Learning Padless Correlation Filters for Boundary-Effect Free Tracking

Dongdong Li, Gongjian Wen, *Member, IEEE*, Yangliu Kuai, and Fatih Porikli, *Fellow, IEEE*

***Abstract*— Recently, discriminative correlation filters (DCFs) have achieved enormous popularity in the tracking community due to high accuracy and beyond real-time speed. Among different DCF variants, spatially regularized discriminative correlation filters (SRDCFs) demonstrate excellent performance in suppressing boundary effects induced from circularly shifted training samples. However, SRDCF have two drawbacks which may be the bottlenecks for further performance improvement. First, SRDCF needs to construct an element-wise regularization weight map which can lead to poor tracking performance without careful tunning. Second, SRDCF does not guarantee zero correlation filter values outside the target bounding box. These small but nonzero filter values away from the filter center hardly contribute to target location but induce boundary effects. To tackle these drawbacks, we revisit the standard SRDCF formulation and introduce padless correlation filters (PCFs) which totally remove boundary effects. Compared with SRDCF that penalizes filter values with spatial regularization weights, PCF directly guarantee zero filter values outside the target bounding box with a binary mask. Experimental results on the OTB2013, OTB2015 and VOT2016 data sets demonstrate that PCF achieves real-time frame-rates and favorable tracking performance compared with state-of-the-art trackers.**

***Index Terms*— Visual tracking, correlation filter, boundary effect, model complexity.**

I. Introduction

VISUAL tracking is a classical computer vision problem with many applications in multimedia such as video surveillance, augmented reality and human-computer interaction [1]–[3]. Generic tracking means single-camera, single-object, short-term and model-free tracking [4], [5], which estimates the trajectory of a target in the whole video given only its initial state (usually an axis-aligned rectangle) in the first frame. *Short-term* implies re-detection modules are unnecessary while *model-free* means neither pre-learned object models nor class-specific prior knowledge are permitted. Despite significant progress in recent years, robust tracking under complicated scenarios is still challenging due to illumination change, self-deformation, partial occlusion, fast motion and background clutter.

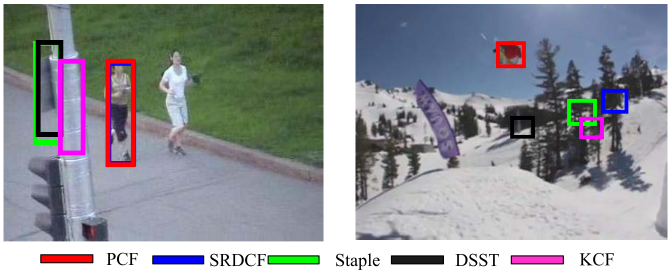

Fig. 1. A brief comparison of our approach with SRDCF [6], Staple [7], DSST [8] and KCF [9] on the *Jogging* and *Skiing* sequences from OTB2015 [4]. Our tracker demonstrates excellent robustness in presence of heavy occlusion in *Jogging* and fast motion in *Skiing*.

Most trackers follow the tracking-by-detection framework to locate the target frame by frame. Generally, tracking-by-detection based trackers can be divided into generative methods [10]–[12] and discriminative methods [9], [13], [14] according to different appearance representation schemes. Generative methods focus on representing target appearance and ignore background information, which leads to drift in complex scenarios. On contrast, discriminative methods pose single object tracking as a binary classification task to discriminate the object from its surrounding background. In recent years, Discriminative Correlation Filters (DCF) based discriminative trackers have achieved enormous popularity due to high computational efficiency and fair robustness. With the circular structure, DCF transform computationally consuming spatial correlation into efficient element-wise operation in the Fourier domain and achieve extremely high tracking speed. Based on standard DCF formulation, different variants of correlation filters have been proposed to boost tracking performance using multi-dimensional features [15], robust scale estimation [8], non-linear kernels [9], long-term memory components [16], complementary cues [7] and target adaptation [17].

However, standard DCF based trackers significantly suffer from boundary effect induced by the periodic assumption. Due to the circularity, correlation filters are trained with wrapped-around circularly shifted versions of the target. As a result, the detection scores are only accurate near the center of the search area, which leads to a very restricted target search area at the detection step. Therefore, in presence of fast motion and heavy occlusion, standard DCF based trackers are easily to drift to the background as show in Fig.1. To tackle this

Manuscript received June 30, 2018; accepted July 24, 2018. Date of publication August 1, 2018; date of current version August 22, 2018. This work was supported by the National Natural Science Foundation of China (NSFC) under Project 41601487. The associate editor coordinating the review of this paper and approving it for publication was Prof. Huang Chen Lee. *(Corresponding author: Dongdong Li.)*

D. Li, G. Wen, and Y. Kuai are with the College of Electronic Science, National University of Defense Technology, Changsha 410073, China (e-mail: moqimubai@sina.cn; wengongjian@sina.com).

F. Porikli is with the Research School of Engineering, The Australian National University, Canberra, ACT 2601, Australia.

Digital Object Identifier 10.1109/JSEN.2018.2861912

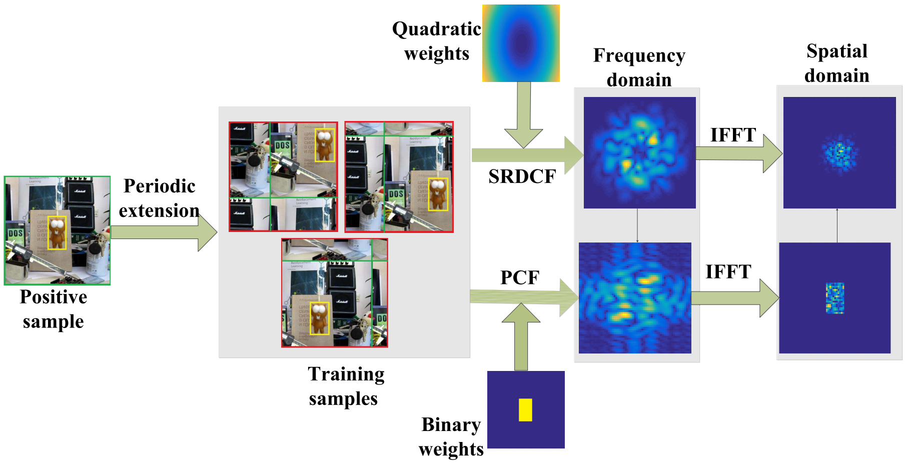

Fig. 2. Comparison of the tracking frameworks in SRDCF and PCF. SRDCF employ the quadratic regularization weights and derives a correlation filter with large spatial support. Nonzero values are assigned to the background region. The influence of the background information in the detection stage leads to drift in challenging tracking scenarios. On contrast, PCF use the binary weights and derives small correlation filters with zero padded in the neighborhood. The nonzero filter values exist only in the target bounding box, which increases the discriminative power of the correlation filter by emphasizing the appearance information in the target region. Moreover, PCF maintain less trainable filter values in the model and reduce the risk of over-fitting.

problem, Spatially Regularized Discriminate Correlation Filters (SRDCF) [6] introduce a spatial regularization component into the standard DCF formulation. Compared with standard DCF, SRDCF have two following advantages. First, SRDCF pad the target image patch with large background padding to guarantee a large search area. Second, the spatial regularization component penalizes filter values away from the target center with large spatial weights to suppress the boundary effects. With the spatial regularization component, the descendant variants of SRDCF, DeepSRDCF [18] and CCOT [19], have won the VOT2015 [20] and VOT2016 [21] Challenges.

Despite the above achievements, we argue that SRDCF has the following drawbacks which would be the bottlenecks for further performance improvement. First, the regularization weights in SRDCF are highly video and dataset dependent, which if not performed correctly can lead to poor tracking performance. Second, the learned correlation filter is a trade-off between the desired correlation response and spatial regularization, and thus SRDCF cannot guarantee the filter values are zero outside of object bounding box. Third, these small but nonzero background filter values hardly contribute to target location but lead to boundary effects and massive trainable parameters, which increase the model complexity and risk of over-fitting.

In this paper, we revisit the core SRDCF formulation and develop Padless Correlation Filter (PCF) for boundary-effect free tracking. Different from SRDCF which penalize background filter coefficients with regularization weights, PCF directly removes background coefficients with a binary mask (see Fig.2). Meanwhile, PCF achieve better generalization capacity with only one regularization parameter to set while SRDCF need to carefully tune a element-wise regularization weight map. We perform comprehensive experiments on three benchmark datasets: OTB2013 [4] with 50 sequences, OTB2015 [5] with 100 videos and VOT2016 [21] with 60 videos. With less trainable parameters and boundary effects, PCF achieves an absolute gain of 3.8% in AUC on OTB2015 and a relative gain of 9.0% in EAO on VOT2016 compared with the baseline SRDCF.

II. Related Works

There are extensive surveys on visual tracking in the literature. We refer interested readers to [5] and [21] for a thorough review of existing tracking algorithms. Due to space limitations, here we focus on correlation filter based trackers.

A. Discriminative Correlation Filters

Discriminative Correlation Filters (DCF) are initially developed for the object detection task [22] and are introduced into visual tracking until recent years. The pioneer work was done by Bolme *et al.* [23] who introduced Minimum Output Sum of Squared Error (MOSSE) filters and achieved a tracking speed of hundred of frames per second (fps). Later, correlation filters have been extended to multi-dimensional features such as HOG [24] or Color-Names [25] and achieve a notable performance improvement. Henriques *et al.* [9] formulated learning correlation filters as a ridge regression problem and introduced the kernel trick into correlation filters. To achieve fast scale estimation, a discriminative scale space tracker (DSST) [26] is proposed to achieve real-time scale adaptive tracking. Staple [7] combines the correlation filter response with the global color histogram score to achieved surprisingly robustness against object deformation. LCT [16] combines DCF with an online trained random fern classifier for re-detection to achieve long-term robust tracking.

B. Variants For Conventional DCF

Recent works have found that some of the inherent limitations of DCF can be overcome directly by modifying the conventional DCF formulation used for training. For example, by adapting the target response (used for ridge regression in DCF) as part of a new formulation, Bibi *et al.* [17] significantly decrease target drift while remaining computationally efficient. Liu *et al.* [27] introduce part-based tracking strategy into DCF to reduce sensitivity to partial occlusion and better preserve object structure. To improve the discriminative capacity of DCF, Mueller *et al.* [28] introduce background context into the DCF tracking framework while maintaining high frame-rates.

C. Reducing Boundary Effects for Correlation Filters

In recent years, numerous tracking benchmarks [4], [29], [30] and tracking challenges [20], [21] have seen continuous performance improvements of visual tracking. However, traditional DCF based trackers seldom demonstrate to be competitive in these benchmarks or challenges. We argue that this inferior performance is induced by boundary effects originated from the periodic assumption. Due to boundary effects, the detection scores of DCF are only accurate near the target center, which leads to a restricted search area. To suppress boundary effects and expand the search area, SRDCF [6] learn a correlation filter with large spatial support and thus maintain a larger search area in the detection stage. Filter values outside the object bounding box are penalized with higher regularization weights to highlight the central area of the correlation filter. Afterwards, the descendant variant of SRDCF, CCOT [19], employs the integration of multi-resolution features in the continuous domain and achieves the top rank on the VOT2016 challenge [21]. Based on CCOT, ECOT [31] improves the tracking speed and robustness by performing feature dimensionality reduction with a factorize convolution operator and reducing training samples in the learning model.

III. Our Approach

In this section, we adopt SRDCF as our baseline and present a theoretical framework for learning padless correlation filters. Our formulation is generic and can be extended to CCOT [19] and ECOT [31] for further performance gain.

A. Baseline SRDCF

Before the detailed discussion of our proposed padless correlation filters, we first revisit the detailed derivation of the SRDCF formulation. For convenience reasons, we adopt the same notation as in [6]. In the SRDCF formulation, the aim is to learn a multi-channel spatially regularized convolution filter h from a set of training examples $\{(x_k, y_k)\}_{k=1}^{t}$. Each training sample x_k is assumed to have the spatial size $M \times N$ and consists of a d-dimensional feature extracted from an image region. We denote feature layer $l \in \{1, \cdots, d\}$ of x_k by x_k^l. The desired output of y_k are scalar values over the domain $\boldsymbol{\Omega}$, which include a label for each location in the sample x_k. The desired correlation filter h is obtained by minimizing the following target function,

$$\varepsilon(h) = \sum_{k=1}^{t} \alpha_k \left\| \sum_{l=1}^{d} x_k^l * h^l - y_k \right\|^2 + \sum_{l=1}^{d} \left\| w \cdot h^l \right\|^2. \quad (1)$$

Here, $\cdot$ denotes point-wise multiplication, * denotes circular convolution and the weights $\alpha_k \geq 0$ determine the impact of each training sample. The regularization weights w get lower values near the target center and higher values away from the target center. In this way, correlation filter values away from the target center are penalized to small values.

By applying Parseval's theorem to (1), the correlation filter h can equivalently be obtained by minimizing the resulting loss function over the DFT coefficients $\hat{h}$,

$$\varepsilon(\hat{h}) = \sum_{k=1}^{t} \alpha_k \left\| \sum_{l=1}^{d} \hat{x}_k^l \cdot \hat{h}^l - \hat{y}_k \right\|^2 + \lambda \sum_{l=1}^{d} \left\| \hat{w} * \hat{h}^l \right\|^2. \quad (2)$$

In (2), $\hat{w} * \hat{h}^l$ follows the convolution property of the inverse Fourier transform. Therefore, a vectorization of (2) gives,

$$\varepsilon(\hat{h}) = \sum_{k=1}^{t} \alpha_k \left\| \sum_{l=1}^{d} \mathcal{D}(\hat{\mathbf{x}}_k^l) \hat{\mathbf{h}}^l - \hat{\mathbf{y}}_k \right\|^2 + \lambda \sum_{l=1}^{d} \left\| \mathcal{C}(\hat{\mathbf{w}}) \hat{\mathbf{h}}^l \right\|^2. \quad (3)$$

Here, bold letters denote a vectorization of the corresponding scalar valued functions and $\mathcal{D}(\mathbf{v})$ denotes the diagonal matrix with the elements of the vector $\mathbf{v}$ in its diagonal. The $MN \times MN$ matrix $\mathcal{C}(\hat{\mathbf{w}})$ represents circular 2D convolution with the weights $\hat{w}$, i.e. $\mathcal{C}(\hat{\mathbf{w}})\hat{\mathbf{h}}^l = vec(\hat{w} * \hat{h}^l)$. Each row in $\mathcal{C}(\hat{\mathbf{w}})$ thus contains a cyclic permutation of $\hat{\mathbf{w}}$.

The loss function in (3) is simplified by defining the fully vectorized filter as the concatenation $\hat{\mathbf{h}} = ((\hat{\mathbf{h}^1})^T \dots (\hat{\mathbf{h}^d})^T)^T$ where T represents the transpose of a vector,

$$\varepsilon(\hat{\mathbf{h}}) = \sum_{k=1}^{t} \alpha_k \left\| D_k \hat{\mathbf{h}} - \hat{\mathbf{y}}_k \right\|^2 + \lambda \left\| W \hat{\mathbf{h}} \right\|^2. \quad (4)$$

Here we have defined the concatenation $D_k = (\mathcal{D}(\hat{\mathbf{x}}_k^1) \dots \mathcal{D}(\hat{\mathbf{x}}_k^d))$ and W to be the $dMN \times dMN$ block diagonal matrix with each diagonal block being equal to $\mathcal{C}(\hat{\mathbf{w}})$.

To obtain a simple expression of (4), we define the sample matrix $D = [D_1^T \dots D_t^T]^T$ the diagonal weight matrix, $\Gamma = \alpha_1 I \oplus \dots \oplus \alpha_t I$ and the label vector $\hat{\mathbf{y}} = [\hat{\mathbf{y}}_1^T \dots \hat{\mathbf{y}}_t^T]^T$. The minimizer of (4) is found by solving the following normal equations,

$$(D^H \Gamma D + \lambda W^H W)\hat{\mathbf{h}} = D^H \Gamma \hat{\mathbf{y}}. \quad (5)$$

It's worth noting that the $dMN \times dMN$ coefficient matrix $D^H \Gamma D + \lambda W^H W$ is a symmetric positive-semidefinite matrix. There, SRDCF and its descendant variants CCOT, ECOT employ the Preconditioned Conjugate Gradient (PCG) method [32] to iteratively solve (5), since PCG was shown to effectively utilize the sparsity structure of the problem.

B. Padless Correlation Filters

As shown in (1), the learned correlation filter is a compromise between the desired correlation response and spatial regularization, and thus SRDCF can't guarantee the filter values are zero outside of object bounding box. To remove the nonzero filter values outside of object bounding box, we introduce a cropping operator b into the SRDCF formulation to substitute the spatial weights w. The cropping operator b is a binary mask with zero values outside the target bounding

Algorithm 1: Each Cycle of Optimization Using the Preconditioned Conjugate Gradient Method

```
Initialize $\hat{\mathbf{f}}$ and the preconditioner.
Repeat
    Apply inverse FFT to $\hat{\mathbf{f}}$ and crop: $B\hat{\mathbf{f}} = \mathbf{vec}(\mathscr{F}(b \cdot \mathscr{F}^{-1}(\hat{f})))$
    Perform element-wise multiplication in the frequency domain: $D^T \Gamma D B\hat{\mathbf{f}}$ and $D^T \Gamma \hat{\mathbf{y}}$
    Apply inverse FFT then crop: $B^T D^T \Gamma D B\hat{\mathbf{f}}$ and $B^T D^T \Gamma \hat{\mathbf{y}}$
    Compute the residual vector: $\mathbf{r} = B^T D^T \Gamma \hat{\mathbf{y}} - B^T D^T \Gamma D B\hat{\mathbf{f}} - \lambda\hat{\mathbf{f}}$
    Compute the search direction $\mathbf{p}$ and search scale $\mathbf{s}$ from $\mathbf{r}$ with the PCG method.
    Update $\hat{\mathbf{f}} = \hat{\mathbf{f}} + \mathbf{sp}$.
Until $\hat{\mathbf{f}}$ has converged or the maximum number of iterations has reached.
```

box (see Fig.2). Compared with SRDCF which penalize the correlation filter values with w in the regularization term, our padless correlation filters impose b directly in the target response regression term as .

$$\varepsilon(f) = \sum_{k=1}^{t} \alpha_k \left\| \sum_{l=1}^{d} x_k^l * (b \cdot f^l) - y_k \right\|^2 + \lambda \sum_{l=1}^{d} \left\| f^l \right\|^2. \tag{6}$$

In (6), the first term is responsible for target response regression while the second term is a regularization term to avoid over-fitting. The correlation filter values in f outside the target bounding box are removed by the cropping operator b. Therefore, in this paper, we term our approach as Padless Correlation Filters (PCF), which means the filter values in the padding area of f are all set to zeros and do not contribute to the target response. Compared with SRDCF, PCF only maintain non-zero filter values inside the target bounding box and thus reduce the number of trainable coefficients in the tracking model. In this way, the computational burden and risk of over-fitting are significantly reduced.

1) PCF Training: Similar to (1), (6) can be transformed into the Fourier domain as following:

$$\varepsilon(\hat{f}) = \sum_{k=1}^{t} \alpha_k \left\| \sum_{l=1}^{d} \hat{x}_k^l \cdot (\hat{b} * \hat{f}^l) - \hat{y}_k \right\|^2 + \lambda \sum_{l=1}^{d} \left\| \hat{f}^l \right\|^2. \tag{7}$$

In (7), $\hat{b} * \hat{f}^l$ follows the convolution property of the inverse Fourier transform. Therefore, a vectorization of (7) gives,

$$\varepsilon(\hat{f}) = \sum_{k=1}^{t} \alpha_k \left\| \sum_{l=1}^{d} \mathcal{D}(\hat{\mathbf{x}}_k^l) \mathcal{C}(\hat{\mathbf{b}}) \hat{\mathbf{f}}^l - \hat{\mathbf{y}}_k \right\|^2 + \lambda \sum_{l=1}^{d} \left\| \hat{\mathbf{f}}^l \right\|^2. \tag{8}$$

Here, the $MN \times MN$ matrix $\mathcal{C}(\hat{\mathbf{b}})$ represents circular 2D convolution with the function $\hat{b}$, i.e. $\mathcal{C}(\hat{\mathbf{b}})\hat{\mathbf{f}}^l = vec(\hat{b} * \hat{f}^l) = vec(\mathscr{F}(b \cdot f))$. Each row in $\mathcal{C}(\hat{\mathbf{b}})$ thus contains a cyclic permutation of $\hat{\mathbf{b}}$.

Similar to (3), the loss function in (8) is simplified by defining the fully vectorized filter as the concatenation $\hat{\mathbf{f}} = ((\hat{\mathbf{f}}^1)^T \ldots (\hat{\mathbf{f}}^d)^T)^T$,

$$\varepsilon(\hat{\mathbf{f}}) = \sum_{k=1}^{t} \alpha_k \left\| D_k B \hat{\mathbf{f}} - \hat{\mathbf{y}}_k \right\|^2 + \lambda \left\| \hat{\mathbf{f}} \right\|^2. \tag{9}$$

Here we have defined the concatenation $D_k = (\mathcal{D}(\hat{\mathbf{x}}_k^1) \ldots \mathcal{D}(\hat{\mathbf{x}}_k^d))$ and B to be the $dMN \times dMN$ block diagonal matrix with each diagonal block being equal to $\mathcal{C}(\hat{\mathbf{b}})$.

Different from (4), the minimizer of (9) is found by solving the following normal equations,

$$(B^H D^H \Gamma D B + \lambda I)\hat{\mathbf{f}} = B^H D^H \Gamma \hat{\mathbf{y}}. \tag{10}$$

It's worth noting that the $dMN \times dMN$ coefficient matrix $B^T D^T \Gamma D B + \lambda I$ is a symmetric positive-semidefinite matrix. Our task is to solve the vectorized correlation filter $\hat{\mathbf{f}}$ from the linear equation system in (10). Following SRDCF and CCOT, we employ the Preconditioned Conjugate Gradient method to iteratively solve (10).

In fact, it is not necessary to form the big $dMN \times dMN$ symmetric positive-semidefinite matrix $(B^T D^T \Gamma D B + \lambda I)$ in memory in each cycle of Conjugate Gradient Optimization. The left-hand side of the normal equation (10) is computed from right to left by performing the matrix-vector and transpose matrix-vector multiplication. B^T and B performs as the augmenting and cropping operators respectively. D is a $t \times d$ block matrix with each block as a $MN \times MN$ diagonal matrix $\mathcal{D}(\hat{\mathbf{x}}_k^1)$. Therefore, the matrix-vector multiplication related to D can be computed as efficient element-wise multiplication. Different from (2) which exhaustively computes the circular correlation of $\hat{w} * \hat{h}^l$ in the frequency domain, PCF computed the element-wise multiplication as $b \cdot f^l$ in the spatial domain. $b \cdot f^l$ can be equally computed by cropping the central area of f^l and padding it with zeros in the neighborhood.

Given the initial guess $\hat{\mathbf{f}}_0$ in each cycle of conjugate gradient optimization, the correlation filter $\hat{\mathbf{f}}$ can be learned with a few iterations in each frame according to the PCG method. A full description of the optimization procedure can be seen in Algorithm 1. It's worth to mention that we adopt the Jacobi preconditioner as $Diag(D^T \Gamma D + \lambda I)$ to ensure a small condition number of (10). Our detailed implementation for the PCG optimization can be found in the source codes available at https://github.com/moqimubai/PCF.

2) PCF Detection: At the detection stage, the location of the target in a new frame t is estimated by applying the filter $\hat{f}_{t-1}$ that has been updated in the previous frame. Let z denote the test sample extracted in the current frame and $\hat{f}$ denote the correlation filter learned in the frequency domain in the previous frame. The correlation scores $S_f(z)$ at all locations in the image region are computed as follows,

$$S_f(z) = \mathscr{F}^{-1}\left\{ \sum_{l=1}^{D} \hat{z}^l \cdot \mathscr{F}(b \cdot \mathscr{F}^{-1}(\hat{f}^l)) \right\}. \tag{11}$$

```
Algorithm 2: Visual Tracking Algorithm Based on Padless Correlation Filters
Input:
    Initial target state in the first frame $p_1 = \{x_1, y_1, w_1, h_1\}$.
Output:
    Estimated target state $p_t = \{x_t, y_t, w_t, h_t\}$ in each frame.
From t=2 to $T$, do
1: Crop the image patch $I_t$ centered at $p_{t-1}$ in frame $t$ and extracted detection features $z_t$ from $I_t$.
2: Estimate the target state $p_t$ in frame $t$ with $z_t$ according to III-B.2.
3: Crop a new image patch $I'_t$ centered at $p_t$ and extracted training features $x_t$ from $I'_t$.
4: Calculate $\hat{f}_t$ from $x_t$ with $\hat{f}_{t-1}$ as the initial guess using the Preconditioned Conjugate Gradient method as described
   in III-B.1.
```

Here, $\cdot$ denotes point-wise multiplication, $\mathscr{F}$ denotes the DFT of a function and $\mathscr{F}^{-1}$ denotes the inverse Fourier transformation.

Note that only the filter values in the target bounding box of f, namely $b \cdot \mathscr{F}^{-1}(\hat{f}^l)$, are activated, which reduces the boundary effects in the detection stage.

In terms of scale estimation, we adopt the same strategy as SRDCF by extracting multi-resolution samples at the previous target location. The scale level with the highest maximal detection score is then used to update the target location and scale. An outline of our proposed method is shown in Algorithm 2.

IV. Experiments

Here, we present a comprehensive evaluation of PCF on the OTB2013, OTB2015 and VOT2016 datasets. Readers are encouraged to read [5] and [21] for more details about each dataset.

Evaluation Methodology: OTB2013 [4] is a popular tracking benchmark dataset containing 50 fully annotated videos with substantial variations. OTB2015 is the extension of OTB2013 and contains 100 video sequences. Compared with OTB2013, more challenging sequences are added into OTB2015. On OTB2013 and OTB2015, we use the precision and success plots in one-pass evaluation (OPE) [5] to rank all the trackers. The precision plots show the percentage of frames whose estimated location is within the given threshold distance of the ground truth. The success plots show the ratios of successful frames when the threshold varies from o to 1, where a successful frame means its overlap with the ground truth is larger than this given threshold. For the VOT2016 dataset, tracking performance is evaluated in terms of both accuracy and robustness. The accuracy score is based on the overlap with ground truth, while the robustness is determined by failure rate. Different from OTB2013 and OTB2015, the trackers in VOT2016 are restarted at each failure.

Comparison Scenarios: In our experiments, we implement two versions of our tracker, namely PCF_HOG with hand-crafted features (HOG) and PCF_deep with convolutional features. On OTB2013 and OTB2015, we compare PCF_HOG and PCF_deep with state-of-the-art trackers in the literature. On VOT2016, we compare PCF_deep with DeepSRDCF [18] and the top 8 trackers on the challenge.

Implementation Details: The regularization parameter λ in (6) is set to 1e-5 in both PCF_HOG and PCF_deep. We set the search area to 4 times the target size and use 5 iterations in each cycle of conjugation gradient optimization. Parameters are fixed for all videos in each datasets. Our trackers implemented in Matlab use Piotr's Matlab Toolbox [33] for hand-crafted feature extraction and Matconvnet [34] for deep feature extraction. The deep features are extracted from the first convolutional layer in the imagenet-vgg-m-2048 model [35]. We perform the experiments on a PC with Intel i7 CPU (3.4 GHz) together with a single NVIDIA GeForce GTX Titan X GPU. A different type of CPU or GPU can lead to a slight difference in tracking performance on benchmarks. The source codes and experimental results are available at https://github.com/moqimubai/PCF.

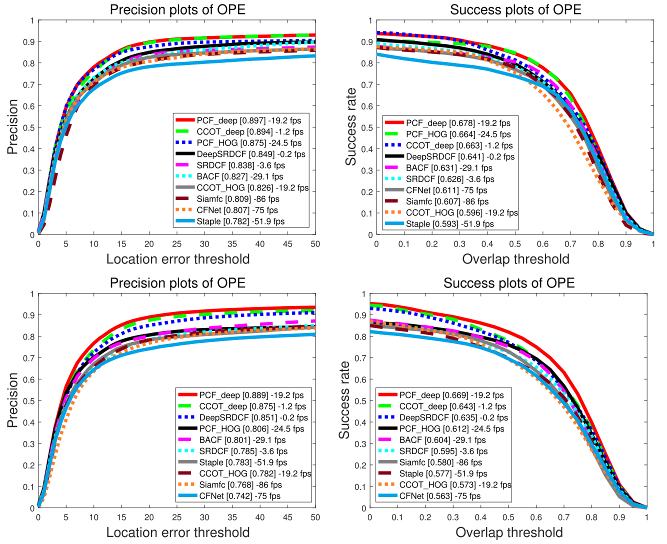

Fig. 3. Precision and success plots for all trackers on OTB2013 (first row) and OTB2015 (second row).

A. OTB2013 and OTB2015

Here, we provide a comparison of PCF_HOG and PCF_deep with state-of-the-art methods from the literature: CCOT [19], DeepSRDCF [18], BACF [36], SRDCF [6], Staple [7], CFNet [37] and Siamfc [38]. Similar to PCF, we also introduce two version of the CCOT tracker, namely CCOT_HOG and CCOT_deep. For fair comparison, all the four trackers (PCF_HOG, PCF_deep, CCOT_HOG and CCOT_deep) employ 50 training samples in filter learning,

TABLE I

QUANTITATIVE COMPARISON OF THE DISTANCE PRECISION (DP), OVERLAP PRECISION (OP) AND TRACKING SPEED (FPS) OF ALL TRACKERS ON OTB2015. THE BEST AND SECOND BEST VALUES ARE HIGHLIGHTED IN COLOR

	PCF_deep	**PCF_HOG**	CCOT_deep	CCOT_*HOG*	DeepSRDCF	SRDCF	BACF	Staple	CFNet	Siamfc
DP (%)	88.9	80.6	87.5	78.2	85.1	78.9	81.4	78.4	74.8	77.1
OP (%)	81.9	76.0	77.8	69.3	76.1	72.8	77.2	69.9	70.0	73.0
Avg. FPS	19.2	24.5	1.2	19.2	0.2	3.6	29.1	51.9	75	86

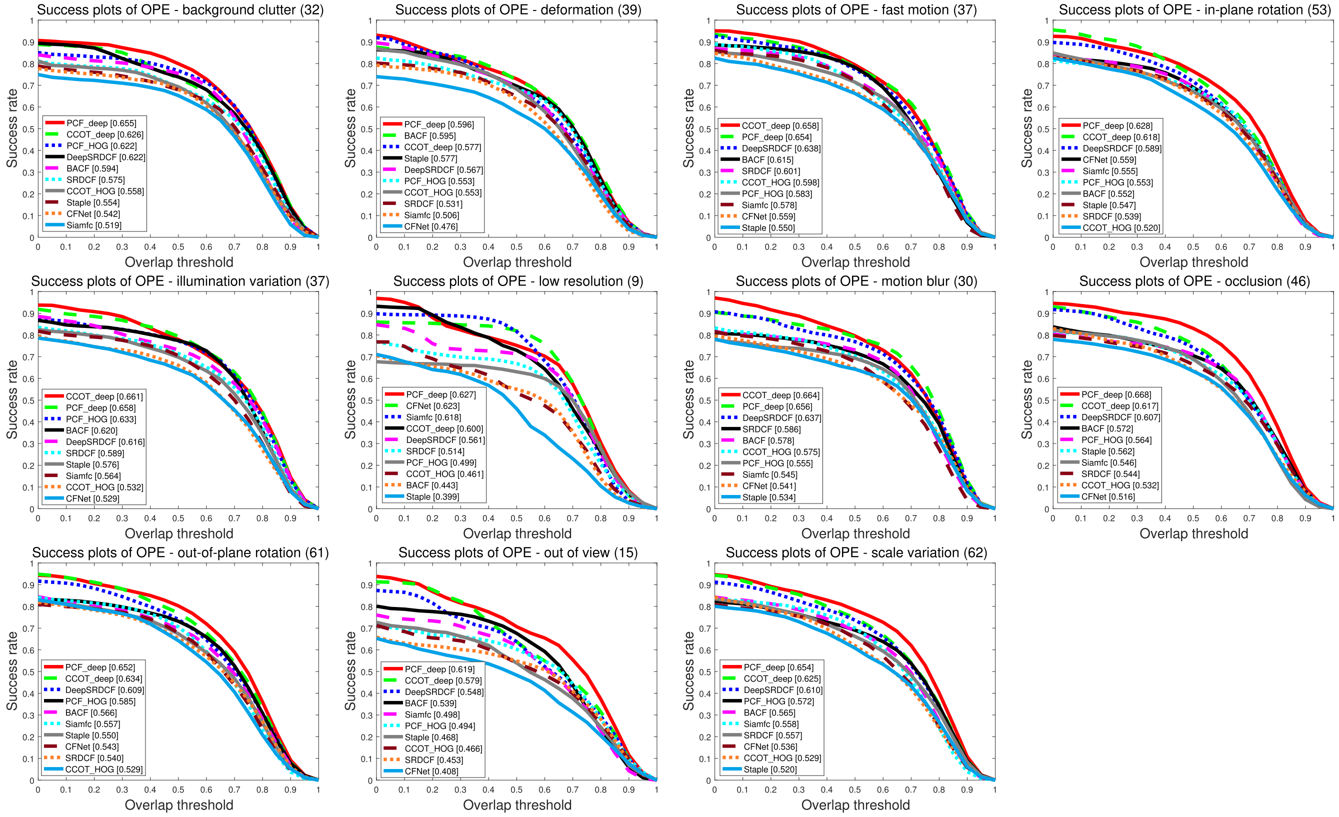

Fig. 4. *Success ratio* plots on 11 attributes of the OTB2015 dataset. Trackers are ranked by their AUC scores. Ours method has achieved consistently the superior performance over the state-of-the-art.

namely setting t to 50 in (7). All HOG-based trackers, CCOT_HOG, PCF_HOG, BACF and SRDCF, employ 31-dimensional HOG features using 4 × 4 cell size while CCOT_deep, PCF_deep and DeepSRDCF employ shallow deep Features extracted from the first convolutional layer in the VGG-m network [35]. CFNet and Siamfc are two trackers which employ the deep architecture and are trained in an end-to-end fashion.

Quantitative Comparison: Fig.3 compares PCF_deep and PCF_HOG with the other trackers on OTB2013 and OTB2015, where PCF_deep achieves the highest area-under-curve (AUC) scores in the precision and success plots over both datasets. On OTB2013, PCF_HOG with hand-crafted features even performs better than DeepSRDCF with convolutional features, which demonstrates the effectiveness of our approach in handling boundary effects and reducing model complexity.

Table I presents distance precision (DP) at 20 pixels, overlap precision (OP) at IoU = 0.5 and tracking speeds (FPS) of all compared trackers on OTB2015. All deep trackers are run with GPU. Due to the lower model complexity, PCF_HOG (24.5 fps) operates faster than CCOT_HOG (19.2 fps) and SRDCF (3.6 fps).

Attribute Based Comparison: Fig.4 illustrates the attribute based evaluation of all trackers on the OTB2015 dataset. All sequences in the OTB2015 dataset are annotated by 11 different visual attributes, namely: illumination variation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, out-of-view, background clutter and low resolution. In Fig.4, PCF_deep achieves the best performance on 8 out of 11 attributes, which demonstrates the superiority of our approach in challenging tracking scenarios. For the rest three attributes (fast motion, illumination variation and motion blur), PCF_deep still achieves higher performance than most trackers including DeepSRDCF, BACF, SRDCF, CCOT_HOG, Staple, CFNet and Siamfc. Even compared with the top tracker CCOT_deep

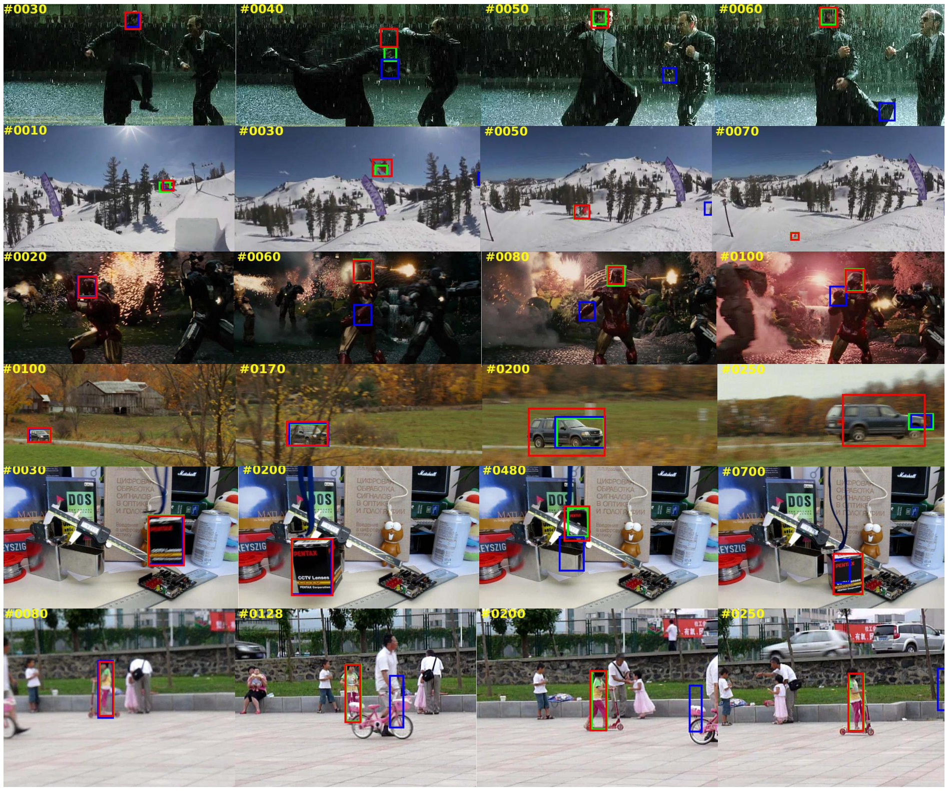

Fig. 5. Tracking screenshots of PCF_deep, CCOT_deep and DeepSRDCF. The videos (from top to bottom) are *Matrix*, *Skiing*, *Ironman*, *CarScale*, *Box* and *Girl2* from the OTB2015 dataset.

on these three attributes, our PCF_deep achieves only a slight absolute decrease of 0.4% on fast motion, 0.3% on illumination variation and 0.8% on motion blur. While our padless correlation filters improve tracking performance on most attributes, there are certain categories that benefit more than others. The most significant improvement is achieved in the case of background clutter, deformation and occlusion, which can be attributed to the excellent discriminative power and large spatial support of our padless correlation filters.

To intuitively exhibit the superiority of our proposed padless correlation filters, Fig.5 shows screenshots of the tracking results from 6 challenging videos on the OTB2015 dataset. For fair comparison, we compare PCF_deep against CCOT_deep and DeepSRDCF. All three trackers employ convolutional features. The videos (from top to bottom) are *Matrix*, *Skiing*, *Ironman*, *CarScale*, *Box* and *Girl2*. It is easy to see that PCF_deep performs better than CCOT_deep and DeepSRDCF in presence of fast motion (*Matrix*,*Skiing*), illumination variation (*Ironman*), scale variation (*CarScale*) and partial or full occlusion (*Box*,*Girl2*).

B. VOT2016

The visual object tracking (VOT) challenge is a competition between short-term, model-free visual tracking algorithms. Different from OTB2015, for each sequence in this dataset, a tracker is restarted whenever the target is lost (*i.e.* at a tracking failure). On VOT2016 dataset, four primary measures are used to analyze tracking performance: expected average overlap (EAO), robustness (R), accuracy (A) and equivalent filter operation (EFO).

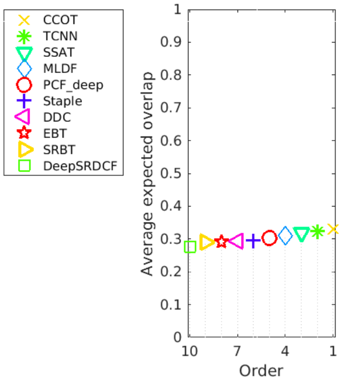

Fig. 6. Comparison of our approach with DeepSRDCF and the top 8 trackers in terms of EAO on VOT2016.

Table II and Fig.6 shows the comparison of PCF_deep with DeepSRDCF and the top 8 participants in the VOT2016 challenge. In the comparison, CCOT (0.331) achieves a higher EAO score than PCF_deep (0.303). This can be attributed to the convolutional features from multiple layers CCOT has adopted. Their feature representation performs better than ours from single layer output. It's worth noting that PCF_deep (0.303) achieves a relative gain of 9.0% in EAO on VOT2016 compared with DeepSRDCF (0.276). As indicated in the VOT2016 report [21], the strict state-of-the-art bound

TABLE II

State-of-the-Art Comparison of Expected Average Overlap (EAO), Robustness (Failure Rate), Accuracy, and Speed (in EFO Units) on VOT2016. The Best and Second Best Values Are Highlighted in Color

	DeepSRDCF	SRBT	EBT	DDC	Staple	**PCF_deep**	MLDF	SSAT	TCNN	CCOT
EAO	0.276	0.290	0.291	0.293	0.295	0.303	0.310	0.320	0.325	0.331
Failure rate	1.05	1.25	0.90	1.23	1.35	1.02	0.83	1.04	0.96	0.85
Accuracy	0.53	0.50	0.44	0.53	0.54	0.55	0.48	0.577	0.54	0.52
EFO	0.38	3.69	3.01	0.20	11.144	11.35	1.48	0.48	1.05	0.51

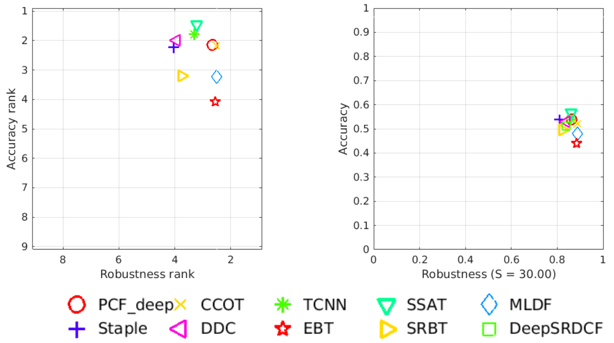

Fig. 7. State-of-the-art comparison on VOT2016. In the ranking plot (left) the accuracy and robustness rank for each tracker is displayed. The AR plot (right) shows the accuracy and robustness scores.

is 0.251 under EAO metrics. Therefore, PCF_deep (0.303) exceeds this bound and can be regarded as state-of-the-art. It's worth noting that PCF_deep ranks second in terms of accuracy and achieves the fastest tracking speed (with GPU) among all the trackers. Fig.7 shows a visualization of the overall results in terms of accuracy and robustness on the VOT2016 dataset.

V. Conclusion

We revisit the core SRDCF formulation and counter the issues of boundary effects and model complexity. We introduce an ideal binary weight function into the DCF formulation and develop Padless Correlation Filters (PCF). Accordingly, new training and detection strategies are designed for model updating and target location during tracking. Without nonzero values outside of the rectangular object area in the correlation filter, PCF remove boundary effects and reduce the number of trainable parameters in the tracking model. Our method demonstrates the competitive accuracy and superior tracking speed compared to state-of-the-art DCF-based and deep trackers over an extensive evaluation. In our future work, PCF will be extended to infrared object tracking [39] or RGBD object tracking [40] for more general applications.

References

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, pp. 1–45, 2006.

[2] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *Int. J. Comput. Vis.*, vol. 63, no. 2, pp. 153–161, 2005.

[3] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1442–1468, Jul. 2014.

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

[5] 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.

[6] 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. (ICCV)*, Dec. 2015, pp. 4310–4318.

[7] 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. (CVPR)*, Jun. 2016, pp. 1401–1409.

[8] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Discriminative scale space tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1561–1575, Aug. 2017.

[9] 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.

[10] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2000, pp. 142–149.

[11] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 125–141, 2008.

[12] X. Mei and H. Ling, "Robust visual tracking using $\ell 1$ minimization," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 1436–1443.

[13] S. Hare *et al.*, "Struck: Structured output tracking with kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2096–2109, Oct. 2016.

[14] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, Aug. 2011.

[15] 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.

[16] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Long-term correlation tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5388–5396.

[17] A. Bibi, M. Mueller, and B. Ghanem, "Target response adaptation for correlation filter tracking," in *Proc. 14th Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The Netherlands, Oct. 2016, pp. 419–433.

[18] 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 (ICCVW)*, Dec. 2015, pp. 621–629.

[19] 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.

[20] M. Kristan *et al.*, "The visual object tracking VOT2015 challenge results," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCV)*, Santiago, Chile, Dec. 2015, pp. 564–586.

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

[22] H. K. Galoogahi, T. Sim, and S. Lucey, "Multi-channel correlation filters," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Sydney, NSW, Australia, Dec. 2013, pp. 3072–3079.

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

[24] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2005, pp. 886–893.

[25] J. van de Weijer, C. Schmid, J. Verbeek, and D. Larlus, “Learning color names for real-world applications,” *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1512–1523, Jul. 2009.

[26] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, “Accurate scale estimation for robust visual tracking,” in *Proc. Brit. Mach. Vis. Conf.* BMVA Press, Sep. 2014, pp. 1–11.

[27] S. Liu, T. Zhang, X. Cao, and C. Xu, “Structural correlation filter for robust visual tracking,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 4312–4320.

[28] M. Mueller, N. Smith, and B. Ghanem, “Context-aware correlation filter tracking,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1387–1395.

[29] P. Liang, E. Blasch, and H. Ling, “Encoding color information for visual tracking: Algorithms and benchmark,” *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5630–5644, Dec. 2015.

[30] M. Mueller, N. Smith, and B. Ghanem, “A benchmark and simulator for UAV tracking,” in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, 2016, pp. 445–461.

[31] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, “ECO: Efficient convolution operators for tracking,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 6131–6139.

[32] J. Nocedal and S. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006.

[33] P. Dollár. *Piotr’s Computer Vision Matlab Toolbox (PMT)*. Accessed: 2014. [Online]. Available: https://github.com/pdollar/toolbox

[34] A. Vedaldi and K. Lenc, “Matconvnet: Convolutional neural networks for matlab,” in *Proc. 23rd ACM Int. Conf. Multimedia*, 2015, pp. 689–692.

[35] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, “Return of the devil in the details: Delving deep into convolutional nets,” in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2014.

[36] H. K. Galoogahi, A. Fagg, and S. Lucey, “Learning background-aware correlation filters for visual tracking,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 21–26.

[37] J. Valmadre, L. Bertinetto, J. F. 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. (CVPR)*, 2017, pp. 5000–5008.

[38] 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. (ECCV)*. Springer, 2016, pp. 850–865.

[39] M. Felsberg *et al.*, “The thermal infrared visual object tracking VOT-TIR2016 challenge results,” *in Proc. Eur. Conf. Comput. Vis. Workshops (ECCV)*. Springer, 2016, pp. 824–849.

[40] S. Song and J. Xiao, “Tracking revisited using RGBD camera: Unified benchmark and baselines,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Sydney, NSW, Australia, Dec. 2013, pp. 233–240.

Dongdong Li received the B.S. degree from Wuhan University in 2012 and the M.S. degree from National University of Defense Technology in 2014. He is currently pursuing the Ph.D. degree with the College of Electronic Science and Technology, National University of Defense Technology, Changsha, Hunan, China. He has been working on camera calibration, object detection, and visual object tracking problems. He serves as a Reviewer for *Optical Engineering* and *Optics and Lasers in Engineering*.

Gongjian Wen received the B.S., M.S., and Ph.D. degrees from the National University of Defense Technology, Changsha, China, in 1994, 1997, and 2000, respectively. Since 2009, he has been a Professor with the College of Electronic Science and Engineering, National University of Defense Technology, where he served as the Head of the fourth department of the National Key Laboratory of Automatic Target Recognition. His research interests include image understanding, remote sensing, and target recognition.

Yangliu Kuai received the B.S. and M.S. degrees from the National University of Defense Technology, Changsha, Hunan, China, in 2013 and 2015, respectively, where she is currently pursuing the Ph.D. degree with the College of Electronic Science. She has been working on camera calibration, object detection, and visual object tracking problems.

Fatih Porikli (F’14) received the Ph.D. degree from NYU. He served as a Distinguished Research Scientist at Mitsubishi Electric Research Laboratories, Cambridge, MA, USA. He is currently a Professor with the Research School of Engineering, Australian National University, Canberra, Australia. He is also acting as the Leader of the Computer Vision Group, NICTA, Australia. He has contributed broadly to object detection, motion estimation, tracking, image-based representations, and video analytics. He has 66 granted patents. He is the coeditor of two books on *Video Analytics for Business Intelligence* and Handbook on *Background Modeling and Foreground Detection for Video Surveillance*. His publications won four Best Paper Awards and also received the R&D 100 Award in the Scientist of the Year category in 2006. He has served as the general and program chair of numerous IEEE conferences. He is an associate editor of five journals.