Dynamic Saliency-Aware Regularization for Correlation Filter-Based Object Tracking

Wei Feng, *Member, IEEE*, Ruize Han, Qing Guo, Jianke Zhu, *Senior Member, IEEE*, and Song Wang, *Senior Member, IEEE*

***Abstract*— With a good balance between tracking accuracy and speed, correlation filter (CF) has become one of the best object tracking frameworks, based on which many successful trackers have been developed. Recently, spatially regularized CF tracking (SRDCF) has been developed to remedy the annoying boundary effects of CF tracking, thus further boosting the tracking performance. However, SRDCF uses a fixed spatial regularization map constructed from a loose bounding box and its performance inevitably degrades when the target or background show significant variations, such as object deformation or occlusion. To address this problem, we propose a new dynamic saliency-aware regularized CF tracking (DSAR-CF) scheme. In DSAR-CF, a simple yet effective energy function, which reflects the object saliency and tracking reliability in the spatial–temporal domain, is defined to guide the online updating of the regularization weight map using an efficient level-set algorithm. Extensive experiments validate that the proposed DSAR-CF leads to better performance in terms of accuracy and speed than the original SRDCF.**

***Index Terms*— Correlation filter, object tracking, saliency, dynamic spatial regularization, level-set optimization.**

I. Introduction

VISUAL object tracking is a very important task in computer vision which is widely used in robotic service, human motion analyses, autonomous driving and many other applications. The main challenge of object tracking comes from a variety of target's unpredictable transformations over time. Such transformations, which occur under object occlusions, object deformations, background clutters, and many other scenarios, may cause the failure of identifying the tracking target in the new frame based on its appearance features in the previous frames.

In recent years, correlation filter (CF) has been used in visual object tracking with a good balance between tracking accuracy and speed. In CF tracking, the target in a new frame is located by a circular convolution operation, which can be implemented by efficient Fast Fourier Transform (FFT). One issue in CF tracking is the undesired boundary effects introduced by the circular shifts of training samples [3], that usually lead to degraded tracking performance [16]. To alleviate the boundary effects, a spatial regularization component can be incorporated to penalize the CF values, resulting in spatially regularized CF tracking (SRDCF) [9]. In SRDCF, the regularization component requires the definition of a spatial weight map to penalize the CF values in non-target regions. In practice, this weight map is simply computed using each pixel's distance to the region center and does not change over time after being initialized in the first frame. However, in real-world tracking tasks, object shape is usually non-centrosymmetrical and irregular in the tracking process and may change frequently over time. From this perspective, it is unconscionable to define a constant regularization weight map using only the spatial distance to the map center since it is likely to learn much undesired background information in the CF filtering. In this paper, we consider object shape and variation information into the regularization component to more accurately penalize the filter values outside the object boundary and overcome the limitation of the fixed regularization weight map in SRDCF [9].

More specifically, we propose a dynamic saliency-aware regularized CF tracking (DSAR-CF), which introduces the saliency information and dynamic variations into the regularization component. As shown in Figure 1, a basketball player keeps running with varying shape along a video and is represented by a bounding box, which contains much background information, in each frame. In the proposed DSAR-CF, we first introduce object saliency information into the regularization weight map to highlight the appearance of the player as well as suppressing the background information around the player in the first frame. We then propose a strategy to dynamically update the regularization weight map to reflect the player's shape variations in the subsequent frames. We then develop a level-set algorithm to iteratively optimize the regularization weight map in each frame. Experimental results show that the proposed DSAR-CF outperforms the baseline tracker SRDCF

Manuscript received May 4, 2018; revised October 1, 2018 and December 24, 2018; accepted January 10, 2019. Date of publication January 25, 2019; date of current version May 14, 2019. This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 61671325, Grant 61572354, Grant 61672376, and Grant U1803264. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Julian Fierrez. *(Wei Feng and Ruize Han contributed equally to this work.) (Corresponding author: Qing Guo.)*

W. Feng, R. Han, and Q. Guo are with the School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University, Tianjin 300350, China, and also with the Key Research Center for Surface Monitoring and Analysis of Cultural Relics, State Administration of Cultural Heritage, Tianjin 300350, China (e-mail: tsingqguo@tju.edu.cn).

J. Zhu is with the College of Computer Science and Technology, Zhejiang University, Hangzhou 310072, China, and also with the Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies, Zhejiang University, Hangzhou 310072, China.

S. Wang is with the School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University, Tianjin 300350, China, and also with the Key Research Center for Surface Monitoring and Analysis of Cultural Relics, State Administration of Cultural Heritage, China, and also with the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208 USA.

Digital Object Identifier 10.1109/TIP.2019.2895411

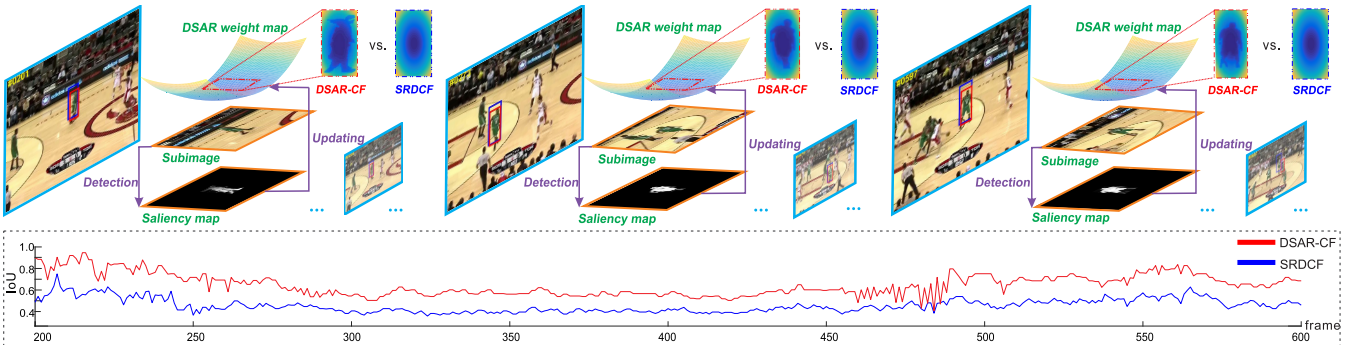


Fig. 1. An illustration of the proposed dynamic saliency-aware regularized CF tracking (DSAR-CF), where saliency information and dynamic shape variation are considered in regularization weight map. On the contrary, the traditional spatially regularized CF tracking (SRDCF) only considers the spatial distance and remains unchanged in tracking. At the bottom of the figure shows the intersection-over-union (IoU) curve between predicted and ground truth bounding boxes of target tracked by DSAR-CF and SRDCF, respectively.

and produces much better performance than several state-of-the-art trackers on standard benchmarks i.e. OTB-2013 [44], OTB-2015 [45] and VOT-2016 [27].

II. Related Work

In this section, we briefly overview the related work on CF tracking, spatially regularized CF tracking and other tracking methods.

A. Correlation Filter for Tracking

Correlation filter (CF) was used for object detection as early as 1980's. However, it was not applied to visual tracking until Bolme *et al.* [3] proposed the Minimum Output Sum of Squared Error (MOSSE) filter based tracker in 2010, which achieved the state-of-the-art performance on a benchmark with fast speed. CF shows two merits in tracking: 1) CF is able to make extensive use of limited training data by using circular-shift operations, and 2) the computational time for training and detection is significantly reduced by computing in the Fourier domain using FFT. In recent years, several subsequent CF tracking methods have shown continuous performance improvement on benchmarks. Two typical strategies have been used to obtain better performance in CF tracking – using more effective features and using the conceptual improvement in filter learning. In the first strategy, multi-channel feature maps [11], [23] were integrated to CF tracking. Henriques *et al.* [23] proposed a CF tracker with multi-channel HOG (Histogram of Oriented Gradient) features while maintaining a high algorithm speed. Danelljan *et al.* [11] applied multi-dimensional color attributes and Li and Zhu [29] applied feature combination for CF tracking. Recently, deep CNN (Convolutional Neural Network) based features have been applied to CF tracking [10], [31] and they further improved the performance but taking more computation time. In the second strategy, recent conceptual improvements in filter learning include non-linear kernelized correlation filter (KCF) proposed in [23], accurate scale estimation in Discriminative Scale-Space Tracking (DSST) [8], and color statistics integration in Sum of Template And Pixel-wise LEarners (Staple) [1]. Based on KCF and DSST, Siena and Vijaya Kumar [37] improved CF based trackers by adapting learning rate of correlation filter with the guidance of an occlusion detection strategy. Although such method helps improve tracking accuracy, it does not consider the main drawback of CF, i.e. boundary effects. Our method, i.e. DSAR-CF, is totally different from [37] and alleviates boundary effects by introducing a dynamic and saliency-aware regularization term into the objective function of CF. As a result, DSAR-CF significantly improves tracking accuracy of CF even under occlusion.

B. Spatial Regularization for CF Tracking

Recently, several methods [5], [7], [12], [15], [16], [18], [22], [30], [47], [48] were proposed to improve the CF tracking performance by highlighting the object appearance while suppressing the background interference. Galoogahi *et al.* [16] proposed a CF tracker with limited boundary (CFLB) to reduce the boundary effects in CF tracking. In [15], the background-aware correlation filter (BACF) based tracking was developed to learn CF from real negative training examples extracted from the background. Lukezic *et al.* [30] proposed discriminative correlation filter with channel and spatial reliability (CSRDCF) using the spatial reliability map to adapt the filter support to the part of object suitable for tracking. SRDCF [9] adopts a spatial regularization component to penalize CF values. More recently, based on SRDCF, Danelljan *et al.* [12] introduced a novel formulation for learning a convolution operator in the continuous spatial domain, which was further enhanced to tackle the problems of computational complexity and over-fitting simultaneously in [7]. Guo *et al.* [18] proposed a method to maintain two online transformations for both target and background.

C. Other Related Work on Tracking

Many non-CF trackers [19], [20], [24], [41], [49] were proposed to deal with various challenges in tracking, such as occlusion, non-rigid deformation, and background clutter. Wang *et al.* [41] used a multi-modal target detection technique to prevent model drift between similar objects or noisy background. A high-confidence updating is employed to avoid the

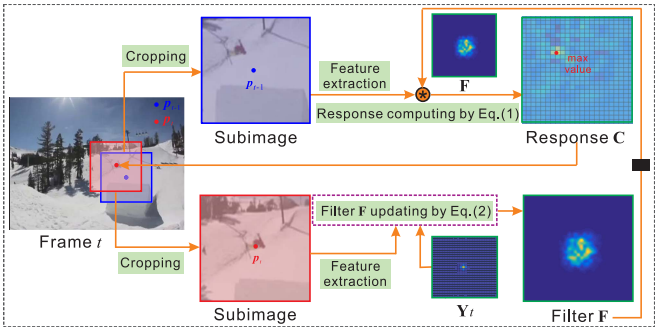


Fig. 2. An illustration of CF tracking from frame $t-1$ to t. The black block denotes a time delay operation.

model corruption once the target is severely occluded or totally missing. Guo *et al.* [19], [20] addressed the object rotation in tracking by incorporating structural regularization and online data-driven sampling. Zhang *et al.* [49] used the correlation particle filter for handling partial and total occlusions or large-scale variation. Huang *et al.* [24] proposed a part-based method that can capture the structure of target for alleviating the problem of object occlusion and deformation in tracking.

In this paper, our main idea is to integrate the object variation information into the spatial weight map to boost the performance of regularized CF tracking. We will include several state-of-the-art CF and non-CF tracking methods described above for our later comparison experiments, especially in challenging scenarios such as target deformation and occlusions.

III. Proposed Method

In this section, we first introduce the background on CF tracking and spatially regularized CF tracking, and then introduce the proposed saliency-aware regularized CF tracking.

A. CF Tracking [3]

As in many previous tracking models, the input of CF tracking is a video sequence, together with a tight bounding box B_1, which specifies the tracking target in the first frame. CF tracking then estimates B_t, the tight bounding box of the target in frames $t = 2, 3, \cdots$, sequentially. The basic idea of CF tracking is to alternately update an $M \times N$ correlation filter (CF) $\mathbf{F}$ and apply the updated CF $\mathbf{F}$ to locate the target frame by frame. Without loss of generality, let us assume that the target has been tracked in frames 1 through $t-1$ and now consider the tracking from frame $t-1$ to t. CF tracking consists of the following steps, as illustrated in Figure 2.

i) Define a target-search region in frame t. This is achieved by dilating the bounding box B_{t-1} derived in frame $t-1$ by a factor $K \geq 1$ and overlaying the dilated box in frame t as the search region.

ii) Compute the feature map of the search region. This is achieved by taking the subimage in frame t inside the dilated box, resizing the subimage into a pre-specified dimension, and applying a feature-extraction method to each pixel of the resized subimage to get a feature map $\mathbf{R}_t \in \mathbb{R}^{M \times N}$.

iii) Estimate the bounding box B_t in frame t. This is achieved by applying the current CF $\mathbf{F} \in \mathbb{R}^{M \times N}$ to the feature map, resulting the response map

$$\mathbf{C}_t = \mathbf{R}_t * \mathbf{F}, \tag{1}$$

where $*$ denotes the circular convolution, computing the peak location in $\mathbf{C}_t \in \mathbb{R}^{M \times N}$, taking the peak's original location in frame t as the target center, and constructing B_t to be of the same size as B_1, but around the identified target center in frame t.

iv) Update the CF $\mathbf{F}$. This is achieved by dilating the bounding box B_t by factor K, taking the subimage in frame t inside the dilated box, resizing the subimage to the pre-specified dimension, extracting a feature map $\mathbf{X}_t \in \mathbb{R}^{M \times N}$ for this subimage, all as in Step i), and all tracked frames are considered, then updating CF $\mathbf{F}$ by minimizing

$$\mathrm{E}_{\mathrm{CF}}(\mathbf{F}) = \sum_{k=1}^{t} \alpha_k \|\mathbf{X}_k * \mathbf{F} - \mathbf{Y}_k\|^2 + \|\mathbf{F}\|^2, \tag{2}$$

where $\mathbf{Y}_t \in \mathbb{R}^{M \times N}$ is a 2D Gaussian-shape response map with peak at its center and $\alpha_k \geq 0$ is the impact of frame k. This optimization problem can be explicitly solved with a closed-form solution in Fourier domain.

v) With the updated CF $\mathbf{F}$, go back to Step i) and track to the next frame $t+1$.

For simplicity, what we describe above is for one-channel feature map. In practice, it can be extended to multiple-channel feature maps by using multiple feature-extraction algorithms [17]. As discussed in [16], the circular convolution in Eq. (1) assumes the periodic shifts of the feature map $\mathbf{X}_t$, which introduces the undesired *boundary effects* to CF based tracking.

B. Spatially Regularized CF Tracking (SRDCF) [9]

SRDCF introduces a spatial regularization component within the CF formulation to address the problem of boundary effects. More specifically, in Step iv) of the above CF-tracking algorithm, the regularization term in Eq. (2) is replaced by a more general Tikhonov regularization in updating the filter $\mathbf{F}$ in frame t, i.e., Eq. (2) is extended to

$$\mathrm{E}_{\mathrm{SR}}(\mathbf{F}) = \sum_{k=1}^{t} \alpha_k \|\mathbf{X}_k * \mathbf{F} - \mathbf{Y}_k\|^2 + \|\mathbf{W} \odot \mathbf{F}\|^2, \tag{3}$$

where $\odot$ denotes the element-wise product.

For Eq. (3), SRDCF [9] suggests the use of $\mathbf{W} = \mathbf{W}_{\mathrm{SR}} \in \mathbb{R}^{M \times N}$, where

$$\mathbf{W}_{\mathrm{SR}}(i, j) = a + b\left(\frac{i - \frac{M}{2}}{\frac{w}{2}}\right)^2 + b\left(\frac{j - \frac{N}{2}}{\frac{h}{2}}\right)^2 \tag{4}$$

is a 2D quadratic-shape regularization map with $i = 1, \cdots, M$ and $j = 1, \cdots, N$. $a, b > 0$ are two pre-specified coefficients and w, h are the width and height of the target bounding box. In SRDCF, $\mathbf{W}_{\mathrm{SR}}$ is calculated in the first frame and does not change any more in the tracking process. Since the target is located at the center of the subimage cropped out in Step iv) of the above CF tracking algorithm, the use of $\mathbf{W}_{\mathrm{SR}}$

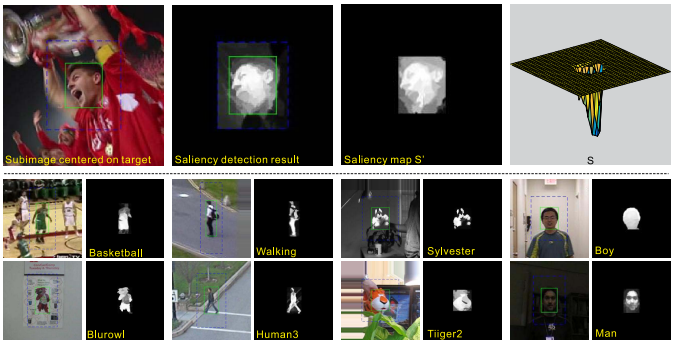


Fig. 3. First row presents meta results of saliency detection. Second row shows the saliency results of eight challenge cases.

suppresses the $\mathbf{F}$ values in the non-target region. An iterative approach based on the Gauss-Seidel algorithm can be used to minimize Eq. (3) for the correlation filter [9]. A scale estimation method [8] is also used in SRDCF to estimate the size of the bounding box in Step iii).

However, in real-world tracking tasks, the fixed weight map $\mathbf{W}_{\mathrm{SR}}$ in SRDCF is not adaptive to the shape irregularity and temporal change over time. In the following, we propose new saliency-aware regularizations into CF tracking to address this problem.

C. Static Saliency-Aware Regularized CF Tracking (SSAR-CF)

As discussed above, we attempt to generate a new target shape related weight map to handle the problem of shape irregularity. Saliency detection aims to detect the salient object and represent its shape with a binary image [25], [26], [39]. From this point, we derive the saliency map of the target and then incorporate it into the weight map $\mathbf{W}$, where we simply multiply the saliency map with the weight map to get a new weight map $\mathbf{W}_{\mathrm{S}}$, to better reflect the shape of the target. Specifically, as shown in first row of Figure 3, with the bounding box of a target, i.e. the green box, that is annotated in the first frame or estimated through target detection, i.e. step iii) in Section III-A, we first crop a region using the blue box, which centers at the green box and is κ times larger than it. We then perform saliency detection on the cropped region with an existing algorithm [35] and abandon the saliency results inside the blue box outside the green box to suppress the background clutters and distractors. We will discuss the advantage of such abandoning strategy in Section IV-D. Saliency result $\mathbf{S}'$ is then computed and used to generate saliency map $\mathbf{S}$. We present the saliency detection results of eight challenging cases in the second row of Figure 3. We then normalize $\mathbf{S}'$ to

$$\mathbf{S}''(i,j) = \frac{M_{\mathbf{S}} - \mathbf{S}'(i,j)}{M_{\mathbf{S}} - m_{\mathbf{S}}}, \tag{5}$$

where $M_{\mathbf{S}}$ and $m_{\mathbf{S}}$ are the maximum and minimum values of $\mathbf{S}'$. As a result, the elements of $\mathbf{S}''$ takes values in $[0, 1]$: target regions have values close to 0 and the surrounding non-target regions have values close to 1. We finally uniformly resize $\mathbf{S}''$ into an $M \times N$ saliency map $\mathbf{S}$ to make it consistent

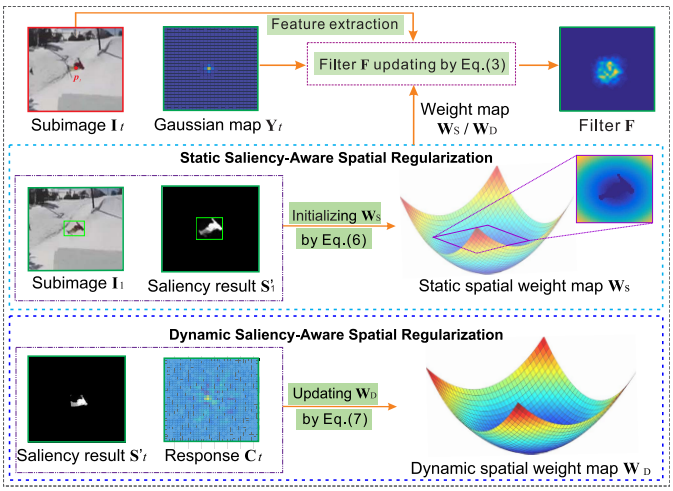


Fig. 4. We propose the saliency-aware spatial regularization in the filter updating process of CF tracking. SSAR-CF introduces the saliency map into the spatial weight map to highlight the object region. DSAR-CF further investigate a dynamic strategy to update the spatial weight map by considering the saliency map and response map.

to the size of weight map and correlation filter, and use $\mathbf{S}$ to regularize the fixed weight map $\mathbf{W}_{\mathrm{SR}}$ of SRDCF and obtain a new weight map

$$\mathbf{W}_{\mathrm{S}} = \mathbf{S} \odot \mathbf{W}_{\mathrm{SR}}. \tag{6}$$

For online object tracking, we add a step before Step i) of CF tracking algorithm in Section III-A by using Eq. (6) to calculate the weight map $\mathbf{W}_{\mathrm{S}}$ in the first frame, in which the target is annotated with a given bounding box.

By setting $\mathbf{W} = \mathbf{W}_{\mathrm{S}} \in \mathbb{R}^{M \times N}$, the energy function of Eq. (3) incorporates the target saliency and its minimization using Gauss-Seidel algorithms leads a new way to update the correlation filter in the Step iv) of the CF tracking algorithm. We call the corresponding CF tracking algorithm Static Saliency-Aware Regularized CF tracking (SSAR-CF) since $\mathbf{W}_{\mathrm{S}}$ is fixed during the tracking process after computed in the first frame. We will show that SSAR-CF improves the tracking accuracy of the classical SRDCF while maintaining the original speed in the later experiments.

As an alternative to Eq. (6), we can set weight map $\mathbf{W}$ as $\mathbf{S}$ directly. However, such setup makes our tracker miss target easily: since $\mathbf{S}$ shown in Figure 3 has small value within the target and large value in the background, this setup introduces strong punishment to the filter in the context of target. As discussed in [2] and [46], the context information helps localize the target accurately. Hence, $\mathbf{S}$ limits the contribution of context to the discriminative power of the learned filter and leads to a poor tracker. In contrast to $\mathbf{S}$, $\mathbf{W}_{\mathrm{SR}}$ is a continuous function of coordinates, and context around the target still helps learn discriminative filter by using $\mathbf{W}_{\mathrm{SR}}$ during tracking. However, $\mathbf{W}_{\mathrm{SR}}$ ignores the content of target, e.g. shape or salient parts. $\mathbf{W}_{\mathrm{S}}$ actually fuses $\mathbf{W}_{\mathrm{SR}}$ and $\mathbf{S}$ by considering both spatial prior and content of the target. We will compare the different setups of $\mathbf{W}$ in Section IV-D to show that $\mathbf{W}_{\mathrm{S}}$ does help get higher accuracy.

D. Dynamic Saliency-Aware Regularized CF Tracking (DSAR-CF)

In this section, we propose a new dynamically updated weight map to tackle the problem of temporal target change. We further extend the weight map $\mathbf{W}$ that dynamically varies over time to better reflect the target shape variation by considering both the saliency map $\mathbf{S}$ and the response map $\mathbf{C}$. The saliency map $\mathbf{S}$, varied from frame to frame, captures object shape and size variation in tracking. The response map $\mathbf{C}$ can help tackle the case of low object saliency, e.g., the tracking target is occluded or surrounded by similar objects [50].

For this purpose, we set $\mathbf{W} = \mathbf{W}_{\mathrm{D}} \in \mathbb{R}^{M\times N}$ to be the optimal solution of

$$\begin{aligned}\mathrm{E}(\mathbf{W}_{\mathrm{D}}, \mu_{\mathrm{obj}}, \mu_{\mathrm{non}}) &= \sum_{(i,j)\in\boldsymbol{\Omega}_{\mathrm{obj}}} (\mathbf{S}(i,j) - \mu_{\mathrm{obj}})^2 \\ &+ \sum_{(i,j)\in\boldsymbol{\Omega}_{\mathrm{non}}} (\mathbf{S}(i,j) - \mu_{\mathrm{non}})^2 \\ &+ \eta \left\|\mathbf{W}_{\mathrm{D}} - \mathbf{W}_{\mathrm{SR}}\right\|^2 + (1-\eta)\left\|\mathbf{W}_{\mathrm{D}} - \mathbf{W}'_{\mathrm{SR}}\right\|^2, \end{aligned} \quad (7)$$

where the target region $\boldsymbol{\Omega}_{\mathrm{obj}}$ and non-target region $\boldsymbol{\Omega}_{\mathrm{non}}$ are defined by

$$\begin{cases} \boldsymbol{\Omega}_{\mathrm{obj}} = \{(i,j)|\mathbf{W}_{\mathrm{D}}(i,j) \le \zeta\}, \\ \boldsymbol{\Omega}_{\mathrm{non}} = \{(i,j)|\mathbf{W}_{\mathrm{D}}(i,j) > \zeta\}, \end{cases} \quad (8)$$

with $\zeta > 0$ being a threshold to partition the considered $M \times N$ domain $\boldsymbol{\Omega}$ into target and non-target regions. In Eq. (7), μ_{obj} and μ_{non} are the mean values of $\mathbf{S}$ in $\boldsymbol{\Omega}_{\mathrm{obj}}$ and $\boldsymbol{\Omega}_{\mathrm{non}}$ respectively. Minimizing the first two terms of Eq. (7) can tune $\mathbf{W}_{\mathrm{D}}$ according to the saliency map $\mathbf{S}$ and embed the shape information of target into $\mathbf{W}_{\mathrm{D}}$. Specifically, we regard $\mathbf{W}_{\mathrm{D}}$ as a level-set function whose ζ-level set corresponds to a contour that is a cross section between $\mathbf{W}_{\mathrm{D}}$ and a horizontal plane defined by ζ [33]. The contour, i.e. ζ-level set, also splits the saliency map into two regions, i.e. $\boldsymbol{\Omega}_{\mathrm{obj}}$ and $\boldsymbol{\Omega}_{\mathrm{non}}$, whose mean values are μ_{obj} and μ_{non}, respectively. Clearly, if $\boldsymbol{\Omega}_{\mathrm{obj}}$ contains background region of the saliency map and $\boldsymbol{\Omega}_{\mathrm{non}}$ contains the target region, the first two terms of Eq. (7) would be very large. Hence, to minimize the first two terms of Eq. (7), $\mathbf{W}_{\mathrm{D}}$ must be tuned to drive its ζ-level set, i.e. the contour, to the target boundary. As a result, the shape information of the target is naturally embedded into $\mathbf{W}_{\mathrm{D}}$. Figure 5 shows an example of the value change of $\mathbf{W}_{\mathrm{D}}$ in the iterative optimization. With decreasing value of Eq. (7), the ζ-level set gradually gets closer to the target boundary which is also embedded into $\mathbf{W}_{\mathrm{D}}$.

And $\mathbf{W}'_{\mathrm{SR}}$ in Eq. (7) is constructed as

$$\mathbf{W}'_{\mathrm{SR}} = M_{\mathbf{W}} + m_{\mathbf{W}} - \mathbf{W}_{\mathrm{SR}}, \quad (9)$$

where $M_{\mathbf{W}}$ and $m_{\mathbf{W}}$ are the maximum and minimum values of $\mathbf{W}_{\mathrm{SR}}$ in Eq. (4). With Eq. (9), $\mathbf{W}'_{\mathrm{SR}}$ has high penalties on target region and low values on background as shown in Figure 6, which avoids corrupting the filter when tracking result is unreliable due to the interferences [50], e.g. occlusion and background clutter. Besides, we use Eq. (9) to calculate $\mathbf{W}'_{\mathrm{SR}}$ to guarantee that $\mathbf{W}'_{\mathrm{SR}}$ has the same value range as $\mathbf{W}_{\mathrm{SR}}$, which makes the value range of learned filter not to change

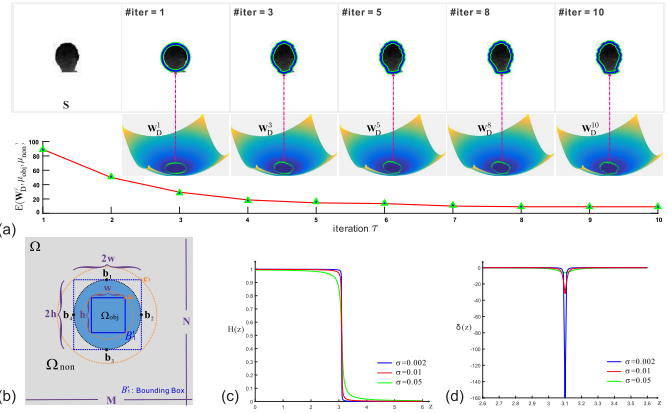


Fig. 5. (a) The value change of $\mathbf{W}_{\mathrm{D}}$ during the optimization. We iteratively optimize $\mathbf{W}_{\mathrm{D}}$ and illustrate the intermediate results after iterations of 1, 3, 5, 8, 10 times, respectively. Vertical axis denotes the function value of the first two terms of Eq. (7). (b) shows the coordinate space to calculate $\mathbf{W}_{\mathrm{SR}}$. The solid bounding box with blue color denotes the location and size of the target. The dashed box is two times larger than the solid bounding box. We can set ζ as the values of $\mathbf{W}_{\mathrm{SR}}$ at the points $\mathbf{b}_1$, $\mathbf{a}_1$ and $\mathbf{c}_1$, respectively. Note that, points $\mathbf{b}_{\{1,2,3,4\}}$ have the same value according the definition of $\mathbf{W}_{\mathrm{SR}}$. (c) and (d) shows the Heaviside step function H(z) and its derived function $\delta(z)$ when setting three different values for σ.

with different weight maps. The impact parameter η in Eq. (7) is produced by the response map $\mathbf{C}$ in Eq. (1). The basic idea is to make $\mathbf{W}_{\mathrm{D}}$ close to $\mathbf{W}_{\mathrm{SR}}$ when the tracking result is reliable, and close to $\mathbf{W}'_{\mathrm{SR}}$ when tracking result is unreliable, e.g., with target occlusions in Figure 6. In this paper, we use the PSR (Peak to Sidelobe Ratio) [3] score of the response map $\mathbf{C}$ to measure the tracking reliability and set the value of η. Given $\mathbf{C}$, we calculate the peak value ρ and the sidelobe that is the rest of the pixels excluding an 11×11 window around the peak location. The PSR score P is defined as $P = \frac{\rho - \mu_{\mathrm{s}}}{\sigma_{\mathrm{s}}}$, where μ_{s} and σ_{s} are the mean and standard deviation of the sidelobe in the response map $\mathbf{C}$. Based on this, we set η as

$$\eta = \begin{cases} 1 & \text{if } \rho > \tau_1\bar{\rho} \ \ \& \ P > \tau_2\bar{P}, \\ 0 & \text{otherwise}, \end{cases} \quad (10)$$

where $\bar{\rho}$ and $\bar{P}$ are the average values of ρ and P in all the tracked frames, $\tau_1, \tau_2 > 0$ are two pre-set constant coefficients [41]. This way, the first two terms in Eq. (7) reflects the saliency map and the last two terms in Eq. (7) reflects the response map.

In the first frame, we initialize $\mathbf{W}_{\mathrm{D}} = \mathbf{W}_{\mathrm{S}}$. During online tracking, we first estimate the bounding box of a target at frame t via step iii) in Section III-A, and obtain a weight map $\mathbf{W}_{\mathrm{D}}$ for spatial regularization. More specifically, after estimating the target bounding box in frame t, we then perform saliency detection as described in Section III-C. The saliency map $\mathbf{S}_t$ and response map $\mathbf{C}_t$, are fed to Eq. (7) to get $\mathbf{W}_{\mathrm{D}}$ using the level-set optimization in Section III-E.

We set $\mathbf{W} = \mathbf{W}_{\mathrm{D}}$ in Eq. (3), which is then optimized by the Gauss-Seidel algorithm for updating correlation filter in the Step iv) of the CF tracking algorithm. We call this tracking Dynamic Saliency-Aware Regularized CF tracking (DSAR-CF) since the weight map dynamically varies frame by frame. As in SRDCF, a scale estimation method [29] can

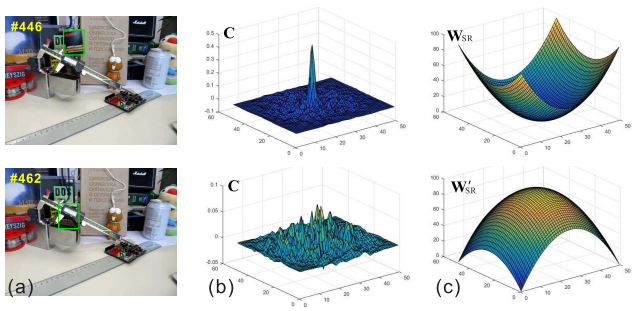


Fig. 6. The tracking target in the green bounding box with and without occlusion are initialized in (a), and the corresponding response maps $\mathbf{C}$ that reflect the reliability of tracking results are shown in (b), on this basic, we make $\mathbf{W}_{\mathrm{D}}$ close to $\mathbf{W}_{\mathrm{SR}}$ when the tracking result is reliable, and close to $\mathbf{W}'_{\mathrm{SR}}$ when tracking result is unreliable.

be used to estimate the size of the bounding box in Step iii). In the following, we give the level-set algorithm for optimizing Eq. (7).

E. Level-Set Optimization for $\mathbf{W}_{\mathrm{D}}$

We use a level-set algorithm [4], [42] to minimize the energy function defined in Eq. (7). Specifically, we first transform Eq. (7) to

$$\begin{aligned}\mathrm{E}(\mathbf{W}_{\mathrm{D}}, \mu_{\mathrm{obj}}, \mu_{\mathrm{non}}) &= \sum_{(i,j)\in\Omega} (\mathbf{S}(i,j) - \mu_{\mathrm{obj}})^2 \mathrm{H}(\mathbf{W}_{\mathrm{D}}(i,j)) \\ &+ \sum_{(i,j)\in\Omega} (\mathbf{S}(i,j) - \mu_{\mathrm{non}})^2 [1 - \mathrm{H}(\mathbf{W}_{\mathrm{D}}(i,j))] \\ &+ \eta \left\|\mathbf{W}_{\mathrm{D}} - \mathbf{W}_{\mathrm{SR}}\right\|^2 + (1-\eta) \left\|\mathbf{W}_{\mathrm{D}} - \mathbf{W}'_{\mathrm{SR}}\right\|^2, \end{aligned} \quad (11)$$

where $\mathrm{H}(\cdot)$ is the Heaviside step function

$$\mathrm{H}(z) = \frac{1}{2} - \frac{1}{\pi}\arctan\left(\frac{z-\zeta}{\sigma}\right), \quad (12)$$

as shown in Figure 5 (c).

By minimizing the energy function Eq. (11) [21], we have

$$\mu_{\mathrm{obj}} = \frac{\sum_{(i,j)\in\Omega} \mathrm{H}(\mathbf{W}_{\mathrm{D}}(i,j))\mathbf{S}(i,j))}{\sum_{(i,j)\in\Omega} \mathrm{H}(\mathbf{W}_{\mathrm{D}}(i,j))}, \quad (13)$$

$$\mu_{\mathrm{non}} = \frac{\sum_{(i,j)\in\Omega} [1 - \mathrm{H}(\mathbf{W}_{\mathrm{D}}(i,j))]\mathbf{S}(i,j)}{\sum_{(i,j)\in\Omega} [1 - \mathrm{H}(\mathbf{W}_{\mathrm{D}}(i,j))]}. \quad (14)$$

By computing the gradient

$$\begin{aligned}\frac{\partial \mathbf{W}_{\mathrm{D}}}{\partial \tau} &= \delta(\mathbf{W}_{\mathrm{D}})[-(\mathbf{S} - \mu_{\mathrm{obj}})^2 + (\mathbf{S} - \mu_{\mathrm{non}})^2] \\ &- 2[\eta(\mathbf{W}_{\mathrm{D}} - \mathbf{W}_{\mathrm{SR}}) + (1-\eta)(\mathbf{W}_{\mathrm{D}} - \mathbf{W}'_{\mathrm{SR}})], \end{aligned} \quad (15)$$

where $\delta(\cdot)$ is the derivative of $\mathrm{H}(\cdot)$, τ is the iteration index, we can iteratively optimize the energy function by the gradient decent [4] until the maximal number of iterations are reached or the solution does not change much between two iterations. The detailed derivation process of Eq. (11) and Eq. (15) can be found in Appendix.

In practice, we update the correlation filter and the spatial weight map in every 2 frames which benefits from the robust learned filter in DSAR-CF. We have analyzed the algorithm speed in subsection IV-F.

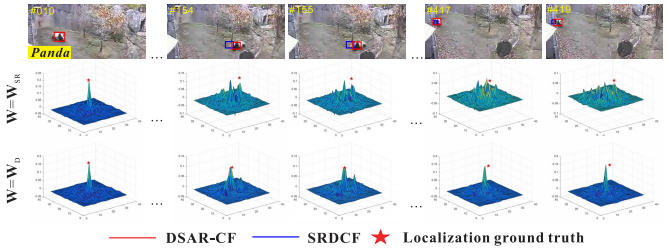


Fig. 7. SRDCF with $\mathbf{W}_{\mathrm{SR}}$ and $\mathbf{W}_{\mathrm{D}}$ are used to track the panda respectively. Tracking results with corresponding response maps are shown for comparison.

F. Saliency-Aware Regularization for Boundary Effects

In this section, we discuss the advantage of multiplying filter $\mathbf{F}$ with the our saliency-aware weight map, i.e. $\mathbf{W} = \mathbf{W}_{\mathrm{D}}$, compared with the original weight map in SRDCF, i.e. $\mathbf{W} = \mathbf{W}_{\mathrm{SR}}$ in Eq. (3). We show that spatial regularization used by SRDCF actually alleviates boundary effects by assigning different training samples weights that only rely on their spatial prior. In our saliency-aware regularization, such weights consider both the content of target and the spatial prior and is able to localize target more accurately. As discussed in [23], correlation filter can be viewed as solving a regression problem where circular shifted versions of a real sample work as the training samples. For example, $\mathbf{X} \in \mathbb{R}^{M\times N}$ in Eq. (3) is a real sample. We can circularly shift it and get MN synthetic samples. We reformulate the objective function of SRDCF, i.e. Eq. (3), by setting $\mathbf{F}' = \mathbf{W} \odot \mathbf{F}$ and get

$$\mathrm{E}'_{\mathrm{SR}}(\mathbf{F}') = \left\|\mathbf{X} * (\frac{1}{\mathbf{W}} \odot \mathbf{F}') - \mathbf{Y}\right\|^2 + \left\|\mathbf{F}'\right\|^2, \quad (16)$$

where the index k in Eq. (3) is ignored for convenient representation. $\mathbf{X} * (\frac{1}{\mathbf{W}} \odot \mathbf{F})$ equals to $(\mathbf{A}\mathrm{diag}(\frac{1}{\mathbf{w}}))\mathbf{f}'$, where $\mathbf{w}$ and $\mathbf{f}'$ are the vectorized $\mathbf{W}$ and $\mathbf{F}'$, respectively. Each row of $\mathbf{A} \in \mathbb{R}^{MN\times MN}$ is a vectorized training sample, i.e. a circularly shifted $\mathbf{X}$. Hence, $\mathbf{A}\mathrm{diag}(\frac{1}{\mathbf{w}})$ is to assign each training sample, i.e. a row of $\mathbf{A}$, a weight that is determined by $\mathbf{W}$.

In the original CF objective function, i.e. Eq. (2), all the elements of $\mathbf{W}$ have the same value, which means both real sample and synthetic samples are of equal importance for learning filter. However, those synthetic samples shifted to be far from target center cannot represent the real scene. Hence, it is difficult to learn discriminative filter via CF. SRDCF generates $\mathbf{W} = \mathbf{W}_{\mathrm{SR}}$ according to the shifting distance of training samples. That is, a synthetic sample with large shifting distance will be assigned a small weight. This effectively removes the influence of useless synthetic samples, thus learns much more discriminative filter than CF. However, SRDCF uses a fixed and rectangle-defined weight map, i.e., $\mathbf{W}_{\mathrm{SR}}$ in Eq. (4), which ignores the content of target, e.g., shape or saliency part, and is not suitable for targets with irregular shapes. In contrast to SRDCF, we online optimize Eq. (7) to make selection of target and non-target region to generate dynamic weight map. Hence, the generated weight map, i.e. $\mathbf{W}_{\mathrm{D}}$, can reduce influences of more useless synthetic samples than $\mathbf{W}_{\mathrm{SR}}$ and help learn more discriminative filter. We show a typical example in Figure 7 where SRDCF with $\mathbf{W}_{\mathrm{D}}$ and $\mathbf{W}_{\mathrm{SR}}$

are used to track a panda respectively. Response maps on six frames are used to evaluate the discriminative power of the learned filter. Clearly, the peak of response map generated by SRDCF with $\mathbf{W} = \mathbf{W}_{\mathrm{D}}$ is more prominent than the one of $\mathbf{W}_{\mathrm{SR}}$, which enables SRDCF with $\mathbf{W}_{\mathrm{D}}$ to localize target more accurately.

IV. Experimental Results

In this section, we validate the proposed method by conducting comprehensive experiments on standard benchmarks OTB-2013 [44], OTB-2015 [45] and VOT-2016 [27], and compare its performance with several existing state-of-the-art trackers. Furthermore, we conduct the analysis experiments to evaluate the proposed algorithm and the ablation study to demonstrate the usefulness of each component.

A. Setup

1) Implementation Details: We implement the proposed method in Matlab and run on a desktop computer with an Intel Core i7 3.4GHz CPU. We apply HOG [6] for extracting feature maps in Steps ii) and iv) in CF tracking. Compared with SRDCF, our method has four new parameters, i.e. dilation factor κ, control parameter ζ in Eq. (8), σ in Eq. (12) and η in Eq. (7). We fix these parameters on all benchmark datasets by setting $\kappa = 1.5$, $\zeta = 3.1$ and $\sigma = 0.01$. η is computed by Eq. (10). All other parameters are inherited from SRDCF. For example, we fix $a = 0.1$, $b = 3.0$ in Eq. (4), and $K = 4$ in Section III-A. We further discuss the influence of the four new parameters in Section IV-C.1.

2) Datasets and Metrics: The experiments are conducted on three standard benchmarks: OTB-2013 [44], OTB-2015 [45] and VOT-2016 [27]. The first two OTB datasets contain 51 and 100 sequences, respectively. For OTB datasets, we use the one-pass evaluation (OPE) with metrics of center location error (CLE) and intersection-over-union (IoU). The CLE and IoU measure distance of predicted locations from the ground truth and the overlap ratio between predicted and ground truth bounding boxes, respectively. For each metric, we can set a threshold to judge if a tracker is successful at each frame and calculate the percentage of successful frames within each sequence. We then calculate average success percentages w.r.t. different thresholds on all sequences and obtain success and precision plots. After that, the area under curve (AUC) of each plot can be calculated. The VOT-2016 has 60 sequences and re-initializes testing trackers when it misses the target. The expected average overlap (EAO) considering both bounding box overlap ratio (accuracy) and the re-initialization times i.e. failures times (robustness) serves as the major evaluation metrics. The VOT-2016 [27] provides the EAO with re-initialization as baseline experiments and without re-initialization as unsupervised experiments, and the overall integrates both the baseline and unsupervised experiments.

3) Comparison Methods: We compare the proposed method with 8 state-of-the-art trackers based on hand-crafted features including KCF [23], DSST [8], SAMF [29], DLSSVM [32], Staple [1], SRDCF [9], CSRDCF [30] and BACF [15], and with 8 deep feature based methods including DeepSRDCF [10], HCF [31], HDT [34], SiamFC [2], CFNet[1], C-COT [12], DSiam [18] and CREST [38]. Among them, DLSSVM is an SVM-based tracker. KCF, DSST, SAMF, SRDCF, Staple, CSRDCF and BACF are correlation filter (CF) based trackers. Moreover, DeepSRDCF, HCF, HDT, C-COT are deep feature based CF trackers. SiamFC, CFNet, and DSiam are Siamese network based trackers. CREST is a convolutional residual based tracker.

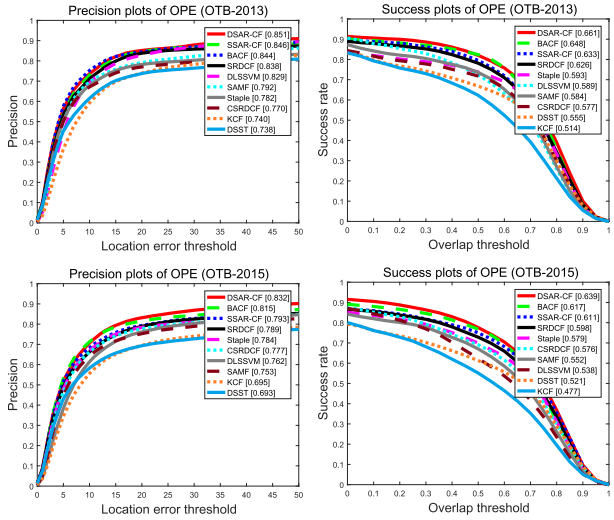


Fig. 8. Precision plots (left) and success plots (right) of both the proposed and comparison methods on OTB-2013 (first row) and OTB-2015 (second row). The legend contains the average distance precision score at 20 pixels and the AUC of success plot of each method.

B. Results

1) Evaluation on OTB Benchmark: The first row of Figure 8 shows comparison results of our methods and eight hand-crafted feature based trackers on OTB-2013. In terms of precision score, our DSAR-CF obtains the highest performance and a gain of 1.3% over SRDCF. In terms of the AUC of success plots, DSAR-CF outperforms all the comparison trackers including recent BACF and CSRDCF and achieves 3.5% improvement over SRDCF. Without online updating the weight map, SSAR-CF also gets better results than the other trackers, except for BACF, which demonstrates the effectiveness of using saliency map to learn more discriminative filter.

We can find similar results on OTB-2015 in the second row of Figure 8. Specifically, DSAR-CF gets gains of 3.9% and 4.1% over SRDCF according to the precision score and success plot AUC. These improvements are higher than the ones obtained on OTB-2013, since OTB-2015 extends OTB-2013 with more challenging sequences where SRDCF easily fails.

We also perform an attribute-based analysis of the proposed method on OTB-2015. The 100 videos of OTB-2015 are

[1]The version of CFNet we use is *Baseline+CF-conv3* [40].

TABLE I

ATTRIBUTES BASED SUCCESS RATE AUC SCORES FOR SSAR-CF, DSAR-CF AND OTHER 8 HAND-CRAFTED FEATURE BASED TRACKERS ON OTB-2015. THE BEST THREE RESULTS ARE MARKED IN RED, GREEN AND BLUE RESPECTIVELY

	OCC	BC	IV	FM	DEF	SV	OPR	IPR	OV	MB	LR
KCF	44.3	49.8	47.9	45.9	43.6	39.4	45.3	46.9	39.3	45.8	30.7
DSST	46.1	52.4	56.1	46.6	43.4	47.9	48.2	51.1	38.5	47.3	39.5
SAMF	53.4	53.4	53.4	52.6	49.5	49.3	52.7	52.7	50.0	52.3	43.1
DLSSVM	50.8	51.7	52.1	54.3	51.2	46.5	53.1	53.3	46.8	57.1	39.9
Staple	54.3	56.1	59.5	54.1	55.0	52.0	53.4	54.9	47.6	54.0	39.9
SRDCF	55.9	58.3	61.3	59.7	54.4	56.1	55.0	54.4	46.0	59.4	49.4
CSRDCF	53.1	56.4	54.2	57.2	53.2	52.0	52.0	51.2	51.8	56.8	43.1
BACF	56.7	62.5	63.0	59.4	57.2	57.6	57.7	57.5	55.2	57.0	53.0
SSAR-CF	56.5	61.3	62.4	59.9	56.0	57.4	56.6	54.8	50.2	61.8	50.0
DSAR-CF	61.1	63.7	64.9	62.0	59.3	62.7	61.5	58.6	55.2	63.3	54.9

TABLE II

ATTRIBUTES BASED SUCCESS RATE AUC SCORES FOR SSAR-CF, DSAR-CF AND OTHER 8 DEEP TRACKERS ON OTB-2015. THE BEST THREE RESULTS ARE MARKED IN RED, GREEN AND BLUE RESPECTIVELY

	OCC	BC	IV	FM	DEF	SV	OPR	IPR	OV	MB	LR	OTB-2013	OTB-2015
HDT	52.8	57.8	53.5	56.8	54.3	48.6	53.4	55.5	47.2	57.4	45.6	60.3	56.4
HCF	52.5	58.5	54.0	57.0	53.0	48.5	53.4	55.9	47.4	58.5	43.9	60.5	56.2
DeepSRDCF	60.1	62.7	62.1	62.8	56.6	60.5	60.7	58.9	55.3	64.2	47.5	64.1	63.5
CFNet	52.7	56.1	54.3	55.5	52.5	54.7	55.6	57.1	45.6	54.0	55.2	61.0	58.9
DSiam	57.5	60.8	59.8	59.4	53.7	58.0	60.9	59.9	55.1	58.2	60.6	65.6	61.1
SiamFC	54.3	52.3	56.8	56.8	50.6	55.2	55.8	55.7	5.06	55.0	59.2	60.7	58.2
CREST	59.2	61.8	64.4	62.7	56.9	57.2	61.5	61.7	56.6	65.5	52.8	67.3	62.3
C-COT	67.4	65.2	68.2	67.6	61.4	65.4	65.2	62.7	64.8	70.6	61.0	67.7	67.3
SSAR-CF	56.5	61.3	62.4	59.9	56.0	57.4	56.6	54.8	50.2	61.8	50.0	63.3	61.1
DSAR-CF	61.1	63.7	64.9	62.0	59.3	62.7	61.5	58.6	53.3	63.3	54.9	66.1	63.9

grouped into 11 subsets according to 11 attributes[2]. Table I shows the success plot AUC of 8 comparison methods and the proposed DSAR-CF and SSAR-CF on 11 subsets. DSAR-CF outperforms all the comparison trackers on all subsets, which demonstrates the advantage of DSAR-CF in addressing various interferences. Particularly, DSAR-CF gets the largest gain, i.e. 5.1%, on subset of scale variation (SV). Although using fixed weight map, SSAR-CF still gets higher accuracy than SRDCF on all subsets, which also illustrates the advantage of saliency map for spatially regularized CF.

In addition to these hand-crafted feature based trackers, we compare DSAR-CF and SSAR-CF with eight well known deep trackers. Table II shows the comparison results on OTB-2013, OTB-2015 and its 11 subsets. On OTB-2013, the performance of DSAR-CF is slightly worse than C-COT and CREST while getting a gain of 2.0% over DeepSRDCF that is an extension of SRDCF by using deep features, which further demonstrates the effectiveness of introducing saliency-aware regularization. On OTB-2015, DSAR-CF gets the second best results and still outperforms DeepSRDCF. However, the gain over DeepSRDCF is reduced to 0.4%, since deep features help get better results on challenging sequences in OTB-2015. Note that, although C-COT and CREST produce better tracking than DSAR-CF, they are implemented on GPU and run at 0.2 FPS and 1 FPS respectively which are much slower than DSAR-CF running at 6 FPS. Furthermore, C-COT is an improved SRDCF, which could help improve DSAR-CF in the future. In terms of results on 11 subsets, DSAR-CF gets the second best results on subsets of OCC, BV, IV, DEF and SV and outperforms DeepSRDCF on 7 subsets including BC, DEF, OCC, LR, IV, OPR, and SV.

2) Evaluation on VOT-2016 Benchmark: Tracking performance on VOT-2016 is shown in Table III, where we compare the proposed DSAR-CF and SSAR-CF with other five methods that participate in the VOT-2016 challenge, i.e. KCF [23], SAMF [29], DSST [8], SRDCF [9], and SiamFC[3] [2]. We can see that SSAR-CF outperforms the baseline SRDCF in accuracy but with lower robustness – the average number of failures increases from 1.50 to 1.52. However, the proposed DSAR-CF improves both the accuracy and robustness of SRDCF and has significant improvement compared with other comparison methods. For the expected average overlap (EAO), DSAR-CF also gets the best performance in supervised (baseline) and unsupervised experiments: it improves the EAO of SRDCF by 5.9% and 4.3% in terms of baseline and overall respectively.

3) Qualitative Analysis: As shown in Figure 9, *Board* and *Sylvester* are selected to show the robustness of trackers against object deformation. The target in sequence *Board* is

[2]The 11 attributes are occlusion (OCC), background clutter (BC), illumination variation (IV), fast motion (FM), deformation (DEF), scale variation (SV), out-of-plane rotation (OPR), in-plane rotation (IPR), out-of-view (OV), motion blur (MB), and low resolution (LR).

[3]The SiamFC version with AlexNet [2].

TABLE III

COMPARATIVE RESULTS ON VOT-2016 IN TERMS OF THE AVERAGE ACCURACY (ACCURACY), ACCURACY RANKING (ACC. RANK), AVERAGE FAILURES (ROBUSTNESS), AND EAO UNDER BASELINE, UNSUPERVISED AND OVERALL EXPERIMENTS, RESPECTIVELY

Trackers	Accuracy & Robustness			EAO		
	Accuracy	Acc.rank	Robustness	Baseline	Unsupervised	Overall
DSAR-CF	**0.5360**	**1.45**	**1.45**	**0.2581**	**0.5145**	**0.3863**
SSAR-CF	**0.5334**	**1.58**	**1.52**	**0.2525**	**0.5009**	**0.3767**
SRDCF	0.5263	1.67	1.50	0.2437	0.4968	0.3702
SiamFC	0.5008	2.28	1.65	0.2352	0.4818	0.3585
SAMF	0.5040	2.00	2.02	0.2007	0.4604	0.3305
DSST	0.4841	2.27	2.52	0.1814	0.4624	0.3219
KCF	0.4787	2.80	2.03	0.1924	0.4142	0.3033

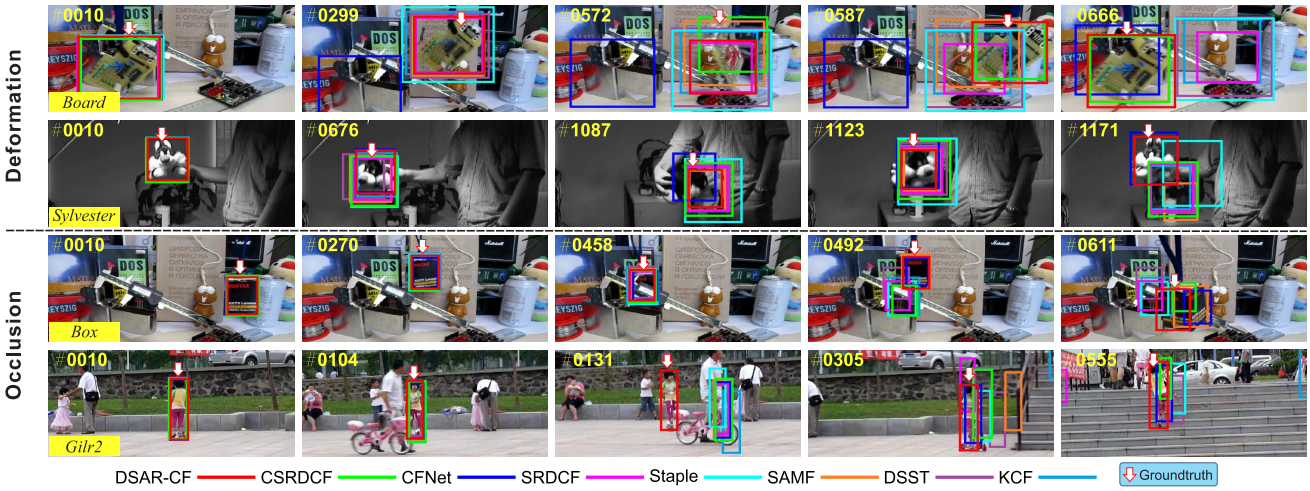


Fig. 9. Tracking results on *Box*, *Girl2* and *Board*, *Sylvester* videos (with deformation or occlusions) in OTB-2015.

TABLE IV

COMPARATIVE STUDY OF DIFFERENT κ, σ AND η ON OTB-2015 VIA THE PRECISION SCORE (%) AT 20 PIXELS AND SUCCESS RATES (%) AUC SCORE

κ	OTB-2015		ζ	OTB-2015		σ	OTB-2015	
	Prec. @20	Succ. AUC		Prec. @20	Succ. AUC		Prec. @20	Succ. AUC
$\kappa = 1$	81.6	62.8	$\zeta = \mathbf{W}_{\mathrm{SR}}(\mathbf{a}_1)$	80.7	62.5	$\sigma = 0.002$	82.4	63.4
$\kappa = 1.5$	**83.2**	**63.9**	$\zeta = \mathbf{W}_{\mathrm{SR}}(\mathbf{b}_1)$	**83.2**	**63.9**	$\sigma = 0.01$	**83.2**	**63.9**
$\kappa = 2$	80.2	62.4	$\zeta = \mathbf{W}_{\mathrm{SR}}(\mathbf{c}_1)$	79.4	61.2	$\sigma = 0.05$	81.5	62.7
SRDCF	78.9	59.8	SRDCF	78.9	59.8	SRDCF	78.9	59.8

a rigid body, but has significant appearance variations as it moves and turns (e.g. #572, #587), most trackers can not track it except for CSRDCF and DSAR-CF. Figure 9 (bottom) shows the tracking results on two representative sequences *Box* and *Girl2* where the target shows severe or long-term occlusion. In the *Box* sequence, the box can be tracked well until it is occluded by the vernier caliper in a long term (e.g. #458). Only three trackers, i.e. SAMF, CFNet and DSAR-CF, can track it continuously after leaving the obstruction (e.g. #492). In the *Girl2* sequence, the girl is occluded by a man severely (e.g. #104) and only the proposed method can track the target successfully (e.g. #131) when the man goes away. Experimental results confirm the strength of the proposed method in tracking the target occluded by other objects. This is due to that DSAR-CF dynamically updates the spatial weight map with the response map by considering the credibility of foreground object and avoiding the disturbance of target losing such as occlusion effectively. Analogously, in the sequence *Sylvester*, a doll moves quickly with rotation and despite heavy deformation in some frames (e.g. #676, #1078, #1123), the proposed method can track the doll well, while most other methods falsely estimate the scale or even lose the target (e.g. #1171). This is due to that the proposed DSAR-CF considers the saliency information in the process of updating the spatial weight map and incorporates more object shape information to improve the robustness in the case of object deformation.

C. Analysis of the Proposed Method

1) Parameter Selection: We investigate the performance changes w.r.t. different setups of the four parameters, i.e. dilation factor κ, control parameter ζ in Eq. (8), σ in Eq. (12) and η in Eq. (7). Specifically, for κ that determines the region size for saliency detection, we compare three variants, i.e. $\kappa = 1, 1.5, 2$, on OTB-2015 dataset by fixing all other parameters. As shown in Table IV, neither a smaller or larger κ can get better tracking accuracy than $\kappa = 1.5$. A larger κ results in a larger region containing more background for saliency detection and easily produces a poor saliency result

TABLE V

COMPARATIVE STUDY OF DIFFERENT τ_1, τ_2 ON OTB-2015 VIA PRECISION SCORE (%) AT 20 PIXELS AND SUCCESS RATES (%) AUC SCORE

τ_1	OTB-2015		τ_2	OTB-2015	
	Prec. @20	Succ. AUC		Prec. @20	Succ. AUC
$\tau_1 = 0.25$	80.9	62.3	$\tau_2 = 0.3$	81.9	62.9
$\tau_1 = 0.275$	80.5	62.1	$\tau_2 = 0.35$	82.4	63.3
$\tau_1 = 0.3$	**83.2**	**63.9**	$\tau_2 = 0.4$	**83.2**	**63.9**
$\tau_1 = 0.325$	82.2	63.1	$\tau_2 = 0.45$	**83.2**	**63.9**
$\tau_1 = 0.35$	82.1	63.1	$\tau_2 = 0.5$	**83.2**	**63.9**

that affects the online learning of filter. Meanwhile, since a target usually fills the whole bounding box, a smaller κ, e.g. 1, usually misses the main boundary of target, thus making limited contribution to the regularization term. Although showing different performance, all three variants improve SRDCF, which validates the effectiveness of introducing saliency into regularization term. ζ is used in Eq. (8) to separate the target from the background according to $\mathbf{W}_{\mathrm{D}}$. We get three variants by setting $\zeta = \mathbf{W}_{\mathrm{SR}}(\mathbf{b}_1)$, $\mathbf{W}_{\mathrm{SR}}(\mathbf{a}_1)$ and $\mathbf{W}_{\mathrm{SR}}(\mathbf{c}_1)$ where $\mathbf{b}_1$, $\mathbf{a}_1$ and $\mathbf{c}_1$ are three points on $\mathbf{W}_{\mathrm{SR}}$ as shown in Figure 5 (b). We evaluate these variants on OTB-2015 and report the results in Table IV. $\zeta = \mathbf{W}_{\mathrm{SR}}(\mathbf{b}_1)$ outperforms the other two variants, i.e. $\mathbf{W}_{\mathrm{SR}}(\mathbf{a}_1)$ and $\mathbf{W}_{\mathrm{SR}}(\mathbf{c}_1)$, with 3.1% and 4.8% relative improvement, respectively. We also show the influence of parameter σ in Eq. (12) that controls the shape of $\mathrm{H}(z)$ and $\delta(z)$ as shown in Figure 5 (c) and (d). We set $\sigma = 0.01$ as the baseline and enlarge/reduce σ five times respectively. As shown in Table IV, the tracking accuracy changes little with the huge displacement of σ. Hence, our method is not very sensitive to σ. η in Eq. (10) works as a weight for two terms in Eq. (7) and is determined by two hyperparameters, i.e. τ_1, τ_2. Table V compares the tracking results on OTB-2015 with different setups of τ_1, τ_2. Clearly, the tracking accuracy changes little with different τ_1 and τ_2. In practice, we set $\tau_1 = 0.3, \tau_2 = 0.4$ for the best performance.

2) Saliency Method Selection: We evaluate the effect of different saliency detection methods on tracking performance and show that our DSAR-CF is not very sensitive to the selection of saliency detection method. We first compare the saliency detection results of four popular methods i.e. SCA [35], wCtr [51], GS [43] and RA [36], on three saliency detection datasets and two cases from OTB. As shown in Figure 10 and Table VI, there are quite large differences between the performance of four saliency detectors. Meanwhile, the performance of a method varies on different datasets. Particularly, SCA gets the best performances on MSRA-5000 and ECSSD while performs worse on PASCAL-S. However, when we equip these methods to DSAR-CF and evaluate the tracking performance on OTB-2015, we find that the four methods lead to similar tracking performance, which are higher than that of the baseline tracker, i.e. SRDCF, as shown in Table VI. Hence, our tracking framework is partly but not highly dependent on the choice of the saliency detection method. In practice, our final version DSAR-CF uses SCA as the saliency detector for its highest performance on OTB-2015.

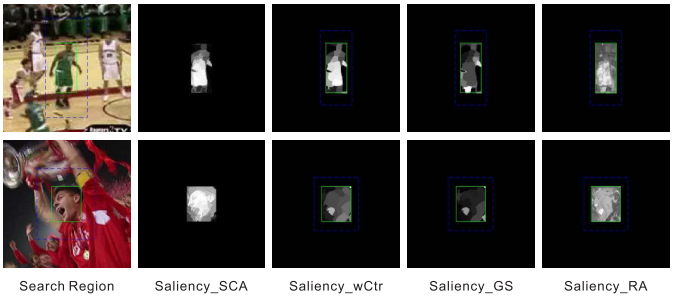


Fig. 10. Saliency results generated by four saliency detection methods (SCA, wCtr, GS, RA) on two sample sequences.

D. Ablation Study

To validate the effectiveness of our method, we compare four variants of our method with two baseline trackers, as shown in Table VII. Specifically, SRDCF-$\mathbf{W}_{\mathrm{SR}}$ and SRDCF-$\mathbf{S}$ are the two baseline trackers that uses $\mathbf{W}_{\mathrm{SR}}$ and $\mathbf{S}$ as the weight map, respectively. SSAR-CF use $\mathbf{W}_{\mathbf{S}}$ in Eq. (6) as weight map. DSAR-CF_mask replaces the saliency map with a target adaptive mask that is online generated by solving the energy function of [30]. DSAR-CF_w/oR uses Eq. (7) to update the weight map while ignoring the influence of response map by fixing $\eta = 1$. DSAR-CF is the final version of our method.

As shown in Table VII, SSAR-CF gets 2.2% relative improvement over SRDCF, i.e. SRDCF-$\mathbf{W}_{\mathrm{SR}}$, according to the success plot AUC on OTB-2015. More importantly, SSAR-CF outperforms SRDCF on all 11 subsets of OTB-2015. Hence, SSAR-CF does help SRDCF with lower sensitivity to various interferences by introducing the saliency maps of targets into the spatial weight map for effective regularization. However, those improvements are not that remarkable except for the one obtained on the subset of out-of-view (OV), since SSAR-CF uses fixed weight map, i.e. $\mathbf{W}_{\mathrm{S}}$, during tracking and does not consider the target variation. When we use Eq. (7) to update the weight map via online updated saliency map, DSAR-CF_w/oR gets much higher performance gain over SRDCF than SSAR-CF. By further introducing guidance of response maps via dynamically calculated η, our final version of DSAR-CF achieves 6.9% relative improvement over SRDCF.

We also compare SSAR-CF with SRDCF-$\mathbf{W}_{\mathrm{SR}}$ and SRDCF-$\mathbf{S}$ to validate the effectiveness of the saliency-aware regularization. As presented in Table VII, by setting $\mathbf{S}$ as weight map directly, SRDCF-$\mathbf{S}$ gets much lower accuracy than SRDCF-$\mathbf{W}_{\mathrm{SR}}$ and almost fails on most of the sequences. However, SSAR-CF using $\mathbf{W}_{\mathrm{S}}$ that combines $\mathbf{W}_{\mathrm{SR}}$ and $\mathbf{S}$ outperforms SRDCF-$\mathbf{W}_{\mathrm{SR}}$ with 2.2% relative improvement according to success plot AUC on OTB-2015, which demonstrates the advantage of adding saliency map into $\mathbf{W}_{\mathrm{SR}}$.

An alternative way of realizing online updated regularization is to generate $\mathbf{W}_{\mathrm{D}}$ by replacing the saliency map with a target adaptive mask that is online generated by solving the energy function in [30]. We denote such variant as DSAR-CF_mask. As shown in Table VII, by online updating the weight map with the target adaptive mask, DSAR-CF_mask gets higher performance than SSAR-CF

TABLE VI

TOP: EVALUATION RESULTS OF FOUR SALIENCY DETECTORS ON THREE STANDARD SALIENCY DETECTION DATASETS (MSRA-5000, ECSSD AND PASCAL-S), WHERE THE MEAN ABSOLUTE ERROR (MAE) IS TAKEN AS EVALUATION METRIC. BOTTOM: TRACKING PERFORMANCE OF DSAR-CF WHEN USING THE FOUR SALIENCY DETECTION METHODS RESPECTIVELY

	SCA	wCtr	GS	RA
MSRA-5000	**0.078**	0.111	0.146	0.323
ECSSD	**0.134**	0.224	0.254	0.367
PASCAL-S	0.180	**0.128**	0.161	0.326
Prec. @20	**83.2**	81.4	80.5	79.8
Succ.AUC	**63.9**	62.7	62.5	61.7

on OTB-2013. However, DSAR-CF_mask is worse than SSAR-CF on OTB-2015 and the subsets of IV, DEF, OV and MB, since the target adaptive masks from [30] usually miss the main boundary of target and leads to less discriminative filter. In contrast to DSAR-CF_mask, our DSAR-CF online updates the weight map with saliency maps and can outperform SSAR-CF on all the subsets, which demonstrates that updating regularization via saliency maps is more effective than that via target adaptive masks for online learning discriminative filter.

In terms of saliency detection, we find that DSAR-CF outperforms DSAR-CF_w/oAB, which does not abandon the saliency result outside the target region, on all subsets and achieves 5.0% relative improvement on the OTB-2015. Hence, abandoning the saliency results of background does help DSAR-CF learn more discriminative filter and track the target more accurately. This is reasonable since the saliency values (< 1) in background would reduce the penalties of $\mathbf{W}_{\mathrm{SR}}$ on background when we use Eq. (6) to reweight $\mathbf{W}_{\mathrm{SR}}$, thus making the learned filter more sensitive to background clutters.

Figure 11 compares DSAR-CF with SRDCF on three typical sequences whose targets are located in background clutter, deformation and occlusion, respectively. In 'Soccer', although the man is surrounded by other players with similar appearances, DSAR-CF still gets reliable saliency maps that capture the main boundary of the target during tracking and achieves higher accuracy than SRDCF. In 'Human5', the woman walks from far to near, which results in large scale variation and deformation. DSAR-CF also gets saliency maps containing the main parts of target and estimates the scale variation more accurately than SRDCF. Particularly, at frame #710, the woman is partially occluded by a wall. The saliency map successfully captures such changes and helps DSAR-CF keep tracking the target. SRDCF however fails. Such situation can be also found in 'Lemming'. All of the three cases show that DSAR-CF can generate meaningful saliency maps even the target is under background clutter, deformation and occlusion. Meanwhile, the introduction of saliency map does help improve the tracking accuracy by estimating scale variation more accurately and overcoming the partial occlusion.

E. Failure Cases

We have shown that saliency map does help improve the tracking accuracy of spatially regularized correlation filter in

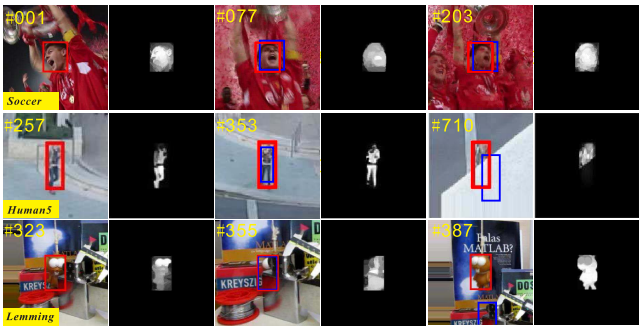


Fig. 11. Comparing SRDCF with DSAR-CF on three challenging sequences with background clutter, deformation and occlusion. The cropped search regions and corresponding saliency results are shown below each frame.

Fig. 12. Three failure cases of DSAR-CF shown in three rows respectively. The cropped search region and corresponding saliency results are also shown. White arrows are ground truth of target location.

Section IV-D. However, saliency map becomes less effective when target moves very fast. Specifically, as shown in Figure 12, when a target, e.g. the high jumper in the top row, moves fast, background around target may be totally different between neighbor frames. As a result, the saliency part in $t-1$ is no longer salient in t due to the significant change of surroundings, which makes the updated filter in $t-1$ less effective when detecting the target at t. We can find similar results on other two cases, i.e. 'Matrix' and 'Dragonbaby' in rows 2 and 3 of Figure 12. Particularly, in 'Matrix', the target is under multiple interferences, e.g. low resolution and background clutter, which leads to the target missing in the saliency map. In 'Dragonbaby', the baby's face suddenly disappears while his arm becomes salient, which leads to target missing in the saliency map. Note that, although saliency map has such disadvantage, our method, i.e. DSAR-CF, still outperforms SRDCF on the subset of fast motion in OTB-2015, as shown in Table I. There are two reasons. First, we use a saliency map to modify the original spatial weight map, i.e. $\mathbf{W}_{\mathrm{SR}}$, and get $\mathbf{W}_{\mathrm{S}}$ (Please refer to Section III-C for details). Hence, $\mathbf{W}_{\mathrm{S}}$ still keeps main information of $\mathbf{W}_{\mathrm{SR}}$ which helps avoid severe accuracy decrease when the saliency map is not correct. Second, the above sudden motion does not happen frequently in a sequence, which means that introducing saliency map could help improve tracking accuracy for most of the time. As shown in Table I, DSAR-CF outperforms baseline tracker SRDCF on all 11 subsets of OTB-2015.

TABLE VII

ABLATION STUDY BY COMPARING FIVE VARIANTS OF OUR METHOD WITH BASELINE TRACKER, SRDCF THAT IS RE-DENOTED AS SRDCF-$\mathbf{W}_{\mathrm{SR}}$ HERE. SRDCF-$\mathbf{W}_{\mathrm{SR}}$, SRDCF-$\mathbf{S}$, SSAR-CF AND DSAR-CF SET $\mathbf{W}_{\mathrm{SR}}$, $\mathbf{S}$, $\mathbf{W}_{\mathrm{S}}$ AND $\mathbf{W}_{\mathrm{D}}$ AS WEIGHT MAP RESPECTIVELY. DSAR-CF_mask UPDATES $\mathbf{W}_{\mathrm{D}}$ USING THE TARGET SIZE ADAPTIVE MASK. DSAR-CF_W/OR ONLINE GETS BY FIXING $\eta = 1$ IN EQ. (7), WHICH IGNORES THE INFLUENCE OF RESPONSE MAP. DSAR-CF_W/OAB GETS THE SALIENCY RESULT $\mathbf{S}'$ WITHOUT ABANDONING THE SALIENCY RESULTS OUTSIDE THE TARGET REGION

	OCC	BC	IV	FM	DEF	SV	OPR	IPR	OV	MB	LR	OTB-2013	OTB-2015
SRDCF-$\mathbf{W}_{\mathrm{SR}}$	55.9	58.3	61.3	59.7	54.4	56.1	55.0	54.4	46.0	59.4	49.4	62.6	59.8
SRDCF-$\mathbf{S}$	34.8	32.5	34.9	41.3	27.5	33.0	33.4	41.8	28.6	43.9	35.3	-21.2=41.4	-21.3=38.5
SSAR-CF	56.5	61.3	62.4	59.9	56.0	57.4	56.6	54.8	50.2	61.8	50.0	+1.3=63.9	+1.3=61.1
DSAR-CF_mask	57.3	61.5	59.6	60.5	55.7	58.1	58.0	57.2	47.3	60.0	51.1	+1.9=64.5	+1.2=61.0
DSAR-CF_w/oR	57.8	61.3	62.1	59.8	58.0	59.8	58.7	57.5	49.8	58.8	**54.9**	+2.4=65.0	+2.5=62.3
DSAR-CF_w/oAB	56.2	59.0	60.1	58.8	57.7	57.8	56.5	55.3	48.3	57.9	54.1	+0.2=62.8	+1.1= 60.9
DSAR-CF	**61.1**	**63.7**	**64.9**	**62.0**	**59.3**	**62.7**	**61.5**	**58.6**	**55.2**	**63.3**	**54.9**	+3.5=66.1	+4.1=63.9

TABLE VIII

TIME COST OF DSAR-CF AND ITS DIFFERENT COMPONENTS ON OTB-2015

	Total	Detection	Updating	Saliency	Avg.FPS
Time (Sec)	0.168	0.023	0.145	0.037	6.0
Proportion	1	0.137	0.863	0.220	

TABLE IX

THE TRACKING PERFORMANCE AND SPEED OF DSAR-CF USING DIFFERENT UPDATE FREQUENCY

	Prec.@20	Succ.AUC	Avg.FPS
KCF (PAMI2015)	69.5	47.7	153.5
CFLB (CVPR2015)	45.7	34.1	87.1
SRDCF (ICCV2015)	78.9	59.8	5.5
CSRDCF (CVPR2017)	77.7	57.6	7.5
DSAR-CF	**83.2**	**63.9**	6.0
DSAR-CF_upW/10	82.1	63.7	9.0
DSAR-CF_upF/5	79.0	61.4	**16.4**

F. Speed Analysis

In this section, we first discuss the time consumption of different components of DSAR-CF. Note that, in SSAR-CF, we only compute the saliency map at the first frame, which has no effect on tracking speed of SRDCF. For DSAR-CF, we evaluate the average time cost of it and its different components on OTB-2015 in Table VIII. Specifically, we compute the average time cost and time proportion of every component on one frame. 'Detection', 'Updating' and 'Saliency' denote the stages of detection, filter and weight map updating, and saliency detection during tracking, respectively.

Algorithm speed is also important in many tracking problems. Table IX compares several related and well-known CF trackers, where the FPS is measured on a desktop computer with an Intel Core i7 3.4GHz CPU. We can see that the proposed DSAR-CF can improve the baseline SRDCF in terms of both success rate and algorithm speed.

We can further speed up our tracker in practice. Specifically, since the saliency map does not change frequently, we can update the saliency-aware weight map after multiple frames, e.g., in every 10 frames. We denote such variant as DSAR-CF_upW/10. As shown in Table IX, although the accuracy decreases slightly, DSAR-CF_upW/10 improves the tracking speed from 6 FPS to 9 FPS while still showing much higher performance than SRDCF. To further improve the tracking speed, we can update the filter in every 5 frames i.e. DSAR-CF_upF/5, which runs near real time at an average of 16 FPS. Currently, our method is implemented on the Matlab platform without any optimization strategies and can be implemented for real time applications by further optimizing the code with GPU acceleration or parallel computing.

V. CONCLUSION

In this paper, we extended the CF tracking and the spatially regularized CF tracking to incorporate target saliency and make the regularization weight map dynamically vary frame by frame to better capture the shape variation of the target. We developed a level-set algorithm to iteratively compute the optimal regularization weight map in each frame. Experimental results on several standard benchmarks verified the effectiveness of the proposed method over many existing state-of-the-art tracking methods. Ablation study verified the usefulness of important components of the proposed method. For future work, we plan to improve the optimization method by replacing the traditional gradient descent with the network-based method, with the aim of further improving the speed while maintaining the tracking accuracy. We can further improve DSAR by using structure information from shape matching [14] and object segmentation [13], [28].

APPENDIX

In the following appendix, we show the detailed steps to transform Eq. (7) into Eq. (11) and optimize Eq. (7) w.r.t. $\mathbf{W}_{\mathrm{D}}$. Note that, the optimization method is inspired by the active contour model [4], [21]. The first two terms of Eq. (7) are defined on two different regions, i.e. Ω_{obj} and Ω_{non}. Thus,

Eq. (7) cannot be directly optimized. We modify Eq. (7) to

$$\begin{aligned}\mathrm{E}(\mathbf{W}_\mathrm{D}, \mu_\mathrm{obj}, \mu_\mathrm{non}) &= \sum_{(i,j)\in\Omega} (\mathbf{S}(i,j) - \mu_\mathrm{obj})^2 \mathrm{H_p}(\mathbf{W}_\mathrm{D}(i,j)) \\ &+ \sum_{(i,j)\in\Omega} (\mathbf{S}(i,j) - \mu_\mathrm{non})^2 \left[1 - \mathrm{H_p}(\mathbf{W}_\mathrm{D}(i,j))\right] \\ &+ \eta \left\|\mathbf{W}_\mathrm{D} - \mathbf{W}_\mathrm{SR}\right\|^2 + (1-\eta) \left\|\mathbf{W}_\mathrm{D} - \mathbf{W}'_\mathrm{SR}\right\|^2, \quad (17)\end{aligned}$$

where $\mathrm{H_p}(\cdot)$ is a piecewise Heavisible function

$$\mathrm{H_p}(z) = \begin{cases} 1 & z \le \zeta \\ 0 & z > \zeta, \end{cases} \quad (18)$$

with $z = \mathbf{W}_\mathrm{D}(i,j)$. $(\mathbf{S}(i,j) - \mu_\mathrm{obj})^2 \mathrm{H_p}(\mathbf{W}_\mathrm{D}(i,j))$ is only valid when $(i,j) \in \Omega_\mathrm{obj}$. Similarly, $(\mathbf{S}(i,j) - \mu_\mathrm{non})^2(1 - \mathrm{H_p}(\mathbf{W}_\mathrm{D}(i,j)))$ has meaningful value if and only if $(i,j) \in \Omega_\mathrm{non}$. However, Eq. (18) is not derivable. We cannot use the gradient descent to optimize Eq. (17) directly. We thus adopt an approximated Heaviside function

$$\mathrm{H}(z) = \frac{1}{2} - \frac{1}{\pi}\arctan\left(\frac{z-\zeta}{\sigma}\right), \quad (19)$$

where σ controls the similarity to $\mathrm{H_p}$. By replacing $\mathrm{H_p}$ with H, we get Eq. (11). The derivative of $\mathrm{H}(\cdot)$ is

$$\delta(z) = \frac{\partial H(z)}{\partial z} = -\frac{1}{\pi} \cdot \frac{\sigma}{\sigma^2 + (z-\zeta)^2}. \quad (20)$$

We show $\mathrm{H}(\cdot)$ with $\sigma = 0.002, 0.01, 0.05$ and their derivative functions in Figure 5 (c) and (d). With $\mathrm{H}(\cdot)$ and $\delta(\cdot)$, we optimize Eq. (11) via gradient descent and calculate derivative of E w.r.t. $\mathbf{W}_\mathrm{D}$ by

$$\begin{aligned}\frac{\partial \mathrm{E}}{\partial \mathbf{W}_\mathrm{D}} &= \delta(\mathbf{W}_\mathrm{D})[(\mathbf{S} - \mu_\mathrm{obj})^2 - (\mathbf{S} - \mu_\mathrm{non})^2] \\ &+ 2[\eta(\mathbf{W}_\mathrm{D} - \mathbf{W}_\mathrm{SR}) + (1-\eta)(\mathbf{W}_\mathrm{D} - \mathbf{W}'_\mathrm{SR})]. \quad (21)\end{aligned}$$

We denote τ as iteration index and update $\mathbf{W}_\mathrm{D}$ along the negative direction of the gradient, i.e.

$$\mathbf{W}_\mathrm{D}^{\tau+1} = \mathbf{W}_\mathrm{D}^{\tau} + \epsilon \frac{\partial \mathbf{W}_\mathrm{D}^{\tau}}{\partial \tau}, \quad (22)$$

where ϵ is learning rate and

$$\begin{aligned}\frac{\partial \mathbf{W}_\mathrm{D}^{\tau}}{\partial \tau} &= -\frac{\partial \mathrm{E}}{\partial \mathbf{W}_\mathrm{D}^{\tau}} = \delta(\mathbf{W}_\mathrm{D}^{\tau})[-(\mathbf{S} - \mu_\mathrm{obj})^2 + (\mathbf{S} - \mu_\mathrm{non})^2] \\ &- 2[\eta(\mathbf{W}_\mathrm{D}^{\tau} - \mathbf{W}_\mathrm{SR}) + (1-\eta)(\mathbf{W}_\mathrm{D}^{\tau} - \mathbf{W}'_\mathrm{SR})]. \quad (23)\end{aligned}$$

ACKNOWLEDGMENT

The authors thank all reviewers and the associate editor for their valuable comments.

REFERENCES

[1] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2015, pp. 1401–1409.

[2] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional Siamese networks for object tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 850–865.

[3] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2544–2550.

[4] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Trans. Image Process.*, vol. 10, no. 2, pp. 266–277, Feb. 2001.

[5] Z. Chen, Q. Guo, L. Wan, and W. Feng, "Background-suppressed correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Jul. 2018, pp. 1–6.

[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 886–893.

[7] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "Eco: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1–3.

[8] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. IEEE Brit. Mach. Vis. Conf.*, Sep. 2014.

[9] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4310–4318.

[10] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop*, Dec. 2016, pp. 621–629.

[11] M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1090–1097.

[12] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 472–488.

[13] W. Feng, J. Jia, and Z.-Q. Liu, "Self-validated labeling of Markov random fields for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 10, pp. 1871–1887, Oct. 2010.

[14] W. Feng, Z.-Q. Liu, L. Wan, C.-M. Pun, and J. Jiang, "A spectral-multiplicity-tolerant approach to robust graph matching," *Pattern Recognit.*, vol. 46, no. 10, pp. 2819–2829, 2013.

[15] H. K. Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1135–1143.

[16] H. K. Galoogahi, T. Sim, and S. Lucey, "Correlation filters with limited boundaries," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4630–4638.

[17] H. K. Galoogahi, T. Sim, and S. Lucey, "Multi-channel correlation filters," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2014, pp. 3072–3079.

[18] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, and S. Wang, "Learning dynamic siamese network for visual object tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 1781–1789.

[19] Q. Guo, W. Feng, C. Zhou, C.-M. Pun, and B. Wu, "Structure-regularized compressive tracking with online data-driven sampling," *IEEE Trans. Image Process.*, vol. 26, no. 12, pp. 5692–5705, Dec. 2017.

[20] Q. Guo, W. Feng, C. Zhou, and B. Wu, "Structure-regularized compressive tracking," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Jul. 2016, pp. 1–6.

[21] Q. Guo, S. Sun, X. Ren, F. Dong, B. Z. Gao, and W. Feng, "Frequency-tuned active contour model," *Neurocomputing*, vol. 275, pp. 2307–2316, Jan. 2018.

[22] R. Han, Q. Guo, and W. Feng, "Content-related spatial regularization for visual object tracking," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Jul. 2018, pp. 1–6.

[23] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

[24] L. Huang, B. Ma, J. Shen, L. Shao, and F. Porikli, "Visual tracking by sampling in part space," *IEEE Trans. Image Process.*, vol. 26, no. 12, pp. 5800–5810, Dec. 2017.

[25] R. Huang, W. Feng, and J. Sun, "Saliency and co-saliency detection by low-rank multiscale fusion," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Jun./Jul. 2015, pp. 1–6.

[26] R. Huang, W. Feng, and J. Sun, "Color feature reinforcement for cosaliency detection without single saliency residuals," *IEEE Signal Process. Lett.*, vol. 24, no. 5, pp. 569–573, May 2017.

[27] M. Kristan *et al.*, "The visual object tracking VOT2016 challenge results," in *Proc. Eur. Conf. Comput. Vis. Workshop*, 2016, pp. 777–823.

[28] L. Li, W. Feng, L. Wan, and J. Zhang, "Maximum cohesive grid of superpixels for fast object localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3174–3181.

[29] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 254–265.

[30] A. Lukežič, T. Vojiř, L. Čehovin, J. Matas, and M. Kristan, "Discriminative correlation filter with channel and spatial reliability," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jul. 2017, pp. 6309–6318.

[31] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3074–3082.

[32] J. Ning, J. Yang, S. Jiang, L. Zhang, and M.-H. Yang, "Object tracking via dual linear structured SVM and explicit feature map," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4266–4274.

[33] S. Osher, R. Fedkiw, and K. Piechor, *Level Set Methods and Dynamic Implicit Surfaces*. New York, NY, USA: Springer, 2003.

[34] Y. Qi *et al.*, "Hedged deep tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4303–4311.

[35] Y. Qin, H. Lu, Y. Xu, and H. Wang, "Saliency detection via cellular automata," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 110–119.

[36] E. Rahtu, J. Kannala, M. Salo, and J. Heikkilä, "Segmenting salient objects from images and videos," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 366–379.

[37] S. Siena and B. V. K. V. Kumar, "Detecting occlusion from color information to improve visual tracking," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2016, pp. 1110–1114.

[38] Y. Song, C. Ma, L. Gong, J. Zhang, R. Lau, and M.-H. Yang, "Crest: Convolutional residual learning for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2574–2583.

[39] Z. Tan, L. Wan, W. Feng, and C.-M. Pun, "Image co-saliency detection by propagating superpixel affinities," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 2114–2118.

[40] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr, "End-to-end representation learning for correlation filter based tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 5000–5008.

[41] M. Wang, Y. Liu, and Z. Huang, "Large margin object tracking with circulant feature maps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4800–4808.

[42] Y. Wang, S. Xiang, C. Pan, L. Wang, and G. Meng, "Level set evolution with locally linear classification for image segmentation," *Pattern Recognit.*, vol. 46, no. 6, pp. 1734–1746, Jun. 2013.

[43] Y. Wei, F. Wen, W. Zhu, and J. Sun, "Geodesic saliency using background priors," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 29–42.

[44] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2411–2418.

[45] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.

[46] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang, "Fast visual tracking via dense spatio-temporal context learning," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 127–141.

[47] P. Zhang, Q. Guo, and W. Feng, "Fast spatially-regularized correlation filters for visual object tracking," in *Proc. Pacific Rim. Int. Conf. Art. Intell.*, 2018, pp. 57–70.

[48] P. Zhang, Q. Guo, and W. Feng, "Fast and object-adaptive spatial regularization for correlation filters based tracking," *Neurocomputing*, 2019. doi: 10.1016/j.neucom.2019.01.060.

[49] T. Zhang, S. Liu, C. Xu, B. Liu, and M.-H. Yang, "Correlation particle filter for visual tracking," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2676–2687, Jun. 2018.

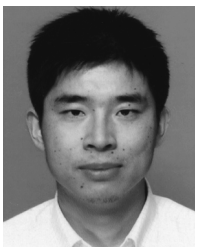
[50] C. Zhou, Q. Guo, L. Wan, and W. Feng, "Selective object and context tracking," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2017, pp. 1947–1951.

[51] W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency optimization from robust background detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2814–2821.

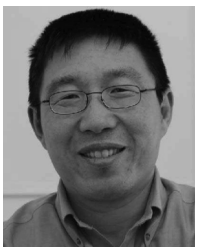
Wei Feng received the B.S. and M.Phil. degrees in computer science from Northwestern Polytechnical University, China, in 2000 and 2003, respectively, and the Ph.D. degree in computer science from the City University of Hong Kong in 2008. From 2008 to 2010, he was a Research Fellow with The Chinese University of Hong Kong and the City University of Hong Kong. He is currently a Full Professor with the School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University, China, and also with the Key Research Center for Surface Monitoring and Analysis of Cultural Relics, State Administration of Cultural Heritage, China. His major research interests are active robotic vision and visual intelligence, specifically including active camera relocalization and lighting recurrence, general Markov random fields modeling, energy minimization, active 3D scene perception, SLAM, and generic pattern recognition. He focuses on solving preventive conservation problems of cultural heritages via computer vision and machine learning. He is a member of the IEEE.

Ruize Han received the B.S. degree in mathematics and applied mathematics from the Hebei University of Technology, China, in 2016, and the M.Eng. degree in computer technology from Tianjin University, China, in 2019. His major research interest is visual intelligence, specifically including video processing and visual object tracking. He is also interested in solving preventive conservation problems of cultural heritages via artificial intelligence.

Qing Guo received the M.E. degree in computer application technology from the College of Computer and Information Technology, China Three Gorges University, in 2014. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University, China, and also with the Key Research Center for Surface Monitoring and Analysis of Cultural Relics, State Administration of Cultural Heritage, China. His research interests include visual object tracking, image and video object segmentation, image denoising, and other related vision problems.



Jianke Zhu received the B.S., M.S., and Ph.D. degrees in computer science and engineering from The Chinese University of Hong Kong, Hong Kong, in 2001, 2005, and 2008, respectively. He held a post-doctoral position at the BIWI Computer Vision Laboratory, ETH Zürich, Zürich, Switzerland. He is currently a Professor with the College of Computer Science, Zhejiang University, Hangzhou, China, and also with the Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies, Zhejiang University, Hangzhou, China. His research interests include computer vision and multimedia information retrieval.



Song Wang (M'02–SM'13) received the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana–Champaign (UIUC) in 2002. From 1998 to 2002, he was a Research Assistant with the Image Formation and Processing Group, Beckman Institute, UIUC. In 2002, he joined the Department of Computer Science and Engineering, University of South Carolina, where he is currently a Professor. He is also with the School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University, Tianjin, China, and also with the Key Research Center for Surface Monitoring and Analysis of Cultural Relics, State Administration of Cultural Heritage, Tianjin. His research interests include computer vision, medical image processing, and machine learning. He is a Senior Member of the IEEE Computer Society. He serves as the Publicity/Web Portal Chair for the Technical Committee of Pattern Analysis and Machine Intelligence, IEEE Computer Society. He also serves as an Associate Editor for *Pattern Recognition Letters*.