Remove Cosine Window from Correlation Filter-based Visual Trackers: When and How

Feng Li, Xiaohe Wu, Wangmeng Zuo, David Zhang, and Lei Zhang

Abstract—Correlation filters (CFs) have been continuously advancing the state-of-the-art tracking performance and have been extensively studied in the recent few years. Nonetheless, the existing CF trackers adopt a cosine window to spatially reweight base image to alleviate boundary discontinuity. However, cosine window emphasizes more on the central region of base image and has the risk of contaminating negative training samples during model learning. On the other hand, spatial regularization deployed in many recent CF trackers plays a similar role as cosine window by enforcing spatial penalty on CF coefficients. Therefore, we in this paper investigate the feasibility to remove cosine window from CF trackers with spatial regularization. When simply removing cosine window, CF with spatial regularization still suffers from small degree of boundary discontinuity. To tackle this issue, binary and Gaussian shaped mask functions are further introduced for eliminating boundary discontinuity while reweighting the estimation error of each training sample, and can be incorporated with multiple CF trackers with spatial regularization. In comparison to the baseline methods with cosine window, our methods are effective in handling boundary discontinuity and sample contamination, thereby benefiting tracking performance. Extensive experiments on four benchmarks show that our methods perform favorably against the state-of-the-art trackers using either handcrafted or deep CNN features.

Index Terms—Visual tracking, correlation filters, cosine window, spatial regularization

I. INTRODUCTION

CORRELATION filter (CF) is a representative framework for visual tracking and has attracted great research interest. Since the pioneering work of MOSSE [1], extensive studies have been given to improve the CF models by incorporating nonlinear kernel [2], [3], scale adaptivity [4], [5], [6], max-margin classification [7], spatial regularization [8], [9], [10], and continuous convolution [11], [12]. Moreover, the use of deep representation [13], [14], [15] and its combination with handcrafted features also significantly boosts the tracking performance. Benefited from the progress in models and feature representation, CFs have continuously advanced the state-of-the-art tracking accuracy and robustness in the recent few years.

F. Li, X. Wu, and W. Zuo are with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, 150001, China. e-mail: (fengli_hit@hotmail.com, xhwu.cpsl.hit@gmail.com, wmzuo@hit.edu.cn).
D. Zhang is with the School of Science and Engineering, The Chinese University of Hong Kong (Shenzhen), Shenzhen, China, e-mail: (cs-dzhang@comp.polyu.edu.hk).
L. Zhang is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, e-mail: (davidzhang@cuhk.edu.cn).
Manuscript received May xx, 2019.

In standard CF, the training set is formed as all the cyclic shifts of a base image and can be represented as a circulant matrix, making that CFs can be efficiently learned via fast Fourier transform (FFT). Albeit such circulant property greatly benefits learning efficiency, it makes the negative samples (i.e., shifted images) suffer from the boundary discontinuity problem. As shown in Fig. 1(a), except for the base image in green box, all the shifted images (e.g., the two patches in cyan and blue boxes) are generated using the circulant property and are not truly negative patches in real images.

In order to alleviate boundary discontinuity, cosine window has been introduced in early CF trackers, e.g., MOSSE [1] and KCF [2], and generally inherited by the subsequent improved models [4], [8], [16]. In particular, cosine window bands on base image as a pre-processing step by multiplying with a cosine shaped function (i.e., larger values for central regions and zeros for boundary pixels). Using KCF [2] as an example, it can be seen from Fig. 1(b) that after deploying cosine window boundary discontinuity can be largely suppressed (e.g., the patch in cyan box). Nonetheless, the shifted images near boundary are still plagued, as shown in the patch in blue box. In addition, when cosine window is deployed to base image, it also suppresses the pixels outside the target bounding boxes, thus has the risk of reducing the quality of training samples away from the target center. Since the model learning is directly dependent on the training set, the contamination on sample quality has a negative impact on learning more robust models. For example, the regions outside the human head in the base image of Fig. 1(b) are forced to approximate zero with the introduction of cosine window. This reduces the quality of the patches in cyan and blue boxes since the surrounding regions of the human head may also be helpful for the model learning. Unfortunately, this issue remains less investigated in the tracking community.

Recently, spatial regularization has also been suggested in numerous CF trackers [8], [12], [9], [16], [17], [18] to alleviate boundary discontinuity, which can be roughly grouped into two categories. On the one hand, SRDCF [8] and its later works [12], [17], [18] penalize the filter coefficients near boundaries to approximate zero. On the other hand, CFLB [9] and its multi-channel extension BACF [16] directly restrict the filter coefficients to be zero outside target bounding boxes. In general, existing CF trackers with spatial regularization still adopt cosine window, and are more effective in handling boundary discontinuity, as illustrated in Fig. 1(c)(e). Even though, the contamination of negative samples remains inevitable, since the cyan patch in Fig. 1(c) is forced to be zero in comparison to its counterpart in Fig. 1(d), and cannot
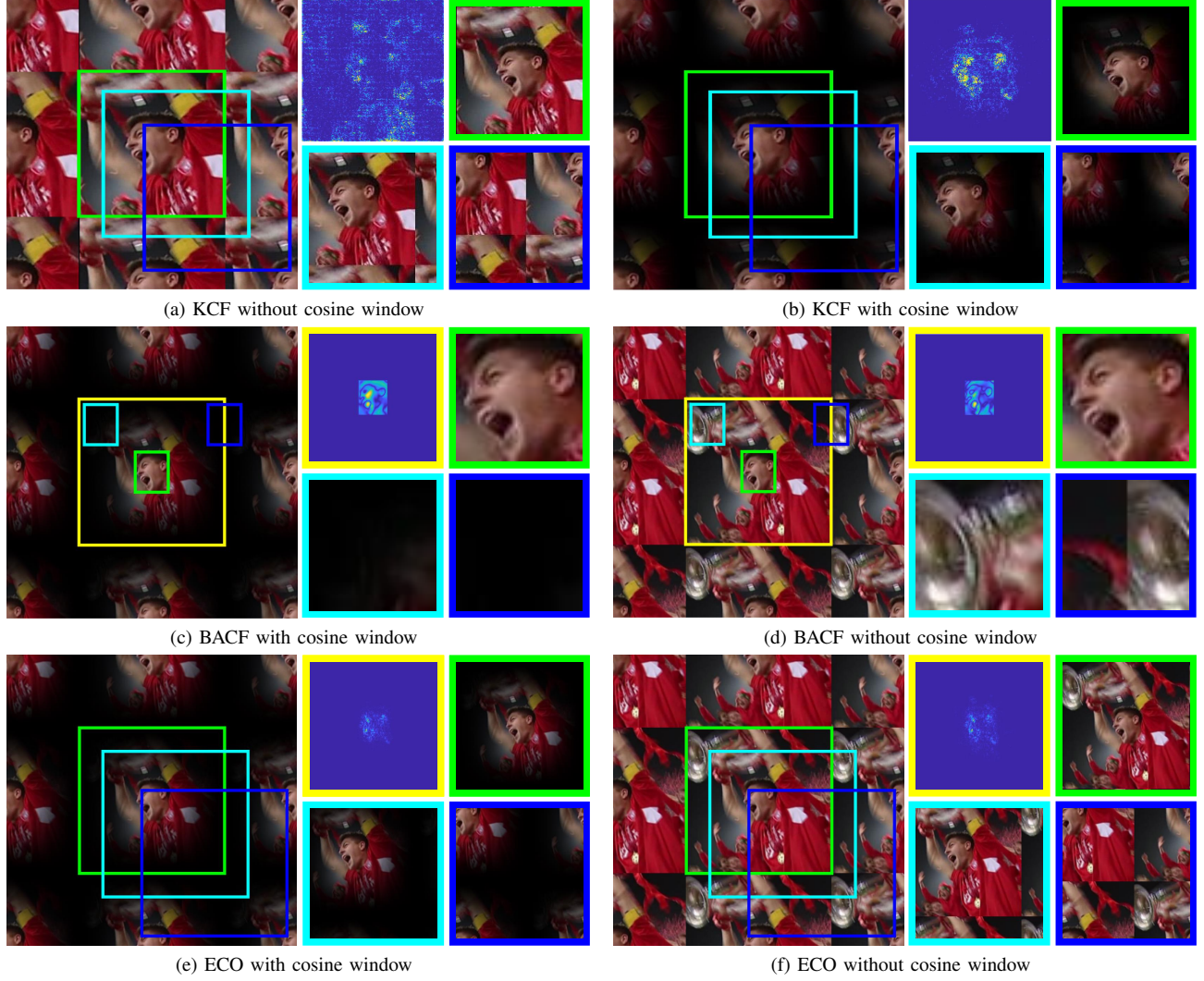
Fig. 1. Illustration of the cyclic extension of base image, representative samples and learned filters by KCF, BACF and ECO with and without cosine window, including KCF (a) without and (b) with cosine window, BACF (c) with and (d) without cosine window, ECO (e) with and (f) without cosine window. In each of (a)(b)(c)(d)(e)(f), the left part shows the cyclic extension of base image, while the right part illustrates the learned filter and three training samples cropped from the left part. One can observe from (a)(b)(c)(e) that while cosine window plays similar roles with spatial regularization on handing boundary discontinuity, it has the risk of reducing the quality of training samples. And when removing cosine window from BACF, ECO in (d) and (f), the learned filter does not change much in its appearance. These results motivate us to remove cosine window from CF trackers with spatial regularization.

contribute much on model learning, thereby leading to the degraded performance.

Comparing the filters in Fig. 1(a)(b)(c)(e), one can see that cosine window plays a similar role as spatial regularization in enforcing the filter coefficients near boundary to approach zero. Therefore, it is interesting to ask the first problem concerned in this work: when can we remove cosine window from CFs? In Fig. 1(d)(f), we present the learned filters by simply removing cosine window from both BACF and ECO trackers. It can be observed that the filters in both Fig. 1(c)(d) and Fig. 1(e)(f) are only moderately different in appearances. Our empirical study further shows that the two trackers without cosine window performs slightly inferior to their baseline methods with cosine window. Thus, our answer to this question is: when spatial regularization is deployed, it is possible to remove cosine window from CF trackers.

The second problem concerned in this work is: how to remove cosine window from CF trackers with spatial regularization. To begin with, Fig. 1(d) illustrates three representative samples used in BACF by simply removing cosine window. While most samples are real image patches (e.g., those in green and cyan boxes), there remain a small percentage of negative samples suffering from boundary discontinuity (e.g., the patch near boundary in blue box). To address this issue, we introduce a binary mask function to eliminate the effect of boundary discontinuous sample. In particular, we assign zero to negative samples with discontinuous boundaries, thereby safely removing cosine window. To further improve tracking performance, a Gaussian shaped mask function is also presented to emphasize more on samples near target center.

To evaluate the feasibility and effectiveness of removing cosine window, we incorporate our methods with several representative CF trackers with spatial regularization, including BACF [16], STRCF [17], ECO [12], and UPDT [18].

Experiments are then conducted on four tracking benchmarks, i.e., OTB-2015 [19], Temple-Color [20], VOT-2018 [21] and LaSOT datasets. In comparison to the baseline methods with cosine window, our methods are effective in handling boundary discontinuity while avoiding sample contamination, and give rise to more robust appearance models as well as tracking performance. Moreover, by incorporating with UPDT [18], our methods achieve the state-of-the-art tracking performance, and attain an EAO score of 0.391 on VOT-2018, surpassing the rank-1 tracker (i.e., LADCF [22]) in the VOT2018 challenge.

To sum up, the main contributions of this paper are:

- The existing CF methods adopt the cosine window to alleviate boundary discontinuity, but neglect its risk of contaminating the negative samples during model training. As far as we are concerned, we make the first attempt to address this issue by removing cosine window from CF trackers to avoid the sample contamination.
- *When to remove cosine window from CF trackers?* we perform analyses and empirical studies to show the similar roles of cosine window with spatial regularization on reducing boundary discontinuity. Based on these results, we find that it is possible to remove cosine window from CF methods with spatial regularization.
- *How to remove cosine window from CF trackers with spatial regularization?* We propose two mask functions to safely remove cosine window from CF trackers with spatial regularization. Since there still exist several samples with discontinuous boundaries after removing cosine window from CF trackers with spatial regularization, we incorporate two mask functions into the CF models to reweight the estimation errors of each sample. In this way, the resulting models can both eliminate boundary discontinuity and avoid sample contamination.
- Our methods are generic and can be incorporated with several representative CF trackers with spatial regularization. The experimental results on multiple datasets show that the proposed methods outperform their baseline methods with cosine window, and perform favorably with state-of-the-art methods.

The remainder of this paper is organized as follows. Section II briefly reviews the CF trackers relevant to this work. Section III provides both qualitative and quantitative analyses to dissect the effect of removing cosine window from CF trackers. Section IV further describes our solutions to remove cosine window, which are then incorporated with multiple CF trackers with spatial regularization. Section V reports the experimental results. Finally, Section VI ends this work with several concluding remarks.

II. RELATED WORK

The core problem of CF trackers is to learn a discriminative filter for the next frame from current frame and historical information. Early methods, e.g., MOSSE [1] and KCF [2], formulate the CF framework with one single base image from the current frame, and update the CFs using the linear interpolation strategy. Denote by the sample pair $\{(\mathbf{x}_t, \mathbf{y}_t)\}$ in frame t, where each sample $\mathbf{x}_t$ consists of L feature maps with

$\mathbf{x}_t = [\mathbf{x}_{t,1}, ..., \mathbf{x}_{t,L}]$, and $\mathbf{y}_t$ represents the Gaussian shaped label. Then the correlation filter $\mathbf{f}$ is obtained by minimizing the following objective,

$$\mathcal{E}(\mathbf{f}) = \frac{1}{2} \left\| \sum_{l=1}^{L} \mathbf{f}_l \star (\mathbf{x}_{t,l} \odot \mathbf{c}) - \mathbf{y}_t \right\|^2 + \lambda \mathcal{R}(\mathbf{f}), \quad (1)$$

where $\star$ and $\odot$ respectively stand for circular convolution and Hadamard product, $\mathbf{c}$ denotes cosine window, and λ denotes the tradeoff parameter of the regularization term $\mathcal{R}(\mathbf{f})$.

Since the pioneering work of MOSSE [1], many improvements have been made to CF trackers. On the one hand, the CF models have been consistently improved with the introduction of non-linear kernel [2], scale adaptivity [4], [5], [6], long-term tracking [23], part-based CFs [24], particle filters [25], spatial regularization [9], [17], continuous convolution [11], [12], and formulation with multiple base images [8], [12], [18]. On the other hand, the progress in feature engineering has also been advancing the state-of-the-art tracking performance, such as HOG [26], color features [27], [28], self-similarity local features [29] and hierarchal CNN features [13], [14], [15], [30], [31].

Among these improvements, we specifically mention a category of CF formulations with multiple base images [8], [12], [18]. Given a set of K base images $\{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=1}^{K}$, CF with multiple base images can then be expressed as,

$$\mathcal{E}(\mathbf{f}) = \frac{1}{2} \sum_{k=1}^{K} \alpha_k \left\| \sum_{l=1}^{L} \mathbf{f}_l \star (\mathbf{x}_{k,l} \odot \mathbf{c}) - \mathbf{y}_k \right\|^2 + \lambda \mathcal{R}(\mathbf{f}), \quad (2)$$

where α_k represents the weight of the k-th base image $\mathbf{x}_k$. For example, SRDCF [8] and CCOT [11] simply adopt the latest K frames as base images. In SRDCFdecon [32], an adaptive decontamination model is presented to downweight corrupted samples while up-weighting faithful ones. ECO [12] and UPDT [18] apply a Gaussian mixture model (GMM) to determine both the weights as well as base images. In general, CF trackers with multiple base images perform much better than those with single base image, and have achieved state-of-the-art tracking performance.

In contrast to CF with single base image in Eqn. (1), the introduction of multiple base images breaks the circulant structure, and generally requires iterative optimization algorithms to solve the resulting formulation in Eqn. (2). Therefore, in this work different solutions are respectively developed for removing cosine window from CF trackers with single and multiple base images.

III. WHEN TO REMOVE COSINE WINDOW

Cosine window is first introduced in the early MOSSE and KCF methods to alleviate the effect of boundary discontinuity, and then adopted in all the subsequent CF trackers. In the recent few years, spatial regularization has also been deployed in CF trackers for handling boundary discontinuity. Albeit cosine window is also adopted in CF with spatial regularization, considering their similar roles, it is natural to ask whether it is possible to remove cosine window from CF when spatial regularization is adopted.

In this section, we use KCF and BACF as two representative examples, and evaluate the performance of CF trackers with and without cosine window on the VOT-2018 dataset. Here we name KCF and BACF without cosine window as KCF_{RC} and $BACF_{RC}$, respectively. Table I lists their EAO, accuracy and robustness on VOT-2018. For KCF, it can be seen that removing cosine window is harmful to tracking performance and gives rise to an obvious EAO drop from 0.106 to 0.069. From Fig. 1(a)(b), the filter learned by KCF is much different from that learned by KCF_{RC} in appearance. Moreover, cosine window also performs similarly in enforcing non-central filter coefficients to approach zero. In contrast to KCF, cosine window actually plays a minor role on improving tracking performance for BACF, and the EAO of $BACF_{RC}$ is only 0.013 lower than that of BACF. From Fig. 1(c)(d), the filters learned by BACF and $BACF_{RC}$ are also similar in appearance. Similar results can also be observed for STRCF [17], ECO [12] and UPDT [18] on the VOT-2018 benchmark in Section V, indicating that it is possible to remove cosine window from CF trackers when spatial regularization is introduced.

We also note that BACF still slightly outperforms $BACF_{RC}$, which can be explained by taking both boundary discontinuity and sample contamination into account. From Fig. 1(c), it can be seen that BACF can well handle boundary discontinuity by incorporating cosine window and spatial regularization. However, cosine window is deployed on the base image, which makes the shifted negative samples contaminated and may be harmful to tracking performance. In $BACF_{RC}$, one can see from Fig. 1(d) that most negative samples are real image patches (e.g., those in green and cyan boxes). However, a small percentage of negative samples still suffer from boundary discontinuity (e.g., that in blue box), which may explain the slight inferiority of $BACF_{RC}$ in comparison to BACF. To sum up, for removing cosine window, it is better to avoid sample contamination as well as eliminate boundary discontinuity for all negative samples. Thus, we turn to the second problem of this work, i.e., how to remove cosine window from CF trackers with spatial regularization, and present our solutions in the next section.

IV. HOW TO REMOVE COSINE WINDOW

Simply removing cosine window from CF trackers with spatial regularization generally cannot outperform the baseline methods because the negative samples near boundary still suffer from boundary discontinuity. To address this issue, we modify the formulation of CF trackers by introducing mask function to deactivate the boundary discontinuous samples. Two mask functions are presented to eliminate boundary discontinuity as well as emphasize more on samples near the target center. Then, optimization algorithms are respectively developed for removing cosine window from CF trackers with single and multiple base images.

A. Problem formulation

Without loss of generality, we use BACF as an example to analyze the positions of negative samples suffering from boundary discontinuity. Suppose that the sizes of the target

TABLE I
THE EAO, ACCURACY AND ROBUSTNESS OF TWO CF TRACKERS (I.E., KCF AND BACF) AND THEIR CORRESPONDING VARIANTS WITHOUT USING COSINE WINDOW DURING TRAINING (I.E., KCF_{RC} AND $BACF_{RC}$) ON THE VOT-2018 DATASET. HERE, ↑ (↓) DENOTES HIGHER (LOWER) IS BETTER.

Methods	KCF [2]	KCF_{RC}	BACF [16]	$BACF_{RC}$
EAO (↑)	0.106	0.069	0.137	0.124
Accuracy (↑)	0.327	0.374	0.432	0.466
Robustness (↓)	1.182	1.823	0.757	0.892

bounding box and base image are $h \times w$ and $H \times W$, respectively. For BACF, we have $H = W = 5\sqrt{hw}$. From Fig. 2(a), it can be seen that only the samples at position (x, y) are with discontinuous boundaries when $\frac{H}{2} \geq |x| > \frac{H}{2} - \frac{h}{2}$ or $\frac{W}{2} \geq |y| > \frac{W}{2} - \frac{w}{2}$. In general, H (W) is much larger than h (w), and thus the majority of samples (e.g., 64% when $h = w$) are real image patches. In order to eliminate the effect of boundary discontinuity, we introduce a binary mask function $\mathbf{M}$ shown in Fig. 2(b) to indicate the samples of real image patches. In particular, a sample at position (x, y) is a real image patch when $M(x, y) = 1$. Then the binary mask $\mathbf{M}$ can be defined as follows,

$$M(x, y) = \begin{cases} 1, & \text{if } |x| \leq \frac{H}{2} - \frac{h}{2}, \ |y| \leq \frac{W}{2} - \frac{w}{2}, \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

Furthermore, the CF model usually is learned from an unbalanced set containing few positive samples and a large amount of negative samples. The binary mask $\mathbf{M}$ treats all boundary continuous samples equally, and has the risk of degrading tracking performance due to vast negative samples. Considering that the samples near the target center are more important than those on image boundaries, we also present a Gaussian shaped mask function $\mathbf{M}_G$ defined as,

$$M_G(x, y) = \begin{cases} e^{-(\frac{x}{h\delta})^2 - (\frac{y}{w\delta})^2}, & \text{if } |x| \leq \frac{H}{2} - \frac{h}{2}, |y| \leq \frac{W}{2} - \frac{w}{2}, \\ 0, & \text{otherwise,} \end{cases}$$
$$(4)$$

where the parameter δ is introduced to control the weight decay speed of training samples. Empirical study also validates that Gaussian shaped mask function $\mathbf{M}_G$ generally performs moderately better than binary mask function $\mathbf{M}$ for CF trackers with spatial regularization.

In the following, we use Eqn. (2) as a general form to illustrate how to eliminate boundary discontinuity while avoiding sample contamination for CF trackers with spatial regularization. In particular, we remove cosine window from Eqn. (2), and incorporate the binary mask $\mathbf{M}$ to deactivate the negative samples suffering from boundary discontinuity, resulting in the following model,

$$\mathcal{E}(\mathbf{f}) = \frac{1}{2} \sum_{k=1}^{K} \alpha_k \left\| \mathbf{M} \odot \left(\sum_{l=1}^{L} \mathbf{f}_l \star \mathbf{x}_{k,l} - \mathbf{y}_k \right) \right\|^2 + \lambda \mathcal{R}(\mathbf{f}). \quad (5)$$

With the introduction of $\mathbf{M}$, the estimation error of the sample with discontinuous boundary can be safely excluded during training. In comparison to CF tracker with cosine window in
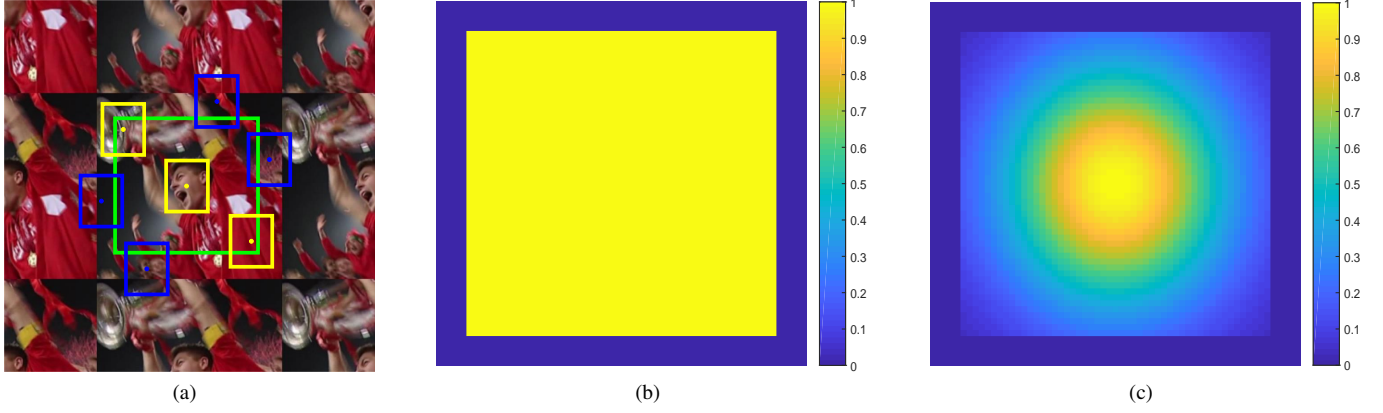
Fig. 2. Illustration of the mask functions. (a) Training samples from the video *soccer* used in BACF, where the samples outside green box suffer from the boundary discontinuity problem, (b) the binary mask function $\mathbf{M}$, (c) the Gaussian shaped mask function $\mathbf{M}_G$. In order to eliminate boundary discontinuity, a binary mask function $\mathbf{M}$ is introduced to deactivate the patches with discontinuous boundaries by setting their values in $\mathbf{M}$ to 0. And another Gaussian shaped mask function $\mathbf{M}_G$ is further proposed to emphasize more on samples near target center.

Eqn. (2), the formulation in Eqn. (5) can circumvent both boundary discontinuity and sample contamination, thereby benefiting tracking performance.

Given a specific CF tracker, we denote the models by (i) removing cosine window, (ii) removing cosine window and incorporating binary mask function, (iii) removing cosine window and incorporating Gaussian mask function as CF_{RC}, CF_{RCB}, and CF_{RCG}, respectively. In the following, we present the optimization algorithms to solve the model in Eqn. (5) for CF trackers with single and multiple base images, respectively.

B. Solution for CF trackers with single base image

For CFLB [33], BACF [16], CSR-DCF [10] and STR-CF [17], the filter is updated by solving a specific CF model defined on a single base image (i.e., the current frame). In this case, the resulting constrained optimization problem can be efficiently solved via alternating minimization, in which each subproblem has the closed-form solution. When removing cosine window from this category of CF trackers with spatial regularization, we rewrite the model in Eqn. (5) as

$$\mathcal{E}(\mathbf{f}) = \frac{1}{2} \left\| \mathbf{M} \odot \left(\sum_{l=1}^{L} \mathbf{f}_l \star \mathbf{x}_l - \mathbf{y} \right) \right\|^2 + \lambda \mathcal{R}(\mathbf{f}). \quad (6)$$

Since the mask function $\mathbf{M}$ is defined as a constant non-negative matrix, the data term $\left\| \mathbf{M} \odot \left(\sum_{l=1}^{L} \mathbf{f}_l \star \mathbf{x}_l - \mathbf{y} \right) \right\|^2$ shares the same convex property with $\left\| \sum_{l=1}^{L} \mathbf{f}_l \star \mathbf{x}_l - \mathbf{y} \right\|^2$ according to the operations preserving convexity of functions in [34]. Suppose Eqn. (6) is a convex function and can be solved via the ADMM algorithm. According to the Eckstein-Bertsekas condition [35], the ADMM algorithm is guaranteed to converge to global optimum when Eqn. (6) is convex and each subproblem in ADMM algorithm has closed-form solution. In the following, we take BACF as an example, and present an alternating direction method of multipliers (ADMM) to optimize the resulting formulation.

With simple algebra, the original formulation of BACF can be equivalently rewritten as,

$$\mathcal{L}(\mathbf{g}) = \frac{1}{2} \left\| \sum_{l=1}^{L} (\mathbf{x}_l \odot \mathbf{c}) \star (\mathbf{P}^T \mathbf{g}_l) - \mathbf{y} \right\|^2 + \frac{\lambda}{2} \|\mathbf{g}\|^2, \quad (7)$$

where $\mathbf{P}$ stands for the binary mask matrix which crops the central D elements of $\mathbf{g}_l$ with the size of T. After removing cosine window and incorporating with the mask function $\mathbf{M}$, we further let $\mathbf{f}_l = \mathbf{P}^T \mathbf{g}_l$, and the modified BACF model can be formulated as,

$$\mathcal{L}(\mathbf{f}, \mathbf{g}) = \frac{1}{2} \left\| \mathbf{M} \odot \left(\sum_{l=1}^{L} \mathbf{x}_l \star \mathbf{f}_l - \mathbf{y} \right) \right\|^2 + \frac{\lambda}{2} \|\mathbf{g}\|^2,$$
$$\text{s.t.} \quad \mathbf{f}_l = \mathbf{P}^T \mathbf{g}_l. \quad (8)$$

The model in Eqn. (8) is still a convex optimization problem, can be solved with the ADMM algorithm. To begin with, we introduce another auxiliary variable $\mathbf{z} = \sum_{l=1}^{L} \mathbf{x}_l \star \mathbf{f}_l - \mathbf{y}$, and reformulate Eqn. (8) as,

$$\mathcal{L}(\mathbf{f}, \mathbf{g}, \mathbf{z}) = \frac{1}{2} \|\mathbf{M} \odot \mathbf{z}\|^2 + \frac{\lambda}{2} \|\mathbf{g}\|^2,$$
$$\text{s.t.} \quad \mathbf{f}_l = \mathbf{P}^T \mathbf{g}_l, \quad \mathbf{z} = \sum_{l=1}^{L} \mathbf{x}_l \star \mathbf{f}_l - \mathbf{y}. \quad (9)$$

Then the augmented Lagrangian function of Eqn. (9) is given as,

$$\mathcal{L}(\mathbf{f}, \mathbf{g}, \mathbf{z}, \boldsymbol{\zeta}, \boldsymbol{\gamma}) = \frac{1}{2} \|\mathbf{M} \odot \mathbf{z}\|^2 + \frac{\lambda}{2} \|\mathbf{g}\|^2$$
$$+ \sum_{l=1}^{L} \boldsymbol{\zeta}_l^T (\mathbf{f}_l - \mathbf{P}^T \mathbf{g}_l) + \frac{\mu}{2} \sum_{l=1}^{L} \left\| \mathbf{f}_l - \mathbf{P}^T \mathbf{g}_l \right\|^2$$
$$+ \boldsymbol{\gamma}^T \left(\sum_{l=1}^{L} \mathbf{x}_l \star \mathbf{f}_l - \mathbf{y} - \mathbf{z} \right) + \frac{\tau}{2} \left\| \sum_{l=1}^{L} \mathbf{x}_l \star \mathbf{f}_l - \mathbf{y} - \mathbf{z} \right\|^2, \quad (10)$$

where $\boldsymbol{\zeta}$, $\boldsymbol{\gamma}$ denote the Lagrangian multipliers, and μ, τ represent the penalty parameters, respectively. Eqn. (10) can be solved iteratively with the ADMM algorithm, in which all the subproblems, i.e., $\mathbf{f}$, $\mathbf{g}$ and $\mathbf{z}$, have their closed-form solutions. In the following, we present the solution of each subproblem.

Subproblem g:

$$\arg\min_{\mathbf{g}} \frac{\lambda}{2}\|\mathbf{g}\|^2 + \sum_{l=1}^{L}\boldsymbol{\zeta}_l^{\mathrm{T}}(\mathbf{f}_l - \mathbf{P}^{\mathrm{T}}\mathbf{g}_l) + \frac{\mu}{2}\sum_{l=1}^{L}\left\|\mathbf{f}_l - \mathbf{P}^{\mathrm{T}}\mathbf{g}_l\right\|^2. \quad (11)$$

Note that each channel of $\mathbf{g}$ in Eqn. (11) can be computed independently, thus the closed-form solution of the l-th channel of $\mathbf{g}$ can be expressed as,

$$\mathbf{g}_l = \left(\lambda\mathbf{I} + \mu\mathbf{P}\mathbf{P}^{\mathrm{T}}\right)^{-1}\left(\mathbf{P}\boldsymbol{\zeta}_l + \mu\mathbf{P}\mathbf{f}_l\right), \quad (12)$$

where $\mathbf{I}$ denotes an identity matrix. Note that $\lambda\mathbf{I} + \mu\mathbf{P}\mathbf{P}^{\mathrm{T}}$ is a diagonal matrix and its inverse matrix can be efficiently computed via element-wise operation.

Subproblem f:

$$\arg\min_{\mathbf{f}} \sum_{l=1}^{L}\boldsymbol{\zeta}_l^{\mathrm{T}}(\mathbf{f}_l - \mathbf{P}^{\mathrm{T}}\mathbf{g}_l) + \frac{\mu}{2}\sum_{l=1}^{L}\left\|\mathbf{f}_l - \mathbf{P}^{\mathrm{T}}\mathbf{g}_l\right\|^2$$
$$+ \boldsymbol{\gamma}^{\mathrm{T}}\left(\sum_{l=1}^{L}\mathbf{x}_l \star \mathbf{f}_l - \mathbf{y} - \mathbf{z}\right) + \frac{\tau}{2}\left\|\sum_{l=1}^{L}\mathbf{x}_l \star \mathbf{f}_l - \mathbf{y} - \mathbf{z}\right\|^2. \quad (13)$$

Using Parseval's theorem, Eqn. (13) can be equivalently expressed in the Fourier domain,

$$\arg\min_{\hat{\mathbf{f}}} \sum_{l=1}^{L}\hat{\boldsymbol{\zeta}}_l^{\mathrm{T}}\left(\hat{\mathbf{f}}_l - \hat{\mathbf{q}}_l\right) + \frac{\mu}{2}\sum_{l=1}^{L}\left\|\hat{\mathbf{f}}_l - \hat{\mathbf{q}}_l\right\|^2$$
$$+ \hat{\boldsymbol{\gamma}}^{\mathrm{T}}\left(\sum_{l=1}^{L}\hat{\mathbf{x}}_l \odot \hat{\mathbf{f}}_l - \hat{\mathbf{y}} - \hat{\mathbf{z}}\right) + \frac{\tau}{2}\left\|\sum_{l=1}^{L}\hat{\mathbf{x}}_l \odot \hat{\mathbf{f}}_l - \hat{\mathbf{y}} - \hat{\mathbf{z}}\right\|^2. \quad (14)$$

Here $\hat{\mathbf{x}} = \sqrt{T}\mathbf{F}\mathbf{x}$ represents the FFT of sample $\mathbf{x}$ where $\mathbf{F}$ is the orthonormal Discrete Fourier Transform (DFT) matrix, and $\hat{\mathbf{q}}_l$ takes the form of $\hat{\mathbf{q}}_l = \sqrt{T}\mathbf{F}\mathbf{P}^{\mathrm{T}}\mathbf{g}_l$. Analogous to BACF [16], the solution for $\hat{\mathbf{f}}$ can be divided into T independent subproblems. Denote by $\mathbf{x}(t) \in \mathbb{R}^L$ the vector consisting of t-th elements of sample $\mathbf{x}$ along all L channels, then the t-th elements $\hat{\mathbf{f}}(t)$ of $\hat{\mathbf{f}}$ can be computed by,

$$\hat{\mathbf{f}}(t) = \left(\tau\hat{\mathbf{x}}(t)\hat{\mathbf{x}}(t)^{\mathrm{T}} + \mu\mathbf{I}\right)^{-1}$$
$$\left(\tau\hat{\mathbf{x}}(t)\hat{\mathbf{y}}(t) + \tau\hat{\mathbf{x}}(t)\hat{\mathbf{z}}(t) - \hat{\mathbf{x}}(t)\boldsymbol{\gamma}(t) - \hat{\boldsymbol{\zeta}}(t) + \mu\hat{\mathbf{q}}(t)\right). \quad (15)$$

Note that $\hat{\mathbf{x}}(t)\hat{\mathbf{x}}(t)^{\mathrm{T}}$ is rank-1 matrix, thus Eqn. (15) can be efficiently solved with Sherman-Morrison formula [36],

$$\hat{\mathbf{f}}(t) = \frac{1}{\mu}\left(\tau\hat{\mathbf{x}}(t)\hat{\mathbf{y}}(t) + \tau\hat{\mathbf{x}}(t)\hat{\mathbf{z}}(t) - \hat{\mathbf{x}}(t)\hat{\boldsymbol{\gamma}}(t) - \hat{\boldsymbol{\zeta}}(t) + \mu\hat{\mathbf{q}}(t)\right)$$
$$- \frac{\hat{\mathbf{x}}(t)}{\mu b}\left(\tau\hat{\mathbf{y}}(t)\hat{s}_{\mathbf{x}}(t) + \tau\hat{\mathbf{z}}(t)\hat{s}_{\mathbf{x}}(t) - \hat{\boldsymbol{\gamma}}(t)\hat{s}_{\mathbf{x}}(t) - \hat{s}_{\boldsymbol{\zeta}}(t) + \mu\hat{s}_{\mathbf{q}}(t)\right). \quad (16)$$

where $\hat{s}_{\mathbf{x}}(t) = \hat{\mathbf{x}}(t)^{\mathrm{T}}\hat{\mathbf{x}}(t)$, $\hat{s}_{\boldsymbol{\zeta}}(t) = \hat{\mathbf{x}}(t)^{\mathrm{T}}\hat{\boldsymbol{\zeta}}(t)$, $\hat{s}_{\mathbf{q}}(t) = \hat{\mathbf{x}}(t)^{\mathrm{T}}\hat{\mathbf{q}}(t)$ and $b = \frac{\mu}{\tau} + \hat{s}_{\mathbf{x}}(t)$. And the solution for $\mathbf{f}$ is further obtained with the inverse DFT operation.

Subproblem z:

$$\arg\min_{\mathbf{z}} \frac{1}{2}\|\mathbf{M}\odot\mathbf{z}\|^2 + \boldsymbol{\gamma}^{\mathrm{T}}\left(\sum_{l=1}^{L}\mathbf{x}_l \star \mathbf{f}_l - \mathbf{y} - \mathbf{z}\right)$$
$$+ \frac{\tau}{2}\left\|\sum_{l=1}^{L}\mathbf{x}_l \star \mathbf{f}_l - \mathbf{y} - \mathbf{z}\right\|^2. \quad (17)$$

Analogous to Eqn. (11), each element in $\mathbf{z}$ can also be computed independently, and its solution is expressed as follows,

$$\mathbf{z} = (\mathrm{Diag}(\mathbf{M}\odot\mathbf{M} + \tau\mathbf{1}))^{-1}\left(\tau(\sum_{l=1}^{L}\mathbf{x}_l \star \mathbf{f}_l - \mathbf{y}) + \boldsymbol{\gamma}\right), \quad (18)$$

where $\mathbf{1}$ defines a vector in which each element equals to the value 1, and $\mathrm{Diag}(\cdot)$ constructs a diagonal matrix from a vector.

Lagrangian Update: The Lagrangian multipliers $\boldsymbol{\zeta}$, $\boldsymbol{\gamma}$ are updated as,

$$\boldsymbol{\zeta}^{(t+1)} = \boldsymbol{\zeta}^{(t)} + \mu\left(\mathbf{f}^{(t+1)} - \mathbf{P}^{\mathrm{T}}\mathbf{g}^{(t+1)}\right),$$
$$\boldsymbol{\gamma}^{(t+1)} = \boldsymbol{\gamma}^{(t)} + \tau\left(\sum_{l=1}^{L}\mathbf{x}_l \star \mathbf{f}_l^{(t+1)} - \mathbf{y} - \mathbf{z}^{(t+1)}\right). \quad (19)$$

where $\mathbf{f}^{(t+1)}$, $\mathbf{g}^{(t+1)}$ and $\mathbf{z}^{(t+1)}$ are the solutions to the above subproblems at iteration $t+1$.

Finally, we also note that the above solutions can be easily extended to remove cosine window from other CF trackers (e.g., STRCF) with a single base image.

C. Solution for CF trackers with multiple base images

Another category of CF trackers with spatial regularization is defined on multiple base images, which inevitably breaks the circulant structure and requires iterative optimization to solve some of the resulting subproblems. Several representative trackers in this category include SRDCF [8], CCOT [11], ECO [12] and UPDT [18]. In this subsection, we use ECO as an example to suggest an iterative optimization method for removing cosine window. Without loss of generality, our solution can be easily extended to remove cosine window from other CF trackers based on multiple base images (e.g., UPDT [18]).

In general, the learning algorithm in ECO consists of two stages. (i) In the first frame, a sample projection matrix is learned with the CF to reduce the number of feature channels in training samples. (ii) In the subsequent frames the projection matrix is fixed and the CFs are further updated with the reduced features. To keep consistent with the ECO tracker [12], we also define the formulation for data on a one-dimension domain. Denote by a collection of K sample pairs $\{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=1}^{K}$, and the feature map size for the l-th channel $\mathbf{x}_{k,l}$ is N_l. The feature map $\mathbf{x}_{k,l}$ in ECO tracker is first transformed into the continuous spatial domain $t \in [0, T)$ with an interpolation operator J_l,

$$J_l\{\mathbf{x}_{k,l}\}(t) = \sum_{n=0}^{N_l-1}\mathbf{x}_{k,l}[n]b_l(t - \frac{T}{N_l}n), \quad (20)$$

where b_l is an interpolation kernel with the period $T > 0$. Suppose the reduced correlation filter $\mathbf{f} = [\mathbf{f}_1, ..., \mathbf{f}_D]$ consists of D feature maps with $D < L$, and the projection matrix $\mathbf{Q} \in \mathbb{R}^{L \times D}$ is represented as $\mathbf{Q} = \{q_{l,d}|l = 1, ..., L, d = 1, ..., d\}$. Then the filter $\mathbf{f}$ and sample projection matrix $\mathbf{Q}$ can be

computed by minimizing the following objective function,

$$
\mathcal{E}(\mathbf{f}, \mathbf{Q}) = \frac{1}{2} \sum_{k=1}^{K} \alpha_k \left\| \sum_{d=1}^{D} \sum_{l=1}^{L} q_{l,d} \mathbf{f}_d \star (J_l\{\mathbf{x}_{k,l}\} \odot \mathbf{c}) - \mathbf{y}_k \right\|^2 \\
+ \frac{1}{2} \sum_{d=1}^{D} \|\mathbf{w} \odot \mathbf{f}_d\|^2 + \frac{\lambda}{2} \|\mathbf{Q}\|^2,
\tag{21}
$$

where $\mathbf{w}$ denotes the spatial regularization matrix.

When removing cosine window and incorporating the mask function $\mathbf{M}$, the ECO model can be modified as,

$$
\mathcal{E}(\mathbf{f}, \mathbf{Q}) = \frac{1}{2} \sum_{k=1}^{K} \alpha_k \left\| \mathbf{M} \odot (\sum_{d=1}^{D} \sum_{l=1}^{L} q_{l,d} \mathbf{f}_d \star J_l\{\mathbf{x}_{k,l}\} - \mathbf{y}_k) \right\|^2 \\
+ \frac{1}{2} \sum_{d=1}^{D} \|\mathbf{w} \odot \mathbf{f}_d\|^2 + \frac{\lambda}{2} \|\mathbf{Q}\|^2.
\tag{22}
$$

To solve Eqn. (22), we introduce a series of auxiliary variables $[\mathbf{z}_1, ..., \mathbf{z}_K]$ with $\mathbf{z}_k = \sqrt{\alpha_k}(\sum_{d=1}^{D} \sum_{l=1}^{L} q_{l,d} \mathbf{f}_d \star J_l\{\mathbf{x}_{k,l}\} - \mathbf{y}_k)$, then it can be relaxed as,

$$
\mathcal{E}(\mathbf{f}, \mathbf{Q}, \mathbf{z}_k) = \frac{1}{2} \sum_{k=1}^{K} \|\mathbf{M} \odot \mathbf{z}_k\|^2 + \frac{1}{2} \sum_{d=1}^{D} \|\mathbf{w} \odot \mathbf{f}_d\|^2 + \frac{\lambda}{2} \|\mathbf{Q}\|^2 \\
+ \frac{\tau}{2} \sum_{k=1}^{K} \alpha_k \left\| \sum_{d=1}^{D} \sum_{l=1}^{L} q_{l,d} \mathbf{f}_d \star J_l\{\mathbf{x}_{k,l}\} - \mathbf{y}_k - \frac{\mathbf{z}_k}{\sqrt{\alpha_k}} \right\|^2,
\tag{23}
$$

where τ is a penalty parameter which is updated along with the iterations.

We suggest an iterative optimization algorithm for solving the problem in Eqn. (23). In particular, we minimize the objective in each iteration by alternating between updating the auxiliary variables $\mathbf{z}_k$ and the model parameters $\{\mathbf{f}, \mathbf{Q}\}$, which is further explained as follows.

Updating $\{\mathbf{f}, \mathbf{Q}\}$**:** Given the auxiliary variables $[\mathbf{z}_1, ..., \mathbf{z}_K]$, we can observe that the subproblem shares similar formulation with Eqn. (21), thus it can be minimized with the optimization method used in the ECO tracker.

Updating z: Analogous to Eqn. (17), the closed-form solution for $\mathbf{z}_k$ can be computed by,

$$
\mathbf{z}_k = (\mathrm{Diag}(\mathbf{M} \odot \mathbf{M} + \tau \mathbf{1}))^{-1} \tau \sqrt{\alpha_k} \left(\sum_{d=1}^{D} \sum_{l=1}^{L} q_{l,d} \mathbf{f}_d \star J_l\{\mathbf{x}_{k,l}\} - \mathbf{y}_k \right).
\tag{24}
$$

V. EXPERIMENTS

In this section, we evaluate the feasibility and effectiveness of removing cosine window by integrating it into five representative CF trackers with spatial regularization, i.e., BACF, STRCF, ECOhc, ECO and UPDT. Then, extensive experiments are conducted to compare our methods with the state-of-the-art methods on four popular tracking benchmarks, i.e., OTB-2015 [19], Temple-Color [20], VOT-2018 [21] and LaSOT datasets.

A. Baseline CF trackers

Our methods are generic and can be integrated to multiple CF trackers with spatial regularization, such as those with single or multiple base images, using handcrafted or deep CNN features. In the experiments, we choose three baseline CF trackers using handcrafted features, i.e., BACF [16], E-COhc [12] and STRCF [17]. Moreover, we also consider two state-of-the-art baseline CF trackers using CNN features, i.e., ECO [12] and UPDT [18]. It is worth noting that we only incorporate our method with UPDT on the VOT-2018 dataset, because UPDT employs the difficult videos from OTB-2015 for parameter tuning and most of these videos also exist in Temple-Color. Besides, another two CF trackers without spatial regularization, i.e., MOSSE and KCF, are also included to illustrate when to remove cosine window from CF trackers.

B. Implementation details

We employ the publicly available codes provided by the authors to reproduce the results of the baseline CF trackers and competing methods. As for our modified trackers by removing cosine window, we keep most of the parameters the same with their baseline methods, and mainly finetune the parameters added by our methods. In particular, we set the penalty parameters τ, μ, and the number of iterations in BACF$_{RCG}$ and STRCF$_{RCG}$ as {2.5, 2.5, 3} and {2, 2, 4}, respectively. The penalty parameters τ, μ are updated along with iterations by $\tau^{(t+1)} = \min(p\tau^{(t)}, \tau_{max})$ and $\mu^{(t+1)} = \min(p\mu^{(t)}, \mu_{max})$, where τ_{max}, μ_{max} and p are set to {100, 100, 1.05} and {100, 100, 1.07} for BACF$_{RCG}$ and STRCF$_{RCG}$, respectively. As for the ECO and UPDT trackers, the parameters τ and the number of iterations are set to {2.2, 4} and {2.5, 5} and {2.5, 4} for ECOhc$_{RCG}$, ECO$_{RCG}$ and UPDT$_{RCG}$. In addition, we assign the standard deviation parameter δ in Eqn. (4) to {1.2, 1.2, 1.4, 2, 1.8} for BACF$_{RCG}$, STRCF$_{RCG}$, ECOhc$_{RCG}$, ECO$_{RCG}$ and UPDT$_{RCG}$, respectively. Note that we employ the same parameter settings for each tracker on all datasets. Our method is implemented on Matlab 2017b with Matconvnet library [37], and all the experiments are run on a PC with Intel i7 CPU, 32GB RAM and a single NVIDIA GTX 1070 GPU.

C. Internal Analyses of our methods

1) Ablation study: In this section, we study the effect of removing cosine window, incorporating binary or Gaussian shaped mask functions into the baseline CF trackers using the VOT-2018 benchmark [21]. To this end, we implement four variants for each baseline CF tracker, i.e., the baseline CF (termed as **Baseline**), directly incorporating with binary mask function (**B**), directly incorporating with Gaussian shaped mask function (**G**), removing cosine window (**RC**), removing cosine window and incorporating binary mask function (**RCB**), and removing cosine window and incorporating Gaussian shaped mask function (**RCG**). In addition, we also include MOSSE and KCF as baseline trackers to show that their performance is degraded by removing cosine window and cannot be remedied by incorporating mask function. Following the protocols in [38], we evaluate the performance of each method using Expected Average Overlap (EAO), accuracy and robustness as performance measures.

Table II presents the results of all the variants on the VOT-2018 dataset. One can observe that except for the BACF

method, the performance of **B** variants generally outperforms the **Baseline** with an improvement of 0.002 in terms of EAO scores. This can be explained by the fact that incorporating with binary mask function is able to deactivate several samples with discontinuous boundaries, thereby benefiting the tracking performance. Meanwhile, although cosine window implicitly emphasizes more on the samples near the target center, the **G** variants can still boost the performance with an EAO gain of 0.001 over **B**. In addition, when removing the cosine window from MOSSE and KCF, the tracking performance degrades significantly. And they still perform inferior to the **Baseline** methods even with the introduction of binary or Gaussian shaped mask functions. Thus, cosine window cannot be removed from the CF trackers without spatial regularization.

As for the CF trackers with spatial regularization, we can make the following observations. (i) In comparison to **Baseline**, the performance of **RC** variant slightly degrades with a drop of ~ 0.015 in terms of EAO. Such performance degradation can be explained by the fact that a small percentage of negative samples still suffer from boundary discontinuity which may be harmful to tracking performance. (ii) By integrating the binary mask function $\mathbf{M}$ into the CF trackers with spatial regularization, the **RCB** variants consistently outperform the **RC** and **Baseline**. In terms of EAO, the performance gain of **RCB** can be about $0.02 \sim 0.04$ against **RC** and about $0.015 \sim 0.03$ against **Baseline**. The performance improvement can be ascribed to the reason that **RCB** is more effective in handling both boundary discontinuity and sample contamination in comparison with **Baseline** and **RC**. (iii) The introduction of Gaussian shaped mask function $\mathbf{M}_G$ can further boost the performance of CF trackers with spatial regularization, indicating that the samples near target center should be emphasized more in the modified CF models. (iv) While **B** and **G** variants perform better than Baseline, they are still much worse than **RCB** and **RCG** on CF trackers with spatial regularization, validating the effectiveness of the proposed methods. (v) Finally, **RCB** and **RCG** significantly improve the robustness against the **Baseline** trackers with lower failure times. In terms of accuracy, **RCB** and **RCG** perform on par with **Baseline** and **RC**, indicating that the gain of mask function should be attributed to the improvement on the robustness of appearance modeling.

To sum up, the results empirically validate our answers to the two problems concerned in this work. (i) It is feasible to remove cosine window for CF trackers with spatial regularization. (ii) By incorporating with mask function, we can not only safely remove cosine window from CF trackers with spatial regularization, but also bring moderate performance gains over their **Baseline** methods with cosine window.

2) Effect of hyper-parameter δ in $\mathbf{M}_G$: The hyper-parameter δ in Gaussian shaped mask function $\mathbf{M}_G$ controls the decay speed of training samples from target center to boundaries. In particular, higher δ indicates the slower decay speed, and more negative samples near boundary will be considered during training. When $\delta \to +\infty$, the Gaussian shaped mask function $\mathbf{M}_G$ degrades to the binary mask function $\mathbf{M}$. Using BACF, ECOhc and ECO, we analyze the effect of the hyper-parameter δ on tracking performance.
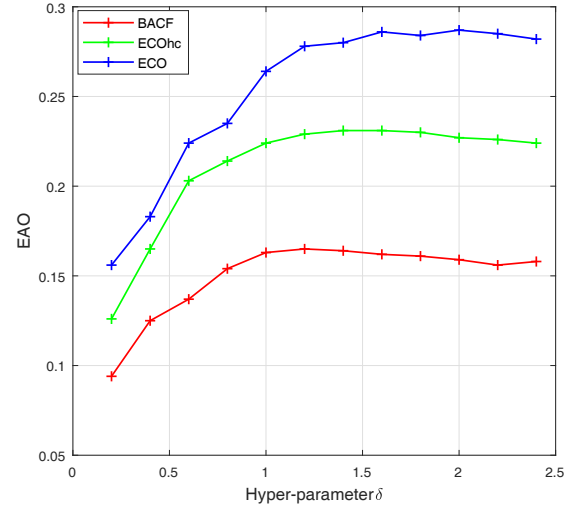

Fig. 3. Effect of the hyper-parameter δ in $\mathbf{M}_G$ for BACF$_{RCG}$, ECOhc$_{RCG}$ and ECO$_{RCG}$ on the VOT-2018 dataset.

Concretely, Fig. 3 shows the EAO plots of the three trackers with different δ values on the VOT-2018 dataset. It can be seen that the choice of δ has a significant effect on EAO score for all the three trackers. For BACF, ECOhc and ECO, the **RCG** methods achieve the best performance when $\delta = \{1.2, 1.4, 2\}$, respectively.

3) The removal of cosine window on tracking stage: In addition to the removal of cosine window on training stage, it is also interesting to investigate the effect of removing cosine window during tracking. Since cosine window can only be removed from CF trackers with spatial regularization in the training stage, we take KCF and BACF$_{RCG}$ methods as two representative examples respectively, and further evaluate their performance of removing cosine window during tracking. Moreover, after removing cosine window during tracking, we also implement two variants by incorporating our mask functions on response maps to penalize large displacements on consecutive frames. In particular, we make four variants for each baseline tracker, i.e., the baseline CF (**Baseline**), removing cosine window during tracking (**TeRC**), removing cosine window and incorporating the binary mask on response map (**TeRCB**), and removing cosine window and incorporating the Gaussian mask on response map (**TeRCG**).

Table III gives the results of all the variants on the VOT-2018 dataset. One can observe that while **TeRC** variants outperform the **Baseline** methods with accuracy gains of 0.051 and 0.024 for KCF and BACF methods, their robustness scores drop significantly than the **Baseline** methods with more tracking failures. And the overall EAO scores of TeRC variants degrade with a drop of 0.042 and 0.048 for KCF and BACF, respectively. It indicates that the cosine window cannot be easily removed in the tracking stage. Moreover, even when banding the binary mask function on the response map, the **TeRCB** variants still perform comparably with **TeRC**, but much worse than **Baseline** methods. However, with the integration of the Gaussian mask function on response map, the **TeRCG** variants significantly improve the performance against **TeRCB**, and perform on par with **Baseline** trackers.

TABLE II
THE EAO, ACCURACY (ACC.) AND ROBUSTNESS (RO.) RESULTS BY PROGRESSIVELY INTEGRATING OUR METHODS INTO THE BASELINE CF TRACKERS ON THE VOT-2018 DATASET. HERE, **Baseline**, **RC**, **RCB**, AND **RCG** RESPECTIVELY REPRESENT THE BASELINE CF TRACKER, THAT BY REMOVING COSINE WINDOW, THAT BY REMOVING COSINE WINDOW AND INCORPORATING WITH BINARY MASK FUNCTION $\mathbf{M}$, AND THAT BY REMOVING COSINE WINDOW AND INCORPORATING WITH GAUSSIAN SHAPED MASK FUNCTION $\mathbf{M}_G$. (*) NOTE THAT THE RESULTS OF ECO AND UPDT ARE REPRODUCED FROM THE RELEASED CODES ON THE VOT-2018 CHALLENGE WEBSITE, AND WE REPORT THE UPDT RESULTS AS THE AVERAGE SCORES OF 15 TIMES RUNNING FOLLOWING THE PROTOCOLS IN [21].

Methods	MOSSE [1]			KCF [2]			BACF [16]			STRCF [17]			ECOhc [12]			ECO* [12]			UPDT* [18]		
	EAO	ACC.	RO.	EAO	ACC.	RO.	EAO	ACC.	RO.	EAO	ACC.	RO.	EAO	ACC.	RO.	EAO	ACC.	RO.	EAO	ACC.	RO.
Baseline	0.067	0.387	1.862	0.106	0.327	1.182	0.137	0.432	0.757	0.174	0.47	0.632	0.212	0.524	0.492	0.262	0.458	0.323	0.352	0.523	0.207
B	0.07	0.372	1.827	0.11	0.304	1.167	0.137	0.432	0.757	0.176	0.464	0.618	0.213	0.522	0.487	0.264	0.454	0.317	0.353	0.522	0.204
G	0.072	0.369	1.804	0.111	0.302	1.162	0.139	0.428	0.749	0.177	0.462	0.613	0.215	0.518	0.483	0.265	0.45	0.311	0.354	0.521	0.202
RC	0.033	0.403	2.438	0.069	0.374	1.823	0.124	0.466	0.892	0.166	0.486	0.683	0.194	0.532	0.521	0.251	0.476	0.334	0.343	0.529	0.221
RCB	0.038	0.408	2.392	0.075	0.377	1.788	0.158	0.462	0.723	0.187	0.478	0.604	0.225	0.528	0.476	0.279	0.464	0.272	0.384	0.524	0.174
RCG	0.042	0.396	2.375	0.081	0.372	1.774	0.165	0.458	0.692	0.192	0.474	0.595	0.231	0.524	0.464	0.287	0.462	0.258	0.391	0.528	0.168

TABLE III
THE EAO, ACCURACY (ACC.) AND ROBUSTNESS (RO.) RESULTS BY COMPARING DIFFERENT VARIANTS OF KCF AND BACF$_{RCG}$ METHODS AFTER REMOVING COSINE WINDOW ON TRACKING STAGE. HERE, **Baseline**, **TeRC**, **TeRCB**, AND **TeRCG** RESPECTIVELY REPRESENT THE BASELINE CF TRACKER (I.E., KCF AND BACF$_{RCG}$), THAT BY REMOVING COSINE WINDOW ON TRACKING STAGE, THAT BY REMOVING COSINE WINDOW AND INCORPORATING BINARY MASK FUNCTION $\mathbf{M}$ ON THE RESPONSE MAP, AND THAT BY REMOVING COSINE WINDOW AND INCORPORATING GAUSSIAN SHAPED MASK FUNCTION $\mathbf{M}_G$ ON THE RESPONSE MAP. (*) NOTE THAT KCF AND BACF$_{RCG}$ ARE TAKEN AS THE **Baseline** METHODS RESPECTIVELY, SINCE COSINE WINDOW CAN ONLY BE REMOVED FROM CF TRACKERS WITH SPATIAL REGULARIZATION IN THE TRAINING STAGE.

Methods	KCF			BACF$_{RCG}$		
	EAO	ACC.	RO.	EAO	ACC.	RO.
Baseline*	0.106	0.327	1.182	0.165	0.458	0.692
TeRC	0.064	0.378	1.847	0.117	0.482	1.236
TeRCB	0.072	0.377	1.822	0.123	0.479	1.207
TeRCG	0.107	0.329	1.183	0.166	0.461	0.695

The results demonstrate that after removing cosine window during tracking, it is vital to enforce strict spatial constraints on response map for reducing the risk of tracking drift. In summary, the empirical studies show that while cosine window cannot be removed directly during tracking, it can be replaced by incorporating the proposed Gaussian mask function on response map.

D. VOT-2018 benchmark

To further assess the proposed methods, we compare our best trackers (i.e., UPDT$_{RCG}$ and ECO$_{RCG}$) with the state-of-the-art trackers on the VOT-2018 dataset. VOT-2018 consists of 60 challenging videos collected from real-life datasets. In the benchmark, a tracker will be re-initialized with the ground-truth bounding boxes whenever it significantly drifts from the target. And the performance is evaluated with three measures: accuracy, robustness and EAO. The accuracy computes the average overlap between estimated bounding boxes and ground-truth annotations. The robustness score counts the times of tracking failures. And EAO measure is a principled combination of accuracy and robustness scores.

Table IV lists the results of our UPDT$_{RCG}$ and ECO$_{RCG}$, ECO, and the top ten best performing trackers on the VOT-2018 challenge. For a fair comparison, we reproduce the results of ECO and UPDT with their publicly available codes on the VOT-2018 challenge website, and the UPDT result is reported as the average score of 15 times running. We also note that the reported EAO score of UPDT on the VOT-2018 challenge is 0.378, while our reproducing result is 0.352 based on the released code on the VOT-2018 challenge. From Table IV, we can observe that UPDT$_{RCG}$ slightly outperforms VOT-2018 challenge winner LADCF and ranks the first among all the competing trackers. UPDT$_{RCG}$ is also superior to its baseline UPDT by an EAO gain of 0.039, indicating the feasibility and benefit of removing cosine window. Not surprisingly, ECO$_{RCG}$ also shows its superiority, i.e., an improvement of 0.025 by EAO score, over the ECO method.

E. OTB-2015 dataset

The OTB-2015 dataset [19] consists of 100 full annotated videos with 11 video attributes, including illumination variations (IV), scale variation (SV), occlusion (OCC), in-plane rotation (IPR), out-of-plane rotation (OPR), motion blur (MB), fast motion (FM), deformation (DEF), background clutter (BC), out of view (OV) and low resolution (LR). Following the settings given in [19], we evaluate the trackers based on the One Pass Evaluation (OPE) protocol, and adopt the overlap precision (OP) metric for calculating the fraction of frames with bounding box overlaps exceeding 0.5 in a sequence. Besides, we also present the overlap success plots with different overlap thresholds for detailed comparison.

To assess our methods, we compare four of them (i.e., STRCF$_{RCG}$, ECOhc$_{RCG}$, BACF$_{RCG}$ and ECO$_{RCG}$) with 24 state-of-the-art trackers, which can be roughly grouped into two categories: (i) trackers using handcrafted features (i.e., STRCF [17], ECOhc [12], BACF [16], DSST [4], SAMF$_{AT}$ [43], Staple [44], TRACA [45], SRDCFDecon [32], SRDCF [8], SKSCF [7] and PTAV [46]), and (ii) trackers using deep CNN features (i.e., ECO [12], CCOT [11], CNN-SVM [47], FCNT [48], CF-Net [49], DeepSTRCF [17], VITAL [50], DeepSRDCF [51], SiameseFC [52], HDT [14], HCF [13], SiamRPN [42] and MDNet [53]). In particular,

TABLE IV
COMPARISON WITH THE STATE-OF-THE-ART TRACKERS IN TERMS OF EAO, ACCURACY, AND ROBUSTNESS ON THE VOT-2018 DATASET. THE FIRST, SECOND AND THIRD BEST RESULTS ARE HIGHLIGHTED IN COLOR. (*) NOTE THAT THE RESULTS OF ECO AND UPDT ARE REPRODUCED FROM THE RELEASED CODES ON VOT-2018 CHALLENGE WEBSITE, AND WE REPORT THE UPDT RESULT AS THE AVERAGE SCORE OF 15 TIMES RUNNING FOLLOWING THE PROTOCOLS IN [21].

Methods	ECO^* [12]	DLSTpp [39]	SA_Siam_R [40]	CPT [21]	DeepSTRCF [17]	$UPDT^*$ [18]	DRT [41]	RCO [21]	SiamRPN [42]	MFT [21]	LADCF [22]	ECO_{RCG} Ours	$UPDT_{RCG}$ Ours
EAO (↑)	0.262	0.325	0.337	0.339	0.345	0.352	0.356	0.376	0.383	0.385	0.389	0.287	0.391
Accuracy (↑)	0.458	0.543	0.566	0.506	0.523	0.523	0.519	0.507	0.586	0.505	0.503	0.462	0.528
Robustness (↓)	0.323	0.224	0.258	0.239	0.215	0.207	0.201	0.155	0.276	0.14	0.159	0.258	0.168

$STRCF_{RCG}$, $ECOhc_{RCG}$, and $BACF_{RCG}$ are compared with the trackers using handcrafted features, while ECO_{RCG} is compared with the trackers using deep CNN features. For a fair comparison, UPDT and $UPDT_{RCG}$ are not included in the comparison because UPDT adopts the difficult videos from OTB-2015 for parameter tuning.

1) Comparison with state-of-the-arts: We compare the proposed methods with the state-of-the-art trackers on OTB-2015. Table V lists the mean OP results of all the competing methods. One can see that our methods are consistently superior to their baseline approaches. Using handcrafted features, $BACF_{RCG}$, $ECOhc_{RCG}$ and $STRCF_{RCG}$ outperform their baseline methods with mean OP gains of 1.1%, 1.3% and 2.3%, respectively. Using deep CNN features, ECO_{RCG} also surpasses its baseline ECO by 1.9% in terms of mean OP. Moreover, our $STRCF_{RCG}$ achieves the best mean OP among the trackers using handcrafted features, while our ECO_{RCG} performs the best among those using deep CNN features. Furthermore, Fig. 4 shows the overlap success curves of the competing methods, which are ranked with the Area-Under-the-Curve (AUC) score. Not surprisingly, our methods perform favorably against the competing trackers using handcrafted and deep CNN features.

2) Attribute comparison: Using the handcrafted features, we further investigate the performance of our methods on all 11 video attributes. Table VI gives the mean OP results of all the trackers. One can see that our ECO_{RCG} and $STRCF_{RCG}$ obtain the rank-1 performance on 9 of all 11 video attributes. For the attributes *motion blur, background clutter, illumination variation and occlusion*, significant improvement can be achieved by our methods. By removing cosine window and incorporating mask function, our methods are more effective in exploiting negative samples for model learning, and benefit the robustness of tracking performance. This may explain the better results of our methods when the target suffers from rapid appearance changes (e.g., motion blur, occlusion, and illumination variation) and background clutter. In addition, Fig. 5 provides the AUC plots of all competing trackers using handcrafted features on all video attributes. It can be seen that our ECO_{RCG} and $STRCF_{RCG}$ also perform favorably against the state-of-the-art methods on most attributes.

3) Running time: Fig. 6 reports the tracking speed (FPS) of the four baseline trackers, i.e., BACF, STRCF, ECOhc and ECO, and their corresponding **RCG** methods on OTB-2015. It can be seen that $BACF_{RCG}$, $STRCF_{RCG}$ achieve
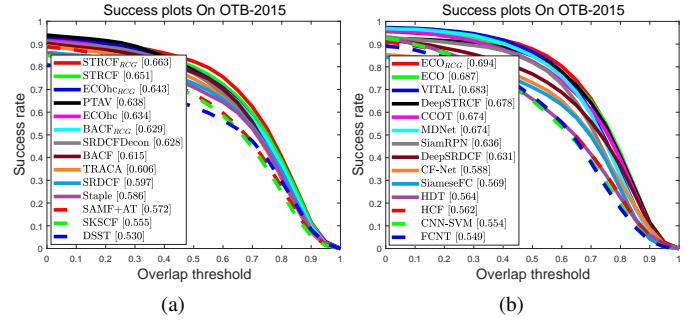

Fig. 4. Comparison of overlap success plots with the state-of-the-art trackers on the OTB-2015 dataset: (a) trackers using handcrafted features, and (b) trackers using deep CNN features.
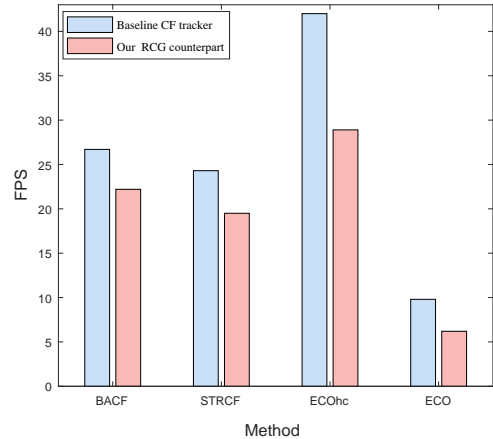

Fig. 6. The tracking speed of four baseline CF trackers, i.e., BACF, STRCF, ECOhc and ECO, and their corresponding **RCG** methods on OTB-2015.

a tracking speed of 22.2 and 19.5 FPS, moderately slower than their baseline methods BACF (26.7 FPS) and STRCF (24.3 FPS). respectively. Thus, while the introduction of mask function increases the model complexity, the two trackers can still be efficiently solved with the ADMM algorithms, and each subproblem has its closed-form solution. As for the trackers with multiple base images, $ECOhc_{RCG}$ runs at approximately 70% speed of the baseline ECOhc (42 FPS), but still maintains real-time tracking performance with 28.9 FPS. When extended to deep CNN features, ECO_{RCG} (5.9 FPS) can run at approximately 70% speed of its baseline ECO method (9.8 FPS).

TABLE V

THE MEAN OP RESULTS (%) OF DIFFERENT TRACKERS USING HANDCRAFTED AND CNN FEATURES ON THE OTB-2015 DATASET. NOTE THAT THE FIRST TWO ROWS COMPARE THE METHODS WITH HANDCRAFTED FEATURES, WHILE THE LAST TWO ROWS CORRESPOND TO THE TRACKERS WITH CNN FEATURES. THE FIRST, SECOND AND THIRD BEST RESULTS ARE HIGHLIGHTED IN COLOR.

Methods	DSST [4]	SKSCF [7]	SAMF$_{AT}$ [43]	Staple [44]	SRDCF [8]	TRACA [45]	SRDCFDecon [32]	BACF [16]	ECOhc [12]	STRCF [17]	BACF$_{RCG}$ Ours	PTAV [46]	ECOhc$_{RCG}$ Ours	STRCF$_{RCG}$ Ours
Mean OP (↑)	62.2	66.5	68	71	72.8	74.7	76.6	76.7	77.2	80	77.8	78.1	78.5	82.3
Methods	CNN-SVM [47]	HCF [13]	HDT [14]	FCNT [48]	SiameseFC [52]	CF-Net [49]	DeepSRDCF [51]	SiamRPN [42]	CCOT [11]	ECO [12]	DeepSTRCF [17]	MDNet [53]	VITAL [50]	ECO$_{RCG}$ Ours
Mean OP (↑)	65.1	65.6	65.8	67.1	71	73	76.8	81.9	82.4	84.8	84.9	84.9	86.6	86.7

TABLE VI

THE MEAN OP RESULTS (%) OF DIFFERENT TRACKERS USING HANDCRAFTED FEATURES ON EACH ATTRIBUTE OF OTB-2015. THE FIRST, SECOND AND THIRD BEST RESULTS ARE HIGHLIGHTED IN COLOR.

Methods	DSST [4]	SKSCF [7]	SAMF$_{AT}$ [43]	Staple [44]	SRDCF [8]	TRACA [45]	SRDCFDecon [32]	BACF [16]	ECOhc [12]	STRCF [17]	BACF$_{RCG}$ Ours	PTAV [46]	ECOhc$_{RCG}$ Ours	STRCF$_{RCG}$ Ours
MB	55.3	63.4	70.8	65	72.9	73.9	79.9	73.5	75.3	79.4	78.4	74.4	77.4	81.5
OCC	56.2	63.4	64.8	68	67.6	71.2	73.5	71.1	74.5	75.1	76.7	73.1	74.5	80.2
IV	65.8	68.6	62.9	72	74.2	76.9	79.3	78.5	76.1	78.3	78.8	79.5	81	80.9
BC	59.9	69	63	67.7	69.2	74.1	78	76	76.5	79.5	79.9	81.5	78.6	82.6
IPR	60.8	65.4	65.5	66.9	66.3	71.5	70	71.5	68.5	73.9	73.9	72.4	70.8	76
OPR	58.3	64.3	64.8	66.5	65.9	72.5	72.4	71.8	72.1	76.6	72.2	73.7	74.8	80.1
SV	55.8	56.3	58.8	61.5	67.1	68.6	74.4	70.2	71.9	76.4	73	70.6	73.6	78.1
FM	55	63.2	66.8	65.9	72.1	70.6	74.6	76	74.5	76	74.6	74.7	72.2	77
DEF	53.1	62.7	58.4	68.6	66.1	70	68.2	71.3	73.7	73.3	68.7	73.9	75.4	73.9
OV	45.5	45.8	60.3	52.3	52.7	67.8	61.8	67.1	63.5	70.9	63.8	68	67.8	70.6
LR	34.7	24	51.4	39.3	64.1	54.9	63.9	62.2	56	69.6	67	56.5	53.8	68.9

4) Qualitative evaluation: Fig. 7 shows the qualitative results of four baseline CF trackers, i.e., BACF, STRCF, ECOhc and ECO, as well as their **RCG** variants. It can be seen from the first row that the target suffers from background clutter and illumination variation. In comparison to baseline BACF, BACF$_{RCG}$ can take the benefit of removing cosine window, thus is able to exploit more useful and uncontaminated training samples for robust model learning, thereby significantly alleviating the tracker drift issue. In the second row, due to the effect of motion blur, fast motion and occlusion challenges, ECOhc cannot track the target throughout the whole sequence while ECOhc$_{RCG}$ still performs well.

In the last two rows, similar phenomena can also be observed when the STRCF and ECO trackers are applied to *coupon* and *freeman4* videos, respectively. In all these videos, the **RCG** variant consistently outperforms its baseline CF method, indicating the effectiveness of removing cosine window and incorporating with mask function.

F. Temple-Color dataset

To further evaluate our methods, comparative experiments are also conducted on the Temple-Color dataset containing 129 color video sequences in total. Fig. 8 shows the overlap success plots of the competing trackers using handcrafted and CNN features. It can be seen from Fig. 8(a) that our methods generally consistently improve the baseline CF trackers using handcrafted features. In particular, BACF$_{RCG}$, ECOhc$_{RCG}$ and STRCF$_{RCG}$ respectively outperform the baseline CF
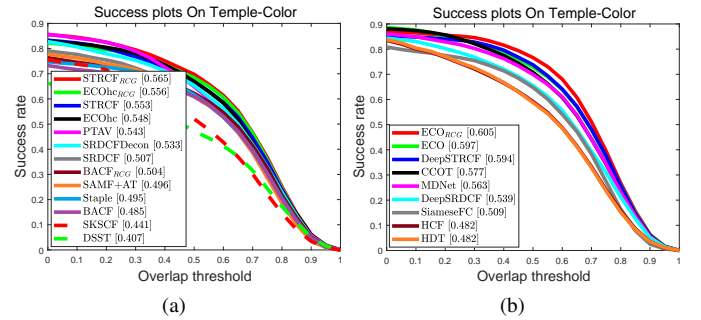

Fig. 8. Comparison of overlap success plots with the state-of-the-art trackers on the Temple-Color dataset: (a) trackers using handcrafted features, and (b) trackers using deep CNN features.

methods with AUC score gains of 1.9%, 0.8% and 1.2%. Moreover, as shown in Fig. 8(b), ECO$_{RCG}$ also performs better than ECO by 0.8% when using deep CNN features, further demonstrating the effectiveness of removing cosine window from CF trackers with spatial regularization.

G. LaSOT dataset

LaSOT is a recent large-scale tracking benchmark consisting of 1400 video sequences, which are further divided into training and test subsets. To assess the proposed methods, we compare them (i.e., BACF$_{RCG}$, STRCF$_{RCG}$, ECOhc$_{RCG}$, ECO$_{RCG}$ and UPDT$_{RCG}$) with 10 recent representative trackers on the test subset of LaSOT (280 videos), including
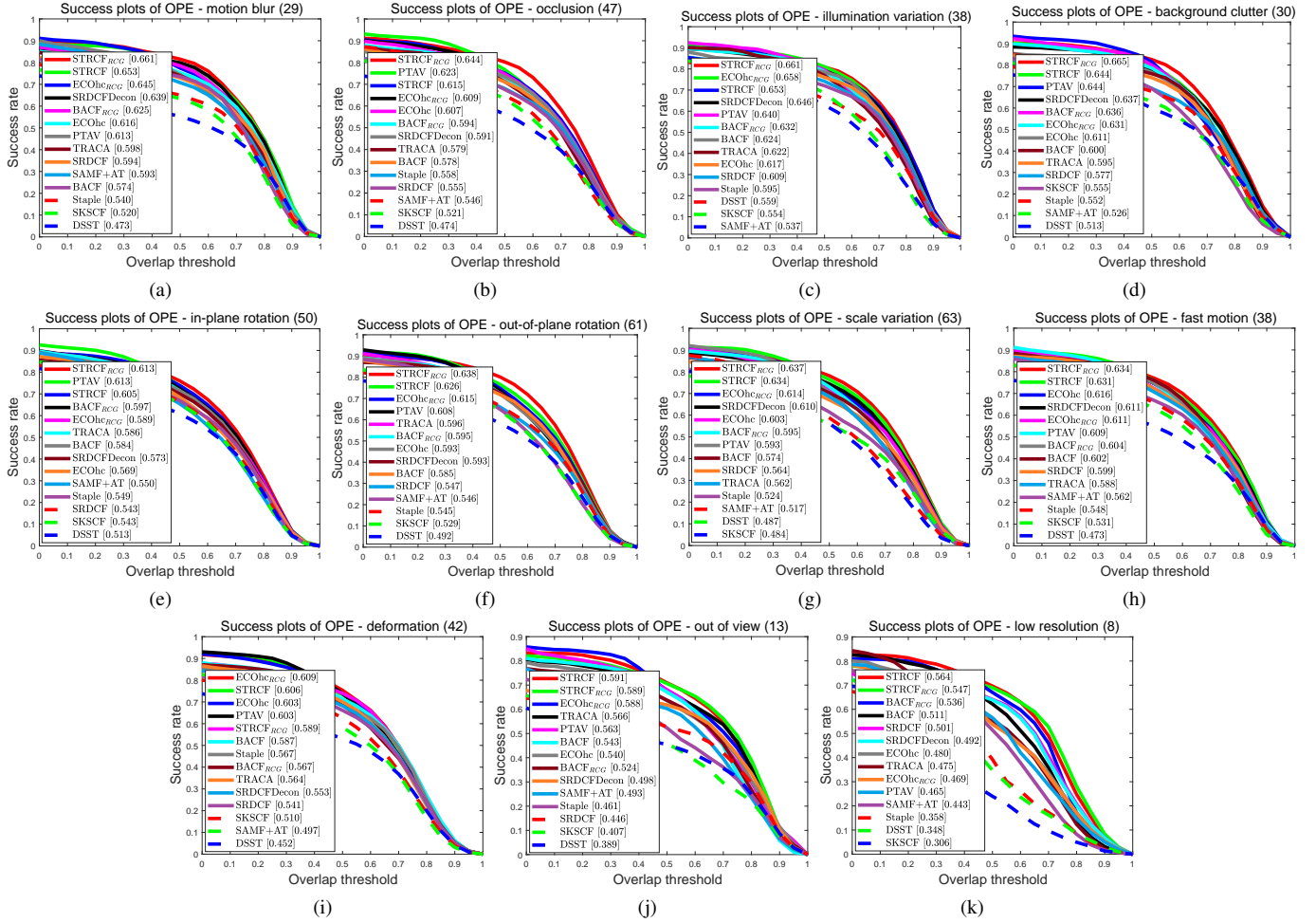
Fig. 5. Overlap success plots of the competing trackers using handcrafted features under all eleven attributes on the OTB-2015 dataset. Our methods achieve the best performance on most attributes.
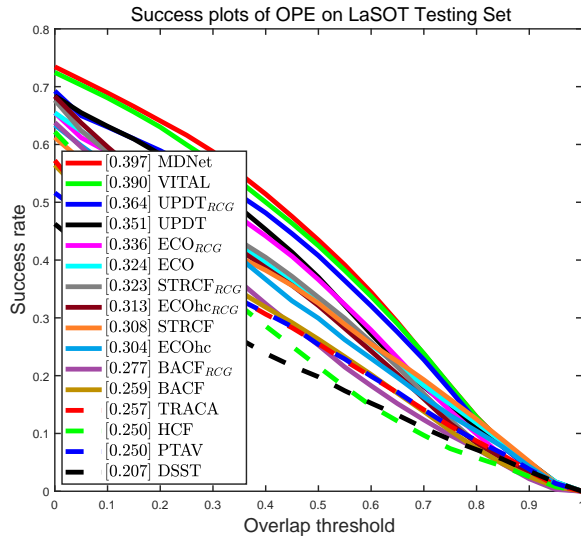

Fig. 9. Comparison of the overlap success plot with the state-of-the-art trackers on the test subset of LaSOT dataset.

MDNet [53], VITAL [50], UPDT [18], ECO [12], STR-CF [17], BACF [16], TRACA [45], HCF [13], PTAV [46] and DSST [4].

Following the protocols in [54], we report the success overlap plot for different trackers in Fig. 9. One can observe that our UPDAT$_{RCG}$ is among the top three best-performed trackers and outperforms its baseline method UPDT with a gain of 1.3% on AUC score. In addition, the proposed **RCG** variants achieve AUC gains of 1.8%, 1.5%, 0.9% and 1.2% for BACF, STRCF, ECOhc and ECO trackers, respectively. These results further validate the superiority of the proposed methods on large tracking benchmarks.

VI. Conclusion

In this paper, we investigated the problems of when and how to remove cosine window from CF trackers. Our empirical analyses showed that both spatial regularization and cosine window can be utilized to alleviate boundary discontinuity. However, cosine window may give rise to sample contamination, while for spatial regularization a small percentage of negative samples still suffer from boundary discontinuity. To remove cosine window from CF trackers with spatial regularization, we introduced a binary mask function to exclude the
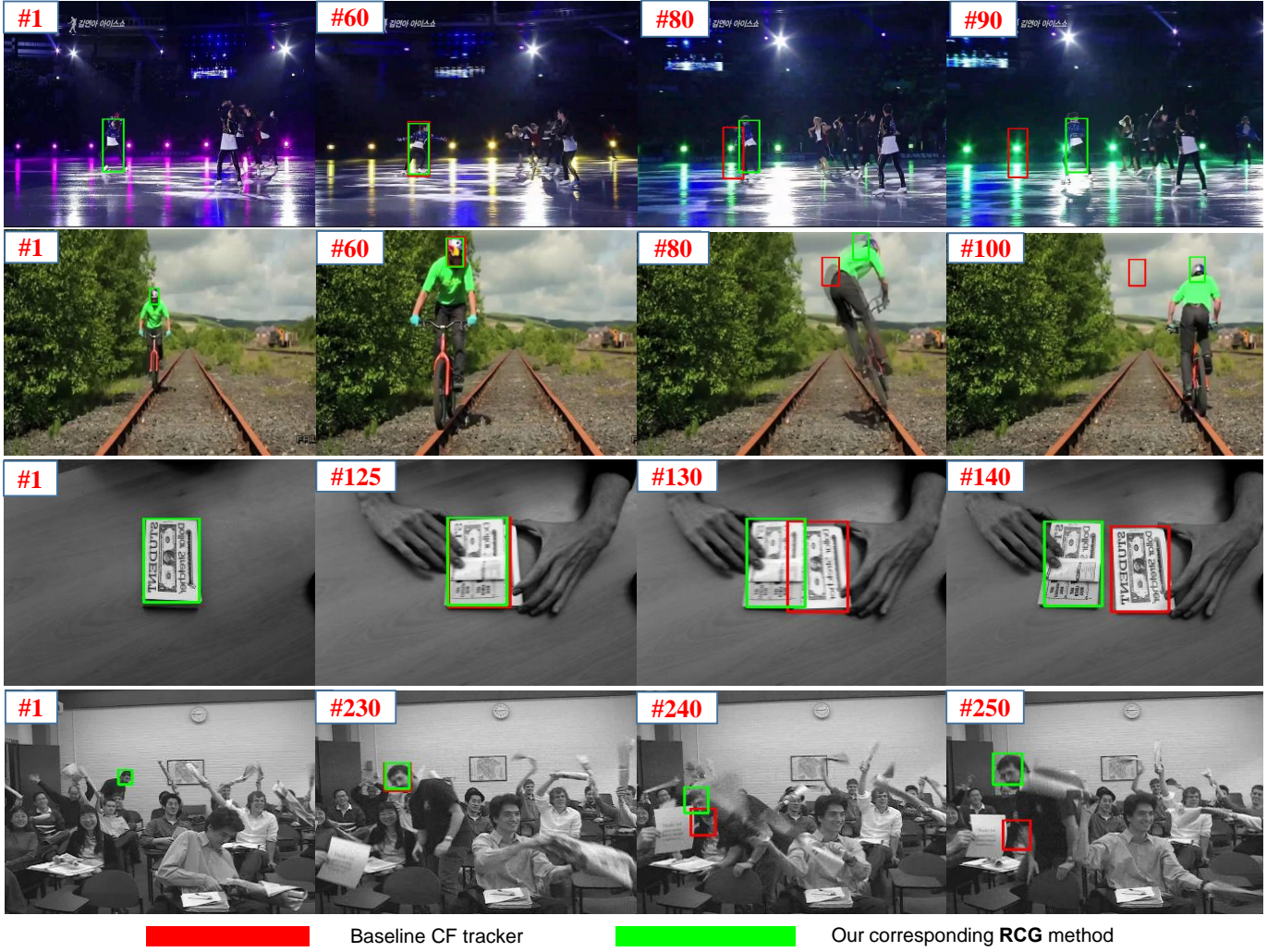
Fig. 7. Visualization of tracking results on four videos for comparing four baseline CF trackers with our corresponding **RCG** methods by removing cosine window and incorporating Gaussian shaped mask function. The videos from top to bottom are *skating1*, *biker*, *coupon* and *freeman4*. In each row, a baseline CF tracker and our corresponding **RCG** method are applied (from top to bottom: BACF [16], ECOhc [12], STRCF [17] and ECO [12]).

negative samples suffering from boundary discontinuity during training. Furthermore, another Gaussian shaped mask function was also introduced to downweight the negative samples far from target center. Then, optimization algorithms were respectively developed for removing cosine window from CF trackers with single and multiple base images. The experiments on OTB-2015, Temple-Color, VOT-2018 and LaSOT datasets showed that our methods are effective in circumventing boundary discontinuity and sample contamination, and bring moderate performance gains over their CF baseline methods with cosine window. Our methods also perform favorably against the state-of-the-art trackers using handcrafted and deep CNN features.

REFERENCES

[1] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2544–2550. 1, 3, 9

[2] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, March 2015. 1, 3, 4, 9

[3] M. Tang and J. Feng, "Multi-kernel correlation filter for visual tracking," in *IEEE International Conference on Computer Vision*, 2015, pp. 3038–3046. 1

[4] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Discriminative scale space tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1561–1575, Aug 2017. 1, 3, 9, 10, 12

[5] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *European Conference on Computer Vision Workshop*, 2014, pp. 254–265. 1, 3

[6] F. Li, Y. Yao, P. Li, D. Zhang, W. Zuo, and M. H. Yang, "Integrating boundary and center correlation filters for visual tracking with aspect ratio variation," in *IEEE International Conference on Computer Vision Workshop*, 2017, pp. 2001–2009. 1, 3

[7] W. Zuo, X. Wu, L. Lin, L. Zhang, and M.-H. Yang, "Learning support correlation filters for visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 5, pp. 1158–1172, May 2019. 1, 9, 10

[8] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *IEEE International Conference on Computer Vision*, 2015, pp. 4310–4318. 1, 3, 6, 9, 10

[9] H. K. Galoogahi, T. Sim, and S. Lucey, "Correlation filters with limited boundaries," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4630–4638. 1, 3

[10] A. Lukezic, T. Vojír, L. Cehovin, J. Matas, and M. Kristan, "Discriminative correlation filter with channel and spatial reliability," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp.

4847–4856. 1, 5

[11] M. Danelljan, A. Robinson, F. Shahbaz Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *European Conference on Computer Vision*, 2016, pp. 472–488. 1, 3, 6, 9, 10

[12] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "Eco: Efficient convolution operators for tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6931–6939. 1, 2, 3, 4, 6, 7, 9, 10, 12, 13

[13] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *IEEE International Conference on Computer Vision*, 2015, pp. 3074–3082. 1, 3, 9, 10, 12

[14] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M. H. Yang, "Hedged deep tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4303–4311. 1, 3, 9, 10

[15] S. Gladh, M. Danelljan, F. S. Khan, and M. Felsberg, "Deep motion features for visual tracking," in *International Conference on Pattern Recognition*, 2016, pp. 1243–1248. 1, 3

[16] H. Kiani Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *IEEE International Conference on Computer Vision*, 2017, pp. 1144–1152. 1, 2, 4, 5, 6, 7, 9, 10, 12, 13

[17] F. Li, C. Tian, W. Zuo, L. Zhang, and M.-H. Yang, "Learning spatial-temporal regularized correlation filters for visual tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4904–4913. 1, 2, 3, 4, 5, 7, 9, 10, 12, 13

[18] G. Bhat, J. Johnander, M. Danelljan, F. S. Khan, and M. Felsberg, "Unveiling the power of deep tracking," in *European Conference on Computer Vision*, 2018, pp. 493–509. 1, 2, 3, 4, 6, 7, 9, 10, 12

[19] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, Sep 2015. 3, 7, 9

[20] P. Liang, E. Blasch, and H. Ling, "Encoding color information for visual tracking: Algorithms and benchmark," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5630–5644, Dec 2015. 3, 7

[21] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pfugfelder, L. Zajc, T. Vojir, G. Bhat, A. Lukezic, A. Eldesokey *et al.*, "The sixth visual object tracking vot2018 challenge results," in *European Conference on Computer Vision Workshop*, 2018, pp. 3–53. 3, 7, 9, 10

[22] T. Xu, Z.-H. Feng, X.-J. Wu, and J. Kittler, "Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual tracking," *arXiv preprint arXiv:1807.11348*, 2018. 3, 10

[23] C. Ma, X. Yang, C. Zhang, and M. H. Yang, "Long-term correlation tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5388–5396. 3

[24] T. Liu, G. Wang, Q. Yang, and L. Wang, "Part-based tracking via discriminative correlation filters," *IEEE Transactions on Circuits and Systems for Video Technology*, 2018. 3

[25] T. Zhang, S. Liu, C. Xu, B. Liu, and M. Yang, "Correlation particle filter for visual tracking," *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 2676–2687, June 2018. 3

[26] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886–893. 3

[27] M. Danelljan, F. S. Khan, M. Felsberg, and J. V. D. Weijer, "Adaptive color attributes for real-time visual tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1090–1097. 3

[28] J. Fan, H. Song, K. Zhang, Q. Liu, and W. Lian, "Complementary tracking via dual color clustering and spatio-temporal regularized correlation learning," *IEEE Access*, vol. 6, pp. 56 526–56 538, 2018. 3

[29] H. Song, Y. Zheng, and K. Zhang, "Robust visual tracking via self-similarity learning," *Electronics Letters*, vol. 53, no. 1, pp. 20–22, 2016. 3

[30] K. Yang, H. Song, K. Zhang, and Q. Liu, "Hierarchical attentive siamese network for real-time visual tracking," *Neural Computing and Applications*, pp. 1–12, 2019. 3

[31] K. Yang, H. Song, K. Zhang, and J. Fan, "Deeper siamese network with multi-level feature fusion for real-time visual tracking," *Electronics Letters*, 2019. 3

[32] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1430–1438. 3, 9, 10

[33] H. K. Galoogahi, T. Sim, and S. Lucey, "Multi-channel correlation filters," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4630–4638. 5

[34] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004. 5

[35] J. Eckstein and D. P. Bertsekas, "On the douglasrachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, pp. 293–318, 1992. 5

[36] K. B. Petersen, M. S. Pedersen *et al.*, "The matrix cookbook," *Technical University of Denmark*, 2008. 6

[37] A. Vedaldi and K. Lenc, "Matconvnet: Convolutional neural networks for matlab," in *ACM international conference on Multimedia*, 2015, pp. 689–692. 7

[38] L. ehovin, A. Leonardis, and M. Kristan, "Visual object tracking performance measures revisited," *IEEE Transactions on Image Processing*, vol. 25, no. 3, pp. 1261–1274, March 2016. 7

[39] L. Yang, R. Liu, D. Zhang, and L. Zhang, "Deep location-specific tracking," in *ACM international conference on Multimedia*, 2017, pp. 1309–1317. 10

[40] A. He, C. Luo, X. Tian, and W. Zeng, "A twofold siamese network for real-time object tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4834–4843. 10

[41] C. Sun, D. Wang, H. Lu, and M.-H. Yang, "Correlation tracking via joint discrimination and reliability learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 489–497. 10

[42] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8971–8980. 9, 10

[43] A. Bibi, M. Mueller, and B. Ghanem, "Target response adaptation for correlation filter tracking," in *European Conference on Computer Vision*, 2016, pp. 419–433. 9, 10

[44] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. Torr, "Staple: Complementary learners for real-time tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1401–1409. 9, 10

[45] J. Choi, H. J. Chang, T. Fischer, S. Yun, K. Lee, J. Jeong, Y. Demiris, and J. Y. Choi, "Context-aware deep feature compression for high-speed visual tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 479–488. 9, 10, 12

[46] H. Fan and H. Ling, "Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking," in *IEEE International Conference on Computer Vision*, 2017, pp. 5486–5494. 9, 10, 12

[47] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *International Conference on Machine Learning*, 2015. 9, 10

[48] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in *IEEE International Conference on Computer Vision*, 2015, pp. 3119–3127. 9, 10

[49] J. Valmadre, L. Bertinetto, J. F. Henriques, A. Vedaldi, and P. H. Torr, "End-to-end representation learning for correlation filter based tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5000–5008. 9, 10

[50] Y. Song, C. Ma, X. Wu, L. Gong, L. Bao, W. Zuo, C. Shen, R. W. Lau, and M.-H. Yang, "Vital: Visual tracking via adversarial learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8990–8999. 9, 10, 12

[51] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *IEEE International Conference on Computer Vision Workshop*, 2015, pp. 621–629. 9, 10

[52] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *European Conference on Computer Vision Workshop*, 2016, pp. 850–865. 9, 10

[53] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4293–4302. 9, 10, 12

[54] H. Fan, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, H. Bai, Y. Xu, C. Liao, and H. Ling, "Lasot: A high-quality benchmark for large-scale single object tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5374–5383. 12