J. Vis. Commun. Image R. 58 (2019) 149–159

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Learning target-aware correlation filters for visual tracking ☆

Dongdong Li [a,*], Gongjian Wen [a], Yangliu Kuai [a], Jingjing Xiao [b], Fatih Porikli [c]

[a] *National University of Defense Technology, Changsha, China*
[b] *Department of Medical Engineering, Xinqiao Hospital, Chongqing, China*
[c] *Australian National University, Canberra, Australia*

ARTICLE INFO

Article history:
Received 8 January 2018
Revised 5 August 2018
Accepted 22 November 2018
Available online 23 November 2018

Keywords:
Correlation filter
Target likelihood map
Visual tracking

ABSTRACT

Discriminative Correlation Filters (DCF) have achieved enormous popularity in the tracking community. Generally, DCF based trackers assume that the target can be well shaped by an axis-aligned bounding box. Therefore, in terms of irregularly shaped objects, the learned correlation filter is unavoidably deteriorated by the background pixels inside the bounding box. To tackle this problem, we propose Target-Aware Correlation Filters (TACF) for visual tracking. A target likelihood map is introduced to impose discriminative weight on filter values according to the probability of this location belonging to the foreground target. According to the TACF formulation, we further propose an optimization strategy based on the Preconditioned Conjugate Gradient method for efficient filter learning. With hand-crafted features (HOG), our approach achieves state-of-the-art performance (62.8% AUC) on OTB100 while running in real-time (24 fps) on a single CPU. With shallow convolutional features, our approach achieves 66.7% AUC on OTB100 and the top rank in EAO on the VOT2016 challenge.

1. Introduction

Visual tracking is a classical and rapidly evolving research topic in computer vision with many applications in multimedia such as video surveillance [1,2], augmented reality [3] and human-computer interaction [4]. Generic tracking means single-camera, single-object, short-term and model-free tracking. It is the task of continuously locating a target given only its initial state (generally an axis-aligned rectangle) in a video sequence. In recent years, multiple tracking benchmarks [5–7] and challenges [8,9] have seen the continuous performance improvement of visual tracking. However, without prior assumptions regarding the object appearance or category, robust tracking under complex scenarios is still challenging due to deformation, occlusion and background clutter.

Recently, Discriminative Correlation Filters (DCF) based trackers have achieved enormous popularity in the tracking community. With the circular structure, standard DCF transforms computationally consuming spatial correlation into efficient element-wise operation in the Fourier domain and achieve extremely high tracking speed. However, standard DCF significantly suffers from boundary effects due to the circulant assumption, which leads to a restricted search area.

To suppress the boundary effects and expand the search area, spatially constrained correlation filters introduce spatial constraints into the standard DCF formulation. Danelljan et al. [10] introduce a spatial regularization component into filter learning to penalize correlation filter values depending on their spatial location. The Spatially Regularized Correlation Filter (SRDCF) allows the correlation filters to be learned on a significantly larger set of negative training samples, without corrupting the positive samples. However, SRDCF can't guarantee zero filter values outside the target bounding box (see Fig. 1). Recently, Galoogahi et al. [11,12] propose Background-Aware Correlation Filters (BACF) which impose direct spatial constraints (a binary mask) on filter learning. BACF maintains nonzero filter values only inside the target bounding box, which significantly reduces boundary effects and trainable parameters in the tracking model. The small correlation filter is padded with zeros in the neighborhood to increase the filter size. However, one limitation of BACF is the assumption that the target shape is well approximated by an axis-aligned bounding box. The binary mask in BACF is manually constructed based on this rectangular shape assumption. For irregularly shaped objects, deformable object and rectangular but partially occluded objects, the learned filter is unavoidably corrupted by the background information inside the bounding box.

To overcome the limitation related to the rectangular shape assumption, we have to exactly know whether a pixel belongs to the background or the foreground target in the search window. In SRDCF, the possibility of a pixel belonging to the foreground

☆ This paper has been recommended for acceptance by Zicheng Liu.
* Corresponding author.
E-mail address: moqimubai@sina.cn (D. Li).

https://doi.org/10.1016/j.jvcir.2018.11.036
1047-3203/© 2018 Published by Elsevier Inc.

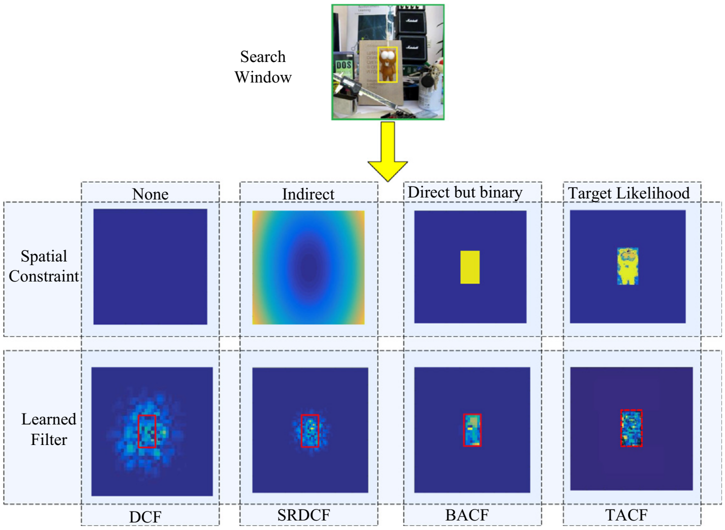
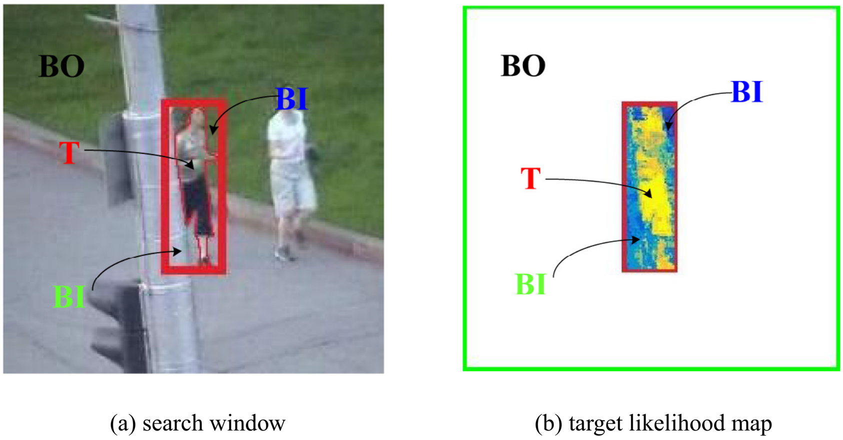


Fig. 1. An comparison of the standard Discriminative Correlation Filter (DCF), Spatially Regularized Correlation Filter (SRDCF), Background-Aware Correlation Filter (BACF) and Target-Aware Correlation Filter (TACF) in terms of spatial constraints and learned filters.

target is measured by its distance to the filter center. In BACF, this possibility is measured by whether this pixel exists inside the target bounding box. Both SRDCF and BACF take pixel location into consideration but ignore pixel values. In this work, we propose Target-Aware Correlation Filters (TACF) for Visual Tracking. As shown in Fig. 2, we classify pixels in the search window into three categories: background pixels outside the bounding box (**BO**), background pixels inside the bounding box (**BI**) and foreground target pixels (**T**). To distinguish target pixels from background pixels, we introduce target likelihood which assigns zero weights to **BO** pixels, low weights to **BI** pixels and high weights to **T** pixels according to the probability of this location belonging to the foreground target. The low and high weights of **BI** and **T** are generated from the foreground/background color models. Therefore, with the pixel-wise weights on the target likelihood map, we can impose discriminative weights on filter values in filter learning. Compared with BACF which coarsely separate background and target pixels with the bounding box and thus regard **BI** as foreground target pixels, our Target-Aware Correlation Filters (TACF) separate the pixels inside the bounding box in a more fine-grained way. In this way, the limitation related to rectangular shape assumption can be overcome with the introduction of target likelihood.

The major contributions of this work are threefold:

- We propose Target-Aware Correlation Filters (TACF) for visual tracking. To overcome the rectangular shape assumption, target likelihood is introduced into the TACF formulation to guided filter learning. In filter training, TACF is guided to focus on the foreground target pixels and reduce the emphasis on the background information inside the target bounding box.
- According to the TACF formulation, we propose an optimization strategy, based on the Preconditioned Conjugate Gradient (PCG) Method, for efficient online filter learning. In Section 3.4, we present the detailed derivation steps for the optimization procedure.
- We perform extensive experiments on OTB100 [13] and VOT2016 [9] benchmark datasets. Our tracker achieves state-of-the-art performance and real-time frame-rates on OTB100 and the top rank in Expected Average Overlap (EAO) on the VOT2016 challenge.



(a) search window

(b) target likelihood map

Fig. 2. The search window and its corresponding target likelihood map. **BO** corresponds to background pixels outside the bounding box in the search window. T corresponds to foreground target pixels inside the red human silhouette in the bounding box. BI corresponds to background pixels caused by partial occlusion while BI corresponds to background pixels caused by irregular target shape. The object likelihood map assigns zero weights to **BO**, high weights to T and low weights to BI and BI. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

2. Related works

There are extensive surveys on visual tracking in the literature. We refer interested readers to [13,9] for a thorough review of existing tracking algorithms. In this section, we only focus on correlation filter based trackers.

Conventional Discriminative Correlation Filters. Compared with traditional tracking-by-detection methods [14–17], discriminative correlation filters draw much attraction in the tracking community due to dense training samples and high computational efficiency. The pioneer MOSSE tracker [18] achieves an impressive tracking speed of over 600 fps. Later, based on the standard DCF formulation, different variants of correlation filters have been proposed to boost tracking performance using multi-dimensional features [19], robust scale estimation [20], non-linear kernels [21], long-term memory components [22], complementary cues [23] and target adaptation [24]. Despite continuous performance improvement, learning correlation filters in the frequency domain significantly suffers from the underlying boundary effects. The boundary effects lead to suboptimal tracking performance and a restricted search area.

Spatially Constrained Correlation Filters. To suppress boundary effects and expand search area, SRDCF [10] learn a correlation filter with large spatial support. Filter values outside the object bounding box are penalized with large regularization weights. Within the spatially regularized framework, CCOT [25] employs the integration of multi-resolution features in the continuous domain and achieves the top rank on the VOT2016 challenge [9]. Despite these achievements, SRDCF can't guarantee that filter values are zero outside the object bounding box. These nonzero values in the background area hardly contribute to target location but increase the computational burden and risk of over-fitting. Recently, Galoogahi et al. propose to directly impose spatial constraints on the filter support and maintain nonzero filter values only within the target bounding box. The method of CFLB [11] was first proposed to train correlation filters from real negative samples densely extracted from the background. However, CFLB is limited to pixel intensities which perform poorly in challenging tracking scenarios. Later, BACF [12] was proposed to extend CFLB from pixel intensities to multi-channel features. BACF learns a small rectangular filter and increase the filter size by padding the filter with zeros in the neighborhood. However, BACF regards the target bounding box as the boundary between the background pixels and foreground target pixels. Therefore, the background pixels inside the bounding box are treated as target pixels in correlation filter learning, which leads to suboptimal tracking performance.

Color Aided Correlation Tracking. Using color cues for helping object tracking is quite common in the tracking community. While color information is known to provide rich discriminative clues for visual tracking, most modern trackers exploit color cues either on the feature level or the response level. Weijer et al. [26] learned Color Names (CN) from real-world pictures which are later employed as hand-crafted features in visual tracking. Danelljan et al. [19] extended the CSK [27] tracker with color attributes as additional features and achieved superior performance for visual tracking. Bertinetto et al. [23] fused the color histogram score with correlation response and improves the robustness of correlation tracking against target deformation. In this work, we try to develop a DCF based tracking framework which exploits color cues on a higher level to guide correlation filter learning.

3. Our approach

In this section, we introduce our Target-Aware Correlation Filters (TACF) in details. We first overview the overall tracking framework of TACF in Section 3.1 and then introduce the concept of target likelihood in Section 3.2. Our detailed TACF formulation is proposed in Section 3.3. Based on the TACF formulation, we derive an efficient iterative optimization procedure for online filter training in Section 3.4. Last but not least, in Section 3.5, we introduce the detection formula of our TACF tracker.

3.1. The TACF framework

The diagram of our tracking framework is demonstrated in Fig. 3. We assume that the tracking window in the first frame is

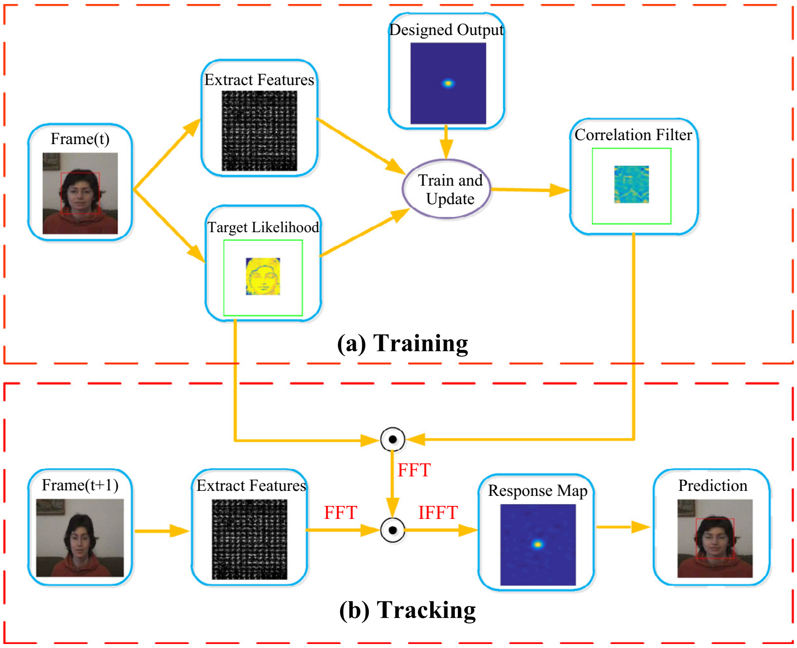


Fig. 3. Main components of our object likelihood guided tracking framework.

given by a detector or manual label. At each frame, we extract features and the target likelihood map from the search window. To guide correlation filter training, the target likelihood map is fed into the iterative optimization procedure along with the extracted features and Gaussian target labels. To predict the target location in the next frame, the learned correlation filter in the current frame is first weighted by the target likelihood map and then convolved with the extracted features in the next frame. The target is located from the maximum of the estimated response map. It's worth noting that our learned correlation filter is derived in the spatial domain in the training stage and transformed into the frequency domain in the detection stage to perform element-wise multiplication.

3.2. Target likelihood

In this work, target likelihood indicates the probability of a pixel belonging to the foreground target. Our target likelihood map is generated from the foreground/background color model for efficiency reasons. For further performance improvement, the target likelihood map can be generated in a more complicated way, such as visual saliency or video segmentation.

Given an image patch I centered at the target (see Fig. 4a), the foreground object histogram H^O and background histogram H^B can be derived from the pixels inside and outside the target bounding box respectively. From H^O and H^B, we can derive the histogram score map w_h as

$$w_h(i,j) = \frac{H^O(b_{i,j})}{H^O(b_{i,j}) + H^B(b_{i,j})}, \quad i \in [0,M], j \in [0,N] \tag{1}$$

where $M \times N$ is the training sample size and $b_{i,j}$ denote the color bin for pixel $I(i,j)$.

As shown in Fig. 4b, the histogram score map w_h maintains high values on target pixels and low values on background pixels, which implies the probability of this location belonging to the foreground target. However, we empirically assume that the target pixels only exist inside the target bounding box. Based on this assumption, we define a spatial prior w_p which is a binary mask with 0 outside the bounding box and 1 inside the bounding box as shown in Fig. 4c. The target likelihood map w_f for the correlation filter is derived as follows

$$w_f = w_p \cdot w_h. \tag{2}$$

From the target likelihood map w_f as Fig. 4d, we can ensure that there are only nonzero filter values inside the target bounding box. Moreover, in filter learning, w_f highlights the filter values with high object likelihood and suppresses the filter values with low object likelihood. In this way, w_f mitigates the emphasis on the background information in the learned correlation filter and thus overcomes the limitation of the rectangular shape assumption.

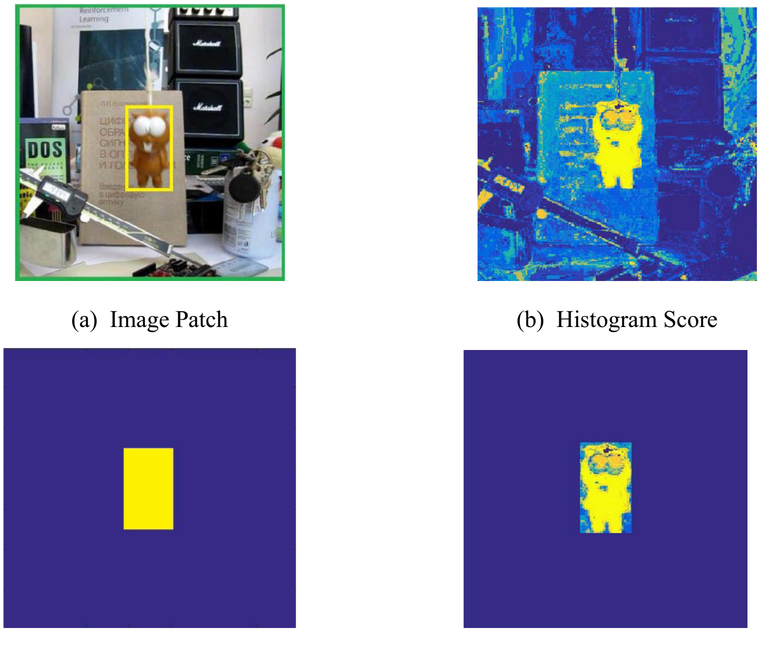
3.3. TACF formulation

To highlight the region that is suitable for correlation filter learning, we impose the target likelihood map w_f on the correlation filter f as following

$$\varepsilon(f) = \sum_{k=1}^{t} \alpha_k \left\| \sum_{l=1}^{d} x_k^l * (w_f \cdot f^l) - y \right\|^2 + \lambda \sum_{l=1}^{d} \left\| f^l \right\|^2. \tag{3}$$

Here, $\alpha_k \geqslant 0$ determine the impact of the k_{th} training sample x_k.

In (3), we impose w_f on f^l as spatial weights and achieve two advantages. First, as mentioned earlier, the spatial weights treat filter values in a discriminate way and thus help to overcome the limitation of the rectangular shape assumption. Second, similar to BACF [12], the learned correlation filter f^l only need to maintain a small filter support within the bounding box and is extended to the size of the search window by padding zeros in the neighborhood. This significantly reduces the number of trainable filter values in the model.



(a) Image Patch

(b) Histogram Score

(c) Spatial Prior

(d) Target Likelihood

Fig. 4. The histogram score (b) is estimated from the image patch (a) using the foreground/background color model. The target likelihood map (d) is derived by imposing spatial prior (c) on the histogram score (b). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

In fact, (3) indicates a more generalized formulation for correlation filter learning. (3) degenerates to DCF if w_f is set to flat weights and to BACF if w_f is set to binary weights (Fig. 4c).

3.4. Correlation filters training

For derivation convenience, here we introduce a binary $MN \times mn$ matrix B which performs a mapping from the vectorization of a $m \times n$ matrix M_1 to the vectorization of a $M \times N$ matrix M_2. M_2 can be derived by padding M_1 with zeros in the neighborhood. On the other hand, the transpose B^T performs an opposite mapping from M_2 to M_1. In fact, we don't have to instantiate B and B^T. In practice, B and B^T equally perform the augmenting and cropping operation as lookup tables.

With the matrix B, we introduce a vectorized version of (3) as (4).

$$\varepsilon(f) = \sum_{k=1}^{t} \alpha_k \left\| \sum_{l=1}^{d} \mathcal{C}(\mathbf{x}^l)\mathcal{D}(\mathbf{w_f})B\mathbf{f}^l - \mathbf{y} \right\|^2 + \lambda \sum_{l=1}^{d} \left\| \mathbf{f}^l \right\|^2. \tag{4}$$

Here, the bold letters $\mathbf{x}_k^l, \mathbf{w_f}, \mathbf{f}^l, \mathbf{y}$ denote a vectorization of the scalar matrices x_k^l, w_f, f^l, y. Particularly, $\mathbf{f}^l$ is a $mn \times 1$ vector and the $MN \times MN$ matrix $\mathcal{C}(\mathbf{x}_k^l)$ represents the circulant matrix of the $M \times N$ matrix x_k^l. Each row in $\mathcal{C}(\mathbf{x}_k^l)$ contains a cyclic permutation of $\mathbf{x}_k^l$. $\mathcal{D}(\mathbf{w_f})$ denotes the diagonal matrix with the elements of $\mathbf{w_f}$ in its diagonal.

To derive the solution to the minimization problem (4) subject to the fully vectorized filter $\mathbf{f} = [(\mathbf{f}^1)^T \dots (\mathbf{f}^d)^T]^T$, we define the concatenated matrix $\mathbf{X}_k = [\mathcal{C}(\mathbf{x}_k^1) \dots \mathcal{C}(\mathbf{x}_k^d)]$, the $d \times d$ block-diagonal matrix $\mathbf{W} = [\mathcal{D}(\mathbf{w_f}) \oplus \dots \oplus \mathcal{D}(\mathbf{w_f})]$ and the $d \times d$ block-diagonal matrix $\mathbf{B} = B \oplus \dots \oplus B$. Each diagonal block of $\mathcal{D}(\mathbf{w_f})$ and $\mathbf{B}$ is equal to $\mathbf{w_f}$ and B respectively. Therefore, (4) can be further simplified as

$$\varepsilon(\mathbf{f}) = \sum_{k=1}^{t} \alpha_k \|\mathbf{X}_k\mathbf{WBf} - \mathbf{y}\|^2 + \lambda\|\mathbf{f}\|^2. \tag{5}$$

The minimizer of (5) is found by solving the following normal equations,

$$(\mathbf{B}^T\mathbf{W}^T\mathbf{X}^T\mathbf{\Gamma}\mathbf{XWB} + \lambda\mathbf{I})\mathbf{f} = \mathbf{B}^T\mathbf{W}^T\mathbf{X}^T\mathbf{\Gamma}\mathbf{y}. \tag{6}$$

Here, we define the sample matrix $\mathbf{X} = [\mathbf{X}_1^T \dots \mathbf{X}_t^T]^T$, the diagonal weight matrix $\mathbf{\Gamma} = \alpha_1 I \oplus \dots \oplus \alpha_t I$ and the label vector $\mathbf{y} = [\mathbf{y}^T \dots \mathbf{y}^T]^T$.

Eq. (6) describes a linear equation system formulated in the spatial domain. The advantages of this equation system is twofold. First, the size of the vector $\mathbf{f}$ to be solved is $mnd \times 1$ instead of MNd which significantly the number of trainable parameters in the tracking model. Second, the coefficient matrix of (6) is symmetric and positive-definite. Therefore, we can employ the Preconditioned Conjugate Gradient (PCG) method [28] to iteratively solve the normal Eq. (6).

In fact, it is unnecessary to form the big $mnd \times mnd$ symmetric positive-semidefinite matrix $(\mathbf{B}^T\mathbf{W}^T\mathbf{X}^T\mathbf{\Gamma}\mathbf{XWB} + \lambda\mathbf{I})$ in memory in each cycle of Conjugate Gradient Optimization. The left-hand side of the normal Eq. (6) is computed from right to left by performing the matrix-vector and transpose matrix-vector multiplication. $\mathbf{B}, \mathbf{B}^T$ and $\mathbf{W}$ perform as the augmenting, cropping and weighting operators respectively. $\mathbf{X}$ is a $t \times d$ block matrix with each block as a $MN \times MN$ circulant matrix $\mathcal{C}(\mathbf{x}_k^l)$. Therefore, the matrix-vector multiplication related to $\mathbf{X}$ can be transformed into the frequency domain as efficient element-wise multiplication related to a $t \times d$ diagonal block matrix $\mathbf{D}$. Each block of $\mathbf{D}$ is a diagonal matrix $\mathcal{D}(\hat{\mathbf{x}}_k^l)$ corresponding to the circulant matrix $\mathcal{C}(\mathbf{x}_k^l)$. Here, ~ denotes the Fourier transform $\mathcal{F}$ of a function.

Given the initial guess of $\mathbf{f}$ by $\mathbf{f}_0$, the search direction $\mathbf{p}$ and search scale $\mathbf{s}$ can be derived from the residual vector $\mathbf{r}$ in each iteration. A full description of the detailed PCG optimization can be seen in Algorithm 1. It's worth to mention that we adopt the Jacobi preconditioner as $Diag(\mathbf{B}^T\mathbf{W}^T\mathbf{X}^T\mathbf{\Gamma}\mathbf{XWB} + \lambda\mathbf{I})$ to ensure a small condition number of (6) and faster convergence in Algorithm 1.

Algorithm 1. Optimization using the PCG method

```
Initialize $\mathbf{f}$
Convert circulant matrices in the spatial domain into diagonal blocks in the frequency domain:$\mathcal{F}(\mathbf{X}) \rightarrow \mathbf{D}$
Apply Fourier transform to gaussian labels:$\mathcal{F}(\mathbf{y}) \rightarrow \hat{\mathbf{y}}$
Repeat
    Apply augmenting, weighting and Fourier transform operation to $\mathbf{f}$:
    $\mathcal{F}(\mathbf{WBf}) \rightarrow \hat{\mathbf{f}}$
    Transform circular correlation in the spatial domain into element-wise multiplication in the frequency domain:
        $\mathcal{F}(\mathbf{X}^T\mathbf{\Gamma}\mathbf{XWBf}) = \mathbf{D}^T\mathbf{\Gamma}\mathbf{D}\hat{\mathbf{f}}$
        $\mathcal{F}(\mathbf{X}^T\mathbf{\Gamma}\mathbf{y}) = \mathbf{D}^T\mathbf{\Gamma}\hat{\mathbf{y}}$.
    Apply inverse Fourier transform, weighting and cropping operation:
    $\mathbf{B}^T\mathbf{W}^T\mathbf{X}^T\mathbf{\Gamma}\mathbf{XWBf} = \mathbf{B}^T\mathbf{W}^T\mathcal{F}^{-1}(\mathbf{D}^T\mathbf{\Gamma}\mathbf{D}\hat{\mathbf{f}})$
        $\mathbf{B}^T\mathbf{W}^T\mathbf{X}^T\mathbf{\Gamma}\mathbf{y} = \mathbf{B}^T\mathbf{W}^T\mathcal{F}^{-1}(\mathbf{D}^T\mathbf{\Gamma}\hat{\mathbf{y}})$
    Compute the residual vector:
    $\mathbf{r} = \mathbf{B}^T\mathbf{W}^T\mathbf{X}^T\mathbf{\Gamma}\mathbf{y} - \mathbf{B}^T\mathbf{W}^T\mathbf{X}^T\mathbf{\Gamma}\mathbf{XWBf} - \lambda\mathbf{f}$
    Compute the search direction $\mathbf{p}$ and search scale $\mathbf{s}$ from $\mathbf{r}$ with the PCG method.
    Update the filter: $\mathbf{f} = \mathbf{f} + \mathbf{sp}$.
Until $\mathbf{f}$ has converged or the maximum number of iterations has reached.
```

3.5. Correlation filter detection

Let z denote the $M \times N \times d$ test sample extracted in the current frame and f denote the $m \times n \times d$ correlation filter learned in the spatial domain in the previous frame. F denotes the augmented filter with zero-padding in the neighborhood of f. The correlation scores $S(z,f)$ at all locations in the image patch are computed as follows,

$$S(z,f) = \mathcal{F}^{-1}\left\{\sum_{l=1}^{d} \hat{z}^l \cdot \mathcal{F}(w_f \cdot F^l)\right\}. \tag{7}$$

Note that the operation $S(z,f)$ corresponds to searching the target over a big search area z with a small matching template f in the spatial domain in a sliding-window fashion. The correlation filter f is weighted by the target likelihood map to highlight the foreground target and suppress the background. The correlation filter f is applied on multiple scales of the searching area to estimate scale changes. We employ the subgrid interpolation strategy to achieve sub-pixel location accuracy. Here, we present an outline of the tracking framework in Algorithm 2.

Algorithm 2. Target-aware correlation filters

```
Input:
  Image patch $I_t$ extracted from frame $t$, object location $p_{t-1}$
  on previous frame, scale $s_{t-1}$, filter $f_{t-1}$, foreground and
  background color histogram $H^O_{t-1}, H^B_{t-1}$.
Output:
  Estimated position $p_t$, scale $s_t$ and filter $f_t$ on the current frame.
Location and scale estimation:
1: Compute the target likelihood map $w_f$ from $H^O_{t-1}$ and $H^B_{t-1}$.
   (Section 3.2)
2: Weight $f_{t-1}$ with $w_f$ and estimate the new target location $p_t$
   and scale $s_t$. (Section 3.5)
Model Update:
1: Estimate the foreground and background color histograms $H^O, H^B$
   from the image patch extracted around $p_t$ in the current frame.
2: Update foreground and background color histograms as
   $H^O_t = (1-\eta)H^O_{t-1} + \eta H^O$ and $H^B_t = (1-\eta)H^B_{t-1} + \eta H^B$.
3: Update the new filter $f_t$ with the PCG method. (Section 3.4)
```

4. Experiments and results

We validate our Target-Aware Correlation Filters (TACF) by performing comprehensive experiments on three tracking benchmarks: OTB50 [5], OTB100 [13] and VOT2016 [9].

Evaluation Methodology: On OTB50 and OTB100, we use the precision plots and success plots in one-pass evaluation (OPE) [13] to rank all the trackers. The precision plots are computed as the percentage of frames in the sequences where Euclidean distance between the ground-truth and the estimated target position is smaller than a certain threshold. The success plots are plotted over the range of intersection over union (IoU) thresholds over all videos. 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 OTB50 and OTB100, the trackers in VOT2016 are restarted at each failure.

Comparison Scenarios: In our experiments, we implement two versions of our target-aware correlation filters, namely TACF with handcrafted features (HOG) and DeepTACF with shallow convolu-

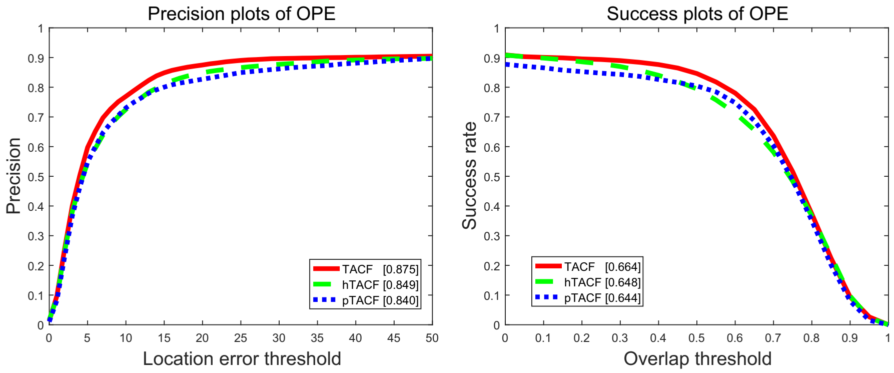


Fig. 5. Precision plots and Success plots for TACF and its baseline trackers (pTACF and hTACF) on OTB50.

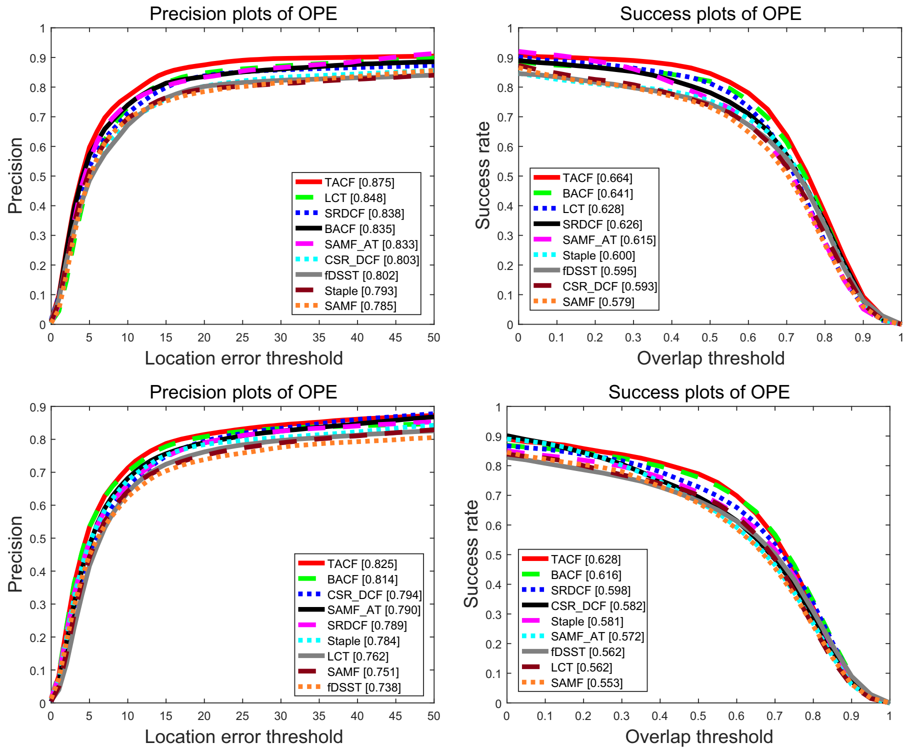


Fig. 6. Precision plots and Success plots for state-of-the-art HOG based trackers on OTB50 (first row) and OTB100 (second row).

tional features. An ablation study is done on OTB50 to demonstrate the effectiveness of target likelihood on guiding correlation filter learning. On OTB100, we compare TACF with state-of-the-art HOG-based trackers and compare DeepTACF with deep trackers employing convolutional features or trained in an end-to-end fashion. On the VOT2016 dataset, we compare DeepTACF with the top 10 trackers in the challenge.

Implementation Details: TACF employs the 31-dimensional HOG features with 4 × 4 cell size. DeepTACF employs the convolutional features extracted from the first convolutional layer in the imagenet-vgg-m-2048 network [29]. The target likelihood map in TACF is extracted from the foreground/background color histograms with 32 bins per color channel. The interpolation parameter η in Algorithm 2 is set to 0.04 following the literature [23]. The regularization parameter λ for the filter in (6) is set to 1e-5. The number of scales is set to 5 with a scale step of 1.02. The region size of the samples to be square and 4.5^2 times the target area. Parameters are fixed for all videos in each dataset. Our tracker is implemented in Matlab and uses Matconvnet [30] for deep feature extraction. The comparison experiments of TACF are performed on a 4-core Intel Core -7-6700 CPU at 3.4 GHz. The comparison experiments of DeepTACF are performed on a GeForce GTX TITAN GPU.

Table 1
Success rates (% at IoU = 0.5) of our approach versus state-of-the-art HOG-based trackers. The first and second best methods are shown in color.

Tracker	TACF	BACF	SRDCF	CSR_DCF
Mean OP (%)	77.9	77.2	72.8	71.9
Avg. FPS	24.2	29.1	3.6	13.0
Tracker	LCT	Staple	fDSST	SAMF
Mean OP (%)	70.1	70.9	68.1	67.4
Avg. FPS	21.2	60.5	82.3	22.5

4.1. Evaluation on OTB

4.1.1. Ablation study

In this subsection, an ablation study on OTB50 is conducted to demonstrate the effectiveness of the target likelihood in guiding filter learning. As analyzed in Section 3.2, the target likelihood map is derived from the spatial prior w_p and the histogram score map w_h. We introduce two baseline trackers (pTACF and hTACF) by setting the target likelihood map w_f to a fixed binary mask w_p and the histogram score w_h respectively. pTACF assumes all the pixels in the bounding box belong to the foreground target and misclassify the background pixels inside the target bounding box in the case of irregularly shaped objects or deformable objects. Compared with pTACF, hTACF imposes high weights on the foreground target pixels inside the bounding box but can't guarantee zero filter values outside the bounding box. Therefore, both pTACF and hTACF suffer from pixel ambiguity in filter learning. On contrast, TACF alleviates pixel ambiguity with the target likelihood map and significantly increases the discriminative power of the learned correlation filter. Fig. 5 shows the precision and success plots of the three trackers (TACF, pTACF, hTACF). Compared with pTACF and hTACF, TACF achieves better performance in both the precision and success plots, which demonstrates the effectiveness of our target likelihood in guiding filter learning.

4.1.2. Comparison with HOG-based trackers

In this subsection, we compare TACF with state-of-the-art HOG-based trackers on the OTB50 and OTB100 datasets. The compared

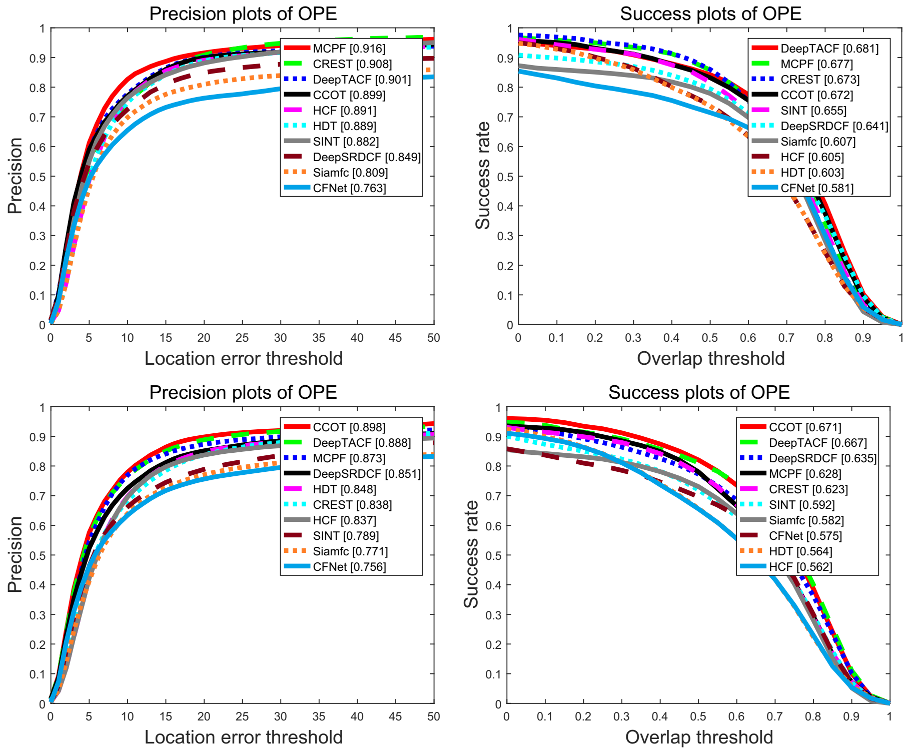


Fig. 7. Precision plots and Success plots for state-of-the-art deep trackers on OTB50 (first row) and OTB100 (second row).

trackers include BACF [12], SRDCF [10], CSR_DCF [31], LCT [22], Staple [23], fDSST [20], SAMF [32] and SAMF_AT [24].

Fig. 6 compare TACF with the above trackers on the OTB50 and OTB100 datasets, where our method achieved the highest score of both precision plots and success plots on two datasets. More particularly, TACF achieved the best AUC (62.8%) of success plots on OTB100 followed by BACF (61.6%) and SRDCF (59.8%), which demonstrates the superiority of our approach against existing spatially constrained correlation filters.

Table 1 shows the mean Overlap Precision (OP) at IoU = 0.5 and tracking speeds (FPS) of all compared HOG-based trackers on OTB100. TACF ranks first in mean overlap precision and runs almost in real-time at a speed of 23.5 fps.

Fig. 9 illustrates the attribute based evaluation of all compared HOG-based trackers in success plots on the OTB100 dataset. All sequences in the OTB100 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. 9, TACF achieves the best performance on 8 of 11 attributes, which demonstrates the robustness of our approach in challenging tracking scenarios.

To intuitively exhibit the superiority of our proposal, Fig. 8 shows screenshots on 7 challenging videos from the OTB100 dataset. Due to page limitation, we only compare TACF against BACF, SRDCF and CSR_DCF. The videos (from top to bottom) are *Human3*, *Diving*, *Skating1*, *Skiing*, *Box*, *Lemming* and *Shaking*. It is easy to see that TACF performs better than the compared trackers in presence of fast motion (*Skiing*), deformation(*Diving*), illumination variation (*Skating1*, *Shaking*) and partial or full occlusion (*Human3*, *Box*, *Lemming*).

4.1.3. Comparison with deep trackers

In this subsection, we compare the deep version of our approach, DeepTACF, with ten state-of-the-art deep trackers on OTB50 and OTB100 datasets. In our implementation, DeepTACF employs shallow convolutional features similar to DeepSRDCF [33]. The compared deep trackers include CREST [34], CFNet [35], Siamfc [36], SINT [37], HCF [38], HDT [39], CCOT [25], DeepSRDCF [33] and MCPF [40]. All the trackers are run on GPU.

Fig. 7 shows the precision and success plots of the compared deep trackers on OTB50 and OTB100. On OTB50, DeepTACF ranks third in precision plots and first in success plots. On OTB100, DeepTACF ranks second in both precision and success plots following

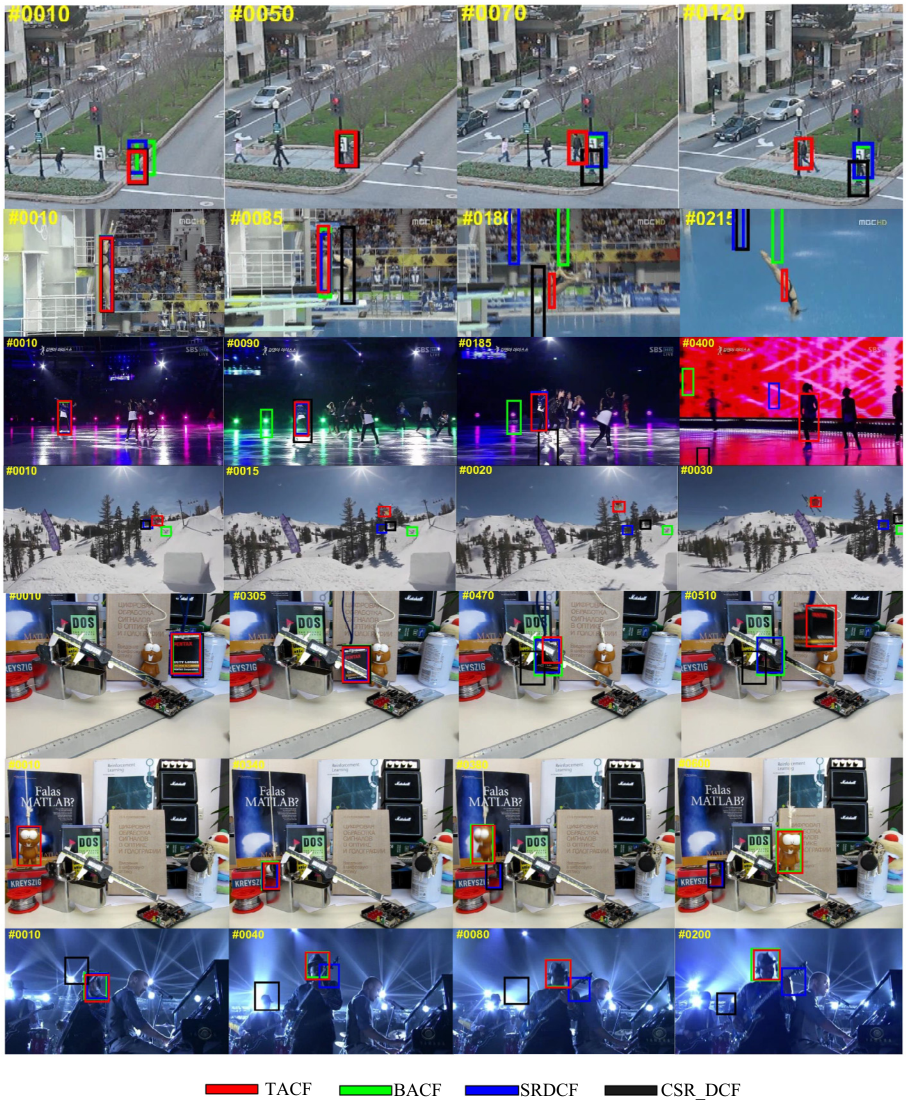


Fig. 8. Tracking screenshots of TACF, BACF, SRDCF and CSR_DCF. The videos (from top to bottom) are *Human3*, *Diving*, *Skating1*, *Skiing*, *Box*, *Lemming* and *Shaking* from the OTB100 dataset.

CCOT. The superior performance of CCOT can be attributed to the convolutional features from multiple layers. With only shallow convolutional features from the first layer, DeepTACF achieves comparable performance against CCOT.

Fig. 10 illustrates the attribute based evaluation of all deep trackers in success plots on the OTB100 dataset. DeepTACF achieves the best performance on 3 of 11 attributes and the second best performance on 8 of 11 attributes following CCOT.

Table 2 reports the mean Overlap Precision (OP) at IoU = 0.5 of DeepTACF and deep trackers as well as their tracking speed reported in the original paper. CCOT (82.0%) achieves the highest mean OP score followed by DeepTACF (81.1%). However, DeepTACF runs almost in real-time (23.5fps) with GPU while CCOT runs with an extremely slow speed of 0.3 fps.

4.2. Evaluation on VOT2016

The visual object tracking (VOT) challenge is a competition between short-term, model-free visual tracking algorithms. Different from OTB, for each sequence in this dataset, a tracker is restarted whenever the target is lost (*i.e.* at a tracking failure). Four primary measures are used to analyze tracking performance: accuracy (A), robustness (R), expected average overlap (EAO) and equivalent filter operation (EFO). A is calculated as the average IoU, while R is expressed in terms of the total number of failures. EAO represents the average IoU with no re-initialization following a failure. EAO reports the tracker speed in terms of a predefined filtering operation that the toolkit carries out prior to running the experiments. We refer readers to [9] for details.

Table 3 shows the comparison of our approach with the top 5 participants in the VOT2016 challenge. In Table 3, DeepTACF outperforms all the top 5 trackers at the EAO score (0.344) and EFO (13.46). As indicated in the VOT2016 report [9], the strict state-of-the-art bound is 0.251 under EAO metrics. For trackers whose EAO values exceed this bound, they will be considered as state-of-the-art trackers. Therefore, DeepTACF can be regarded as state-of-the-art. Fig. 11 shows a visualization of the accuracy and robustness ranking plot for the compared trackers on the VOT2016 dataset.

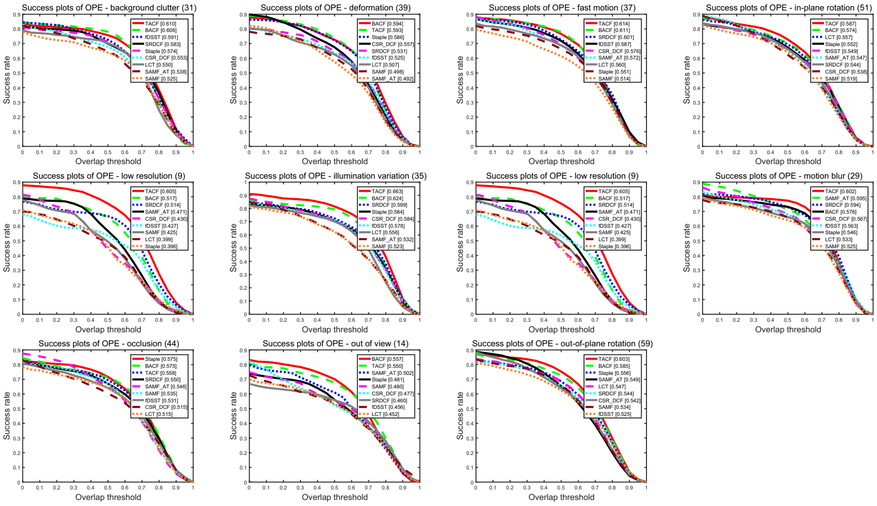


Fig. 9. *Success ratio* plots on 11 attributes of the OTB100 dataset. These HOG-based trackers are ranked by their AUC scores. Our method has achieved consistently the superior performance over the state-of-the-art.

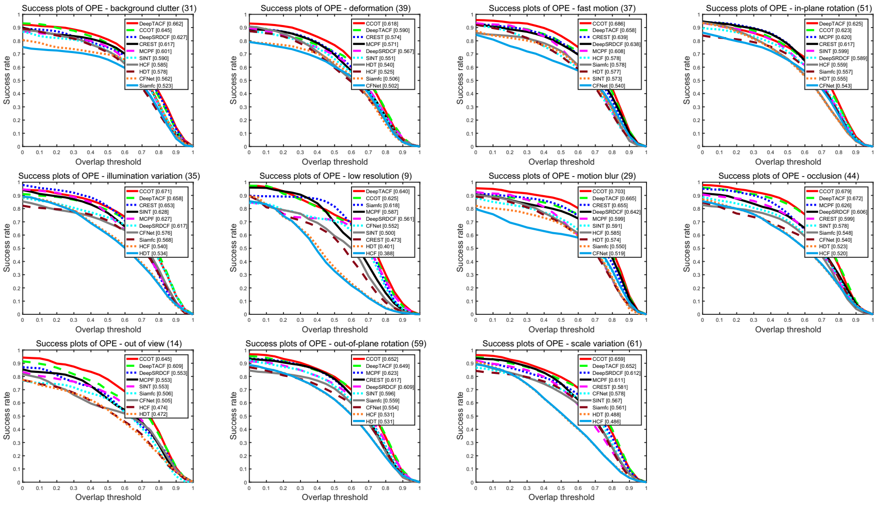


Fig. 10. *Success ratio* plots on 11 attributes of the OTB100 dataset. These deep trackers are ranked by their AUC scores. Our method has achieved consistently the superior performance over the state-of-the-art.

Table 2
Success rates (% at IoU = 0.5) of our approach versus state-of-the-art deep trackers. The first and second best methods are shown in color.

Tracker	**DeepTACF**	CCOT	DeepSRDCF	MCPF	SINT
Mean OP (%)	81.1	82.0	77.2	78.0	71.9
Avg. FPS	23.5	0.3	0.5	4	58
Tracker	Siamfc	CREST	CFNet	HDT	HCF
Mean OP (%)	73.0	77.6	69.6	65.7	65.5
Avg. FPS	86	2.1	75	10	11

Table 3
State-of-the-art comparison in terms of expected average overlap (EAO), robustness (failure rate), accuracy, and speed (in EFO units) on the VOT 2016 dataset. Only the top-5 best compared trackers are shown. The best and second best values are highlighted by red and green fonts.

	Staple	MLDF	SSAT	TCNN	C-COT	**DeepTACF**
EAO	0.295	0.310	0.320	0.325	0.331	0.344
Failure rate	1.35	0.83	1.04	0.96	0.85	0.96
Accuracy	0.544	0.490	0.577	0.554	0.539	0.540
EFO	11.144	1.483	0.475	1.049	0.507	13.46

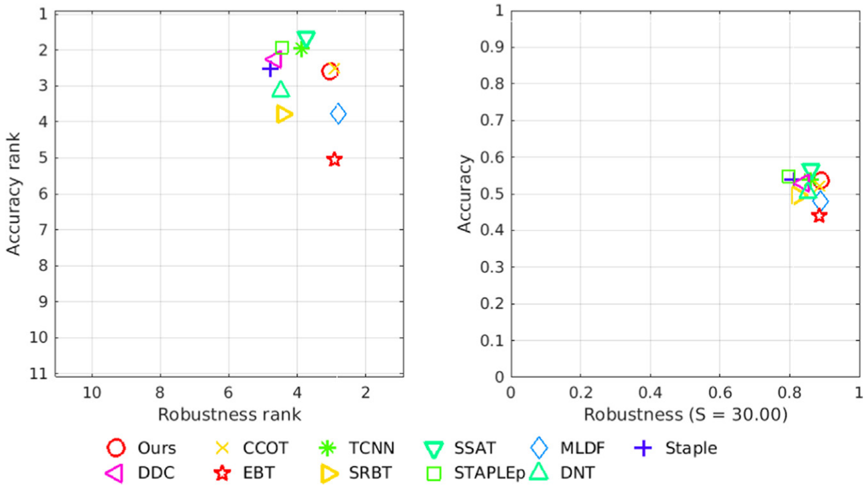


Fig. 11. A state-of-the-art comparison on the VOT2016 benchmark. In the ranking plot (left) the accuracy and robustness rank for each tracker is displayed. The AR plot (right) shows the accuracy and robustness scores.

5. Conclusion

We propose Target-Aware Correlation Filters (TACF) for robust visual tracking. An target likelihood map is introduced to overcome the limitation of the rectangular shape assumption. Our approach imposes the target likelihood map on the correlation filter and discriminatively adjusts the contribution of each filter value to circular correlation. An iterative optimization procedure is designed based on the preconditioned conjugate gradient method for correlation filter training. Experiments on the standard benchmarks demonstrate superior performance of our approach against state-of-the-art trackers with improved frame rates.

Our approach can be extended to exploit generic spatial constraints. Our future work will focus on visual saliency based target likelihood for robust tracking under challenging scenarios.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (NSFC) (No. 41601487, 61701506).

References

[1] K.H. Lee, J.N. Hwang, On-road pedestrian tracking across multiple driving recorders, IEEE Trans. Multimedia 17 (9) (2015) 1429–1438.

[2] L. Wang, X. Tang, D. Li, Robust tracking via spatio-temporally weighted multiple instance learning, in: Proceedings the IEEE Conference on Digital Image Computing: Techniques and Applications (DICTA), 2017, pp. 1–8.

[3] T. Guan, C. Wang, Registration based on scene recognition and natural features tracking techniques for wide-area augmented reality systems, IEEE Trans. Multimedia 11 (8) (2009) 1393–1406.

[4] G. Wu, W. Kang, Vision-based fingertip tracking utilizing curvature points clustering and hash model representation, IEEE Trans. Multimedia 19 (8) (2017) 1730–1741.

[5] Y. Wu, J. Lim, M.-H. Yang, Online object tracking: a benchmark, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2411–2418.

[6] P. Liang, E. Blasch, H. Ling, Encoding color information for visual tracking: algorithms and benchmark, IEEE Trans. Image Process. 24 (12) (2015) 5630–5644.

[7] M. Mueller, N. Smith, B. Ghanem, A benchmark and simulator for uav tracking, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 445–461.

[8] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernandez, T. Vojir, G. Hager, G. Nebehay, R. Pflugfelder, The visual object tracking vot2015 challenge results, in: Proceedings of the IEEE Conference on Computer Vision workshops, 2015, pp. 1–23.

[9] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Čehovin, T. Vojir, G. Häger, A. Lukežič, G. Fernandez, The Visual Object Tracking vot2016 Challenge Results, Springer, 2016.

[10] M. Danelljan, G. Hager, F. Shahbaz Khan, M. Felsberg, Learning spatially regularized correlation filters for visual tracking, in: Proceedings of the IEEE Conference on Computer Vision, 2015, pp. 4310–4318.

[11] H. Kiani Galoogahi, T. Sim, S. Lucey, Correlation filters with limited boundaries, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4630–4638.

[12] H.K. Galoogahi, A. Fagg, S. Lucey, Learning background-aware correlation filters for visual tracking, in: Proceedings of the IEEE Conference on Computer Vision, 2017, pp. 1144–1152.

[13] Y. Wu, J. Lim, M.-H. Yang, Object tracking benchmark, IEEE Trans. Pattern Anal. Mach. Intell. 37 (9) (2015) 1834–1848.

[14] D. Comaniciu, V. Ramesh, P. Meer, Real-time tracking of non-rigid objects using mean shift, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2000, pp. 142–149.

[15] D.A. Ross, J. Lim, R.S. Lin, M.H. Yang, Incremental learning for robust visual tracking, Int. J. Comput. Vision 77 (1–3) (2008) 125–141.

[16] B. Babenko, M.-H. Yang, S. Belongie, Robust object tracking with online multiple instance learning, IEEE Trans. Pattern Anal. Mach. Intell. (2011) 1619–1632.

[17] X. Mei, H. Ling, Robust visual tracking using l1 minimization, in: Proceedings of the IEEE Conference on Computer Vision, 2009, pp. 1436–1443.

[18] D.S. Bolme, J.R. Beveridge, B.A. Draper, Y.M. Lui, Visual object tracking using adaptive correlation filters, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 2544–2550.

[19] M. Danelljan, F. Shahbaz Khan, M. Felsberg, J. Van de Weijer, Adaptive color attributes for real-time visual tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1090–1097.

[20] M. Danelljan, G. Häger, F.S. Khan, M. Felsberg, Discriminative scale space tracking, IEEE Trans. Pattern Anal. Mach. Intell. 39 (8) (2017) 1561–1575.

[21] J.F. Henriques, R. Caseiro, P. Martins, J. Batista, High-speed tracking with kernelized correlation filters, IEEE Trans. Pattern Anal. Mach. Intell. 37 (3) (2015) 583–596.

[22] C. Ma, X. Yang, C. Zhang, M.-H. Yang, Long-term correlation tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5388–5396.

[23] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, P.H. Torr, Staple: Complementary learners for real-time tracking, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2016, pp. 1401–1409.

[24] A. Bibi, M. Mueller, B. Ghanem, Target response adaptation for correlation filter tracking, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 419–433.

[25] M. Danelljan, A. Robinson, F.S. Khan, M. Felsberg, Beyond correlation filters: learning continuous convolution operators for visual tracking, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 472–488.

[26] J. Van De Weijer, C. Schmid, J. Verbeek, D. Larlus, Learning color names for real-world applications, IEEE Trans. Image Process. 18 (7) (2009) 1512–1523.

[27] J.F. Henriques, R. Caseiro, P. Martins, J. Batista, Exploiting the circulant structure of tracking-by-detection with kernels, in: Proceedings of the European Conference on Computer Vision, 2012, pp. 702–715.

[28] J. Nocedal, S.J. Wright, Numerical Optimization, second ed., Springer, New York, NY, USA, 2006.

[29] K. Chatfield, K. Simonyan, A. Vedaldi, A. Zisserman, Return of the devil in the details: delving deep into convolutional nets, in: British Machine Vision Conference, 2014.

[30] A. Vedaldi, K. Lenc, Matconvnet: Convolutional neural networks for matlab, in: Proceedings of the 23rd ACM International Conference on Multimedia, 2015, pp. 689–692.

[31] A. Lukezic, T. Vojir, L.C. Zajc, J. Matas, M. Kristan, Discriminative correlation filter with channel and spatial reliability, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4847–4856.

[32] Y. Li, J. Zhu, A scale adaptive kernel correlation filter tracker with feature integration, in: Proceedings of the European Conference on Computer Vision, 2014, pp. 254–265.

[33] M. Danelljan, G. Hager, F. Shahbaz Khan, M. Felsberg, Convolutional features for correlation filter based visual tracking, in: Proceedings of the IEEE Conference on Computer Vision Workshops, 2015, pp. 58–66.

[34] Y. Song, C. Ma, L. Gong, J. Zhang, R.W. Lau, M.H. Yang, Crest: Convolutional residual learning for visual tracking, in: Proceedings of the IEEE Conference on Computer Vision, 2017, pp. 2574–2583.

[35] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, P.H. Torr, End-to-end representation learning for correlation filter based tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5000–5008.

[36] L. Bertinetto, J. Valmadre, J.F. Henriques, A. Vedaldi, P.H. Torr, Fully-convolutional siamese networks for object tracking, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 850–865.

[37] R. Tao, E. Gavves, A.W. Smeulders, Siamese instance search for tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1420–1429.

[38] C. Ma, J.-B. Huang, X. Yang, M.-H. Yang, Hierarchical convolutional features for visual tracking, in: Proceedings of the IEEE Conference on Computer Vision, 2015, pp. 3074–3082.

[39] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, M.-H. Yang, Hedged deep tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4303–4311.

[40] T. Zhang, C. Xu, M.-H. Yang, Multi-task correlation particle filter for robust object tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4819–4827.