

High performance visual tracking with circular and structural operators

Peng Gao, Yipeng Ma, Ke Song, Chao Li, Fei Wang*, Liyi Xiao*, Yan Zhang

Shenzhen Graduate School, Harbin Institute of Technology, China

ARTICLE INFO

Keywords:

Visual tracking
Circular and structural operators
Ensemble post-processor
Collaborative optimization

ABSTRACT

Visual tracking algorithms based on structured output support vector machine (SOSVM) have demonstrated excellent performance. However, sampling methods and optimization strategies of SOSVM undesirably increase the computational overloads, which hinder real-time application of these algorithms. Moreover, due to the lack of high-dimensional features and dense training samples, SOSVM-based algorithms are unstable to deal with various challenging scenarios, such as occlusions and scale variations. Recently, visual tracking algorithms based on discriminative correlation filters (DCF), especially the combination of DCF and features from deep convolutional neural networks (CNN), have been successfully applied to visual tracking, and attains surprisingly good performance on recent benchmarks. The success is mainly attributed to two aspects: the circular correlation properties of DCF and the powerful representation capabilities of CNN features. Nevertheless, compared with SOSVM, DCF-based algorithms are restricted to simple ridge regression which has a weaker discriminative ability. In this paper, a novel circular and structural operator tracker (CSOT) is proposed for high performance visual tracking, it not only possesses the powerful discriminative capability of SOSVM but also efficiently inherits the superior computational efficiency of DCF. Based on the proposed circular and structural operators, a set of primal confidence score maps can be obtained by circular correlating feature maps with their corresponding structural correlation filters. Furthermore, an implicit interpolation is applied to convert the multi-resolution feature maps to the continuous domain and make all primal confidence score maps have the same spatial resolution. Then, we exploit an efficient ensemble post-processor based on relative entropy, which can coalesce primal confidence score maps and create an optimal confidence score map for more accurate localization. The target is localized on the peak of the optimal confidence score map. Besides, we introduce a collaborative optimization strategy to update circular and structural operators by iteratively training structural correlation filters, which significantly reduces computational complexity and improves robustness. Experimental results demonstrate that our approach achieves state-of-the-art performance in mean AUC scores of 71.5% and 69.4% on the OTB2013 and OTB2015 benchmarks respectively, and obtains a third-best expected average overlap (EAO) score of 29.8% on the VOT2017 benchmark.

1. Introduction

Visual tracking is one of the fundamental research problems in the field of computer vision, it has a variety of applications such as video surveillance, assistant driving systems, service robots and human-computer interaction. Given the initial position and size of an arbitrary target, a high performance tracker should discriminate the target from the background and locate the target in subsequent frames [1]. Despite significant progress have been made in the past decade, there are still numerous unsolved visual tracking challenges due to various negative factors, such as motion blur, illumination variations, deformations and occlusions.

In general, visual tracking algorithms can be categorized as

discriminative or generative approaches. Generative tracking algorithms [2–5] establish a reference appearance model to characterize the target in current frame, and then an image patch most similar to the reference model is searched as the target in the next frame. The searching process can be guided by a probabilistic motion model. However, the background information contained in the initial target bounding box is also modeled, which weakens the similarity measurement and thus limits the performance of generative tracking algorithms. In contrast, discriminative tracking algorithms have received unprecedented research interest in the last decade. Most discriminative algorithms follow the tracking-by-detection paradigm [6–11], which treats the tracking task as a detection problem. They employ a classifier or a regressor to process both target and background representations,

* Corresponding authors.

E-mail addresses: 1392135844@qq.com (F. Wang), xiaoly@hit.edu.cn (L. Xiao).

<https://doi.org/10.1016/j.knosys.2018.08.008>

Received 22 March 2018; Received in revised form 6 August 2018; Accepted 8 August 2018

Available online 09 August 2018

0950-7051/ © 2018 Elsevier B.V. All rights reserved.

and produce an optimal decision boundary that can efficiently discriminate the target from the background.

Support vector machine (SVM) is a popular representative and discriminative appearance classifier [12–14]. Existing SVM-based trackers mostly have two modules: a sampler and a learner. The sampler generates a set of positive and negative training samples, and the learner updates the classifier based on training samples. SOSVM-based trackers outperform other SVM-based trackers considerably [7,15]. However, using dense training samples and high dimensional features increases computational complexity, therefore limiting the performance of these SOSVM-based trackers, and the optimization and detection process may further degrade the computational speed [7,16].

Recently, discriminative correlation filters (DCF) based trackers [17–21] have achieved excellent results in terms of accuracy, robustness and speed [15,22,23], because they treat tracking tasks as similarity learning problems. Since DCF exploits all circular shifts of training samples to solve a ridge regression in the Fourier frequency domain [9], it avoids time-consuming correlation operations. Recent advancements in DCF-based tracking performance are driven by multi-dimensional features [9,18], adaptive scale estimation [24,25], robust long-term memory components [26,27], reducing boundary effects [28,29] and continuous-domain features [30,31]. Most DCF-based trackers employ conventional handcrafted appearance features, such as raw pixels [32], Histograms of Oriented Gradients (HOG) [9,33], ColorNames (CN) [18,34] or their combinations [35,36]. Nevertheless, compared with SOSVM-based trackers, DCF-based trackers are restricted to simple ridge regression which has a weaker discriminative ability.

Due to their powerful representation capabilities, deep CNN features have been successfully applied to many computer vision tasks, including image classification [37,38] and object detection [39,40]. A CNN usually composes of several layers of convolution, local normalization and pooling operations. Different layers capture different levels of features. Some recent works combine DCF with deep features have achieved state-of-the-art performance on many tracking benchmarks [23,41]. Two most successful deep CNN features extracted from pre-trained networks are deep appearance features [42] and deep motion features [43]. Despite deep CNN features have strong capacities to represent targets, extracting them requires intensive computations.

To deal with challenging scenarios such as occlusions and deformations, several recent approaches [44–47] exploit ensemble methods to combine multiple different types of features, models or trackers. Many algorithms take such ensemble strategies, including interactive Markov Chain Monte Carlo [48], factorial hidden Markov model [49], entropy minimization [13], random ferns [50], sharing convolutional layers [51] and linear interpolation [35]. Although the ensemble method is effective and robust for visual tracking, it still suffers from high computational cost and thus has not been applied widely.

In this paper, a novel circular and structural operator based tracker (CSOT) is proposed by incorporating DCF and SOSVM, which not only possesses the powerful discriminative capability of SOSVM [7] but also efficiently inherits the superior computational efficiency of DCF [9]. So that it can employ higher-dimensional deep features and denser circular shifted samples. We employ deep complementary features including both deep appearance features and deep motion features to enhance the diversity of training samples. Similar to [43], the shallow (*Conv*-1) and deep (*Conv*-5) convolutional layers of the VGG-M [38] network are exploited to extract deep appearance feature maps, and they have 128 and 512 feature channels respectively. Deep motion feature maps are extracted from the *Conv*-5 layer of the pre-trained optical flow network as described in *ActionCubes* [53],¹ which has 384 feature channels. In CSOT, a set of primal confidence score maps can be obtained by circular correlating deep complementary feature maps with their corresponding structural correlation filters using circular and structural operators.

Moreover, we have observed the fact that some approaches merely sum up all primal confidence score maps to estimate target positions. Unfortunately, they are not reliable for visual tracking. To address this issue, we introduce an ensemble post-processor based on relative entropy, which can coalesce primal confidence score maps and create an optimal confidence score map for more accurate localization. The target can be localized by searching the peak value on the optimal confidence score map. However, the ensemble post-processor requires all primal confidence score maps have the same spatial resolution, explicitly re-sampling all primal confidence score maps to the same resolution will introduce distortion and even decrease the tracking performance. Therefore, we exploit an implicit interpolation to extract continuous-domain deep feature maps and make all primal confidence score maps have the same spatial resolution [30,31]. Besides, we introduce a collaborative optimization strategy to update circular and structural operators by iterative training structural correlation filters. To boost the tracking accuracy, we adopt a robust scale estimation for our CSOT. The main contributions of this paper can be summarized as follows:

- A novel and efficient CSOT is proposed, which possesses the superior computational efficiency of DCF and the powerful discriminative capability of SOSVM. By employing circular and structural operators, continuous-domain deep feature maps and structural correlation filters are gathered to calculate primal confidence score maps with the same spatial resolution.
- An online collaborative optimization strategy is suggested to train structural correlation filters, which decomposes the training problem into two independent sub-problems and can be iteratively solved online.
- An ensemble post-processor is introduced, which combines primal confidence score maps to create an optimal confidence score map. Furthermore, a robust scale estimation scheme is adopted, which enables our tracker to adapt to target scale variations efficiently.

Finally, experimental evaluations are conducted on several recent visual tracking benchmarks [15,22,23]. As shown in Fig. 1, the results demonstrate that our tracker outperforms most state-of-the-art trackers both in terms of accuracy and robustness. To the best of our knowledge, we are the first to propose circular and structural operators for visual tracking with multiple deep complementary features in the continuous spatial domain.

The rest of the paper is organized as follows. Section 2 briefly reviews related works. Section 3 details the proposed approach. Section 4 demonstrates the experimental results. Finally, Section 5 concludes the paper.

2. Related works

In this section, we give a brief review of related works in four categories: SVM-based trackers, DCF-based trackers, CNN-based trackers and ensemble-based trackers.

2.1. SOSVM-based trackers

SOSVM is a popular and successful backbone for visual tracking in the past decade [15]. It treats visual tracking as a structured estimation that admits a consistent target representation for both optimization and detection. Struck [7] is the first tracker employs SOSVM and demonstrates superior performance on the original OTB benchmark (OTB2013) [15]. But it is time-consuming and difficult to be extended to higher-dimensional features. DLSSVM [16] presents a dual linear SOSVM framework that approximates intersection kernels for feature representations based on explicit feature maps to improve tracking performance. Nonetheless, DLSSVM still suffers from high computational complexity, and it is hard to be used for real-time tracking. SCF [14] derives an equivalent formulation of a SOSVM model with circulant samples to learn support correlation

¹ <https://gkioxari.github.io/ActionTubes>

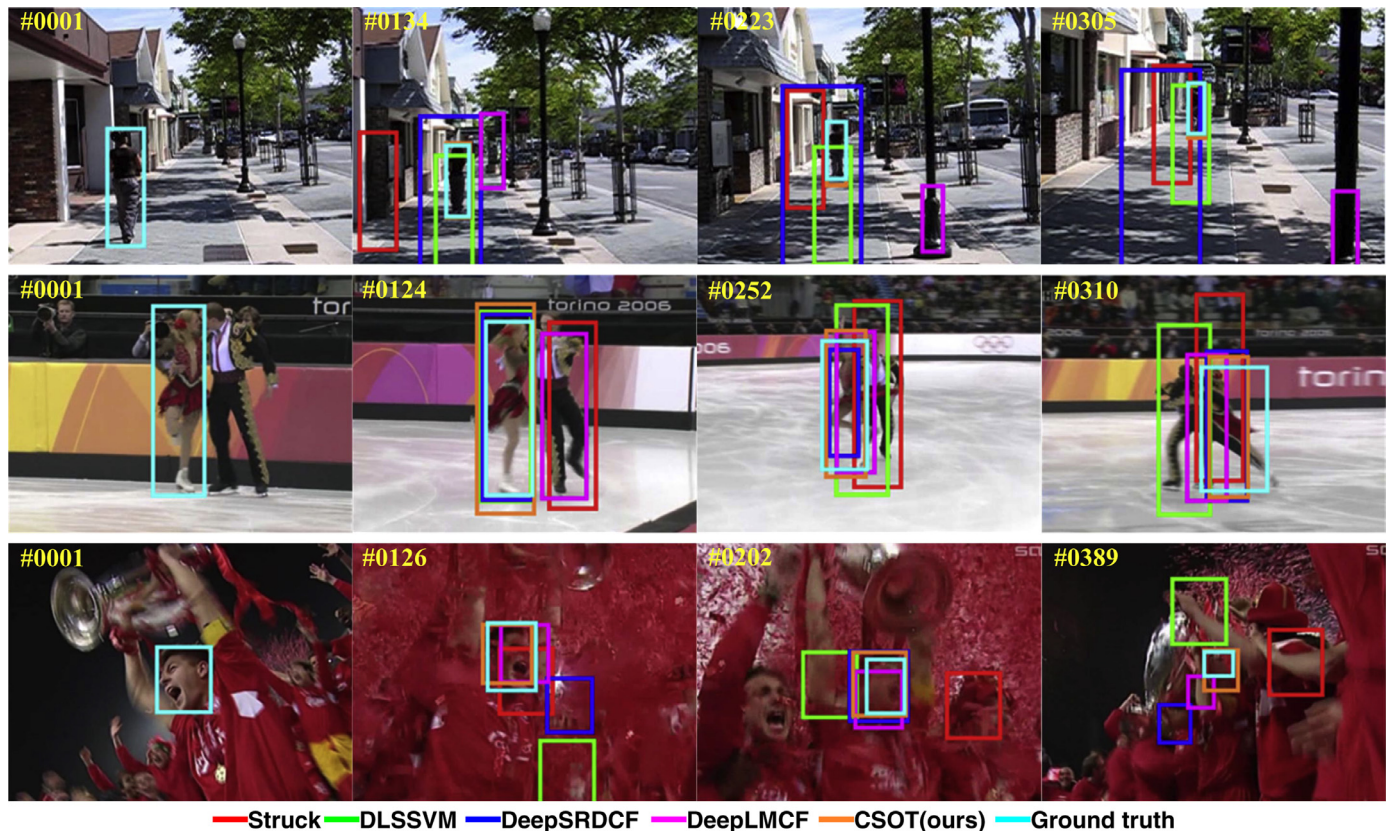


Fig. 1. Comparison of our method CSOT (in orange) with four state-of-the-art trackers Struck [7] (in red), DLSSVM [16] (in green), DeepSRDCF [42] (in blue) and DeepLMCF [52] (in pink) on three example sequences *Human9* (top row), *Skating* (middle row) and *Soccer* (bottom row) from OTB2015 benchmark [22], respectively. The ground-truth bounding boxes of these sequences are also given (in cyan). Best viewed in color. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

filters. LMCF [52] exploits DCF to speed up SOSVM-based tracking models. Despite of their outstanding performances, employing high-dimensional features and dense training samples are computationally expensive, which make them difficult to be applied for real-time tracking.

2.2. DCF-based trackers

In recent years, DCF-based approaches have been successfully applied to visual object tracking. MOSSE [32] is the first tracker which exploits Fast Fourier Transform (FFT) to train single-channel correlation filters based on raw pixel samples of both the target and the background. Since MOSSE uses linear classifier and grayscale features, it has limited capability to track targets. To address this issue, Henriques et al. [9] introduce a kernelized correlation filter-based approach (KCF), which exploits kernel tricks and multi-dimensional HOG features for visual tracking. Subsequently, several DCF-based trackers are proposed to address the inherent limitations of DCF. Both SAMF [24] and DSST [25] use scale pyramid representations to adaptively estimate scale variations of targets. Another deficiency of the DCF-based trackers is that since dense sampling strategies employ periodical training samples, thereby incurring unwanted boundary effects. It severely compromises the robustness of DCF-based tracking approaches, especially when targets are in the challenging scenario of out-of-view. To solve this problem, SRDCF [29] adds a spatially regularized component to penalize DCF close to searching boundaries, while BACF [28] adopts a zero-padding scheme to ensure that correlation filters have the same size as sample patches. Some recent works [30,31,54,55] integrates multi-resolution features into formulations and learn a set of convolution filters to generate continuous-domain confidence score maps of the target. These trackers have shown excellent performance on recent visual tracking benchmarks [22,23,41].

2.3. CNN-based trackers

Because of their impressive representation power of deep features, some recent works combine them with DCF, and show state-of-the-art performance both in terms of accuracy and robustness. MDNet [56] follows a paradigm of offline training and online fine-tuning. It has two purposes: (a) to learn domain-independent representations from pre-trained networks, and (b) to capture domain-specific information through online learning. Inspired by the Siamese network which has five convolutional layers, SiamFC [57] introduces two identical branches, each branch has two conv5 layers. It then employs a new cross-correlation layer to connect the two conv5 layers together. Due to its high tracking speed, SiamFC is claimed to be the best real-time tracker in recent VOT challenge [23]. However, most CNN-based trackers solely employ deep appearance features, but lack of high-level motion cues. Based on SRDCF method [29], DMSRDCF [43] investigates the fusion of conventional handcrafted features, deep appearance features and deep motion features, and it outperforms SRDCF significantly. In general, due to their advantages both in robustness and computational efficiency, CNN-based trackers have shown that deep features are more suitable for visual tracking.

2.4. Ensemble-based trackers

Recent years have witnessed significant advances of ensemble-based methods on visual tracking. Kwon and Lee [48] use principal component analysis to construct several basic models and exploit an interactive Markov Chain Monte Carlo method, which combines those basic models together. EBT [49] formulates an ensemble tracking framework based on a factorial hidden Markov model, which leverages structured

crowdsourced time series data of five independent trackers. Hong et al. [45] present an efficient tracking approach based on a multi-task and multi-view joint sparse method. They employ a particle filter framework to combine each feature view linearly. MEEM [13] proposes an entropy-regularized restoration scheme to restore the historical trackers, and then selects estimations of these trackers based on the minimum entropy criterion to address undesirable model updates. Staple [35] makes use of two complementary tracking models in a ridge regression framework and linearly fuses two different results. MFCMT [21] combines the response maps of DSST and DAT based on a simple relative entropy criterion. All ensemble-based trackers have shown superior performance compared with their basic models.

C-COT [30] and SCF [14] are two works most closely related to ours. They focus on learning continuous convolution filters and support correlation filters, respectively. Different from these works, our work mainly aims to bridge the gap between DCF and SOSVM by using a novel and efficient circular and structural operator for visual tracking. Both the two works employ deep appearance features and conventional handcrafted features. In contrast, our approach exploits both deep appearance features and deep motion features extracted from a pre-trained VGG network [38] and a pre-trained optical flow network [53], respectively. Furthermore, our approach exploits interpolation to build continuous-domain feature maps, while SCF uses merely the explicitly resampling strategy to make feature maps have the same resolution.

3. The proposed approach

In this section, we describe our proposed CSOT in detail. The overall pipeline of CSOT is shown in Fig. 2, which is composed of four portions, i.e., the feature extractor, circular and structural operator, ensemble post-processor and online collaborative optimizer. In the pipeline, we firstly obtain searching samples and optical flows of the input image. Then in the feature extractor, the interpolated continuous deep complementary features of the target can be extracted from two independent deep networks. All these deep features are fed into the circular and structural operator to circular correlate with their corresponding structural correlation filters, primal confidence score maps with the same spatial resolution can be created. In the ensemble post-processor, all these primal confidence maps are combined to form a more discriminative optimal confidence score map. The estimated target size and position can be determined by searching the peak value on the optimal confidence score maps. Finally, we utilize the tracking results to update circular and structural operators by iterative training of structural correlation filters using the online collaborative optimizer.

3.1. Circular and structural operators

In our proposed approach, the aim is to learn the circular and structural operator S_w parameterized by structural correlation filters w from a set of circular input-output pairs (X_i, Y_i) , where X_i denotes an input space and Y_i indicates an arbitrary circular output space. For the sake of clarity, we formulate our approach for data defined in the one-dimensional domain in this section, and it can be generalized to higher dimensions in the same way [30,52]. All circular shifted versions of the target image patch x_i of size $M_i \times N_i$ centered around the estimated target position y_i are considered as training samples [9] at frame i . Hence, the input space X_i contains the estimated target image patches, and the output space Y_i is now representing the estimated target positions on the circular shifted image patches, i.e. $Y_i = \{y_{i(p)} | p \in \{0, \dots, \Lambda - 1\}\}$, where $\Lambda = M_i \times N_i$ denotes the number of circular shifted target positions and p indicates the relative pixel number of circular shift according to y_i , e.g., $y_{i(0)}$ denotes the estimated target position with no circular shift and $y_{i(1)}$ is obtained by circular shifting y_i by 1 pixel. We use (X_i, Y_i) to represent all training samples and $(x_i, y_{i(p)})$ denotes one training sample at frame i . The feature maps of these samples are denoted as $\phi(x_i, y_{i(0)}) \in \mathbb{R}^{\Lambda \times D}$, e.g., the deep feature map extracted from *Conv-1* of the pre-trained *vgg-m* network, D indicates the number of feature map channels.

Since $(X_i, Y_i)_i = [(x_i, y_{i(0)}), (x_i, y_{i(1)}), \dots, (x_i, y_{i(\Lambda-1)})]$ is a circular matrix formed by enumerating all shifted training samples, we can calculate the primal confidence score map $S_i \in \mathbb{R}^\Lambda$ by using the circular and structural operator S_w to circular correlate structural correlation filters w with the feature map $\phi(x_i, y_{i(0)})$,

$$\begin{aligned} S_{i(Y_i)} &= S_w(X_i, Y_i)_i \\ &= \sum_{d=1}^D (\phi^d(X_i, Y_i)_i)^T w^d \\ &= \sum_{d=1}^D w^d \otimes \phi^d(x_i, y_{i(0)}) \end{aligned} \quad (1)$$

where the symbol \otimes denotes the circular correlation, and $w^d \in \mathbb{R}^\Lambda$ indicates the corresponding structural correlation filter of the d th channel feature map $\phi^d(x_i, y_{i(0)})$. The circular and structural operator can be learned by training structural correlation filters w following the optimization problem,

$$\begin{aligned} \min_w \sum_{d=1}^D \|w^d\|^2 + C \sum_{y_{i(p)} \in Y_i} \xi_{i(y_{i(p)})}^2 \\ \text{s. t. } \forall p, \forall y_{i(p)} \in Y_i: \xi_{i(y_{i(p)})} \geq J_i(y_{i(p)}) - (S_{i(y_{i(0)})} - S_{i(y_{i(p)})}) \end{aligned} \quad (2)$$

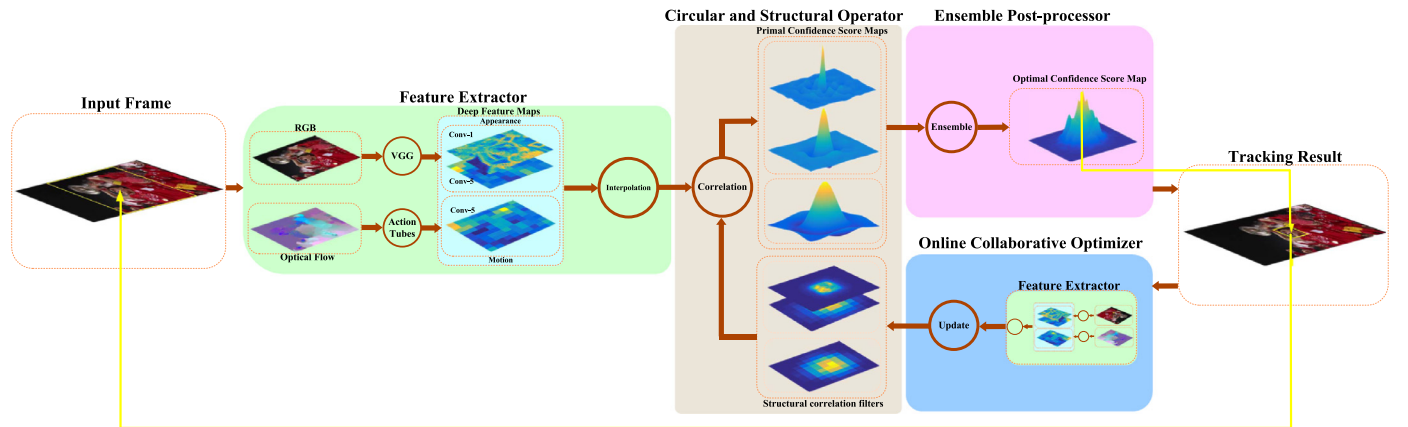


Fig. 2. The overall pipeline of our proposed approach.

Here, $S_{i(\mathbf{y}_{i(p)})} = \sum_{d=1}^D (\varphi^d(\mathbf{x}_i, \mathbf{y}_{i(p)})^T \mathbf{w}^d \in \mathbb{R}$ denotes the primal confidence score of each cyclic shifted training sample $(\mathbf{x}_i, \mathbf{y}_{i(p)})$, the slack variables $\xi_{i(p)}$ represents the penalty assigned to each sample for margin violations. For the desired training samples, $\xi_{i(p)}$ will be 0. The regularization parameter C penalizes the complicated functions which are prone to over-fitting, and biases Eq. (2) towards training error minimization and margin maximization. $J_{i(\mathbf{y}_{i(p)})} \in \mathbb{R}$ is the cost function which denotes the training error associated with $\mathbf{y}_{i(p)}$ and $\mathbf{y}_{i(0)}$. The cost function can be defined as,

$$J_{i(\mathbf{y}_{i(p)})} = 1 - m_{\mathbf{x}}(\mathbf{y}_{i(p)}), \quad p = 0, 1, \dots, \Lambda - 1 \quad (3)$$

The desired confidence score output $m_{\mathbf{x}}(\mathbf{y}_{i(p)})$ is supposed to have a Gaussian form, i.e., $m_{\mathbf{x}}(t) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(t - \mathbf{y}_{i(0)})^2}{2\sigma^2}\right)$, which takes a peak value of 1 located at the estimated target position $\mathbf{y}_{i(0)}$ and smoothly reduces to 0 for larger circular shifts [9].

However, feature maps extracted from each convolutional layer of a deep network frequently have different resolution. In order to combine the primal confidence score for target detection in Section 3.3, all primal confidence score maps must have the same spatial resolution. To achieve this, we take advantage of an implicit interpolation to convert those feature maps to the continuous spatial domain. Suppose feature map $\varphi(\mathbf{x}_i, \mathbf{y}_{i(p)})$ extracted from a specific convolutional layer has an independent resolution N . Then, the feature map can be rewritten as a function $\varphi(\mathbf{x}_i, \mathbf{y}_{i(p)})[n]$ indexed by the discrete spatial variable $n \in \{0, 1, \dots, N-1\}$. Consequently, for each feature channel d , we convert the feature map from discrete spatial domain to the continuous spatial domain $t \in [0, T)$ by employing an interpolation operator Φ ,

$$\Phi^d\{\varphi(\mathbf{x}_i, \mathbf{y}_{i(p)})\}(t) = \sum_{n=0}^{N-1} \varphi^d(\mathbf{x}_i, \mathbf{y}_{i(p)})[n] b\left(t - \frac{T}{N}n\right) \quad (4)$$

Here, b represents the interpolation function with the period of $T > 0$, and $\Phi\{\varphi(\mathbf{x}_i, \mathbf{y}_{i(p)})\}$ is the interpolated feature map, which viewed as a continuous T -periodic function. In our framework, the cubic spline kernel is applied to construct the interpolation function as described in C-COT [30].

Therefore, we can obtain primal confidence score maps with the same resolution by exploiting interpolated feature maps $\Phi\{\varphi(\mathbf{x}_i, \mathbf{y}_{i(0)})\}$ and structural correlation filters \mathbf{w} . Since the circular correlation is equivalent to an element-wise multiplication in the Fourier frequency domain, we can fast compute Eq. (1) with the circulant property of circular matrix [58,59] as,

$$S_{i(\mathbf{y}_i)} = \mathcal{F}^{-1}\left(\sum_{d=1}^D \hat{\mathbf{w}}^d \odot \hat{\Phi}^d\{\varphi(\mathbf{x}_{i+1}, \mathbf{y}_{i(0)})\}\right) \quad (5)$$

where the hat symbol $\hat{\cdot}$ of a T -periodic function represents the discrete Fourier transform (DFT), i.e., $\hat{\mathbf{w}}^d[k] = \frac{1}{T} \int_0^T \mathbf{w}^d(t) e^{-i\frac{2\pi k}{T}t} dt$, $\hat{\Phi}^d\{\varphi(\mathbf{x}_i, \mathbf{y}_{i(0)})\}[k] = \sum_{n=0}^{N-1} \varphi^d(\mathbf{x}_i, \mathbf{y}_{i(0)})[n] e^{-i\frac{2\pi k}{N}n} \hat{b}[k]$ and $\hat{b}[k] = \frac{1}{N} \exp\left(-i\frac{\pi}{N}\right) \hat{b}\left(\frac{k}{N}\right)$. \mathcal{F}^{-1} stands for the inverse discrete Fourier transform (IDFT), i.e., $S_{i(\mathbf{y}_i)}(t) = \mathcal{F}^{-1}(\hat{S}_{i(\mathbf{y}_i)}[k]) = \sum_{k=-\infty}^{+\infty} \hat{S}_{i(\mathbf{y}_i)}[k] e^{i\frac{2\pi k}{T}t}$, $\hat{S}_{i(\mathbf{y}_i)}[k]$ are Fourier coefficients which can be obtained as $\hat{S}_{i(\mathbf{y}_i)}[k] = \sum_{d=1}^D \hat{\mathbf{w}}^d[k] \hat{\Phi}^d\{\varphi(\mathbf{x}_i, \mathbf{y}_{i(0)})\}[k]$. The symbol \odot denotes the element-wise multiplication.

Similar to SRDCF [29], we add a penalty function γ in Eq. (2) to mitigate drawbacks of the periodic assumption and control the spatial extent of structural correlation filters \mathbf{w} . The learning problem can be equivalently formulated as,

$$\begin{aligned} \min_{\mathbf{w}} \sum_{d=1}^D \|\gamma \odot \mathbf{w}^d\|^2 + C \sum_{\mathbf{y}_{i(p)} \in \mathcal{Y}_i} \xi_{i(\mathbf{y}_{i(p)})}^2 \\ \text{s. t. } \forall p, \forall \mathbf{y}_{i(p)} \in \mathcal{Y}_i: \xi_{i(\mathbf{y}_{i(p)})} \geq J_{i(\mathbf{y}_{i(p)})} - (S_{i(\mathbf{y}_{i(0)})} - S_{i(\mathbf{y}_{i(p)})}) \end{aligned} \quad (6)$$

This penalty function γ not only makes structural correlation filters \mathbf{w} reduce smoothly from the center of the sample to boundaries but also suppress \mathbf{w} which resides close to boundaries.

3.2. Online collaborative optimization

In order to optimize Eq. (6) efficiently, a new parameter $\epsilon_{i(\mathbf{y}_i)} \in \mathbb{R}^{\Lambda}$ is defined, where $\xi_{i(\mathbf{y}_i)} = \epsilon_{i(\mathbf{y}_i)} + J_{i(\mathbf{y}_i)} - (S_{i(\mathbf{y}_{i(0)})} - S_{i(\mathbf{y}_i)})$, $\epsilon_{i(\mathbf{y}_i)} \geq 0$. Therefore, the minimization of Eq. (6) is equivalent to the following problem,

$$\begin{aligned} \min_{\mathbf{w}} \sum_{d=1}^D \|\gamma \odot \mathbf{w}^d\|^2 + C \left\| \epsilon_{i(\mathbf{y}_i)} + J_{i(\mathbf{y}_i)} - (S_{i(\mathbf{y}_{i(0)})} - S_{i(\mathbf{y}_i)}) \right\|_2^2 \\ \text{s. t. } \epsilon_{i(\mathbf{y}_i)} \geq 0 \end{aligned} \quad (7)$$

There are two variables $\epsilon_{i(\mathbf{y}_i)}$ and \mathbf{w} in this formulation have to be solved. When \mathbf{w} is known, the subproblem on $\epsilon_{i(\mathbf{y}_i)}$ has a closed-form solution. However, when $\epsilon_{i(\mathbf{y}_i)}$ is known, the subproblem on \mathbf{w} does not have the closed-form solution, since we exploit the spatial regularization and implicit interpolation. We employ the Conjugate Gradient (CG) to iteratively solve the subproblem on \mathbf{w} . Motivated by SCF [14], we propose an online collaborative optimization that solves these problems efficiently by iterating between the following two steps.

Update ϵ . Given \mathbf{w} , the subproblem on $\epsilon_{i(\mathbf{y}_i)}$ becomes

$$\begin{aligned} \min_{\epsilon_i} \left\| \epsilon_{i(\mathbf{y}_i)} + J_{i(\mathbf{y}_i)} - (S_{i(\mathbf{y}_{i(0)})} - S_{i(\mathbf{y}_i)}) \right\|_2^2 \\ \text{s. t. } \epsilon_{i(\mathbf{y}_i)} \geq 0 \end{aligned} \quad (8)$$

Therefore, this subproblem has a closed-form solution,

$$\epsilon_{i(\mathbf{y}_i)} = \max\{0, S_{i(\mathbf{y}_{i(0)})} - S_{i(\mathbf{y}_i)} - J_{i(\mathbf{y}_i)}\} \quad (9)$$

Update \mathbf{w} . Given ϵ_i , the subproblem on \mathbf{w} becomes

$$\min_{\mathbf{w}} \sum_{d=1}^D \|\gamma \odot \mathbf{w}^d\|^2 + C \left\| S_{i(\mathbf{y}_i)} - (S_{i(\mathbf{y}_{i(0)})} - J_{i(\mathbf{y}_i)} - \epsilon_{i(\mathbf{y}_i)}) \right\|_2^2 \quad (10)$$

In order to obtain a simple expression of the normal equations, $\rho_{i(\mathbf{y}_i)} = S_{i(\mathbf{y}_{i(0)})} - J_{i(\mathbf{y}_i)} - \epsilon_{i(\mathbf{y}_i)}$ is defined as the confidence label for each training sample. Then, using Parseval's formula, Eq. (10) can be transformed to the Fourier frequency domain as,

$$\min_{\mathbf{w}} \sum_{d=1}^D \|\hat{\gamma} \otimes \hat{\mathbf{w}}^d\|^2 + C \|\hat{S}_{i(\mathbf{y}_i)} - \hat{\rho}_{i(\mathbf{y}_i)}\|_2^2, \quad (11)$$

where the Fourier coefficients of confidence labels and penalty function are $\hat{\rho}_{i(\mathbf{y}_i)}[k] = \hat{S}_{i(\mathbf{y}_{i(0)})}[k] - \hat{J}_{i(\mathbf{y}_i)}[k] - \hat{\epsilon}_{i(\mathbf{y}_i)}[k]$ and $\hat{\gamma}[k] = \frac{1}{T} \int_0^T \gamma(t) e^{-i\frac{2\pi k}{T}t} dt$ respectively, and $\hat{J}_{i(\mathbf{y}_i)}[k]$ can be straightforwardly deduced from $\hat{m}_{\mathbf{x}}[k] = \frac{\sqrt{2\pi\sigma^2}}{T} \exp\left(-2\sigma^2\left(\frac{\pi k}{T}\right)^2 - i\frac{2\pi k}{T} \mathbf{y}_{i(0)}\right)$.

Finally, subproblem on \mathbf{w} can be addressed by the following normal equation

$$\left(\sum_{d=1}^D (\hat{\Phi}_{i(0)}^d)^H \hat{\Phi}_{i(0)}^d + \frac{1}{C} (\hat{\gamma})^H \hat{\gamma} \right) \hat{\mathbf{w}}^d = (\hat{\Phi}_{i(0)}^d)^H (\hat{\rho}_{i(\mathbf{y}_i)})^H \quad (12)$$

Here, $\hat{\Phi}_{i(0)}^d = \hat{\Phi}^d\{\varphi(\mathbf{x}_i, \mathbf{y}_{i(0)})\}$, and \mathbf{a}^H represents the complex conjugation of a complex matrix \mathbf{a} . In our approach, we apply the CG method mentioned in C-COT [30] to solve Eq. (12) iteratively, since it can utilize the sparsity structure effectively.

Similar to the original DCF framework, we also derive a training equation that does not exploit the penalty function γ . For computational efficiency, we set $\gamma = \mathbf{I}$, where \mathbf{I} is an identity matrix. The optimization problem Eq. (11) can be rewritten as,

$$\min_{\mathbf{w}} \sum_{d=1}^D \|\hat{\mathbf{w}}^d\|^2 + C \|\hat{S}_{i(\mathbf{y}_i)} - \hat{\rho}_{i(\mathbf{y}_i)}\|_2^2 \quad (13)$$

In this case, Eq. (13) is similar to conventional DCF-based trackers which aim to find optimal correlation filters by minimizing a mean squared error between the estimated confidence score map and the

desired output. The normal equation Eq. (13) of \mathbf{w} can be expressed as,

$$\left(\sum_{d=1}^D (\hat{\Phi}_{i(0)}^d)^H \hat{\Phi}_{i(0)}^d + \frac{1}{C} \mathbf{I} \right) \hat{\mathbf{w}}^d = (\hat{\Phi}_{i(0)}^d)^H (\hat{\Phi}_{i(y_i)}^d)^H \quad (14)$$

\mathbf{w} can be obtained by performing element-wise operations in the Fourier frequency domain,

$$\hat{\mathbf{w}}^d = \frac{(\hat{\Phi}_{i(0)}^d)^H (\hat{\Phi}_{i(y_i)}^d)^H}{\sum_{d=1}^D (\hat{\Phi}_{i(0)}^d)^H \hat{\Phi}_{i(0)}^d + \frac{1}{C} \mathbf{I}}, \quad (15)$$

where the \div symbol denotes the element-wise division. We can easily recover structural correlation filters \mathbf{w} using Fourier transforms in the continuous spatial domain. Compared with canonical DCF-based trackers, our collaborative optimization performs convolution in the continuous-domain by exploiting interpolated feature maps. Further, the proposed approach can create confidence score maps of the target by solving the continuous structured function directly, which is more discriminative than ridge regression models [9,32].

3.3. Ensemble post-processor

To detect the target in a new frame $(i+1)$, all circular shifts of the possible target image patch \mathbf{x}_{i+1} of the previous size $M_i \times N_i$ centered around the previous target position \mathbf{y}_i are considered as searching samples. This is performed using the same procedure as for training samples in Section 3.1.

We define searching samples as circular input-output pairs $(\mathcal{X}_s, \mathcal{Y}_s)_{i+1}$, hence, the input space \mathcal{X}_s denotes the possible target image patches, and the output space $\mathcal{Y}_s = \{\mathbf{y}_{i(p)} \mid p \in \{0, \dots, \Lambda - 1\}\}$ is now indicating the possible target positions on the circular shifted image patches. Since circular and structural operators are updated by iterative training structural correlation filters in the Fourier frequency domain, the primal confidence score map S is obtained as,

$$S = \mathcal{F}^{-1} \left(\sum_{d=1}^D \mathcal{P}_T(\hat{S}_{i(\mathcal{Y}_s)}) \right) \quad (16)$$

Notice that before performing IDFT, $\Phi^d\{\varphi(\mathbf{x}_i, \mathbf{y}_{i(0)})\}$ has been interpolated in the continuous spatial domain, we use a padding operator \mathcal{P}_T to zero-padding the high frequencies of $\hat{S}_{i(\mathcal{Y}_s)}[k] = \sum_{d=1}^D \hat{\mathbf{w}}^d[k] \hat{\Phi}^d\{\varphi(\mathcal{X}_s, \mathcal{Y}_s)_{i+1}\}[k]$, hence, S has a constant size of T .

However, primal confidence score maps obtained from only a single-layer deep feature map may sometimes be too weak to deal with challenging scenarios. Robust trackers should imperatively have the better diversity of features. We propose an effective ensemble post-processor that leverages multi-layer deep feature maps extracted from multiple deep networks for visual tracking. The ensemble post-processor can substantially improve tracking performance by coalescing primal confidence score maps based on relative entropy.

For primal confidence score map set $S = \{S_1, S_2, \dots, S_L\}$ based on L -layer deep feature maps, the primal confidence score map $S_l \in S$ consists of a probability distribution $S_{l, \mathbf{y}_{i(p)}}$, which also can be considered as a probability map. The probability distribution is subjected to $\sum_{\mathbf{y}_{i(p)} \in \mathcal{Y}_s} S_{l, \mathbf{y}_{i(p)}} = 1$ and presents the possibility that the possible position $\mathbf{y}_{i(p)}$ becomes the estimated target position. Further, in order to find the optimal confidence score map R , we can minimize the relative entropy, i.e., Kullback–Leibler divergence, between the primal confidence score map S_l of each feature layer and the optimal confidence score map R . Relative entropy can be calculated as,

$$KL(S_l \| R)_{i+1} = \sum_{\mathbf{y}_{i(p)} \in \mathcal{Y}_s} S_{l, \mathbf{y}_{i(p)}} \log \frac{S_{l, \mathbf{y}_{i(p)}}}{R_{\mathbf{y}_{i(p)}}} \quad (17)$$

Then, we can obtain the optimal confidence score map R by

$$\begin{aligned} \arg \min_R \sum_{l=1}^L KL(S_l \| R)_{i+1} \\ \text{s. t. } \sum_{\mathbf{y}_{i(p)} \in \mathcal{Y}_s} R_{\mathbf{y}_{i(p)}} = 1 \end{aligned} \quad (18)$$

where $S_{l, \mathbf{y}_{i(p)}}$ indicates the probability associated with the searching target position $\mathbf{y}_{i(p)}$ on the l th layer primal confidence score map S_l and $R_{\mathbf{y}_{i(p)}}$ indicates the probability on the optimal confidence score map R .

Nonetheless, deep features extracted from pre-trained networks always contain various noise, so that primal confidence score maps obtained using Eq. (5) also have much noise. In order to filter these noise and obtain a more reliable confidence score map, we propose an efficient collaborative filtering operation. Similar to MFCMT [21], since all primal confidence score maps have the same resolution because of exploiting interpolation, we can directly weight a primal confidence score map using all other primal confidence score maps. Therefore, the confidence score map created by fusing any two different layers is more reliable and accurate. The collaborative filter can be formulated as

$$S_{m,n} = S_m S_n, \quad (19)$$

where $m \in \{1, 2, \dots, L-1\}$ and $n \in \{m+1, m+2, \dots, L\}$. After filtering L primal confidence score maps, we get $\frac{L(L-1)}{2}$ weighted confidence score maps $\Theta = \{S_{1,2}, S_{1,3}, \dots, S_{1,L}, S_{2,3}, \dots, S_{L-1,L}\}$ with less noise. With this collaborative filter, if two different primal confidence score maps have similar confidence score at the same position, the weighted confidence score at that position will be higher, while at other positions will have lower weighted confidence scores. Using these weighted confidence score maps, Eq. (18) can be reformulated as

$$\begin{aligned} \arg \min_R \sum_{S_l \in \Theta} \sum_{\mathbf{y}_{i(p)} \in \mathcal{Y}_s} S_{l, \mathbf{y}_{i(p)}} \log \frac{S_{l, \mathbf{y}_{i(p)}}}{R_{\mathbf{y}_{i(p)}}} \\ \text{s. t. } \sum_{\mathbf{y}_{i(p)} \in \mathcal{Y}_s} R_{\mathbf{y}_{i(p)}} = 1 \end{aligned} \quad (20)$$

We exploit the Lagrange multiplier method [20] to solve this problem. The optimal confidence score map can be calculated as

$$R = \frac{2}{L(L-1)} \sum_{S_l \in \Theta} S_l \quad (21)$$

Interestingly, although the optimal confidence score map R is obtained by averaging all weighted confidence score maps, it enhances the robustness of the tracking result. Finally, the target position \mathbf{y}_{i+1} is determined by maximizing the optimal confidence score map R .

In a deep appearance feature network, the shallow layer contains high-resolution and low-level spatial information, while the deep layer involves low-resolution and high-level semantic information. In a deep motion feature network, the deep layer encodes high-level motion information at coarse resolution. Therefore, we can get more discriminative confidence score maps by combining these multiple layers deep features. In Section 4, we will show that our approach outperforms several state-of-the-art trackers both in terms of accuracy and efficiency.

Some works [24,25] demonstrate that effective scale estimation methods are indispensable. In order to improve the accuracy of our proposed approach, we construct a rectangular image patch pyramid by exploiting scale adaptation scheme proposed in CFWCR [54]. The pyramid is centered at the estimated target position in the previous frame. For scale factors $\{a^\tau \mid \tau = \lfloor -\frac{\Delta-1}{2} \rfloor, \dots, \lfloor \frac{\Delta-1}{2} \rfloor\}$, where Δ denotes the number of scale layers, we can obtain a set of input-output pairs $(\mathbf{x}^\delta, \mathbf{y}_p^\delta)$ of size $a^\tau M \times a^\tau N$ on the δ th scaled input image. After that, we extract multiple interpolated deep feature maps according to Eq. (4). Structural correlation filters \mathbf{w} trained from the previous frame can be applied to each deep feature map $\Phi(\mathbf{x}^\delta, \mathbf{y}_p^\delta)$ to get the primal

confidence score map. Next, according to Eq. (21), we can get an optimal confidence score map R^δ of each scaled layer δ . Finally, the estimated target position y_i of the current frame i is determined by the peak value over all optimal confidence score maps R^δ .

For feature maps with $M \times N \times D$ dimensions, we should solve MN subproblems, and the complexity of each subproblem is a linear equation with D variables. Due to the computational efficiency of CG method, each subproblem can be solved in $O(D)$. Since the optimizing process requires DFT, IDFT and element-wise operations in each iteration, the complexity of solving \mathbf{w} is $O(DMN \log(MN))$. The ϵ update problem can be solved in element-wise, which has the cost of $O(MN)$. Thus, the overall complexity to compute the online collaborative optimization algorithm is $O(DMNI_{CG} \log(MN))$, where I_{CG} indicates the number of CG iterations. In a word, our approach is especially suitable for high-dimensional features and dense training samples. We summarize our proposed approach in Algorithm 1.

3.4. Nonlinear extension

It is well known that the kernel trick can improve tracking accuracy further by allowing detection on richer high-dimensional features [9,14]. By exploiting the kernel function $K_{mn} = \kappa(\Phi(\mathbf{x}_i, \mathbf{y}_{i(m)}), \Phi(\mathbf{x}_i, \mathbf{y}_{i(n)}))$, e.g., a Gaussian RBF kernel $\kappa(\Phi(\mathbf{x}_i, \mathbf{y}_{i(m)}), \Phi(\mathbf{x}_i, \mathbf{y}_{i(n)})) = \exp\left(-\frac{1}{2\sigma_k^2} \left\| \Phi(\mathbf{x}_i, \mathbf{y}_{i(m)}) - \Phi(\mathbf{x}_i, \mathbf{y}_{i(n)}) \right\|^2\right)$, the proposed CSOT can be easily posed to the implicit nonlinear space [9]. Structural correlation filters are determined as the weighted sum of circular training samples $\mathbf{w} = \sum_{\mathbf{y}_{i(p)} \in \mathcal{Y}_i} \alpha_{\mathbf{y}_{i(p)}} \Phi(\mathbf{x}_i, \mathbf{y}_{i(p)})$, where $\mathbf{y}_{i(0)}$ is the estimated target position at frame i and α is the parameter vector to be learned. Hence, we have

$$\begin{aligned} \|\mathbf{w}\|^2 &= (\alpha)^T \mathbf{K}_i \alpha \\ &= (\alpha)^T \mathcal{F}^{-1}(\hat{k}_{\Phi_i^{(0)}\Phi_i^{(0)}} \odot \hat{\alpha}) \end{aligned} \quad (22)$$

where $\Phi_i^{(0)} = \Phi(\mathbf{x}_i, \mathbf{y}_{i(0)})$, \mathbf{K}_i denotes a circular kernel matrix whose elements are K_{mn} , and $\hat{k}_{\Phi_i^{(0)}\Phi_i^{(0)}}$ which obtained by the kernel auto-correlation of $\Phi_i^{(0)}$ is the first row of \mathbf{K}_i .

Based on Eq. (10) and Eq. (22), The optimization of kernelized CSOT is formulated as,

$$\begin{aligned} \min_{\alpha} \alpha^T \mathbf{K}_i \alpha \|\gamma\|_2^2 + C \left\| S_{i(\mathcal{Y}_i)} - (S_{i(\mathbf{y}_{i(0)})} - J_{i(\mathcal{Y}_i)} - \epsilon_{i(\mathcal{Y}_i)}) \right\|_2^2 \\ \text{s. t. } \epsilon_{i(\mathcal{Y}_i)} \geq 0 \end{aligned} \quad (23)$$

where $S_{i(\mathcal{Y}_i)} = (\Phi(\mathbf{x}_i, \mathcal{Y}_i))^T \mathbf{w} = \mathbf{K}_i \alpha$ is the primal confidence score map.

Similar to Eq. (11), $\rho_{i(\mathcal{Y}_i)}$ is plugged into Eq. (23)

$$\min_{\alpha} \alpha^T \mathcal{F}^{-1}(\hat{k}_{\Phi_i^{(0)}\Phi_i^{(0)}} \odot \hat{\alpha}) \|\gamma\|_2^2 + C \left\| \mathcal{F}^{-1}(\hat{k}_{\Phi_i^{(0)}\Phi_i^{(0)}} \odot \hat{\alpha}) - \hat{\rho}_{i(\mathcal{Y}_i)} \right\|_2^2 \quad (24)$$

Then, CG method is employed to iteratively solve α in the Fourier frequency domain as,

$$\hat{\alpha} = \left(\hat{k}_{\Phi_i^{(0)}\Phi_i^{(0)}} + \frac{1}{C} (\hat{\gamma})^H \hat{\gamma} \right)^{-1} (\hat{\rho}_{i(\mathcal{Y}_i)})^H \quad (25)$$

The closed-form solution to optimize α also can be expressed without exploiting the spatial penalty function γ similar to Eq. (13),

$$\hat{\alpha} = \frac{\hat{\rho}_{i(\mathcal{Y}_i)}^H}{\hat{k}_{\Phi_i^{(0)}\Phi_i^{(0)}} + \frac{1}{C} \mathbf{I}} \quad (26)$$

When a new frame $(i+1)$ comes out, after obtaining the searching samples $(\mathcal{X}_s, \mathcal{Y}_s)_{i+1}$, the primal confidence score maps for detection is

$$S = \mathcal{F}^{-1}(\mathcal{P}_T(\hat{k}_{\Phi_{i+1}^{(0)}\Phi_i^{(0)}} \odot \hat{\alpha})) \quad (27)$$

where $\hat{k}_{\Phi_{i+1}^{(0)}\Phi_i^{(0)}}$ is the kernel circular correlation of $\Phi_{i+1}^{(0)}$ and $\Phi_i^{(0)}$, and

$\Phi_{i+1}^{(0)} = \Phi(\mathbf{x}_{i+1}, \mathbf{y}_{i(0)})$. Finally, similarly to Section 3.3, we employ the ensemble post-processor to fuse all primal confidence score maps S_l , $l \in \{1, 2, \dots, L\}$, the target position is localized by searching the peak value of the optimal confidence score map R .

4. Experiments

In this section, experimental evaluations are conducted to validate the effectiveness of the proposed CSOT. We first introduce the experimental setup and metrics. Then, we investigate the impact of the various components described in Section 3 by evaluating several variants of the proposed approach. Finally, our proposed approach is compared with several state-of-the-art trackers on the OTB and the VOT benchmarks.

4.1. Experimental setup

The proposed CSOT tracker is implemented in MATLAB R2014b using MatConvNet toolbox [60].² All experiments are performed on an Intel Core i5-4590 CPU @ 3.3 GHz with 8 GB RAM and a NVIDIA Tesla K80 GPU. In our experiments, we crop the searching sample to 5^2 times the previously estimated target size and set the size of searching samples in a restricted area [200,300]. The cubic spline is used as the kernel to construct the interpolation function b as described in the supplementary material of C-COT [30]. The regularization parameter C in Eq. (2) is set as 20000, and the spatial bandwidth σ in the cost function Eq. (3) is set as 0.1. Similar to SRDCF [29], the spatial regularizer γ in Eq. (6) is constructed using a quadratic function $\gamma(m, n) = 0.1 + 3(m/M)^2 + 3(n/N)^2$, where 0.1 is the minimum value of structural correlation filters and 3 is the impact of the spatial regularizer. The kernel function in the nonlinear extension uses Gaussian Radial Basis Function with a spatial bandwidth of 0.2. Similar to CFWCR [54], to conduct adaptive scale estimation, the number of scale layers Δ is set as 10 and the scale factor a is set as 1.03. Online collaborative optimization is performed 25 iterations in the first frame. In the sequential frames, to converge to a desirable estimation of structural correlation filters, online collaborative optimization is iterated 3 times per frame. Each online collaborative optimization iteration involves two CG-iterations to learn structural correlation filters \mathbf{w} using Eq. (12). And the CG is initialized with structural correlation filters learned in the previous frame. All parameters are fixed in the following experiments.

4.2. Experimental metrics

All experiments are conducted on OTB50/2013/2015 [15,22]³ and VOT2017 [23]⁴ benchmarks. Two evaluation metrics are exploited on OTB benchmarks, i.e., distance precision (DP) and overlap precision (OP). DP is computed as the percentage of frames where the average Euclidean distance between the estimated target position and the ground truth is smaller than a preset threshold of 20 pixels. OP is the percentage of frames where the intersection-over-union (IoU) exceeds a fixed threshold of 0.5, IoU is defined as $\frac{|B_t \cap G_t|}{|B_t \cup G_t|}$, where B_t and G_t are the estimated target bounding box and the ground truth, and $|\cdot|$ indicates the number of pixels in the overlap area. We use one-pass evaluation (OPE) that initialize the target position in the first frame and size, then run trackers throughout entire test sequences. Results are reported in precision and success plots. In success plots, trackers are ranked according to the area under the curve (AUC). We also report the speed of trackers by the average frames per second (FPS) over all sequences.

For the VOT2017 benchmark, three evaluation metrics are exploited

² <http://www.vlfeat.org/matconvnet>

³ http://cvlab.hanyang.ac.kr/tracker_benchmark

⁴ <http://www.votchallenge.net/challenges.html>

Input: image sequence $\{\mathbf{I}_i\}_{i=1}^I$, initial target position \mathbf{y}_1 and size $M_1 \times N_1$, scale layers Δ , feature layers L

Output: estimated target position $\{\mathbf{y}_i\}_{i=2}^I$ and size $\{M_i \times N_i\}_{i=2}^I$

- 1: Initialize structural correlation filters $\{\mathbf{w}_{0,l}\}_{l=1}^L = 0$,
- 2: \triangleright Update structural correlation filters using 25 collaborative optimization iterations in the first frame
- 3: Crop training samples \mathbf{x}_1 of size $M_1 \times N_1$ from \mathbf{I}_1 at the initial target position \mathbf{y}_1 ,
- 4: Extract L layers continuous-domain deep feature maps $\{\Phi_{i,l}\}_{l=1}^L$ of searching samples \mathbf{x}_1 using Eq. (4),
- 5: **for** $j = 1, \dots, 25$ **do**
- 6: Update ϵ_j with \mathbf{w}_{j-1} using Eq. (9),
- 7: Update structural correlation filters \mathbf{w}_j with \mathbf{w}_{j-1} , Φ_1 , ϵ_j and \mathbf{y}_1 using Eq. (5), Eq. (9) and Eq. (12),
- 8: **end for**
- 9: $\mathbf{w}_0 \leftarrow \mathbf{w}_{25}$,
- 10: **for** $i = 2, \dots, T$ **do**
- 11: \triangleright Estimate the target position and size from Δ different scaled searching samples
- 12: **for** $\delta = 1, \dots, \Delta$ **do**
- 13: Crop searching samples \mathbf{x}_i of size $M_{i-1}^\delta \times N_{i-1}^\delta$ from \mathbf{I}_i at the previously estimated target position \mathbf{y}_{i-1} ,
- 14: Extract L layers continuous-domain deep feature maps $\{\Phi_{i,l}\}_{l=1}^L$ of searching samples \mathbf{x}_i using Eq. (4)s,
- 15: **for** $l = 1, \dots, L$ **do**
- 16: Calculate the primal confidence score map S_l of each layer continuous-domain deep features map $\Phi_{i,l}$ with $\mathbf{w}_{0,l}$ using circular and structural operator S_w as Eq. (5),
- 17: **end for**
- 18: Calculate the scaled optimal confidence score map R^δ using Eq. (19) and Eq. (21),
- 19: **end for**
- 20: Set \mathbf{y}_i to the position corresponding to the maximum score over all scaled optimal confidence score map R^Δ
- 21: $M_i \times N_i \leftarrow M_{i-1}^\delta \times N_{i-1}^\delta$, where R^δ contains the maximum confidence score,
- 22: Crop training samples \mathbf{x}_i of size $M_i \times N_i$ from \mathbf{I}_i at the current estimated target position \mathbf{y}_i ,
- 23: Extract L layers continuous-domain deep feature maps $\{\Phi_{i,l}\}_{l=1}^L$ of the search region \mathbf{x}_i using Eq. (4),
- 24: \triangleright Update structural correlation filters using three collaborative optimization iterations
- 25: **for** $j = 1, \dots, 3$ **do**
- 26: Update ϵ_j with \mathbf{w}_{j-1} using Eq. (9),
- 27: Update structural correlation filters \mathbf{w}_j with \mathbf{w}_{j-1} , Φ_i , ϵ_j and \mathbf{y}_i using Eq. (5), Eq. (9) and Eq. (12),
- 28: **end for**
- 29: $\mathbf{w}_0 \leftarrow \mathbf{w}_3$,
- 30: **end for**

Algorithm 1. The proposed approach.

as the VOT committee suggested [61]: (a) the accuracy, which measures the average overlap between the estimated bounding box and the ground truth, (b) the robustness, which measures the average failure times during tracking, (c) the expected average overlap (EAO), which measures the average overlap a tracker is expected to attain without re-initialization following a failure on a large collection of short-term sequences. Trackers are ranked according to EAO scores on the VOT2017 benchmark.

4.3. Ablation experiments

To verify the effectiveness of each component proposed in Section 3, we implement and evaluate six variants of our approach on OTB50/2013/2015 benchmarks, which are respectively named as (a) CSOT-HC, (b) CSOT-KHC, (c) CSOT-DCNN, (d) CSOT-CNN, (e) CSOT-CNN2 and (f) CSOT.

Characteristics of all variants are summarized in Table 1. CSOT-HC without ensemble post-processor is implemented as the baseline, where handcrafted features such as HOG and CN are exploited. CSOT-KHC is developed using a Gaussian kernel with conventional handcrafted features. CSOT-CNN and CSOT-DCNN, without the ensemble post-processor, exploit only deep appearance features. CSOT-CNN2 is designed by incorporating deep motion features into CSOT-CNN. CSOT is implemented by integrating the ensemble post-processor into CSOT-CNN2. A noteworthy is CSOT, CSOT-CNN2, CSOT-CNN, CSOT-KHC and CSOT-HC employ both spatial regularization and implicit interpolation operators to incorporate continuous-domain features, while CSOT-DCNN exploits an explicit resampling strategy to extract features.

Success plots of CSOT and its variants on OTB2015 (with 100 sequences), OTB2013 (with 51 sequences) and OTB50 (with 50 sequences) benchmarks are illustrated in Fig. 3. Experimental results show that CSOT-HC gives AUC scores of 63.1%, 64.4% and 58.3% on OTB2015, OTB2013 and OTB50 benchmarks, respectively. For the sake of clarity, the mean AUC score of these three OTB benchmarks is exploited as the evaluation metric in the following evaluations. Therefore, CSOT-HC obtains a mean AUC score of 61.9%. Interestingly, CSOT-KHC obtains a mean AUC score of 62.7%, the nonlinear extension is not helpful to improve results satisfactorily and reduces the tracking speed from 47.1 FPS to 21.6 FPS. Meanwhile, CSOT-CNN obtains a mean AUC score of 66.8% which is 4.9% higher than that of CSOT-HC. It demonstrates that deep appearance features can improve tracking performance obviously than conventional handcrafted features. We also employ CSOT-CNN to evaluate the efficiency of multi-resolution interpolation methods, it provides an absolute gain of 2.1% compared with the discrete resampling version, i.e., CSOT-DCNN. By involving deep motion features to the CSOT-CNN, the mean AUC score of CSOT-CNN2 can be increased by 6.0% compared to CSOT-HC. According to our analysis, it is apparent that involving deep motion features can boost tracking performance. The best result is provided by CSOT. Compared with CSOT-HC, CSOT achieves an absolute gain of 6.7% in the mean AUC score. The results clearly demonstrate that deep appearance features and deep motion features are complementary, and lead to obtain the best results. Deep features extracted from only a single network have limited diversity, which is not sufficient to deal with various challenges. Moreover, CSOT adopts the ensemble post-processor to

enhance diversity, which can enhance the performance to address problems of various challenging factors substantially.

For detailed analyses, we employ CSOT to evaluate the impact of the online collaborative optimization on the OTB2013 benchmark. Fig. 4 shows the convergence plot of initial online collaborative optimization and the success plot of different CG-iterations during online collaborative optimizations. It can be observed that after 25 iterations of online collaborative optimization in the first frame, the relative residual falls close to zero, and this can lead to the desirable initial estimate of structural correlation filters. The success plot illustrates that after three iterations of the online collaborative optimization, where each optimization involves two CG-iterations, the AUC score tends to converge on the OTB2013 benchmark.

Obviously, all experiment results demonstrate that those methods we proposed in Section 3 are reliable and effective. We will compare CSOT with many state-of-the-art tracking approaches in the following evaluations.

4.4. Experiments on OTB

We compare CSOT with nine state-of-the-art trackers: ECO [31], C-COT [30], DeepLMCF [52], DeepSRDCF [42], DLSSVM [16], MEEM [13], Staple [35], MFCMT [21] and Struck [7] on OTB50/2013/2015 benchmarks. Among them, DeepLMCF and DeepSRDCF use deep appearance features as a substitute to conventional handcrafted features employed by LMCF and SRDCF. Struck, DLSSVM, DeepLMCF and MEEM are all SOSVM-based methods. ECO and C-COT are the most popular approaches based on correlation filters and continuous convolution operations. It is worth to mention that MEEM, Staple and MFCMT are three different ensemble-based trackers. The success plots and precision plots of all participating trackers are shown in Fig. 5.

The top row of Fig. 5 presents success plots of all participating trackers. Struck which is the best tracker in the original OTB benchmark [15] obtains a mean AUC score of 43.9%. MEEM provides a mean AUC score of 52.3%. DLSSVM further improves the mean AUC score to 55.8%. Ensemble-based trackers which employing conventional handcrafted features, i.e., Staple and MFCMT, achieve mean AUC scores of 56.3% and 57.2%, respectively. The deep features-based versions of SRDCF and LMCF, namely DeepSRDCF and DeepLMCF, have gained improvements of 4.0% and 4.9% compared to MFCMT, respectively. By employing continuous convolution, C-COT achieves a mean AUC score of 65.2%. ECO provides the best results with a mean AUC score of 68.1%. CSOT obtains state-of-the-art results with a mean AUC score of 68.6% on OTB benchmarks, which is a significant gain of 0.5% over the mean AUC score of ECO. The precision plots of these ten trackers are shown in the bottom row of Fig. 5. Among them, ECO provides the best performance with a mean DP score of 90.4%. Our CSOT approach significantly outperforms ECO and obtains a state-of-the-art result with an absolute gain of 1.5% in mean DP score. Moreover, ECO and C-COT can only run at speeds of 1.1 FPS and 0.2 FPS while our CSOT is superior to them in all evaluations. Interestingly, we notice that CSOT-HC which implemented based on conventional handcrafted features obtains a mean AUC score of 61.9%. It outperforms most existing trackers, even DeepLMCF and DeepSRDCF. All these experimental results illustrate that our proposed approach is effective and achieves appealing

Table 1
Characteristics of CSOT and its variants.

Variants	CSOT-HC	CSOT-KHC	CSOT-DCNN	CSOT-CNN	CSOT-CNN2	CSOT
Deep appearance features	—	—	✓	✓	✓	✓
Deep motion features	—	—	—	—	✓	✓
Handcrafted features	✓	✓	—	—	—	—
Multi-resolution features	✓	✓	—	✓	✓	✓
Ensemble post-processor	—	—	—	—	—	✓
Nonlinear kernel	—	✓	—	—	—	—

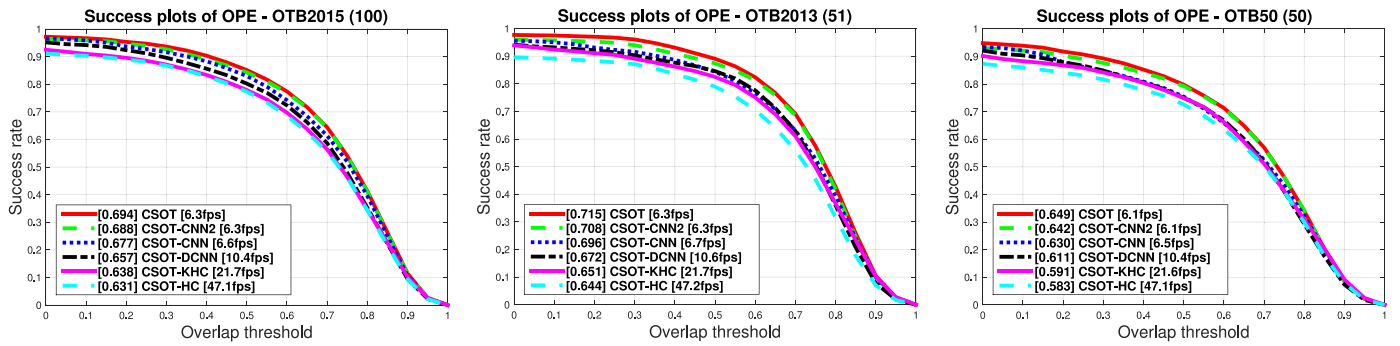


Fig. 3. Success plots of CSOT and its variants on OTB benchmarks [15,22]. The first value in the legend indicates the AUC score for each tracker. We also use the last number in the legend to present the mean speed of each tracker. Best viewed in color.

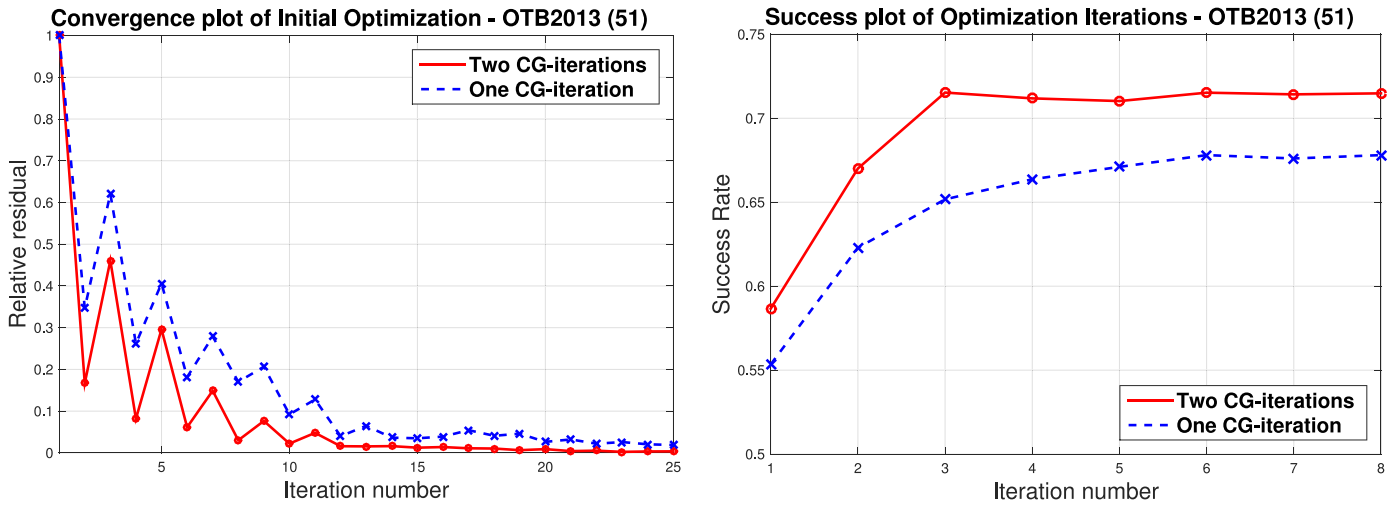


Fig. 4. Convergence plot of initial online collaborative optimization (left) and success plot of different CG-iterations during online collaborative optimizations (right) on the OTB2013 benchmark.

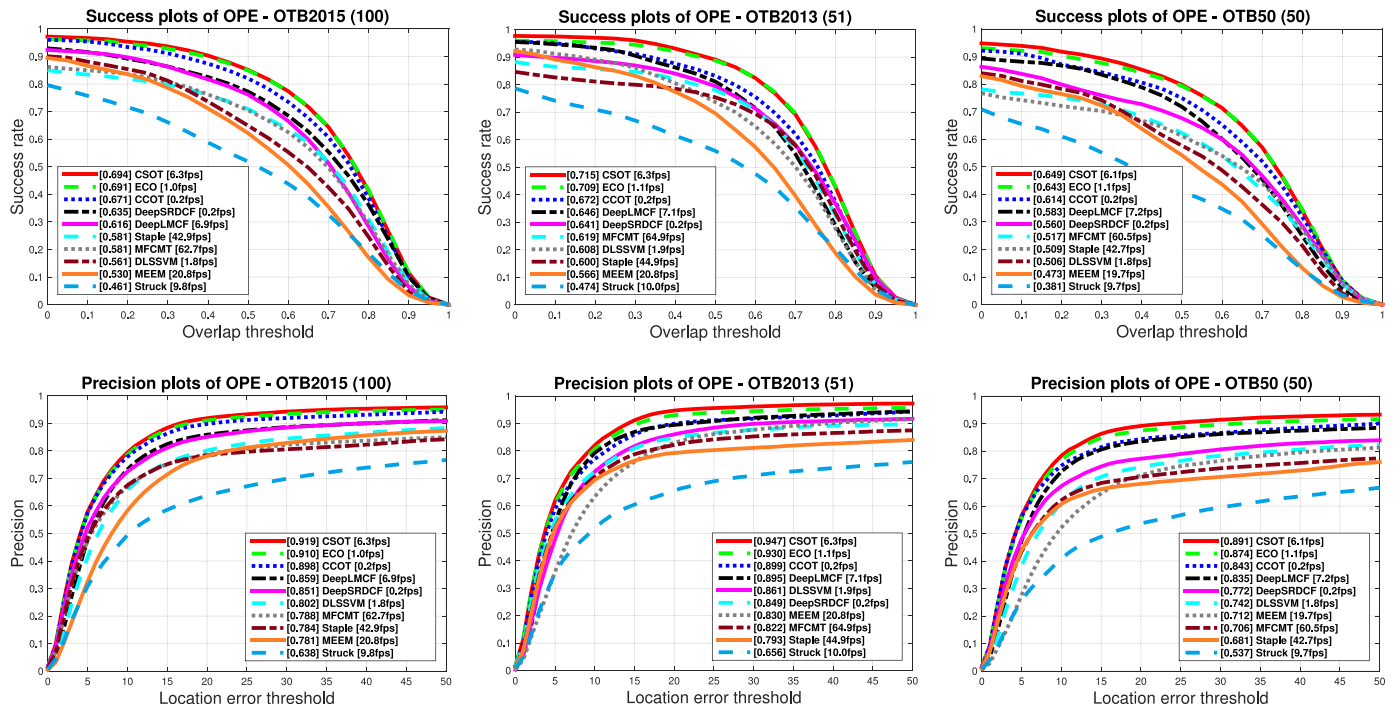


Fig. 5. Success plots (top row) and precision plots (bottom row) of CSOT and nine state-of-the-art trackers over three OTB benchmarks. The first value in the legend indicates the AUC or DP score for each tracker. We also show the speed of the trackers in mean FPS as the last number in the legend. Best viewed in color.

Table 2

Attribute-based comparison of CSOT with nine state-of-the-art trackers on the OTB2015 benchmark [22]. The results are presented in terms of AUC scores (%). Top three results are highlighted in red, blue and green, respectively. Best viewed in color.

Attribute	CSOT(ours)	ECO [31]	CCOT [30]	DeepSRDCF [42]	DeepLMCF [52]	DLSSVM [16]	MEEM [13]	Struck [7]
Fast motion (39)	67.8	68.3	67.5	62.8	60.4	53.9	54.2	46.7
Background clutter (31)	69.2	70.0	64.5	62.7	60.8	54.7	51.9	43.3
Motion blur (29)	70.7	70.9	70.3	64.2	62.7	57.6	55.6	46.3
Deformation (44)	64.1	63.3	61.5	56.6	56.2	51.3	48.9	38.3
Illumination variation (38)	72.2	71.3	67.6	62.1	62.5	56.1	51.7	42.0
In-plane rotation (51)	66.2	65.5	62.3	58.9	59.8	55.6	42.9	45.1
Low resolution (9)	61.3	59.1	62.5	56.1	56.8	43.6	38.2	31.3
Occlusion (49)	62.7	68.0	67.0	60.1	58.9	53.3	50.4	39.1
Out-of-plane rotation (63)	63.6	67.3	64.9	60.7	60.5	54.7	52.5	42.4
Out of view (14)	65.8	66.0	64.5	55.3	60.8	46.7	48.8	37.4
Scale variation (64)	67.1	66.6	65.1	60.5	58.8	48.8	47.0	40.2

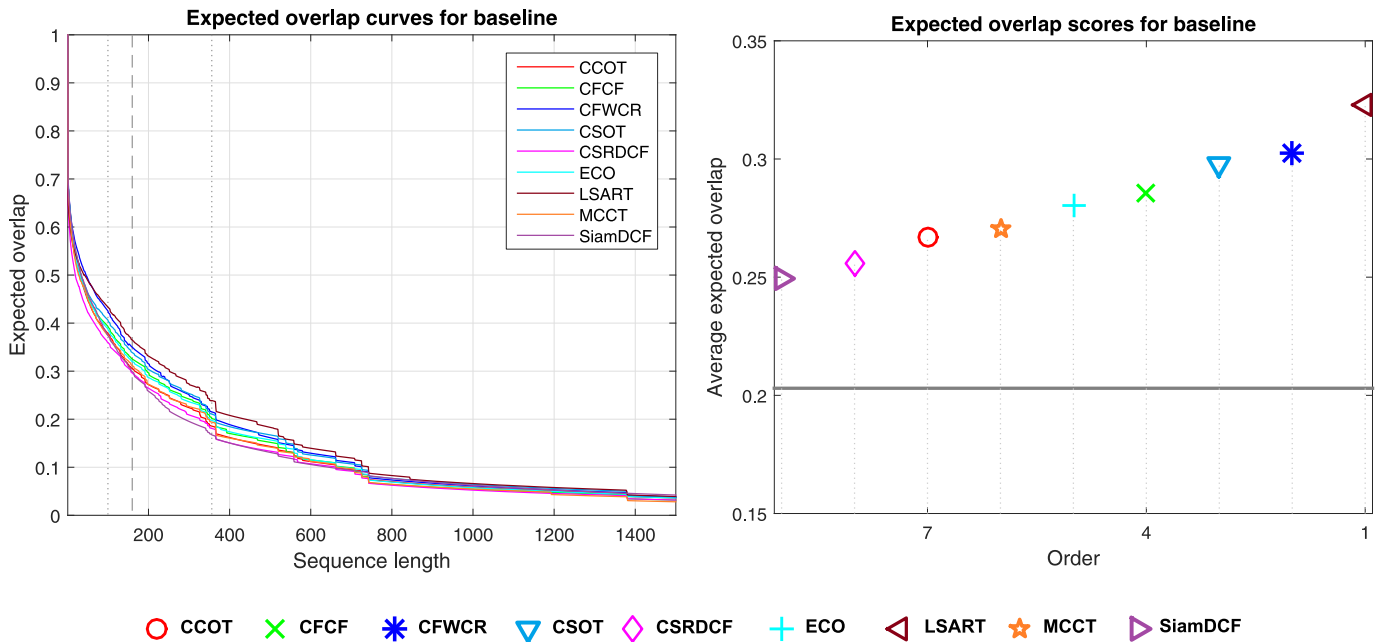


Fig. 6. EAO curve (left) and EAO graph (right) of all participating trackers for experiment baseline on the VOT2017 benchmark [23]. In the EAO graph, the better performance a tracker achieves, the closer to the right of the graph. The horizontal dashed line denotes the state-of-the-art bound in VOT2017 challenge. Best viewed in color.

Table 3

Comparison of state-of-the-art trackers on the VOT2017 benchmark [23]. Performance results are presented in terms of EAO, Accuracy and Robustness. Top three results of each metric are highlighted in red, blue and green respectively.

Tracker	EAO	Accuracy	Robustness
C-COT [30]	0.267	0.493	1.315
CFCF [62]	0.286	0.509	1.169
CFWCR [54]	0.303	0.484	1.210
CSOT (ours)	0.298	0.498	1.089
CSRDCF [63]	0.256	0.488	1.309
ECO [31]	0.281	0.483	1.117
LSART [64]	0.323	0.493	0.943
MCCT [23]	0.270	0.525	1.126
SiamDCF [23]	0.250	0.500	1.866

performance.

For more detailed comparisons, we perform attribute-based analysis of CSOT and nine state-of-the-art trackers on the OTB2015 benchmark. Video sequences contained in OTB benchmarks are annotated with 11 different attributes that represent a variety of challenging factors including fast motion, background clutter, motion blur, deformation,

illumination variation, in-plane rotation, low resolution, occlusion, out-of-plane rotation, out of view and scale variation. The results of AUC scores are summarized in Table 2. It is clear that with the exception of occlusions and out-of-plane rotations, CSOT obtains top two results compared to other trackers on nine out of eleven attributes. Moreover, CSOT achieves significant improvements in four scenarios compared to

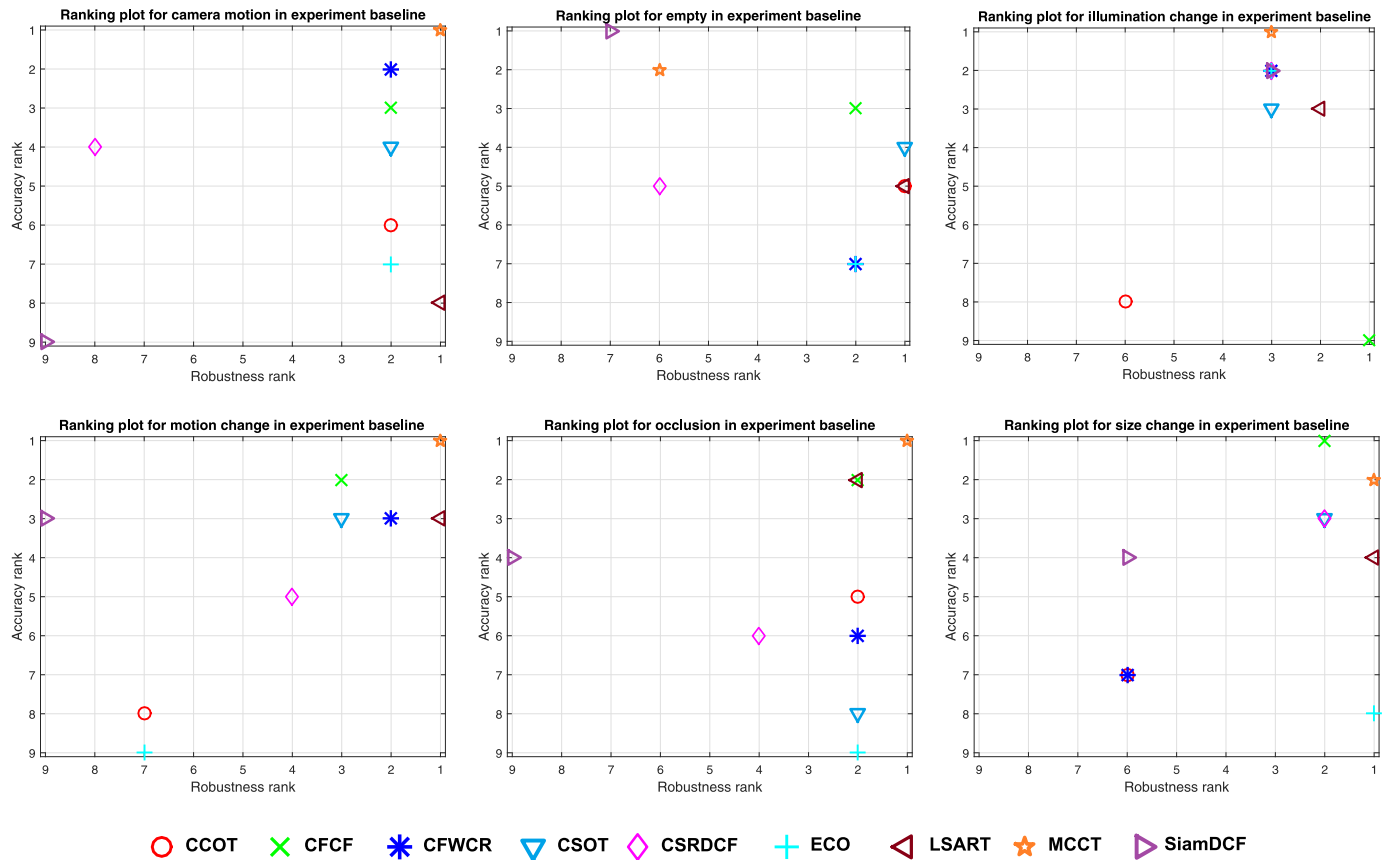


Fig. 7. Ranking plots of the compared trackers in experiment baseline for each attribute (i.e. *camera motion*, *empty*, *illumination change*, *motion change*, *occlusion* and *size change*) on the VOT2017 benchmark [23]. Here, *empty* denotes frames without labeled attribute. The better performance a tracker achieves, the closer to the top-right corner. Best viewed in color.

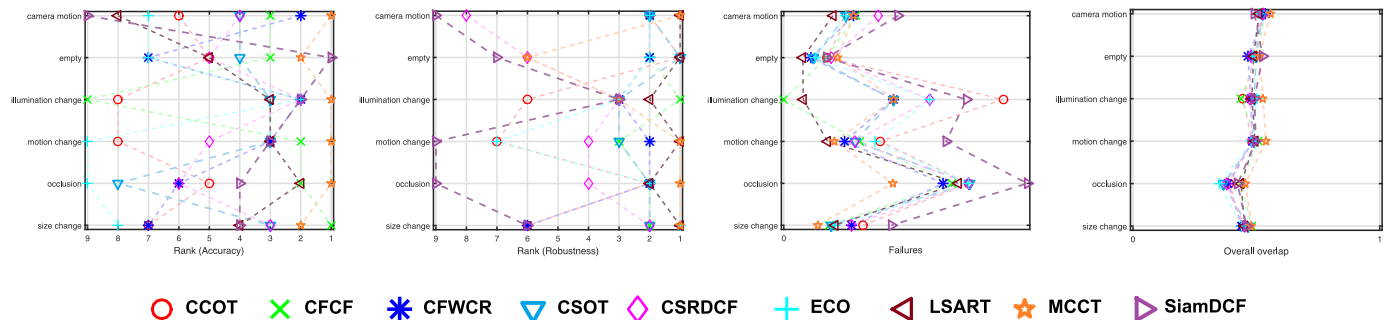


Fig. 8. Ranking of all participating trackers in experiment baseline for each attribute (i.e. *camera motion*, *empty*, *illumination change*, *motion change*, *occlusion* and *size change*) on the VOT2017 benchmark [23]. Here, *empty* denotes frames without the labeled attribute. The better performance a tracker achieves, the closer to the right of the plot. Best viewed in color.

the best existing method and provides the best performance: deformations (0.8%), illumination variations (0.9%), in-plane rotations (0.7%) and scale variations (0.5%). These experimental evaluations illustrate that the combination of multiple deep features and the ensemble post-processor have discriminative capabilities superior to other approaches.

4.5. Experiments on VOT

The VOT challenge is the largest annual competition in the field of visual tracking. We use the VOT2017 benchmark [23] which contains 60 video sequences to evaluate our proposed CSOT and eight state-of-the-art trackers, including C-COT [30], CFCF [62], CFWR [54], CSRDCF [63], ECO [31], LSART [64], MCCT [23] and SiamDCF [23]. For fair comparisons, we use the original results provided by the VOT

challenge committee.⁵

To evaluate overall performance of all participating trackers, we adopt the EAO curve and EAO graph to visualize tracking performance as shown in Fig. 6. The EAO curve demonstrates the average of per-frame overlaps, including the zero overlaps after failures, on different lengths of long sequences. The EAO graph illustrates the EAO score of each tracker, which is computed as the average of the EAO curve values over an interval [108, 371] [65] of typical short-term sequence lengths. In the Expected Average Overlap graph, the gray horizontal line indicates the state-of-the-art bound suggested by the VOT2017 challenge committee [23].

⁵ <http://www.votchallenge.net/vot2017/>

Table 3 shows comparison results on the VOT2017 benchmark. Among compared trackers, LSART achieves the most favorable results in terms of EAO and robustness, but its accuracy is unsatisfactory. MCCT achieves the best accuracy of 52.5%. It is clear that CSOT is ranked as the third-best with an EAO score of 29.8%. Compared with ECO and C-COT, CSOT achieves absolute gains of 1.7% and 2.9%, respectively. Moreover, CSOT obtains the second-best robustness of 1.089 and the fourth-best accuracy of 49.8%.

In the VOT2017 benchmark, all sequences are labeled with five different attributes: camera motion, illumination change, motion change, occlusion and size change. For a more extensive comparison, we further compare each attribute of all participating trackers on the corresponding sub-dataset of VOT2017 benchmark, results are shown in Fig. 7. It can be seen that CSOT obtains satisfactory results on all attributes except occlusion. Especially for the empty attribute, our method achieves the best robustness. With regard to camera motion, occlusion and size change challenges, our approach gets the second-best robustness. CSOT also provides the third-best robustness and accuracy with regard to motion change and illumination change.

Fig. 8 individually visualizes the ranking of all participating trackers in terms of accuracy, robustness, failures and overall overlap for each attribute. Our approach achieves promising results on all five attributes. Compared with ECO and C-COT, all metrics are significantly improved. According to Fig. 8, it is shown that CSOT is effective and robust to deal with various challenging attributes.

5. Conclusions

In this paper, we have proposed a novel circular and structural operator tracker (CSOT) for high performance visual tracking. The superior computational efficiency of DCF complemented with the powerful discriminative capability of SOSVM make it possible to employ higher-dimensional deep features and denser circular samples. To obtain the primal confidence score maps, we utilize circular and structural operators to circular correlate deep complementary features with structural correlation filters. The resulting tracker greatly benefits from the heterogeneity of multiple deep features. To improve tracking performance, we propose an ensemble post-processor based on relative entropy, which fuses primal confidence score maps to get an optimal confidence score map. Besides, we utilize an online collaborative optimization strategy to efficiently update circular and structural operators by training structural correlation filters. Experimental evaluations clearly demonstrate that our CSOT outperforms far above most state-of-the-art trackers both in terms of accuracy and robustness on OTB benchmarks, and it also achieves impressive results on the VOT2017 benchmark.

Acknowledgments

This work was supported in part by the Science and Technology Planning Program of Guangdong Province, China under Grant 2013B090600105 and Grant 2016B090918047. The authors would like to thank all the anonymous reviewers for their insightful comments and suggestions that have helped to significantly improve the quality of this paper.

References

- [1] N. Wang, J. Shi, D.-Y. Yeung, J. Jia, Understanding and diagnosing visual tracking systems, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2015, pp. 3101–3109.
- [2] A.W. Smeulders, D.M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, M. Shah, Visual tracking: an experimental survey, *IEEE Trans. Pattern Anal. Mach. Intell.* 36 (7) (2014) 1442–1468.
- [3] T. Vojir, J. Noskova, J. Matas, Robust scale-adaptive mean-shift for tracking, *Pattern Recognit. Lett.* 49 (2014) 250–258.
- [4] M. Godec, P.M. Roth, H. Bischof, Hough-based tracking of non-rigid objects, *Comput. Vis. Image Underst.* 117 (10) (2013) 1245–1256.
- [5] H. Possegger, T. Mauthner, H. Bischof, In defense of color-based model-free tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2015, pp. 2113–2120.
- [6] Z. Kalal, K. Mikolajczyk, J. Matas, Tracking-learning-detection, *IEEE Trans. Pattern Anal. Mach. Intell.* 34 (7) (2012) 1409–1422.
- [7] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S.L. Hicks, P.H. Torr, Struck: structured output tracking with kernels, *IEEE Trans. Pattern Anal. Mach. Intell.* 38 (10) (2016) 2096–2109.
- [8] K. Zhang, L. Zhang, M.-H. Yang, D. Zhang, Fast tracking via spatial-temporal context learning, in: Proceedings of the European Conference on Computer Vision (ECCV), Springer, 2012, pp. 127–141.
- [9] J.F. Henriques, R. Caseiro, P. Martins, J. Batista, High-speed tracking with kernelized correlation filters, *IEEE Trans. Pattern Anal. Mach. Intell.* 37 (3) (2015) 583–596.
- [10] M. Mueller, N. Smith, B. Ghanem, Context-aware correlation filter tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017, pp. 1387–1395.
- [11] X. Ma, Q. Liu, W. Ou, Q. Zhou, Visual object tracking via coefficients constrained exclusive group lasso, *Mach. Vis. Appl.* (2018) 1–15.
- [12] S. Avidan, Support vector tracking, *IEEE Trans. Pattern Anal. Mach. Intell.* 26 (8) (2004) 1064–1072.
- [13] J. Zhang, S. Ma, S. Sclaroff, Meem: robust tracking via multiple experts using entropy minimization, in: Proceedings of the European Conference on Computer Vision (ECCV), Springer, 2014, pp. 188–203.
- [14] W. Zuo, X. Wu, L. Lin, L. Zhang, M.-H. Yang, Learning support correlation filters for visual tracking, *arXiv:1601.06032* (2016).
- [15] Y. Wu, J. Lim, M.-H. Yang, Online object tracking: a benchmark, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2013, pp. 2411–2418.
- [16] J. Ning, J. Yang, S. Jiang, L. Zhang, M.-H. Yang, Object tracking via dual linear structured SVM and explicit feature map, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 4266–4274.
- [17] J.F. Henriques, R. Caseiro, P. Martins, J. Batista, Exploiting the circulant structure of tracking-by-detection with kernels, in: Proceedings of the European conference on computer vision (ECCV), Springer, 2012, pp. 702–715.
- [18] M. Danelljan, F. Shahbaz Khan, M. Felsberg, J. Van de Weijer, Adaptive color attributes for real-time visual tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2014, pp. 1090–1097.
- [19] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, D. Tao, Multi-store tracker (mster): A cognitive psychology inspired approach to object tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2015, pp. 749–758.
- [20] Q. Liu, X. Lu, Z. He, C. Zhang, W.-S. Chen, Deep convolutional neural networks for thermal infrared object tracking, *Knowl. Based Syst.* 134 (2017) 189–198.
- [21] P. Gao, Y. Ma, K. Song, C. Li, F. Wang, L. Xiao, A complementary tracking model with multiple features, *arXiv:1804.07459* (2018).
- [22] Y. Wu, J. Lim, M.-H. Yang, Object tracking benchmark, *IEEE Trans. Pattern Anal. Mach. Intell.* 37 (9) (2015) 1834–1848.
- [23] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Čehovin, T. Vojir, et al., The visual object tracking VOT2017 challenge results, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), IEEE, 2017, pp. 1949–1972.
- [24] Y. Li, J. Zhu, A scale adaptive kernel correlation filter tracker with feature integration, in: Proceedings of the European Conference on Computer Vision (ECCV), Springer, 2014, pp. 254–265.
- [25] M. Danelljan, G. Häger, F.S. Khan, M. Felsberg, Discriminative scale space tracking, *IEEE Trans. Pattern Anal. Mach. Intell.* 39 (8) (2017) 1561–1575.
- [26] C. Ma, X. Yang, C. Zhang, M.H. Yang, Long-term correlation tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2015, pp. 5388–5396.
- [27] C. Ma, J. Huang, X. Yang, M.H. Yang, Adaptive correlation filters with long-term and short-term memory for object tracking, *arXiv:1707.02309* (2017).
- [28] H.K. Galoogahi, A. Fagg, S. Lucey, Learning background-aware correlation filters for visual tracking, *arXiv:1703.04590* (2017).
- [29] M. Danelljan, G. Häger, F.S. Khan, M. Felsberg, Learning spatially regularized correlation filters for visual tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2015, pp. 4310–4318.
- [30] M. Danelljan, A. Robinson, F.S. Khan, M. Felsberg, Beyond correlation filters: Learning continuous convolution operators for visual tracking, in: Proceedings of the European Conference on Computer Vision (ECCV), Springer, 2016, pp. 472–488.
- [31] M. Danelljan, G. Bhat, S.F. Khan, M. Felsberg, ECO: efficient convolution operators for tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017.
- [32] D.S. Bolme, J.R. Beveridge, B.A. Draper, Y.M. Lui, Visual object tracking using adaptive correlation filters, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 2544–2550.
- [33] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, *IEEE Trans. Pattern Anal. Mach. Intell.* 32 (9) (2010) 1627–1645.
- [34] J. Van De Weijer, C. Schmid, J. Verbeek, D. Larlus, Learning color names for real-world applications, *IEEE Trans. Image Process.* 18 (7) (2009) 1512–1523.
- [35] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, P.H. Torr, Staple: Complementary learners for real-time tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 1401–1409.
- [36] P. Gao, Y. Ma, C. Li, K. Song, Y. Zhang, F. Wang, L. Xiao, Adaptive object tracking

- with complementary models, *IEICE Trans. Inf. Syst.* E101-D (11) (2018).
- [37] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2012, pp. 1097–1105.
- [38] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, *arXiv:1409.1556v6*(2015).
- [39] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2014.
- [40] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, A.C. Berg, SSD: single shot multibox detector, in: *Proceedings of the European Conference on Computer Vision (ECCV)*, Springer, 2016.
- [41] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, The visual object tracking VOT2016 challenge results, in: *Proceedings of the European Conference on Computer Vision (ECCV)*, Springer, 2016.
- [42] M. Danelljan, G. Häger, F.S. Khan, M. Felsberg, Convolutional features for correlation filter based visual tracking, in: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, IEEE, 2015, pp. 621–629.
- [43] S. Gladh, M. Danelljan, F.S. Khan, M. Felsberg, Deep motion features for visual tracking, *arXiv:1612.00615*(2016).
- [44] H.Y. Ju, M.H. Yang, K.J. Yoon, Interacting multiview tracker, *IEEE Trans. Pattern Anal. Mach. Intell.* 38 (5) (2016) 903–917.
- [45] Z. Hong, X. Mei, D. Prokhorov, D. Tao, Tracking via robust multi-task multi-view joint sparse representation, in: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2014.
- [46] B. Han, J. Sim, H. Adam, Branchout: regularization for online ensemble tracking with convolutional neural networks, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 521–530.
- [47] H. Li, Y. Li, F. Porikli, Convolutional neural net bagging for online visual tracking, *Comput. Vis. Image Underst.* 153 (2016) 120–129.
- [48] J. Kwon, K.M. Lee, Tracking by sampling and integrating multiple trackers, *IEEE Trans. Pattern Anal. Mach. Intell.* 36 (7) (2014) 1428–1441.
- [49] N. Wang, D. Yeung, Ensemble-based tracking: Aggregating crowdsourced structured time series data, in: *Proceedings of the International Conference on Machine Learning (ICML)*, 2014.
- [50] C. Rao, C. Yao, X. Bai, W. Qiu, W. Liu, Online random ferns for robust visual tracking, in: *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2012, pp. 1447–1450.
- [51] H. Nam, M. Baek, B. Han, Modeling and propagating cnns in a tree structure for visual tracking, *arXiv:1608.07242*(2016).
- [52] M. Wang, Y. Liu, Z. Huang, Large margin object tracking with circulant feature maps, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2017.
- [53] G. Gkioxari, M. Jitendra, Finding action tubes, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2015.
- [54] Z. He, Y. Fan, J. Zhuang, Y. Dong, H. Bai, Correlation filters with weighted convolution responses, in: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, IEEE, 2017.
- [55] P. Gao, Y. Ma, K. Song, C. Li, F. Wang, L. Xiao, Large margin structured convolution operator for thermal infrared object tracking, in: *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2018.
- [56] H. Nam, B. Han, Learning multi-domain convolutional neural networks for visual tracking, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2016, pp. 4293–4302.
- [57] L. Bertinetto, J. Valmadre, J.A. F. Henriques, A. Vedaldi, P.H.S. Torr, Fully-convolutional siamese networks for object tracking, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2017.
- [58] P.J. Davis, *Circulant matrices*, Am. Math. Soc., 1994.
- [59] R.M. Gray, *Toeplitz and Circulant Matrices: A Review*, Now Publishers Inc., 2006.
- [60] A. Vedaldi, K. Lenc, Matconvnet - convolutional neural networks for matlab, *arXiv:1412.4564v3* (2014).
- [61] M. Kristan, J. Matas, A. Leonardis, T. Vojř, R. Pflugfelder, G. Fernandez, G. Nebehay, F. Porikli, L. Čehovin, A novel performance evaluation methodology for single-target trackers, *IEEE Trans. Pattern Anal. Mach. Intell.* 38 (11) (2016) 2137–2155.
- [62] E. Gundogdu, A. Alatan, Good features to correlate for visual tracking, *arXiv:1704.06326* (2017).
- [63] A. Lukežič, T. Vojř, L. Čehovin, J. Matas, M. Kristan, Discriminative correlation filter with channel and spatial reliability, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2017, pp. 4847–4856.
- [64] C. Sun, H. Lu, M.-H. Yang, Learning spatial-aware regressions for visual tracking, *arXiv:1706.07457* (2017).
- [65] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Čehovin, R. Pflugfelder, et al., The visual object tracking VOT2015 challenge results, in: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, IEEE, 2015, pp. 1949–1972.