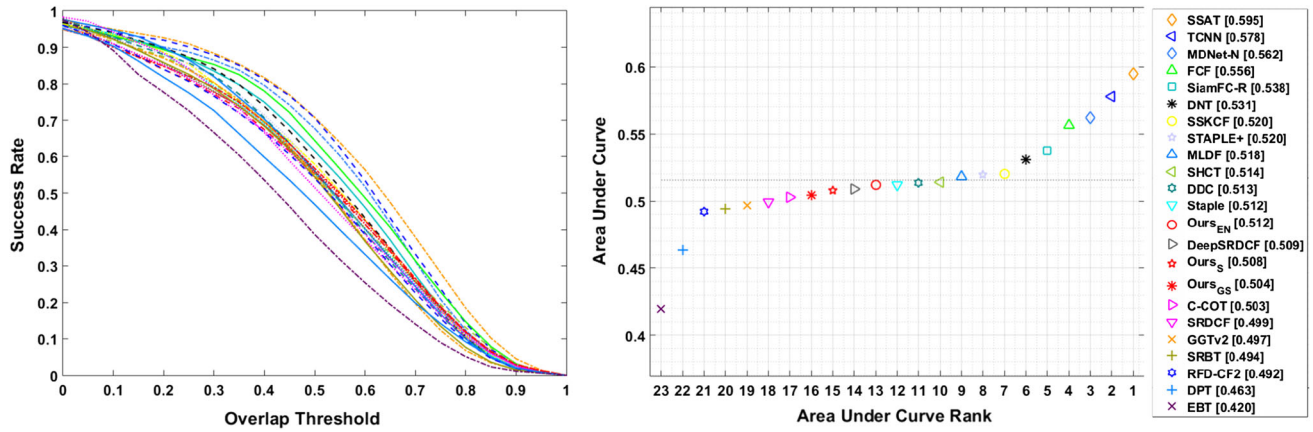


Fig. 9 One pass evaluation (OPE) of the proposed trackers on the VOT 2016 benchmark against the top 20 trackers. (left) Success rate plot. (right) AUC score plot. In the legend of the plots, the AUC scores are shown after the names of the trackers

score. They rank 15, 20, and 22 in terms of EAO, as shown in Fig. 11. The EAO curves of the 23 trackers are also plotted in Fig. 11. Under the OPE protocol, the three proposed trackers rank 10, 15, and 17 in terms of AUC score, as shown in Fig. 12. The success rate plots for the OPE of the 23 trackers are also shown in Fig. 12.

In summary, the three proposed trackers performs competitively on the three latest VOT benchmarks against the

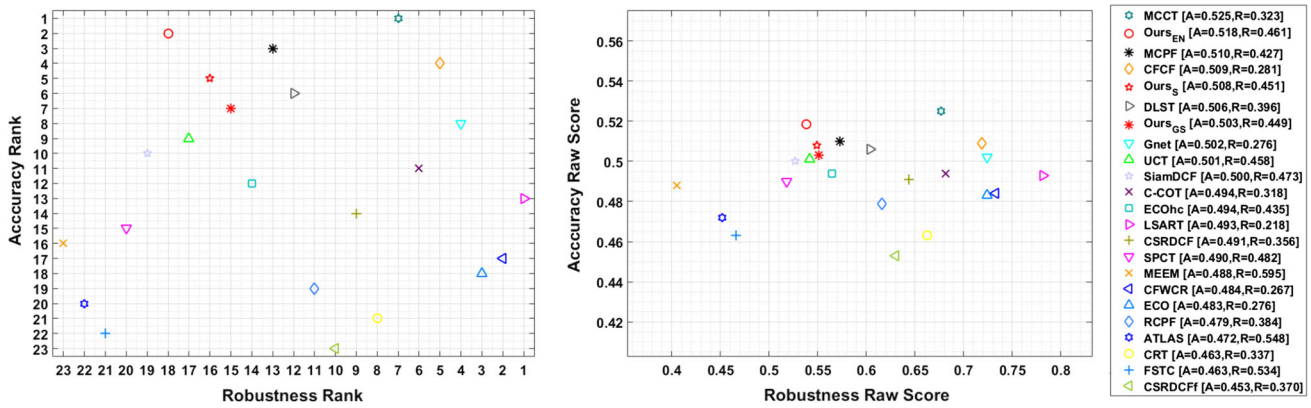


Fig. 10 Tracking performance of the proposed trackers on the VOT 2017 benchmark against the top 20 trackers. (left) Accuracy–robustness raw score plot. (right) Accuracy–robustness rank plot. In the legend of the plots, the accuracy and the robustness raw scores are shown after the names of the trackers

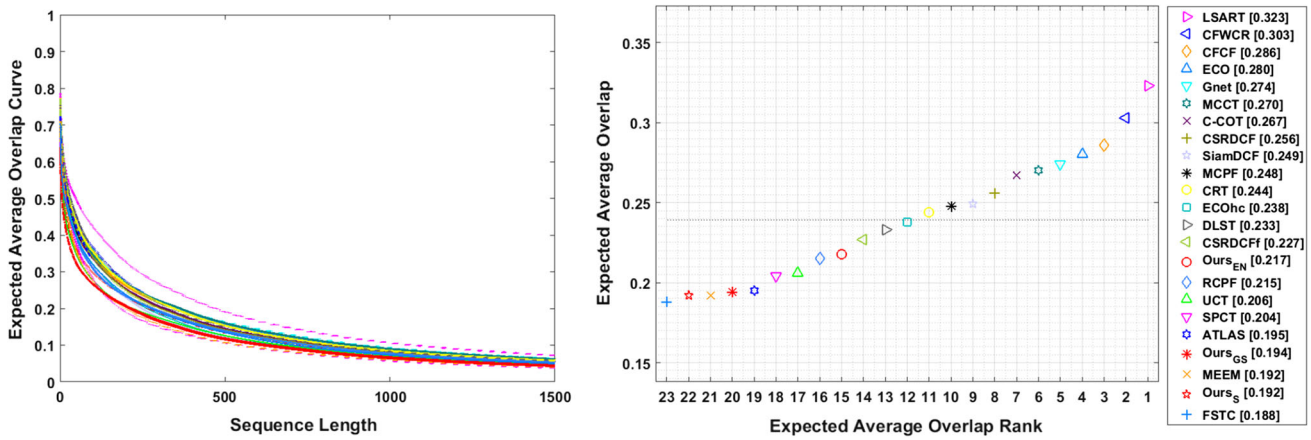


Fig. 11 Tracking performance of the proposed trackers on the VOT 2017 benchmark against the top 20 trackers. (left) EAO curve plot. (right) EAO score plot. In the legend of the plots, the EAO scores are shown after the names of the trackers

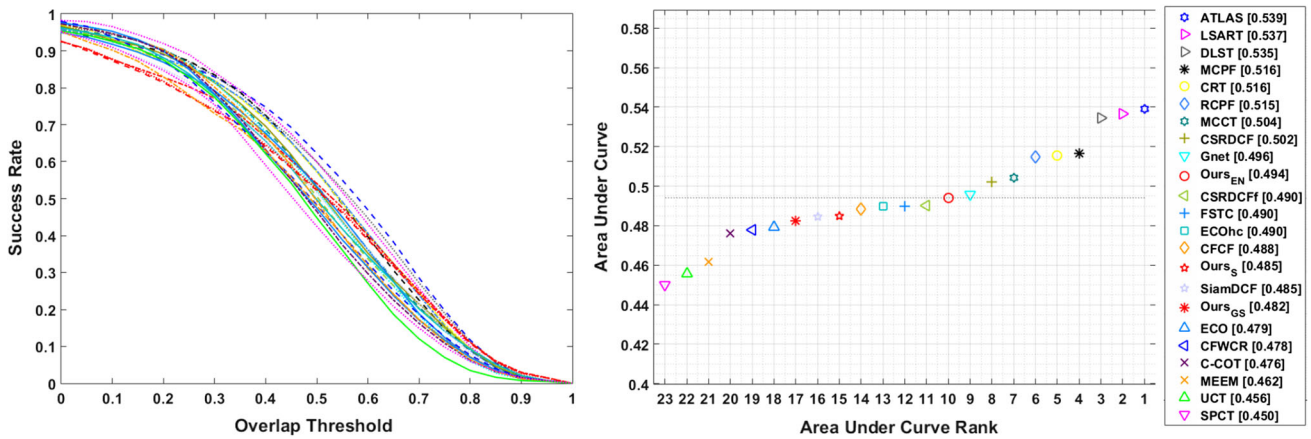


Fig. 12 One pass evaluation (OPE) of the proposed trackers on the VOT 2017 benchmark against the top 20 trackers. (left) Success rate plot. (right) AUC score plot. In the legend of the plots, the AUC scores are shown after the names of the trackers

top 20 trackers from the respect challenges. These baseline trackers utilize the most advanced techniques to address tracking, such as deep learning, convolutional feature maps, and complex tracking model. However, the proposed tracking algorithms rely on the basic correlation filtering framework and leverage the manually designed plain HOG features.

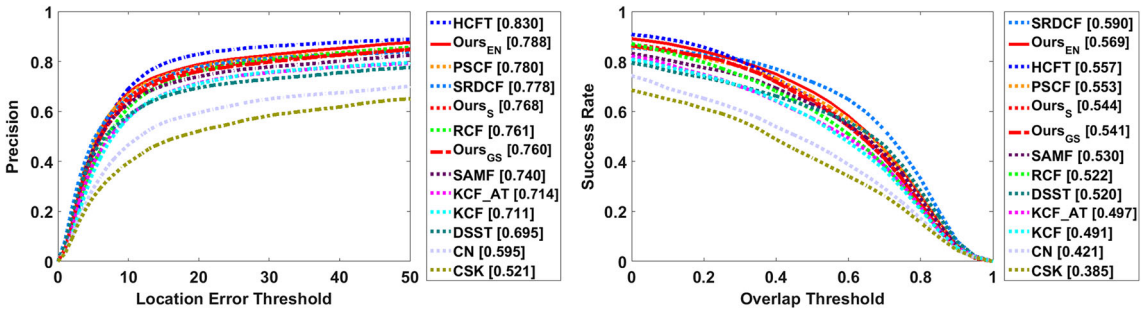


Fig. 13 Tracking performance of the proposed and other 10 popular correlation filtering based trackers on all the 100 video sequences of the OTB 2015 benchmark

Table 1 Tracking performance on the 100 video sequences of the OTB 2015 benchmark

Tracker	Precision		Success rate	
	$\rho = 20$	Average	$\phi = 0.5$	Average
Ours_{EN}	*0.788*	*0.726*	*0.680*	*0.569*
Ours_{S}	0.768	0.705	0.647	0.544
Ours_{GS}	0.760	0.699	0.640	0.541
Correlation filtering based trackers				
PSCF (Sui et al. 2018b)	0.780	0.717	0.659	0.553
RCF (Sui et al. 2016b)	0.761	0.694	0.607	0.522
KCF_AT (Bibi et al. 2016)	0.714	0.653	0.573	0.497
HCFT (Ma et al. 2015a)	**0.830**	**0.746**	0.648	0.557
SRDCF (Danelljan et al. 2015)	0.778	0.714	**0.718**	**0.590**
KCF (Henriques et al. 2015)	0.711	0.650	0.572	0.491
SAMF (Li and Zhu 2014)	0.740	0.684	0.629	0.530
DSST (Danelljan et al. 2014a)	0.695	0.644	0.614	0.520
CN (Danelljan et al. 2014b)	0.595	0.553	0.472	0.421
CSK (Henriques et al. 2012)	0.521	0.495	0.415	0.385

ρ and ϕ denote location error threshold and overlap threshold, respectively. The best and the second best results are marked in bold-face and italic fonts, respectively

Although we do not employ the features learened from large datasets, like deep features, we obtain the comparable tracking performance on the three challenging VOT benchmarks. Specifically, according to the results of the respective VOT challenges, the proposed trackers perform in the top 8% on the VOT 2015, top 20% on the VOT 2016, and top 30% on the VOT 2017, respectively. We note that the above ranks are obtained by comparing to the visual trackers submitted to the respective VOT challenges. These baseline trackers reflect the best performance at the time when the challenges held. The state-of-the-art boundary on these VOT benchmarks is continuously refreshed and getting lower. This work aims at improving the correlation filter learning for visual tracking. The three proposed trackers outperform many state-of-the-art correlation filtering based visual trackers on these three latest VOT benchmarks. It is demonstrated that the proposed approach leads to improved tracking results through exploiting the anisotropy of the correlation filter learning.

4.3 Evaluations on OTB Benchmark

We evaluate the tracking performance of the proposed trackers on the OTB 2015 benchmark. This benchmark uses an OPE protocol to evaluate visual trackers. It contains 100 fully labeled video sequences. We evaluate the proposed trackers within the correlation filtering paradigm. other 10 state-of-the-art correlation filtering based visual trackers are employed as the baseline approaches. The precision plots and the success rate plots are shown in Fig. 13, respectively, which are obtained by the proposed trackers and the 10 correlation filtering based trackers on all the 100 video sequences of the OTB 2015 benchmark. The detailed quantitative evaluation results are shown in Table 1. It can be seen that the proposed tracker, Ours_{EN}, performs the second best in terms of both precision and success rate. The other two proposed trackers also obtain comparable performance on this benchmark.

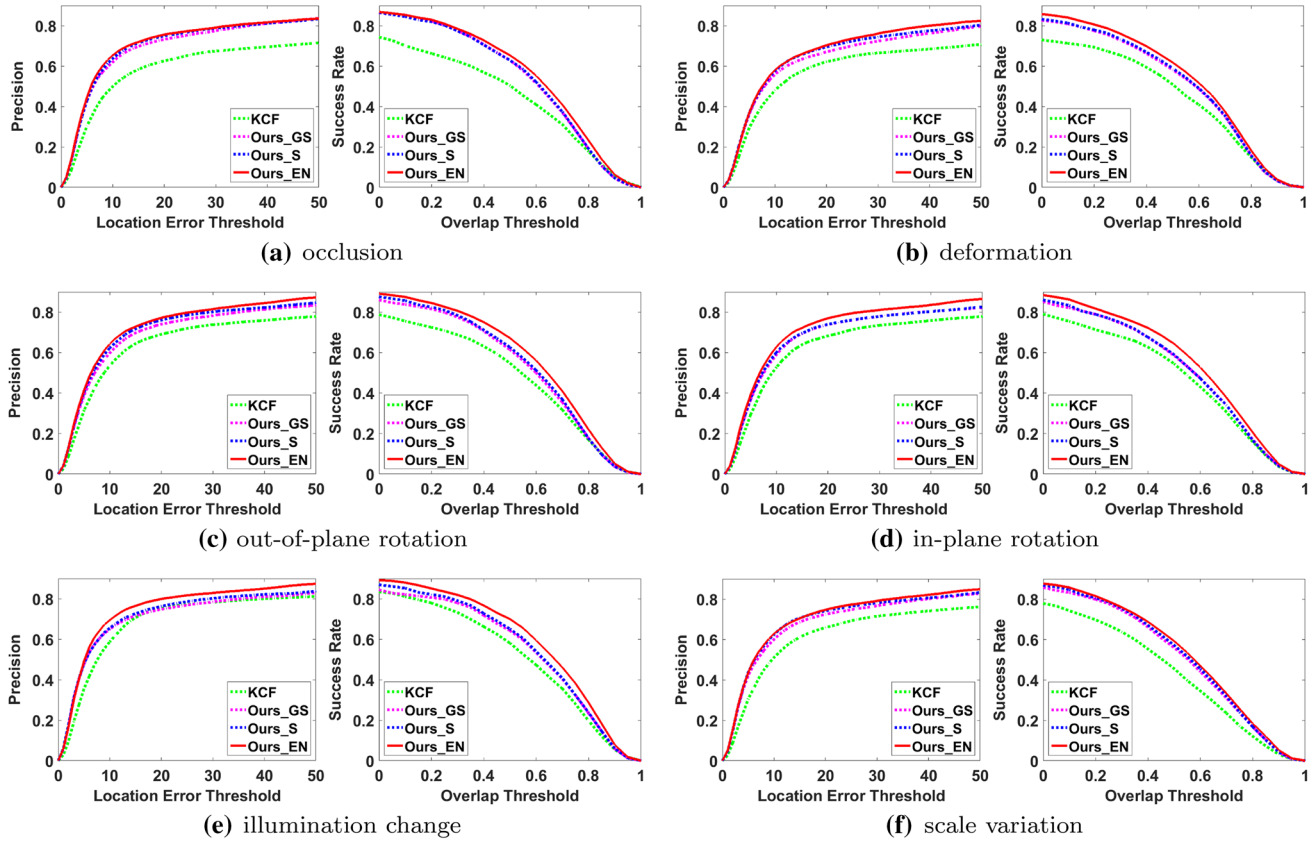


Fig. 14 Tracking performance of the three proposed trackers and the KCF tracker in various challenging situations on the OTB 2015 benchmark

4.4 Analysis of the Proposed Approach

4.4.1 Performance in Various Situations

We investigate the tracking performance in various challenging situations in order to comprehensively evaluate the proposed trackers. To analyze the effectiveness of the three (ℓ_1-, $\ell_1\ell_2$- and $\ell_{2,1}$-) loss functions, the KCF tracker (ℓ_2-loss), that leverages the ℓ_2-loss, is employed as the baseline method. Figures 14, 15 and 16 show the evaluation results in the 11 challenging situations, respectively, on the OTB 2015 benchmarks.

Occlusion Due to the influence of occlusions, the target appearance might change abruptly. In this case, it can be seen that the three proposed trackers significantly outperform the KCF tracker. This is attributed to that the sparsity based loss functions used by the proposed trackers can compensate the occlusions by assigning large errors in the correlation filter learning, i.e., the occlusions are treated as the outliers in the training samples. Thus, these sparsity based loss functions are more robust to the abrupt appearance changes than the squared loss, leading to more reliable filter response.

Deformation In this case, the target deforms non-rigidly due to some complicated factors, like pose change, motion, and viewpoint variation. The local appearance changes often occur in this case. Note that the sparsity based loss functions perform more robustly in the presence of local appearance change, while the squared loss is more effective to handle globally uniform appearance change. As a result, it can be seen from the results that the proposed tracker, Ours_{EN}, obtains the best performance in this case, since the significant local changes are well dealt with by the sparsity constraint (the third term in Eq. (4)) and the small changes are handled by the squared approximation (the first term in Eq. (4)).

In-plane/Out-of-Plane Rotation During tracking, the target motion and/or viewpoint change often cause the in-plane/out-of-plane rotation in the target appearance. It can be seen that, in this case, the three proposed trackers outperform the KCF tracker. This is attributed to the improved robustness from the sparsity related loss functions.

Illumination Change In the case that the lighting condition of the tracking scene varies, the target appearance suffers from a globally uniform change, i.e., the illumination

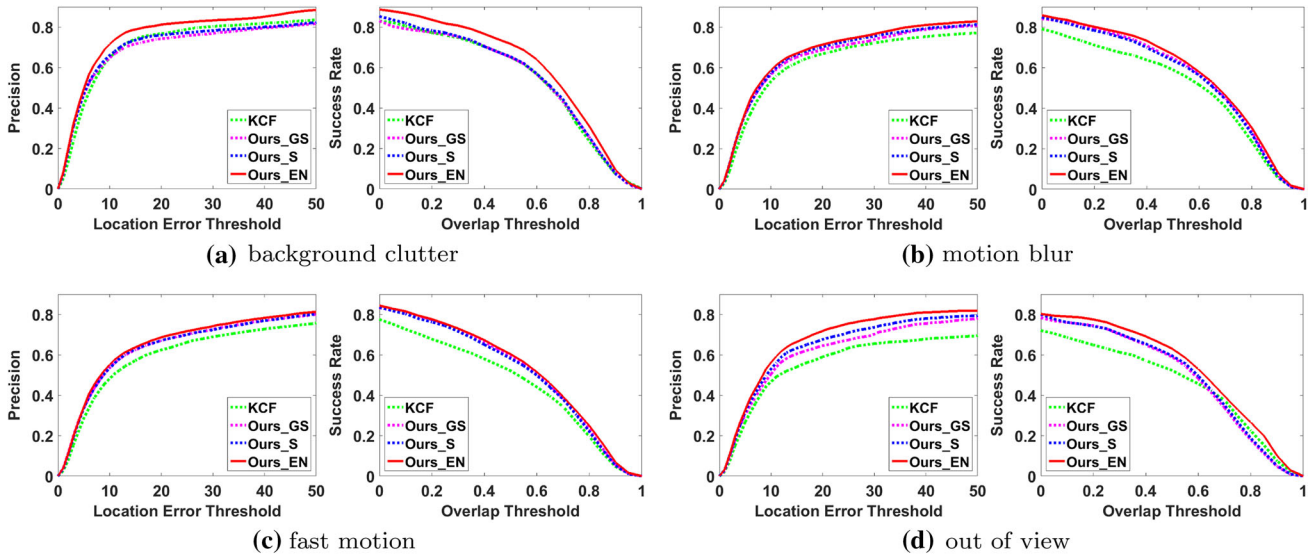


Fig. 15 Tracking performance of the three proposed trackers and the KCF tracker in various challenging situations on the OTB 2015 benchmarks

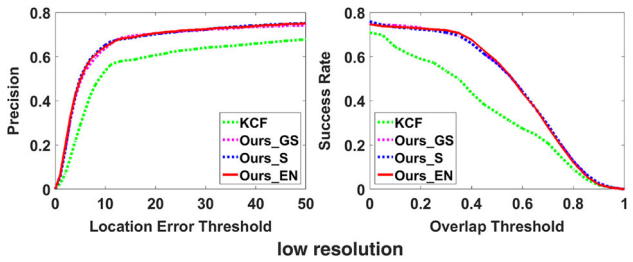


Fig. 16 Tracking performance of the three proposed trackers and the KCF tracker in various challenging situations on the OTB 2015 benchmarks

change influences the entire target appearance. Note that it has demonstrated that the squared loss is very effective to handle the globally uniform change. For this reason, the proposed approach does not make significant improvement over the KCF tracker.

Scale Variation Due to some complicated factors, such as the motion of the target and/or the camera, and viewpoint change, the scale of the target appearance always varies over frames. Visual tracker needs to adjust the size of the target window appropriately; otherwise, tracking failure may happen because 1) only partial information of target is acquired when the size of the target grows; and 2) more background information is unexpectedly acquired when the size of the target decreases. Considering the computational efficiency, the KCF tracker is unable to incorporate with a scale estimation for the target. The proposed trackers employ multiple kernels to conduct the correlation to online estimate the target scale. It is evident from the results that the proposed trackers significantly outperform the KCF tracker in this case.

Background Clutter The cluttered background often distracts the tracker, easily leading to tracking failure. A robust tracker is required to have good discriminative capability to distinguish the target from the surrounding background. The correlation filtering framework is derived from the tracking-by-detection approach. It is thus effective to deal with this situation. As a result, the difference of the performance between the three proposed trackers and the KCF tracker is not that significant.

Motion Blur and Fast Motion During tracking, due to the motion of the target and/or the camera, the appearances of the target and the background may be blurred significantly. Furthermore, it is challenging for the motion model in the case of fast motion. Because the sparsity based loss functions are leveraged in the proposed approach, the motion blur can be handled effectively. Meanwhile, since the motion model of the correlation filtering framework is implemented as a dense sampling around the possible target location, the fast motion can be handled efficiently.

Out of View The target may move out of view during tracking. This is a big challenge for visual trackers. In this case, a good tracker needs to remember the target appearance and have the capability to re-acquire the target as soon as the target reappears in the view. In the proposed approach, an incremental strategy, as presented in Eq. (14), is adopted to update correlation filter in each frame, in order to prevent the newly learned correlation filter from the significant changes. For this reason, the proposed trackers can remember the target appearance in a short period in the case that the target moves out of view.

Low Resolution The video sequences may be captured with low resolution. This needs good robustness for visual trackers. The proposed approach leverages different robust loss functions to promote the robustness of the correlation filter learning. As a result, the performance in the case of low resolution is greatly improved on the two benchmarks.

In summary, the three proposed trackers promote the robustness of the correlation filter learning, leading to significantly improved tracking performance on the two benchmarks. Ours$_{EN}$ combines the ℓ_1- and ℓ_2-loss functions and obtains the best results. The ℓ_1-loss function allows large fitting errors during the correlation filter learning. It is thus robust to abrupt appearance changes, e.g., caused by occlusion and out-of-plane rotation. ℓ_2-loss function groups the fitting errors and produces small and dense errors. It is thus effective to small appearance changes, e.g., caused by illumination change and deformation. $\ell_{2,1}$-loss function exploits the relationship between the fitting errors, resulting in group structured errors. It is thus efficient to complicated appearance changes, e.g., caused by cluttered background. With the properties of the three loss functions, combining the above experimental analysis, we make the following suggestions on the use of the three proposed trackers: in the cases of occlusion and out-of-plane rotation, we recommend Ours$_{EN}$ and Ours$_S$; in the cases of illumination change and deformation, we recommend Ours$_{EN}$ for high accuracy and KCF for high running speed; in the case of background clutter, we recommend Ours$_{GS}$ to handle various complicated appearance; and in the case of low resolution, we recommend the three proposed trackers for robustness consideration.

4.4.2 Peak Sensitivity of the Filter Response

The proposed approach leverages different loss functions to promote the robustness of the correlation filter learning, leading to different anisotropy of the filter responses. In this section, we explain how the loss functions essentially influence the tracking performance via the anisotropic filter responses.

Intuitively, the peak values of the online learned correlation filters should be as stable as possible in consecutive frames in various challenging situations because it is responsible for the accuracy of the target localization in each frame. To this end, we investigate the peak values of the correlation filter on three representative video sequences, including the cases of occlusion, illumination variation, and non-rigid deformation, respectively. Since other complicated challenges are also included on the video sequences of *faceocc2* and *david*, only the first 200 and 100 frames are selected for the investigations, respectively. For the convenience of the interpretation, we employ the KCF tracker (Henriques et al. 2015) as a baseline tracker in the analysis. Note that the KCF tracker is unable to handle scale adaptation during tracking. To make a fair comparison, we include only one scale coefficient with the original scale in the scale pool $\mathcal{S} = \{1\}$ to disable the scale adaptation for the proposed approach. As a result, the investigation aims at the improvement from the loss functions rather than the incorporation with the scale adaptation scheme.

Figure 17 plots the peak values of the learned correlation filter and the filter responses, respectively, obtained by the KCF tracker and the proposed trackers in each frame of the three experimental video sequences. An interesting observation is obtained from the plots: the drastic changes in the peak values just correspond to the significant changes in the target appearance in the corresponding frames. It indicates that, if the peak values are sensitive (i.e., have drastic fluctuation) in successive frames, the corresponding filter responses will be unstable, leading to a lower accuracy of the target location.

As shown in Fig. 17a, the peak values of the filter with respect to the $\ell_1\ell_2$-loss (Ours$_{\mathrm{EN}}$) varies the most slowly over frames in the case of occlusion. The ℓ_1-loss (Ours$_{\mathrm{S}}$) also has relatively smoother curve of the peak values than the $\ell_{2,1}$-(Ours$_{\mathrm{GS}}$) and the ℓ_2-loss (the KCF tracker). It is evident that the investigation results on the peak sensitivity in the successive frames are consistent with the tracking performance evaluations shown in Fig. 14a.

It can be seen from Fig. 17b that, in the case of illumination variation, the filter peaks obtained by the four trackers have the similar sensitivity values in the successive frames. It is also verified from the evaluations shown in Fig. 14e where the four trackers obtain similar tracking performance.

As shown in Fig. 17c, in the presence of non-rigid deformation, the proposed tracker, Ours$_{\mathrm{EN}}$, performs the most stably in terms of the filter peak values, and obtains the best tracking performance. This result is also consistent with the one shown in Fig. 14b. In contrast, the filter peaks obtained by the KCF tracker are very sensitive in the successive frames, leading to the interior tracking performance, which is consistent with the performance evaluations reported in Fig. 14b.

To characterize the above observation quantitatively, a metric is required to measure the sensitivity (drastic fluctuation) of the learned correlation filter. Note that it has been discussed in Bolme et al. (2010) that a good correlation filter often has a large peak-to-sidelobe ratio (PSR) value from a signal processing point of view. However, the PSR only takes account for the performance of the learned filter in a single frame, while a measurement focusing on the performance in successive frames is more desirable for tracking analysis. To this end, from a visual tracking perspective, we define the metric

$$s = \sum_{i=1}^{n} (p_i - p_m)^2, \tag{17}$$

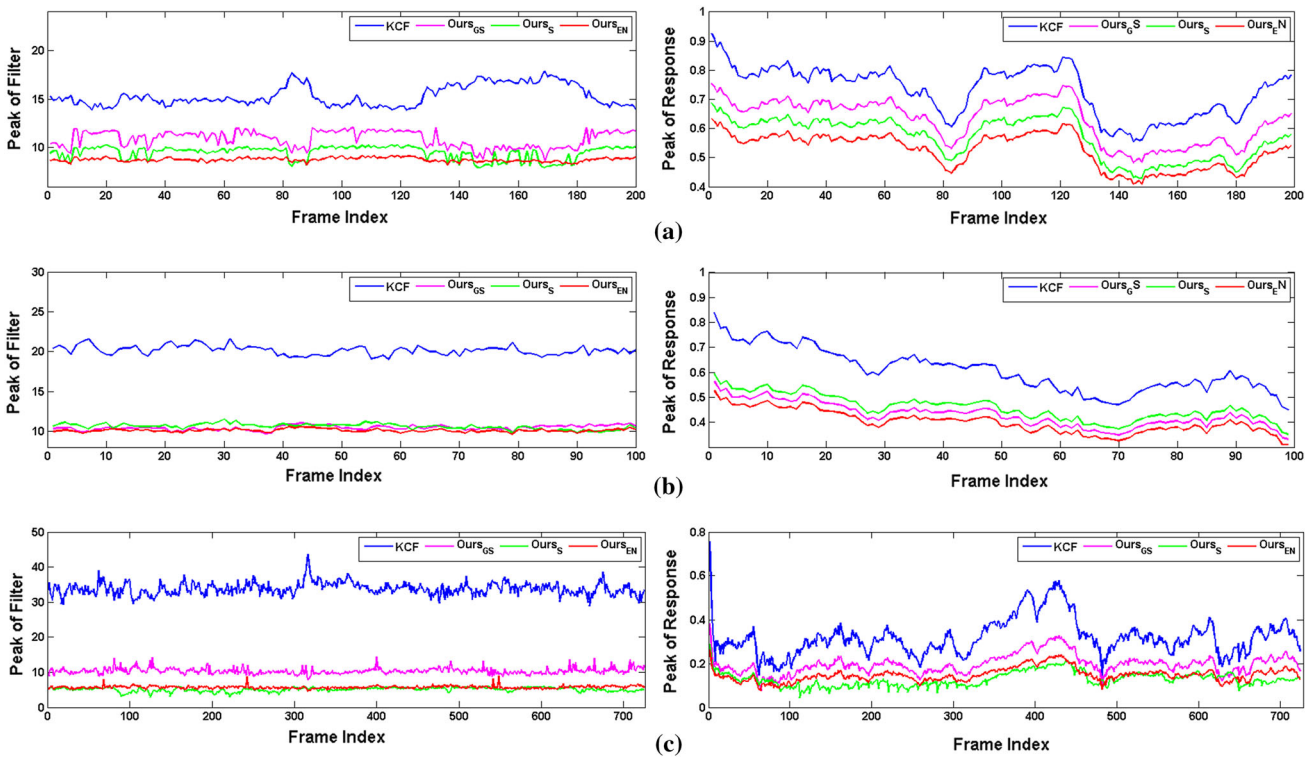


Fig. 17 Peak values of the online learned filters (left column) and the responses (right column) obtained by the KCF tracker and the three proposed trackers in different challenging cases. The curves are expected to be as smooth as possible. **a** occlusion (first 200 frames of *faceocc2*); **b** illumination change (first 100 frames of *david*); and **c** deformation (all the 725 frames of *basketball*)

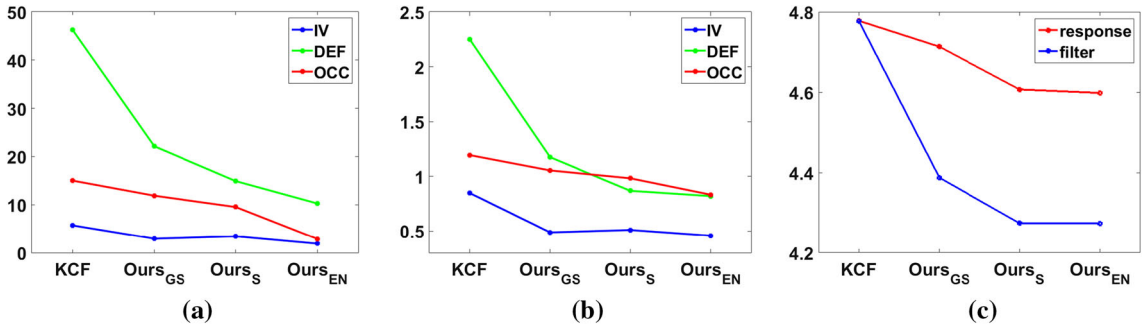


Fig. 18 Peak sensitivity of **a** the filters and **b** the responses obtained by the KCF tracker and the three proposed trackers in different challenging cases. **c** Average peak sensitivity of the filters and the responses obtained by the KCF tracker and the three proposed trackers on all the 100 video sequences of the OTB 2015 benchmark, respectively

to measure the sensitivity of the correlation filter, where p_i denotes the peak value of the filter response in the ith frame, p_m denotes the mean of the filter peaks over the n frames, and the n peak values are normalized by their squared norm. Note that the above metric is actually similar to the variance of the peak values of the n filter responses except for the average and the normalization. Basically, this metric is required to focus on the changes in the peak values of the filter responses in successive frames, which correspond to the target location in the tracking algorithm. We found that the peak values significantly shake up and down when the tracker is losing the target, because in that case the base image changes abruptly. To this end, a variance-like metric can reflect the shakes very well. We thus define the metric as the above presented. To emphasis that this metric is used for investigating tracking performance, we call it the peak sensitivity instead of the peak variance. The sensitivity also means that we expect the peak values have less shakes for a better designed correlation filter, i.e., non-sensitive to the target appearance changes. As

addressed above, the value of s is expected to be small for a good correlation filter, i.e., low sensitivity.

The sensitivity s of the learned correlation filters and the filter responses obtained by the KCF tracker and the three proposed trackers in the above three challenging situations are plotted in Fig. 18a, b, respectively. For a comprehensive verification on the effectiveness of the sensitivity, we analyze the average sensitivity of the correlation filter and the filter response on all the 100 video sequences of the OTB 2015 benchmark, as shown in Fig. 18c. It can be seen that the sensitivity analysis of the correlation filter in successive frames is consistent with the tracking performance evaluations shown in Figs. 14, 15 and 16.

From both the quantitative and the qualitative investigations, a conclusion can be drawn to interpret how the loss functions essentially influence the tracking performance: the lower the sensitivity s of the learned correlation filter in successive frames is, the higher the tracking performance is obtained. Note that this metric can be adopted as a useful reference criterion for designing a robust correlation filter for visual tracking.

In addition, to demonstrate the peak sensitivity thoroughly, we refer to another metric, proposed in Sui et al. (2018b) named the peak strength, which is an analysis method on the response peak from both a discrimination and tracking accuracy perspectives. Its definition is shown as follows.

$$ps = \frac{1}{n}\left(\sum_{k=1}^{n}(p - r_k)^2\right)^{\frac{1}{2}} - \left\|\mathbf{c}_p - \mathbf{c}_{gt}\right\|_2, \tag{18}$$

where p denotes the peak values of a response map, r_k denotes the kth neighbors of p, $\mathbf{c}_p$ and $\mathbf{c}_{gt}$ denote the peak location and the ground truth target location respectively. The peak strength is expected to be large for a good correlation filter. We use 8 neighbors of the peak, i.e., set $n = 8$ in the above equation. We calculate the peak strength values of the response maps in all the 58,935 frames of the OTB 2015 benchmark for the ℓ_2-loss that uses in the plain correlation filter learning framework (i.e., the KCF tracker) and the three robust loss functions proposed in this work. The distributions of these peak strength values are shown in Fig. 19. It can be seen that the three loss functions used in this work have obviously larger peak strength than the ℓ_2-loss. The results are consistent with the peak sensitivity analyzed above.

Revisiting the proposed approach, the sparsity based loss functions can smooth the drastic fluctuation in the response peaks by compensating large errors, resulting in low sensitivity values in the correlation filter learning. In contrast, the squared loss adopted by the KCF tracker enforces small and dense errors in the correlation filter learning. As a result, the filter has to always adjusts itself to fit all the small appearance changes, resulting in high sensitivity values. This reveals, from the sensitivity perspective, why the proposed trackers perform better than the KCF tracker.

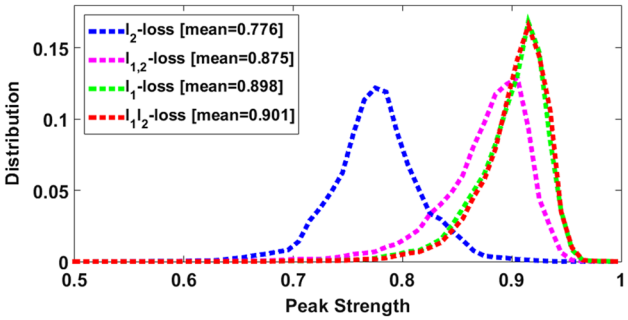


Fig. 19 Distributions of peak strength of the filter responses obtained from different loss functions in all the 58,935 frames of the OTB 2015 benchmark

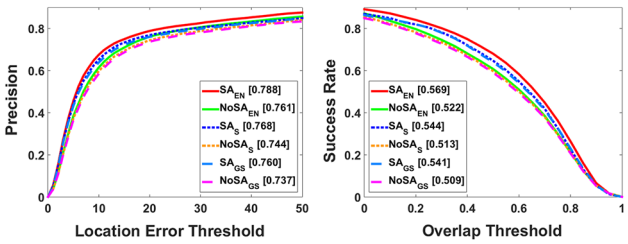


Fig. 20 Tracking performance of the trackers with scale adaptation (SA) and without scale adaptation (NoSA) on the OTB 2015 benchmarks

4.4.3 Scale Adaptation

The proposed tracking algorithm leverages a multi-resolution based scale adaptation method to improve its performance in the target scale estimation for the correlation filtering framework. Note that this method is quite popular and intuitive for the scale estimation. It is straightforward to sample the target scale and evaluate the scale according to the maximum filter response over all the target candidates. We demonstrate the contribution of the scale adaptation to the final tracking performance on the OTB 2015 benchmark. The comparison results are shown in Fig. 20. It can be seen that, with the scale adaptation strategy, the performance of the three proposed trackers is improved consistently by about 3% on this benchmark, in terms of both the precision and the success rate.

4.4.4 Parameter Investigation

The parameter τ in Eq. (4) is investigated on the four benchmarks since τ is a critical parameter which controls the importance of the proposed loss functions. We select the value of τ in the set $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$.

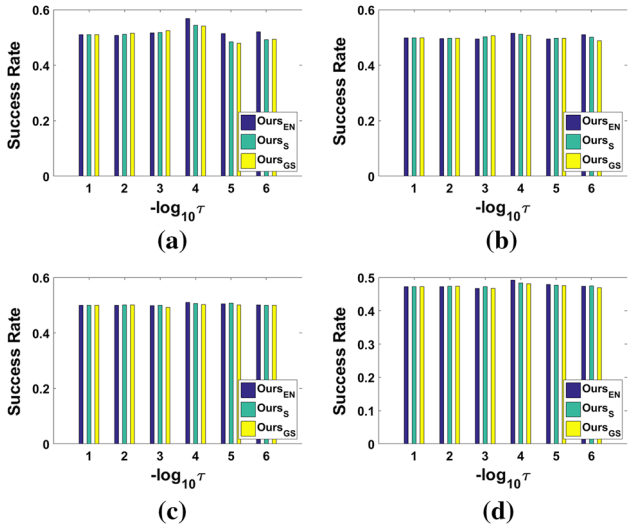


Fig. 21 Investigations on the parameter τ in Eq. (4) in terms of success rate on the benchmarks. **a** OTB 2015; **b** VOT 2015; **c** VOT 2016; and **d** VOT 2017

The tracking performance with respect to different τ values are shown in Fig. 21. It is evident that, on the four benchmarks, τ does not influence the tracking results significantly in terms of success rate. It can also be seen that relatively large values lead to relatively low success rate scores because e_i is too sparse to reflect the anisotropy of the loss functions, while relatively small τ values also result in relatively low performance since the dense e_i values decrease the impact of the anisotropy. As a result, we set τ to 10^{-4} in the experiments since $\tau = 10^{-4}$ leads to the best results on the four benchmarks.

4.4.5 Tracking Speed

The proposed trackers run at 7 and 37 fps respectively with and without the scale adaptation. They are implemented in MATLAB without any code optimization. The scale adaption, presented in Eq. (16) where 7 scales are employed, can improve the tracking performance, as demonstrated in the experimental evaluations, but it slows down the tracking speed correspondingly about 7 times. Note that the computations over the 7 scales for the target localization are independent to each other. As a result, the scale adaption can be exactly equivalently re-implemented by a map-reduce framework, where the target localization is evaluated in parallel over the 7 scales. It indicates that the proposed trackers can achieve real-time performance with the map-reduce framework. In this work, we aim at demonstrating the proposed correlation filter learning method is effective to tracking accuracy and robustness. In practice, for those tracking speed sensitive applications, the proposed tracking algorithm supports various acceleration and parallelization techniques like the map-reduce framework.

4.5 Tracking in Noise Contaminated Frames

We have various imaging devices nowadays, such as mobile phone, and low-definition digit camera. The quality of the video sequences cannot be guaranteed in practical applications, i.e., the frames are often corrupted by noise. A typical case is the white noise corruptions because the heating of imaging devices/units. For this reason, a visual tracker is expected to perform robustly with the noise contaminated frames. To comprehensively investigate the robustness of the proposed approach, the tracking is analyzed in the noise contaminated frames. The representative frames with noise contamination are shown in Fig. 22. The KCF tracker (Henriques et al. 2015) is again employed as the baseline tracker. Note that only one scale coefficient is utilized in the scale pool $\mathcal{S} = \{1\}$ to reveal how robust the loss functions are against the noise contamination rather than their incorporation with the scale adaptivity, leading to a fair comparison with the KCF tracker. We use a synthetic additional white noise to simulate the noise contamination. We choose different portions of pixels at random in each frame, and add these pixels by white noise yielding a Gaussian with a zero mean and a small standard deviation (std = 0.07 in this work). The comparison results under the different noise levels are shown in Fig. 23. It can be seen that the performance of the KCF tracker significantly decreases in the case that even a small number of pixels are corrupted, while in contrast, the proposed trackers are rarely influenced. Note that the performance of the proposed trackers does not decrease sharply until a relative large number of pixels (20% in the comparisons) are corrupted. As a result, it is evident that, in the noise contaminated frames, the proposed trackers perform more robustly than the KCF tracker. This suggests that the proposed approach is suitable for practical applications beyond the benchmark data sets.

5 Conclusion

This study focuses on improving the robustness of the correlation filter learning. The anisotropy of the filter response has been observed and analyzed for the correlation filtering based tracking model, through which the overfitting issue of previous methods has been alleviated. Three sparsity related loss functions have been proposed to exploit the anisotropy, resulting in improved overall tracking performance, correspondingly leading to three implementations of visual trackers within the paradigm of correlation filtering. Extensive experiments have demonstrated that the robustness of the learned correlation filter is greatly improved via the proposed approach. More importantly, this study has

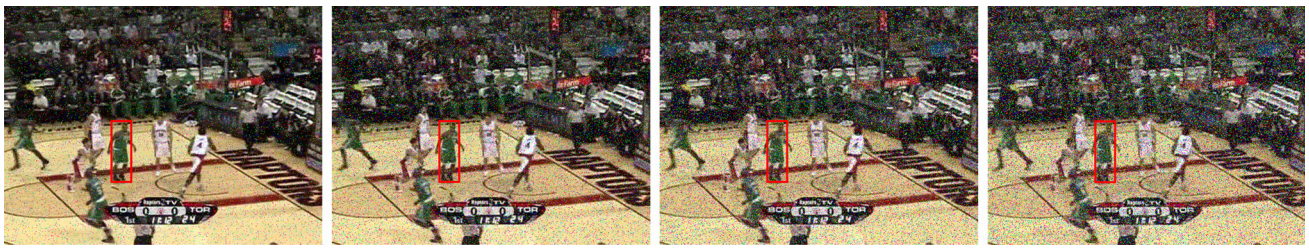


Fig. 22 Representative frames with 5%, 10% , 15% and 20% corrupted pixels (from left to right)

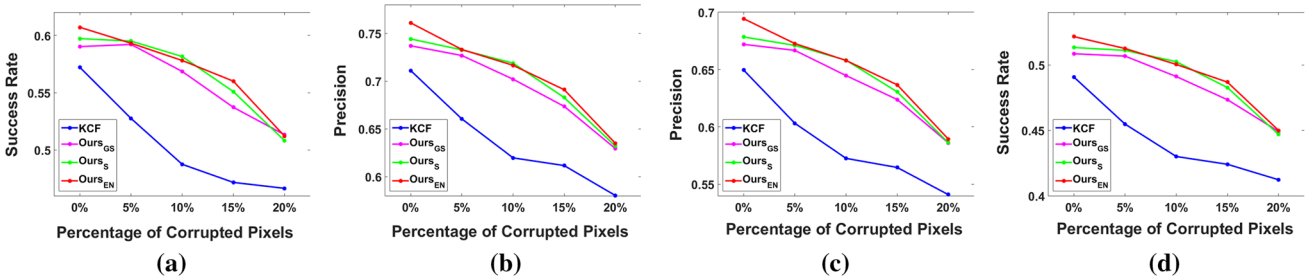


Fig. 23 Tracking performance of the three proposed trackers and the KCF tracker in the presence of noise with different amounts on all the 100 video sequences of the OTB 2015 benchmark. **a** precision plots with $\theta = 20$; **b** success plot with $\rho = 0.5$; **c** precision plots in average; and **d** success plot in average

empirically revealed how different loss functions essentially influence the tracking performance. A metric, the sensitivity of the filter response peak, has been proposed, under which an important conclusion has been drawn that the sensitivity of the peak values of the filter response in successive frames is consistent with the tracking performance. As a result, the sensitivity can be employed as a useful reference criterion for designing a robust correlation filter in visual tracking.

Appendix: Baseline Trackers

Extensive visual trackers are employed in the experimental evaluations as the baseline trackers. In this appendix section, we present the citations of these baseline trackers.

Baseline Trackers on the OTB 2015 Benchmark

PSCF (Sui et al. 2018b), RCF (Sui et al. 2016b), KCF_AT (Bibi et al. 2016), SRDCF (Danelljan et al. 2015), HCFT (Ma et al. 2015a), SAMF (Li and Zhu 2014), DSST (Danelljan et al. 2014a), KCF (Henriques et al. 2015), CN (Danelljan et al. 2014b), and CSK (Henriques et al. 2012).

Baseline Trackers on the VOT 2015 Benchmark

MDNet (Nam and Han 2016b), DeepSRDCF (Danelljan et al. 2015), EBT (Zhu et al. 2016), SRDCF(Danelljan et al. 2015), LDP (Lukezic et al. 2015), sPST (Hua et al. 2015), SC-EBT (Wang et al. 2015b), NSAMF (Li and Zhu 2015), Struck (Hare et al. 2011), RAJSSC (Zhang et al. 2015a), S3Tracker (Lee et al. 2015), SumShift (Lee and Yu 2011), SODLT (Wang et al. 2015c), DAT (Possegger et al. 2015), MEEM (Zhang et al. 2014a), RobStruck (Bogun and Ribeiro 2015), OACF (Bertinetto et al. 2015), MCT (Duffner and Garcia 2015), HMMTxD (Vojir et al. 2015), ASMS (Vojir et al. 2014).

Baseline Trackers on the VOT 2016 Benchmark

C-COT (Danelljan et al. 2016), TCNN (Nam and Han 2016a), SSAT (Qi et al. 2016a), MLDF (Wang et al. 2016a), Staple (Bertinetto et al. 2016b), DDC (Gao et al. 2016), EBT (Zhu et al. 2016), SRBT (Lee and Kim 2016), STAPLE+ (Xu et al. 2016), DNT (Chi et al. 2016), SSKCF (Lee et al. 2016), SiamFC-R (Bertinetto et al. 2016a), DeepSRDCF (Danelljan et al. 2015), SHCT (Wen et al. 2016), MDNet-N (Nam and Han 2016b), FCF (Zhang et al. 2016), SRDCF (Danelljan et al. 2015), RFD-CF2 (Walsh and Mederios 2016), GGTv2 (Hu et al. 2016), DPT (Lukezic et al. 2016).

Baseline Trackers on the VOT 2017 Benchmark

LSART (Sun et al. 2017), CFWCR (He et al. 2017), CFCF (Gundogdu and Alatan 2017), ECO (Danelljan et al. 2017b), Gnet (Singh and Mishra 2017), MCCT (Wang et al. 2017a), CCOT (Danelljan et al. 2016), CSRDCF (Lukezic et al. 2017a), SiamDCF (Wang et al. 2017b), MCPF (Zhang et al. 2017b), CRT (Chen and Tao 2016), ECOhc (Danelljan et al. 2017a), DLST (Yang et al. 2017), CSRDCFf (Lukezic et al.

2017b), RCPF (Zhang et al. 2017a), UCT (Zhu et al. 2017), SPCT (Poostchi et al. 2017), ATLAS (Mocanu et al. 2017), MEEM (Zhang et al. 2014a), FSTC (Chen et al. 2017).

References

Bach, F., Jenatton, R., Mairal, J., & Obozinski, G. (2011). Convex optimization with sparsity-inducing norms. *Optimization for Machine Learning*, *5*, 19–53.

Beck, A., & Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, *2*(1), 183–202.

Bertinetto, L., Henriques, J., Valmadre, J., Torr, P., & Vedaldi, A. (2016a). SiameseFC-ResNet. In *ECCV VOT workshop*.

Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., & Torr, P. (2016b). Staple: Complementary learners for real-time tracking. In *CVPR*.

Bertinetto, L., Valmadre, J., Miksik, O., Golodetz, S., & Torr, P. H. (2015). The importance of estimating object extent when tracking with correlation filters. In *ICCV VOT workshop*.

Bibi, A., Mueller, M., & Ghanem, B. (2016). Target response adaptation for correlation filter tracking. In *ECCV*.

Bogun, I., & Ribeiro, E. (2015). Structure tracker with the robust Kalman filter. In *ICCV VOT workshop*.

Bolme, D., Beveridge, J. R., Draper, Ba., & Lui, Y. M. (2010). Visual object tracking using adaptive correlation filters. In *CVPR*.

Chen, B., Wang, L., & Lu, H. (2017). FSTC. In *ICCV VOT workshop*

Chen, K., & Tao, W. (2016). Convolutional regression for visual tracking. arXiv.

Chi, Z., Lu, H., Wang, L., & Sun, C. (2016). Dual deep network tracker. In *ECCV VOT workshop*.

Danelljan, M., Bhat, G., Khan, S., & Felsberg, M. (2017a). Efficient convolution operator tracker: Hand crafted. In *ICCV VOT workshop*.

Danelljan, M., Ghat, G., Khan, F., & Felsberg, M. (2017b). ECO: Efficient convolution operators for tracking. In *CVPR*.

Danelljan, M., Gustav, H., Khan, F. S., & Felsberg, M. (2015). Learning spatially regularized correlation filters for visual tracking. In *ICCV*.

Danelljan, M., Häger, G., Khan, F.S., & Felsberg, M. (2014a). Accurate scale estimation for robust visual tracking. In *BMVC*.

Danelljan, M., Khan, F. S., Felsberg, M., & Weijer, J. V. D. (2014b). Adaptive color attributes for real-time visual tracking. In *CVPR*.

Danelljan, M., Robinson, A., Shahbaz, K., & Felsberg, M. (2016). Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *ECCV*.

Duffner, S., & Garcia, C. (2015). Using discriminative motion context for online visual object tracking. *IEEE Transactions on Circuits and Systems for Video Technology (TCVST)*, *26*(12), 2215–2225.

Gao, J., Zhang, T., Xu, C., & Liu, B. (2016). Discriminative deep correlation tracking. In *ECCV VOT workshop*.

Gundogdu, E., & Alatan, A. (2017). Good features to correlate for visual tracking. arXiv.

Hare, S., Saffari, A., & Torr, P. (2011). Struck: Structured output tracking with kernels. In *ICCV*.

He, Z., Fan, Y., & Zhuang, J. (2017). CFWCR. In *ICCV VOT workshop*.

Henriques, F., Caseiro, R., Martins, P., & Batista, J. (2012). Exploiting the circulant structure of tracking-by-detection with kernels. In *ECCV*.

Henriques, J., Caseiro, R., Martins, P., & Batista, J. (2015). High-speed tracking with kernelized correlation filters. *IEEE TPAMI*, *37*(3), 583–596.

Hu, T., Du, D., Wen, L., Li, W., Qi, H., & Lyu, S. (2016). Geometric structure hyper-graph based tracker version 2. In *ECCV VOT Workshop*.

Hua, Y., Alahari, K., & Schmid, C. (2015). Online object tracking with proposal selection. In *ICCV*.

Kalal, Z., Mikolajczyk, K., & Matas, J. (2012). Tracking–learning-detection. *IEEE TPAMI*, *34*(7), 1409–1422.

Kristan, M., Matas, J., Leonardis, A., Vojir, T., Pflugfelder, R., Fernandez, G., et al. (2016). A novel performance evaluation methodology for single-target trackers. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, *38*(11), 2137–2155.

Kwon, J., & Lee, K. (2010). Visual tracking decomposition. In *CVPR*.

Lee, H., & Kim, D. (2016). Salient region based tracker. In *ECCV VOT workshop*.

Lee, J. Y., Choi, S., Jeong, J. C., Kim, J. W., & Cho, J. I. (2015). Scaled SumShift tracker. In *ICCV VOT workshop*.

Lee, J. Y., Choi, S., Jeong, J. C., Kim, J. W., & Cho, J. I. (2016). SumShift tracker with kernelized correlation filter. In *ECCV VOT workshop*.

Lee, J. Y., & Yu, W. (2011). Visual tracking by partition-based histogram backprojection and maximum support criteria. In *IEEE international conference on robotics and biomimetics*.

Li, Y., & Zhu, J. (2014). A scale adaptive kernel correlation filter tracker with feature integration. In *ECCV workshop*.

Li, Y., & Zhu, J. (2015). NSAMF. In *ICCV VOT workshop*.

Liu, S., Zhang, T., Cao, X., & Xu, C. (2016). Structural correlation filter for robust visual tracking. In *CVPR*.

Liu, T., Wang, G., & Yang, Q. (2015). Real-time part-based visual tracking via adaptive correlation filters. In *CVPR*.

Lukezic, A., Cehovin, L., & Kristan, M. (2015). Layered deformable parts tracker. In *ICCV VOT workshop*.

Lukezic, A., Cehovin, L., & Kristan, M. (2016). Deformable parts correlation filters for robust visual tracking. arXiv.

Lukezic, A., Vojir, T., Cehovin, L., Matas, J., & Kristan, M. (2017a). Discriminative correlation filter with channel and spatial reliability. In *CVPR*.

Lukezic, A., Vojir, T., Cehovin, L., Matas, J., & Kristan, M. (2017b). Discriminative correlation filter with channel and spatial reliability: Fast. In *ICCV VOT workshop*.

Ma, C., Huang, J. B., Yang, X., & Yang, M. H. (2015a). Hierarchical convolutional features for visual tracking. In *ICCV*.

Ma, C., Yang, X., Zhang, C., & Yang, Mh. (2015b). Long-term correlation tracking. In *CVPR*.

Mei, X., & Ling, H. (2011). Robust visual tracking and vehicle classification via sparse representation. *IEEE TPAMI*, *33*(11), 2259–2272.

Mocanu, B., Tapu, R., & Zaharia, T. (2017). Adaptive single object tracking using offline learned motion and visual similar patterns. In *ICCV VOT workshop*.

Nam, B. M. H., & Han, B. (2016a). Modeling and propagating CNNs in a tree structure for visual tracking. arXiv.

Nam, H., & Han, B. (2016b). Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*.

Poostchi, M., Palaniappan, K., Seetharaman, G., & Gao, K. (2017). Spatial pyramid context-aware tracker. In *ICCV VOT workshop*.

Possegger, H., Mauthner, T., & Bischof, H. (2015). In defense of color-based model-free tracking. In *CVPR*.

Qi, Y., Qin, L., Zhang, S., & Huang, Q. (2016a). Scale-and-state aware tracker. In *ECCV VOT workshop*.

Qi, Y., Zhang, S., Qin, L., Yao, H., Huang, Q., Lim, J., et al. (2016b). Hedged deep tracking. In *CVPR*.

Singh, S., & Mishra, D. (2017). gNetTracker. In *ICCV VOT workshop*.

Smeulders, A. W. M., Chu, D. M., Cucchiara, R., Calderara, S., Dehghan, A., & Shah, M. (2014). Visual tracking: An experimental survey. *IEEE TPAMI*, *36*(7), 1442–1468.

Sui, Y., Tang, Y., & Zhang, L. (2015a). Discriminative low-rank tracking. In *ICCV*.

Sui, Y., Tang, Y., Zhang, L., & Wang, G. (2018a). Visual tracking via subspace learning: A discriminative approach. *International Journal of Computer Vision (IJCV)*, *126*(5), 515–536.

Sui, Y., Wang, G., Tang, Y., & Zhang, L. (2016a). Tracking completion. In *ECCV*.
Sui, Y., Wang, G., & Zhang, L. (2018b). Correlation filter learning toward peak strength for visual tracking. *IEEE Transactions on Cybernetics*, *48*(4), 1290–1303.
Sui, Y., Wang, G., Zhang, L., & Yang, M. H. (2018c). Exploiting spatial–temporal locality of tracking via structured dictionary learning. *IEEE Transactions on Image Processing (TIP)*, *27*(3), 1282–1296.
Sui, Y., & Zhang, L. (2015). Visual tracking via locally structured Gaussian process regression. *IEEE SPL*, *22*(9), 1331–1335.
Sui, Y., & Zhang, L. (2016). Robust tracking via locally structured representation. *IJCV*, *119*(2), 110–144.
Sui, Y., Zhang, S., & Zhang, L. (2015b). Robust visual tracking via sparsity-induced subspace learning. *IEEE TIP*, *24*(12), 4686–4700.
Sui, Y., Zhang, Z., Wang, G., Tang, Y., & Zhang, L. (2016b). Real-time visual tracking: Promoting the robustness of correlation filter learning. In *ECCV*.
Sui, Y., Zhao, X., Zhang, S., Yu, X., Zhao, S., & Zhang, L. (2015c). Self-expressive tracking. *Pattern Recognit.*, *48*(9), 2872–2884.
Sun, C., Liu, J., Lu, H., & Yang, M. H. (2017). Learning spatial-aware regressions for visual tracking. In *ICCV VOT workshop*.
Tang, M., & Feng, J. (2015). Multi-kernel correlation filter for visual tracking. In *ICCV*.
Vojir, T., Matas, J., & Noskova, J. (2015). Online adaptive hidden Markov model for multi-tracker fusion. arXiv.
Vojir, T., Noskova, J., & Matas, J. (2014). Robust scale-adaptive mean-shift for tracking. *Pattern Recognit. Lett.*, *40*, 250–258.
Walsh, R., & Mederios, H. (2016). CF2 with Response Information Failure Detection. In *ECCV VOT workshop*.
Wang, D., Lu, H., & Yang, M. H. (2013). Least soft-thresold squares tracking. In *CVPR*.
Wang, L., Lu, H., Wang, Y., & Sun, C. (2016a). Multi-level deep feature tracker. In *ECCV VOT workshop*.
Wang, L., Ouyang, W., Wang, X., & Lu, H. (2015a). Visual tracking with fully convolutional networks. In *ICCV*.
Wang, L., Ouyang, W., Wang, X., & Lu, H. (2016b). STCT: Sequentially training convolutional networks for visual tracking. In *CVPR*.
Wang, N., Huang, Z., Li, S., & Yeung, D. Y. (2015b). Ensemble-based tracking: Aggregating crowdsourced structured time series data. In *ICML*.
Wang, N., Li, S., Gupta, A., & Yeung, D. Y. (2015c). Transferring rich feature hierarchies for robust visual tracking. arXiv.
Wang, N., Zhou, W., & Li, H. (2017a). Dual deep network tracker. In *ICCV VOT workshop*.
Wang, Q., Gao, J., Xing, J., Zhang, M., Z. Z., & Hu, W. (2017b). SiamDCF. In *ICCV VOT workshop*.
Wen, L., Du, D., Li, S., Chang, C.M., Lyu, S., & Huang, Q. (2016). Structure hyper-graph based correlation filter tracker. In *ECCV VOT workshop*.
Wright, J., Ma, Y., Mairal, J., & Sapiro, G. (2010). Sparse representation for computer vision and pattern recognition. *Proceedings of The IEEE*, *98*(6), 1031–1044.
Wu, Y., Lim, J., & Yang, M. H. (2013). Online object tracking: A benchmark. In *CVPR*.
Wu, Y., Lim, J., & Yang, M. H. (2015). Object tracking benchmark. *IEEE TPAMI*, *37*(9), 1834–1848.
Xu, Z., Li, Y., & Zhu, J. (2016). An improved STAPLE tracker with multiple feature integration. In *ECCV VOT Workshop*.
Yang, L., Liu, R., Zhang, D., & Zhang, L. (2017). Deep location-specific tracking. In *ICCV VOT workshop*.
Yilmaz, A., Javed, O., & Shah, M. (2006). Object tracking: A Survey. *ACM Computing Surveys*, *38*(4), 13–57.
Zhang, J., Ma, S., & Sclaroff, S. (2014a). MEEM: Robust tracking via multiple experts using entropy minimization. In *ECCV*.
Zhang, K., Zhang, L., Liu, Q., Zhang, D., & Yang, M. H. (2014b). Fast visual tracking via dense spatio-temporal context learning. In *ECCV*.
Zhang, M., Xing, J., Gao, J., & Hu, W. (2016). Fully-functional correlation filtering-based tracker. In *ECCV VOT workshop*.
Zhang, M., Xing, J., Gao, J., Shi, X., Wang, Q., & Hu, W. (2015a). Rotation adaptive joint scale-spatial correlation filter based tracker. In *ICCV VOT workshop*.
Zhang, S., Sui, Y., Zhao, S., Yu, X., & Zhang, L. (2015b). Multi-local-task learning with global regularization for object tracking. *Pattern Recognit.*, *48*(12), 3881–3894.
Zhang, S., Zhao, S., Sui, Y., & Zhang, L. (2015c). Single object tracking with fuzzy least squares support vector machine. *IEEE TIP*, *24*(12), 5723–5738.
Zhang, T., Gao, J., & Xu, C. (2017a). Robust correlation particle filter. In *ICCV VOT workshop*.
Zhang, T., Ghanem, B., & Liu, S. (2012a). Robust visual tracking via multi-task sparse learning. In *CVPR*.
Zhang, T., Ghanem, B., Liu, S., & Ahuja, N. (2012b). Low-rank sparse learning for robust visual tracking. In *ECCV*.
Zhang, T., Liu, S., Xu, C., Yan, S., Ghanem, B., Ahuja, N., & Yang, Mh. (2015d). Structural sparse tracking. In *CVPR*.
Zhang, T., Xu, C., & Yang, M. H. (2017b). Multi-task correlation particle filter for robust object tracking. In *CVPR*.
Zhu, G., Porikli, F., & Li, H. (2016). Beyond local search: Tracking objects everywhere with instance-specific proposals. *CVPR*.
Zhu, Z., Huang, G., Zou, W., & Du, D., Huang, C. (2017). UCT. In *ICCV VOT workshop*.