Integrating Boundary and Center Correlation Filters for Visual Tracking with Aspect Ratio Variation

Feng Li[1], Yingjie Yao[1], Peihua Li[2], David Zhang[3], Wangmeng Zuo [*,1], and Ming-Hsuan Yang[4]

[1]School of Computer Science and Technology, Harbin Institute of Technology, China
[2]School of Information and Communication Engineering, Dalian University of Technology, China
[3]Department of Computing, The Hong Kong Polytechnic University, China
[4]School of Engineering, University of California, Merced, USA

fengli_hit@hotmail.com, yyjhit@sina.com, peihuali@dlut.edu.cn
csdzhang@comp.polyu.edu.hk, wmzuo@hit.edu.cn, mhyang@ucmerced.edu

Abstract

The aspect ratio variation frequently appears in visual tracking and has a severe influence on performance. Although many correlation filter (CF)-based trackers have also been suggested for scale adaptive tracking, few studies have been given to handle the aspect ratio variation for CF trackers. In this paper, we make the first attempt to address this issue by introducing a family of 1D boundary CFs to localize the left, right, top, and bottom boundaries in videos. This allows us cope with the aspect ratio variation flexibly during tracking. Specifically, we present a novel tracking model to integrate 1D Boundary and 2D Center CFs (IBCCF) where boundary and center filters are enforced by a near-orthogonality regularization term. To optimize our IBCCF model, we develop an alternating direction method of multipliers. Experiments on several datasets show that IBCCF can effectively handle aspect ratio variation, and achieves state-of-the-art performance in terms of accuracy and robustness. The source code of our tracker is available at https://github.com/lifeng9472/IBCCF/.

1. Introduction

Visual tracking is one of the most fundamental problems in computer vision. And it plays a critical role in versatile applications such as video surveillance, intelligent transportation, and human-computer interaction [36, 31, 35, 39, 5]. Given its annotation on the initial frame, visual tracking aims to estimate the trajectory of a target with large appearance variations caused by many factors, *e.g*., scale, rotation, occlusion, and background clutter. Although great advance
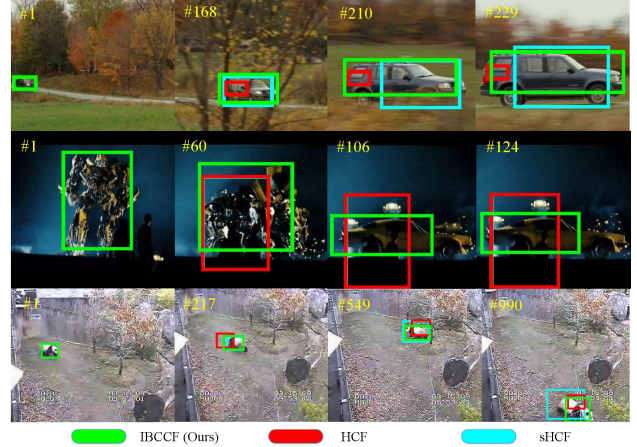
---
[*]Corresponding author.

Figure 1: Three example sequences (*CarScale*, *Trans* and *Panda*) with aspect ratios variation in OTB benchmark. Given the initial target position in the first column, we compare our method (IBCCF) with its counterparts HCF [23] and sHCF. The results show our IBCCF is superior to HCF and sHCF in the case of aspect ratio variation.

has been made, it remains a challenging issue to develop an accurate and robust tracker to handle all these factors. Recently, correlation filter (CF)-based approaches have received considerable attention in visual tracking [3, 14, 7]. In these methods, a discriminative correlation filter is trained to generate 2D Gaussian shaped responses centered at target position. Benefited from the circulant matrix and fast Fourier transform (FFT), the CF-based trackers usually perform very efficiently. With the introduction of deep features [23, 29] and spatial regularization [9, 22] and continuous convolution [11], the performance of CF-based trackers have been persistently improved, and lead to the state-of-the-art performance.

Despite the advances in CF-based tracking, the aspect ratio variation remains an open problem. The changes in aspect ratio can be caused by variations in-plane/out-of-plane rotation, deformation, occlusion, or scale variation, and usually have a severe effect on tracking performance. To handle scale variation, Li and Zhu [17] propose a Scale Adaptive with Multiple Features tracker (SAMF). Danelljan et al. [7] suggest a discriminative scale space tracking (DSST) method to learn separate CFs for translation and scale estimation on scale pyramid representation. Besides, scale variation issue can also be handled by the part-based CF trackers [21, 18, 20]. These methods, however, can only cope with scale variation issue but cannot well address the the aspect ratio variation issue. Fig. 1 illustrates the tracking results on three sequences with aspect ratio variation. It clearly shows that, even with deep CNN features, neither the standard CF (*i.e.* HCF [23]) tracker nor its scale-adaptive version (sHCF) can address the issue of aspect ratio variation caused by in-plane/out-of-plane rotation and deformation.

In this paper, we present a visual tracking model to handle aspect ratio variation by integrating boundary and center correlation filters (IBCCF). The standard CF estimates the trajectory by finding the highest response in each frame to locate the center of the target, and can be seen as a center tracker. In contrast, we introduce a family of 1D boundary CFs to localize the positions of the left, right, top and bottom boundaries (*i.e.* m_l, m_r, n_t, n_b) in the sequences, respectively. By treating 2D boundary region as a multi-channel representation of 1D vectors, a boundary CF is learned to generate 1D Gaussian shaped responses centered at target boundary (See Fig. 2). By using boundary CFs, the left, right, top and bottom boundaries can be flexibly tuned in the image sequences. Thus the aspect ratio variation can be naturally handled during tracking.

We empirically analyze and reveal the near-orthogonality property between the center and boundary CFs. Then, by enforcing the orthogonality with an additional regularization term (*i.e.* near-orthogonality constraint), we present a novel IBCCF tracking model to integrate 1D boundary and 2D center CFs. Meanwhile, an alternating direction method of multipliers (ADMM) [4] is then developed to optimize the proposed IBCCF model.

To evaluate our IBCCF, extensive experiments have been conducted on OTB-2013, OTB-2015 [35] , Temple-Color [19], VOT-2016 [5] and VOT-2017 datasets. The results validate the effectiveness of IBCCF on handling aspect ratio variation. Compared with several state-of-the-art trackers, our IBCCF achieves comparable performance in terms of accuracy and robustness. As shown in Fig. 1, by using CNN features, our IBCCF can well adapt to the aspect ratio variation in the three sequences, yielding better tracking performance.

To sum up, the contributions of this paper are three-fold:

- A novel IBCCF model is developed to address the aspect ratio variation in CF-based trackers. To achieve this, we first introduce a family of boundary CFs to track the left, right, top and bottom boundaries besides tracking the target center. Then, we combines boundary and center CFs by encouraging orthogonality between them for accurate tracking.

- An ADMM algorithm is suggested to optimize our IBCCF model, where each subproblem has the closed-form solution. Our algorithm alternates between updating center CFs and updating boundary CFs, and empirically converges with very few iterations.

- The extensive experimental results demonstrate the effectiveness of our proposed IBCCF, and it achieves comparable tracking performance against several state-of-the-art trackers.

2. Related Work

In this section, we provide a brief survey on CF-based trackers, and discuss several scale adaptive and part-based CF trackers close to our method.

2.1. Correlation Filter Trackers

Denote by $\mathbf{x}$ an image patch of $M \times N$ pixels. Let $\mathbf{y}$ be a 2D Gaussian shaped labels. The correlation filter $\mathbf{w}$ is then learned by minimizing the ridge regression objective:

$$\mathbf{w} = \arg \min_{\mathbf{w}} \left\{ \mathcal{E}(\mathbf{w}) = \|\mathbf{w} \otimes_2 \mathbf{x} - \mathbf{y}\|^2 + \lambda \|\mathbf{w}\|^2 \right\}, \quad (1)$$

where λ denotes the regularization parameter, and $\otimes_2$ is the 2D convolution operator. Denote by $\hat{\mathbf{x}}$ the Fourier transform of $\mathbf{x}$, and $\hat{\mathbf{x}}^*$ the complex conjugate of $\hat{\mathbf{x}}$. Using fast Fourier transform (FFT), the closed-form solution to Eqn. (1) can be given as:

$$\mathbf{w} = \mathcal{F}^{-1} \left(\frac{\hat{\mathbf{x}}^* \odot \hat{\mathbf{y}}}{\hat{\mathbf{x}}^* \odot \hat{\mathbf{x}} + \lambda} \right), \quad (2)$$

where $\odot$ denotes the element-wise multiplication, and $\mathcal{F}^{-1}(\cdot)$ represents the inverse Discrete Fourier transform operator.

From the pioneering MOSSE by Bolme et al. [3], great advances have been made in CF-based tracking. Henriques et al. [14] extend MOSSE to learn nonlinear CF via kernel trick. And the multi-channel extension of CF has been studied in [15]. Driven by feature engineering, HOG [6], color names [10] and deep CNN features [23, 29] have been successively adopted in CF-based tracking. Other issues, such as long-term tracking [24], continuous convolution [11], spatial regularization [9, 22], and boundary effect [16], are also investigated to improve tracking accuracy and robustness. Besides ridge regression, other learning models, e.g.,

support vector machine (SVM) [40, 27] and sparse coding [32], are also introduced. Due to the page limits, in the following, we further review the scale adaptive and part-based CFs which are close to our work.

2.2. Scale Adaptive and Part-based CF Trackers

The first family of methods close to our approach are scale adaptive CF trackers, which aim to estimate target scale changes during tracking. Among the current scale adaptive CF trackers [9, 11, 24], SAMF [17] and DSST [7] are two commonly used methods for scale estimation. They apply the learned filter to samples of multi-resolutions around the target, and compute the response for each scale of the sample whose the maximum response is seen as the optimal scale. However, such strategy is time consuming in the case of large scale space, and many improvements over them have been proposed. Tang and Feng [33] employ bisection search and fast feature scaling method to speed up scale space searching. Bibi and Ghanem [2] maximize the posterior probability rather than the likelihood (*i.e.* maximum response map) in different scales for more stable detections. Additionally, Zhang et al. [38] also suggest a robust scale estimation method by averaging the scales over n consecutive frames. Despite these successes in isometric scale variation, such kind of methods cannot well address aspect ratio variation. Different from the aforementioned methods, the proposed IBCCF approach can handle aspect ratio variation effectively with the introduction of boundary CFs.

Our IBCCF also shares some philosophy with the part-based CF methods, which divide the entire target into several parts and merge the results from all parts for final prediction. For example, Liu et al. [21] divide the target into five parts which are assigned with five independent CF trackers, and the final target position estimation is obtained by merging five CF trackers using Bayesian inference methods. Different from simple dividing parts, Li et al. [18] propose to exploit reliable parts, which estimate their probability distributions under a sequential Monte Carlo framework and employ a Hough voting scheme to locate the target. In the similar line, Liu et al. [20] propose to jointly learn multiple parts from the target with CF trackers in an ADMM framework. Compared with the part-based trackers, the proposed IBCCF has several merits: (1) IBCCF chooses to track meaningful boundary regions, which is more general than the fixed partition based method [21] and easier to be handled than the learned parts based method [20]; (2) With the introduction of 1D boundary CFs, IBCCF can naturally deal with the aspect ratio variation problem; (3) The near-orthogonality constraint between boundary and center CFs encourages IBCCF better performance than part-based ones.
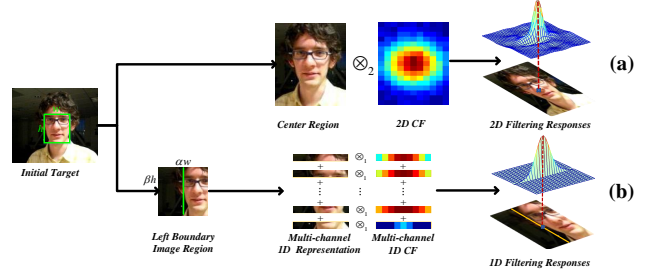

Figure 2: Comparison between standard CF and the proposed 1D boundary CF. Note that $\otimes_1$ and $\otimes_2$ represent the 1D and 2D convolution, respectively. (a): Standard CF convolves the center region with a 2D CF to generate 2D responses for finding the center position. (b): 1D boundary CF crops the boundary region centered at target boundary, then 2D boundary region is reshaped as a multi-channel representation of 1D vectors and convolved with 1D CF to produce 1D responses for finding the boundary position.

3. The Proposed IBCCF Model

In this section, we first introduce the boundary correlation filters. Then, we investigate the near-orthogonality property between the boundary and center CFs, and finally present our IBCCF model.

3.1. Boundary Correlation Filters

In standard CF, the bounding box of a target with fixed size is uniquely characterized by its center (m_c, n_c). By incorporating with scale estimation, the target bounding box can be determined by both the center (m_c, n_c) and the scale factor α. However, both standard and scale adaptive CFs cannot address the aspect ratio variation issue, so better description of bounding box is required. For CNN-based object detection [12], the bounding box is generally parameterized by center coordinate, its height and width. Although such parameterization scheme can cope with aspect ratio variation, it is difficult to predict target height and width in the CF framework.

In this work, the bounding box is parameterized with its left, right, top and bottom boundaries $\mathcal{B} = \{m_l, m_r, n_t, n_b\}$. It is natural to see that such parameterization is able to handle aspect ratio variation with dynamically adjusting four boundaries of target. Moreover, for each boundary of $\mathcal{B}$, a 1D boundary CF (BCF) is learned to estimate the left, right, top or bottom boundary, respectively. Taking the left boundary m_l as an example, Fig. 2(b) illustrates the process of 1D boundary CF. Given a target bounding box, let (m_c, n_c) be the center, h and w be the height and width. Its left boundary can be represented as $m_l = m_c - \frac{w}{2}$. Then we crop a left boundary image region $\mathbf{x}_l$ centered at (m_l, n_c) with width $w_l = \alpha w$ and height $h_l = \beta h$.

To learn 1D boundary CF, the left boundary image region is treated as a multi-channel (*i.e.* h_l) representation of 1D $w_l \times 1$ vectors $\mathbf{x}_l = [\mathbf{x}_l^1; ...; \mathbf{x}_l^{h_l}]$. Denote by $\mathbf{y}_l$ a 1D

Gaussian shaped labels centered at m_l. Then the 1D left boundary CF model can then be formulated as,

$$\mathbf{w}_l = \arg\min_{\mathbf{w}_l} \mathcal{L}_l(\mathbf{w}_l) = \left\| \sum_{j=1}^{h_l} \mathbf{x}_l^j \otimes_1 \mathbf{w}_l^j - \mathbf{y}_l \right\|^2 + \lambda \|\mathbf{w}_l\|^2 \quad (3)$$

where $\otimes_1$ denotes the 1D convolution operator. For each channel of $\mathbf{w}_l$, its closed form solution $\mathbf{w}_l^j$ can be obtained by,

$$\mathbf{w}_l^j = \mathcal{F}^{-1}\left(\frac{\hat{\mathbf{x}}_l^{j*} \odot \hat{\mathbf{y}}_l}{\sum_{j=1}^{h_l} \hat{\mathbf{x}}_l^{j*} \odot \hat{\mathbf{x}}_l^{j*} + \lambda} \right). \quad (4)$$

As shown in Fig. 2(a), the center region is convolved with a 2D CF to generate a 2D filtering responses. Then, the target center is determined by the position with the maximum response. Thus standard CF can be seen as a center CF (CCF) tracker. In contrast, as shown in Fig. 2(b), the left boundary region is first equivalently written as a multi-channel representation of 1D vectors. The multi-channel 1D vectors are then convolved with multi-channel 1D correlation filters to produce 1D filtering responses. And the left boundary is determined by finding the position with the maximum response. Analogously, the other boundaries can also be obtained to track with the right, top and bottom boundary CFs, respectively. Fig. 3 shows the setting of the boundary regions based on the target bounding box.

When a new frame comes, we first crop the boundary regions, which are convolved with the corresponding boundary CFs. The left, right, top and bottom boundaries are then determined based on the corresponding 1D filtering responses. Note that each boundary is estimated independently. Thus, our BCF approach can adaptively fit target scale and aspect ratio.

3.2. Near-orthogonality between Boundary and Center CFs

It is natural to see that the boundary and center CFs are complementary and can be integrated to boost tracking performance. To locate target in the current frame, we can first detect an initial position with CCF, then BCFs are employed to further refine the boundaries and position. To update the tracker, we empirically investigate the relationship between the CCF and BCFs, and then suggest to include a near-orthogonality regularizer for better integration.

Suppose that the size of left boundary region for BCF is the same with that of center region for CCF. Without loss of generality, we let $[\mathbf{x}]_{\text{vec}}$, $[\mathbf{w}]_{\text{vec}}$, and $[\mathbf{w}_l]_{\text{vec}}$ be the vectorization of the center region, center CF, and left boundary CF, respectively. On one hand, the filtering responses of left boundary CF should be higher at the left boundary and near zero otherwise. So we have that $[\mathbf{w}_l]_{\text{vec}}^T [\mathbf{x}]_{\text{vec}} \approx 0$, which indicates that $[\mathbf{w}_l]_{\text{vec}}$ and $[\mathbf{x}]_{\text{vec}}$ are nearly orthogonal. On the other hand, the filtering responses of center CF achieve
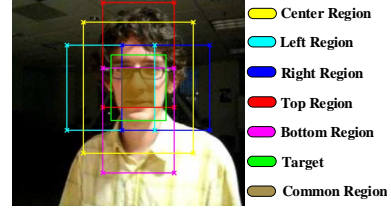

Figure 3: An illustration of the generated center, boundary and common region based on the target bounding box.
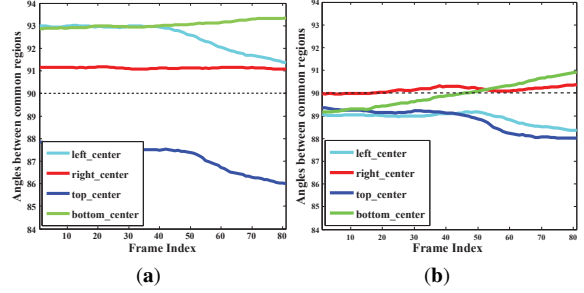

Figure 4: The angles (in degree) between the common regions of center and boundary CFs on the sequence *Skiing*. (a) The angles computed by training CCF and BCFs independently. (b) Using the near-orthogonality constraint, the angles obtained by IBCCF can be more near to $90°$, and 4.7% gains by overlap precision is attained during tracking.

its maximum at the center position, and thus the angle between $[\mathbf{x}]_{\text{vec}}$ and $[\mathbf{w}]_{\text{vec}}$ should be small. Therefore, $[\mathbf{w}_l]_{\text{vec}}$ should be nearly orthogonal with $[\mathbf{w}]_{\text{vec}}$.

However, in general the sizes of left boundary region and center region are not the same. From Fig. 3, one can see that they share a common region. Let $[\tilde{\mathbf{w}}]_{\text{vec}}$ be the vectorization of center CF in the common region, and so do $[\tilde{\mathbf{x}}]_{\text{vec}}$ and $[\tilde{\mathbf{w}}_l]_{\text{vec}}$. We then extend the near-orthogonality property to the common region, and expect that $[\tilde{\mathbf{w}}_l]_{\text{vec}}$ and $[\tilde{\mathbf{w}}]_{\text{vec}}$ are also nearly orthogonal, *i.e.*, $[\tilde{\mathbf{w}}_l]_{\text{vec}}^T [\tilde{\mathbf{w}}]_{\text{vec}} \approx 0$. Analogously, we also expect that $[\tilde{\mathbf{w}}_r]_{\text{vec}}^T [\tilde{\mathbf{w}}]_{\text{vec}} \approx 0$, $[\tilde{\mathbf{w}}_t]_{\text{vec}}^T [\tilde{\mathbf{w}}]_{\text{vec}} \approx 0$, $[\tilde{\mathbf{w}}_b]_{\text{vec}}^T [\tilde{\mathbf{w}}]_{\text{vec}} \approx 0$. Fig. 4 shows the angles between the center CF and boundary CFs in common region on the sequence *Skiing*. From Fig. 4(a), one can note that it is roughly hold true that the boundary and center CFs are nearly orthogonal. Thus, as illustrated in Fig. 4(b), we expect better near-orthogonality and tracking accuracy can be attained by imposing the near-orthogonality constraint on the training of CCF and BCFs. Empirically, for *Skiing*, the introduction of near-orthogonality constraint does bring 4.7% gains by overlap precision during tracking.

3.3. Problem Formulation of IBCCF

By enforcing the near-orthogonality constraint, we propose our IBCCF model to integrate boundary and center CFs, resulting in the following objective,

$$\arg\min_{\mathbf{W}} \sum_{k \in \Psi} \mathcal{L}_k(\mathbf{w}_k) + \mathcal{E}(\mathbf{w}) + \mu \sum_{k \in \Psi} \left([\tilde{\mathbf{w}}]_{\text{vec}}^T [\tilde{\mathbf{w}}_k]_{\text{vec}} \right)^2 \quad (5)$$

where $\Psi = \{l, r, t, b\}$, and $\mathbf{W} = \{\mathbf{w}, \mathbf{w}_l, \mathbf{w}_r, \mathbf{w}_t, \mathbf{w}_b\}$. $\mathcal{L}_k(\mathbf{w}_k)$ and $\mathcal{E}(\mathbf{w})$ are defined in Eqns. (3) and (1).

Comparison with DSST [7]. Even DSST also adopts 1D and 2D CFs, it learns separate 2D and 1D CFs for translation and scale estimation on scale pyramid representation, respectively. And our IBCCF is distinctly different with DSST from three aspects: (i) While DSST formulates scale estimation as 1D CF, BCF is among the first to suggest a novel parameterization of bounding box, and formulates boundary localization as 1D CFs. (ii) For DSST, the inputs to 1D CF are image patches at different scales, while the input for BCF is four regions covering the edges of bounding boxes. (iii) In DSST, the 1D CF and 2D translation CF are separately trained. While in IBCCF, 1D BCFs and 2D CCF are jointly learned by solving the IBCCF model in Eqn. (5).

4. Optimization

In this section, we propose an ADMM method to minimize Eqn. (5) by alternately updating center CF $\mathbf{w}$ and boundary CFs $\mathbf{w}_k$ ($k \in \Psi$), where each subproblem can be easily solved with a close-form solution.

We first employ variable splitting method to change Eqn. (5) into a linear equality constrained optimization problem:

$$\arg\min_{\mathbf{W}} \sum_{k \in \Psi} \mathcal{L}_k(\mathbf{w}_k) + \mathcal{E}(\mathbf{w}) + \mu \sum_{k \in \Psi} \left([\tilde{\mathbf{w}}]_{\text{vec}}^T [\tilde{\mathbf{w}}_k]_{\text{vec}}\right)^2 \quad (6)$$

$$\text{s.t.} \quad \mathbf{g} = \mathbf{w}, \mathbf{u}_k = \mathbf{w}_k, k \in \Psi.$$

Hence, the Augmented Lagrangian Method (ALM) can be applied to solve Eqn. (6), and its augmented lagrangian form [4] is reformulated as:

$$\arg\min_{\mathbf{H}} \sum_{k \in \Psi} \mathcal{L}_k(\mathbf{w}_k) + \mathcal{E}(\mathbf{w}) + \mu \sum_{k \in \Psi} \left([\tilde{\mathbf{g}}]_{\text{vec}}^T [\tilde{\mathbf{u}}_k]_{\text{vec}}\right)^2 \quad (7)$$

$$+ \rho \|\mathbf{w} - \mathbf{g} - \mathbf{p}\|^2 + \sum_{k \in \Psi} \gamma_k \|\mathbf{w}_k - \mathbf{u}_k - \mathbf{q}_k\|^2,$$

where $\mathbf{H} = \{\mathbf{W}, \mathbf{g}, \mathbf{p}, \mathbf{u}_k, \mathbf{q}_k\}$, $\mathbf{p}$ and $\mathbf{q}_k$ represent the Lagrange multiplier, ρ and γ_k are penalty factors, respectively. For multivariable non-convex optimization, ADMM iteratively updates one of variables while keeping the rest fixed, hence the convergence can be guaranteed [4]. By using ADMM, Eqn. (7) is divided into the following subproblems:

$$\begin{cases}
\mathbf{w}^{(i+1)} = \arg\min_{\mathbf{w}} \mathcal{E}(\mathbf{w}) + \rho^{(i)} \left\|\mathbf{w} - \mathbf{g}^{(i)} - \mathbf{p}^{(i)}\right\|^2 \\
\mathbf{g}^{(i+1)} = \arg\min_{\mathbf{g}} \mu \left\|(\mathbf{S}^{(i)})^T [\tilde{\mathbf{g}}]_{\text{vec}}\right\|^2 \\
\qquad\qquad + \rho^{(i)} \left\|\mathbf{w}^{(i)} - \mathbf{g} - \mathbf{p}^{(i)}\right\|^2 \\
\mathbf{w}_k^{(i+1)} = \arg\min_{\mathbf{w}_k} \mathcal{L}(\mathbf{w}_k) + \gamma_k^{(i)} \left\|\mathbf{w}_k - \mathbf{u}_k^{(i)} - \mathbf{q}_k^{(i)}\right\|^2 \\
\mathbf{u}_k^{(i+1)} = \arg\min_{\mathbf{u}_k} \mu \left\|[\tilde{\mathbf{g}}^{(i)}]_{\text{vec}}^T [\tilde{\mathbf{u}}_k]_{\text{vec}}\right\|^2 \\
\qquad\qquad + \gamma_k^{(i)} \left\|\mathbf{w}_k^{(i)} - \mathbf{u}_k - \mathbf{q}_k^{(i)}\right\|^2 \\
\mathbf{p}^{(i+1)} = \mathbf{p}^{(i)} + \mathbf{g}^{(i+1)} - \mathbf{w}^{(i+1)} \\
\mathbf{q}_k^{(i+1)} = \mathbf{q}_k^{(i)} + \mathbf{u}_k^{(i+1)} - \mathbf{w}_k^{(i+1)}
\end{cases}$$

$$(8)$$

where $\mathbf{S}^{(i)} = [[\tilde{\mathbf{u}}_l^{(i)}]_{\text{vec}}, [\tilde{\mathbf{u}}_r^{(i)}]_{\text{vec}}, [\tilde{\mathbf{u}}_t^{(i)}]_{\text{vec}}, [\tilde{\mathbf{u}}_b^{(i)}]_{\text{vec}}]$ and $k \in \Psi$. From Eqn. (5), we can see that the boundary CFs are independent of each other, so each pair of $\mathbf{w}_k$ and $\mathbf{u}_k$ can be updated in parallel for efficiency. Next, we detail the solution to each subproblem as follows:

Subproblem w. Using the properties of circulant matrix and FFT, the closed form solution of $\mathbf{w}$ is given as:

$$\mathbf{w} = \mathcal{F}^{-1} \left(\frac{\hat{\mathbf{x}}^* \odot \hat{\mathbf{y}} + \hat{\mathbf{g}} + \hat{\mathbf{p}}}{\hat{\mathbf{x}}^* \odot \hat{\mathbf{x}} + \lambda + \rho} \right) \quad (9)$$

Subproblem g. The second row of Eqn. (8) is rewritten as:

$$\arg\min_{[\mathbf{g}]_{\text{vec}}} \mu \left\|\mathbf{Q}^T [\mathbf{g}]_{\text{vec}}\right\|^2 + \rho \|[\mathbf{w}]_{\text{vec}} - [\mathbf{g}]_{\text{vec}} - [\mathbf{p}]_{\text{vec}}\|^2 \quad (10)$$

where the matrix $\mathbf{Q} = \begin{bmatrix} \mathbf{S} \\ \mathbf{0} \end{bmatrix}$ is obtained by padding zeros to each column of $\mathbf{S}$. Then $[\mathbf{g}]_{\text{vec}}$ can be computed by:

$$[\mathbf{g}]_{\text{vec}} = \left(\mu \mathbf{Q}\mathbf{Q}^T + \rho I\right)^{-1} \rho \left([\mathbf{w}]_{\text{vec}} - [\mathbf{p}]_{\text{vec}}\right) \quad (11)$$

Note that matrix $\mathbf{Q}$ only contains four columns, thus Singular Value Decomposition (SVD) can be used for improving the efficiency. By performing SVD of $\mathbf{Q}$ with $\mathbf{Q} = \mathbf{U}\Sigma\mathbf{V}^T$, we have:

$$[\mathbf{g}]_{\text{vec}} = \mathbf{U}\Lambda\mathbf{U}^T \left([\mathbf{w}]_{\text{vec}} - [\mathbf{p}]_{\text{vec}}\right) \quad (12)$$

where $\Lambda = \left(\frac{\mu}{\rho}\Sigma\Sigma^T + \mathbf{I}\right)^{-1}$. Let the nonzero elements in matrix Σ be $[\lambda_1, \lambda_2, \lambda_3, \lambda_4]$, the nonzero elements of diagonal matrix Λ become $[\frac{\rho}{\mu\lambda_1^2+\rho}, \frac{\rho}{\mu\lambda_2^2+\rho}, \frac{\rho}{\mu\lambda_3^2+\rho}, \frac{\rho}{\mu\lambda_4^2+\rho}, ..., 1]$. Hence, Eqn. (12) can be written as:

$$[\mathbf{g}]_{\text{vec}} = \left(\mathbf{I} - \mathbf{U}(\mathbf{I} - \Lambda)\mathbf{U}^T\right) \left([\mathbf{w}]_{\text{vec}} - [\mathbf{p}]_{\text{vec}}\right) \quad (13)$$

Since diagonal matrix $(\mathbf{I} - \Lambda)$ only contains four nonzero elements, we have $\mathbf{U}(\mathbf{I} - \Lambda)\mathbf{U}^T = \hat{\mathbf{U}}\text{Diag}(f)\hat{\mathbf{U}}^T$, where $\hat{\mathbf{U}}$ is the first four columns of matrix $\mathbf{U}$ and $\text{Diag}(f)$ denotes the diagonal matrix of the nonzero elements in $(\mathbf{I} - \Lambda)$. Such special case can be solved efficiently [1].

Subproblem $\mathbf{w}_k$. The solution of $\mathbf{w}_k$ shares similar solution with one of Eqn. (9):

$$\mathbf{w}_k = \mathcal{F}^{-1} \left(\frac{\hat{\mathbf{x}}_k^* \odot \hat{\mathbf{y}}_k + \hat{\mathbf{u}}_k + \hat{\mathbf{q}}_k}{\hat{\mathbf{x}}_k^* \odot \hat{\mathbf{x}}_k + \lambda + \gamma_k} \right) \quad (14)$$

Subproblem $\mathbf{u}_k$. The fourth row of Eqn. (8) is written as:

$$[\mathbf{u}_k]_{\text{vec}} = \arg\min_{[\mathbf{u}_k]_{\text{vec}}} \mu \left\|[\mathbf{s}]_{\text{vec}}^T [\mathbf{u}_k]_{\text{vec}}\right\|^2 \qquad (15)$$
$$+ \gamma_k \|[\mathbf{w}_k]_{\text{vec}} - [\mathbf{u}_k]_{\text{vec}} - [\mathbf{q}_k]_{\text{vec}}\|^2$$

where $[\mathbf{s}]_{\text{vec}} = \begin{bmatrix} [\tilde{\mathbf{g}}]_{\text{vec}} \\ \mathbf{0} \end{bmatrix}$ and the close-form solution of $[\mathbf{u}_k]_{\text{vec}}$ is:

$$[\mathbf{u}_k]_{\text{vec}} = (\mu[\mathbf{s}]_{\text{vec}}[\mathbf{s}]_{\text{vec}}^T + \gamma_k\mathbf{I})^{-1}\gamma_k([\mathbf{w}_k]_{\text{vec}} - [\mathbf{q}_k]_{\text{vec}}) \quad (16)$$

Since $[\mathbf{s}]_{\text{vec}}[\mathbf{s}]_{\text{vec}}^T$ is rank-1 matrix, Eqn. (16) can be efficiently solved with Sherman-Morrsion formula [28], so we have:

$$[\mathbf{u}_k]_{\text{vec}} = (1 - \frac{\mu[\mathbf{s}]_{\text{vec}}[\mathbf{s}]_{\text{vec}}^T}{\gamma_k + \mu[\mathbf{s}]_{\text{vec}}^T[\mathbf{s}]_{\text{vec}}})([\mathbf{w}_k]_{\text{vec}} - [\mathbf{q}_k]_{\text{vec}}) \quad (17)$$

---

[1]Please refer to SVD function with "economy" mode in Matlab.

Convergence. To verify the effectiveness of the proposed ADMM, we illustrate the convergence curve with ADMM on sequence *Skiing*. As shown in Fig. 5, although IBCCF model is a non-convex problem, we can see that it converges within very few iterations (four iterations in this case). This phenomenon is ubiquitous in our experiments, and most of the sequences converges within five iterations.
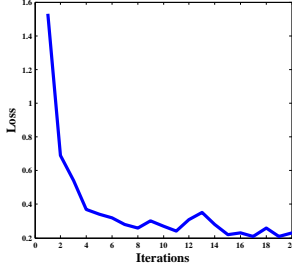

Figure 5: Convergence curve of the proposed IBCCF with ADMM on the fifth frame of sequence *Skiing* on OTB benchmark.

5. Experiments

In this section, we first compare IBCCF with state-of-the-art trackers on OTB dataset [35]. Then we validate the effects of each component on IBCCF, and analyze the time cost using OTB dataset. Finally, we conduct comparative experiments on Temple-Color [19] and VOT benchmarks.

Following the common settings in HCF [23], we implement IBCCF by using the outputs of layers *conv*3-4, *conv*4-4 and *conv*5-4 of VGG-Net-19 [30] for feature extraction. To combine the responses from different layers, we follow the HCF setting and assign the weights of three layers in the center CF to 0.02, 0.5 and 1, respectively. For boundary CFs, we omit the layer of *conv*3-4 and set the weights for layers of *conv*4-4 and *conv*5-4 both to 1. The regularization parameters λ and μ are set to 10^{-4} and 0.1, respectively. Note that we employ a subset of 40 sequences from Temple-Color dataset as the validation set to choose the above parameters. Detailed description about the subset and corresponding experiments are given in Section 5.3. Our approach is implemented with Matlab by using MatConvNet Library. The average running time is about 1.25fps on a PC equipping with a Intel Xeon(R) 3.3GHz CPU, 32GB RAM and NVIDIA GTX 1080 GPU.

5.1. OTB benchmark

OTB benchmark consists of two subsets, i.e., OTB-2013 and OTB-2015. OTB-2013 contains 51 sequences annotated with 11 different attributes, such as scale variation, occlusion and low resolution. OTB-2015 extends OTB-2013 to 100 videos. We quantitatively evaluate our method with One-Pass Evaluation (OPE) protocol, where overlap precision (OP) metrics is used by computing the fraction of frames with bounding box overlaps exceeding 0.5 in a se-

quence. Besides, we also provide overlap success plots containing the OP metrics over a range of thresholds.

5.1.1 Comparison with state-of-the art trackers

We compare our algorithm with 13 state-of-the-art methods: HCF [23], C-COT [11], SRDCF [9], DeepSRDCF [8], SINT+ [34], RPT [18], SCF [20], SAMF [17], Scale-DLSSVM [27], Staple [1], DSST [7], MEEM [37] and SKSCF [40]. Among them, most trackers except HCF and MEEM perform scale estimation during tracking. And both RPT and SCF methods exploit part-based models. In addition, for verifying the effectiveness of BCFs on handling aspect ratio variation, we implement a HCF variant with five scales (denoted by sHCF) under the SAMF framework. Note that we employ publicly available codes of compared trackers or copy the results from the original paper for fair comparison.

Table 1 lists a comparison of mean OP on OTB-2013 and OTB-2015 datasets. From it we can draw the following conclusions: (1) Our IBCCF outperforms most trackers except C-COT [11] and surpasses its counterpart HCF (i.e., only center CF) by 10.2% and 12.8% on OTB-2013 and OTB-2015, respectively. We owe these significant improvements to integration of boundary CFs. C-COT achieves higher mean OP than IBCCF on OTB-2015. It should be noted that spatial regularization is considered to suppress the boundary effect in both DeepSRDCF [8] and C-COT. Furthermore, C-COT also extends DeepSRDCF by learning multiple convolutional filters in continuous spatial domain. In contrast, our IBCCF does not consider the spatial regularization and continuous convolution, and can yield favorable performance against the competing trackers. (2) our IBCCF is consistently superior to sHCF and other scale estimation based methods (e.g., DeepSRDCF) on both datasets. It indicates our boundary CFs are more helpful than scale estimation to CF-based trackers. (3) Compared with part-based trackers (e.g., SCF), our IBCCF also shows its superiority, i.e., 4% gains over SCF on OTB-2013 dataset.
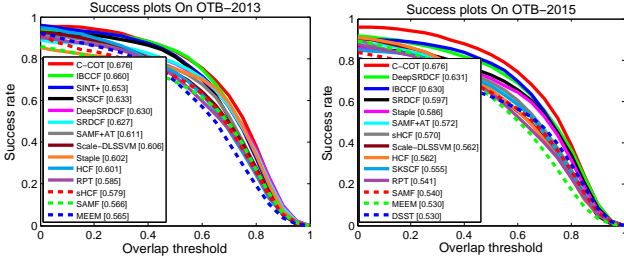
Next, we show the overlap success plots of different trackers, which are ranked using the Area-Under-the-Curve (AUC) score. As shown in Fig. 6, our IBCCF tracker is among the top three trackers on both datasets and outperforms HCF by 5.9% and 6.8% on OTB-2013 and OTB-2015 datasets, respectively.

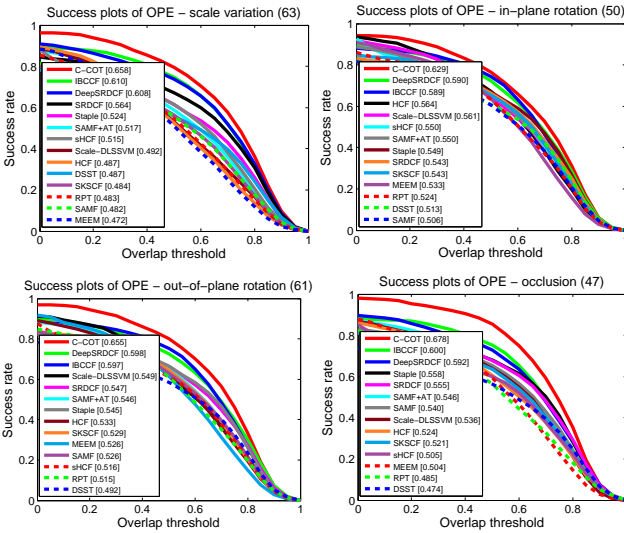#### 5.1.2 Video attributes related to aspect ratio variation

In this subsection, we perform analysis of attributes influencing aspect ratio variation on OTB-2015 dataset. Here we only provide the overlap success plots for four attributes which have great influence on aspect ratio variation and the rest results can be found in the supplementary material.

	C-COT [11]	SINT+ [34]	SKSCF [40]	Scale-DLSSVM [27]	Staple [1]	SRDCF [9]	DeepSRDCF [8]	RPT [18]	MEEM [37]	DSST [7]	SAMF [17]	HCF [23]	SCF [20]	sHCF	IBCCF
OTB-2013	83.4	81.3	80.9	73.2	74.9	78	79.2	71.4	69.1	67.7	68.9	73.5	79.7	73.4	83.7
OTB-2015	82.7	-	67.4	65.2	71.3	72.7	77.6	64	62.3	62.2	64.5	65.6	-	69.2	78.4

Table 1: Mean OP metrics (in %) of different trackers on OTB-2013 and OTB-2015. The best three results are shown in red, blue and green fonts, respectively.

Figure 6: Comparison of overlap success plots with state-of-the-art trackers on OTB-2013 and OTB-2015.

Figure 7: Overlap success plots of all competing trackers with four attributes great influencing aspect ratio variation on the OTB-2015.

	Mean OP	AUC Score
BCFs	50.2	39
Center CF	65.6	56.2
IBCCF (w/o constraints)	72	58.9
IBCCF	78.4	63

Table 2: Evaluation results of component experiments on both mean OP and AUC score metrics (in %).

Occlusion: Obviously, the partial occlusions can lead to aspect ratio variation of target. And complete occlusions also have an adverse impact on the boundary prediction. Despite of these negative effects, IBCCF still outperforms most of the competing trackers and brings 7.6% gains over the center CF tracker (*i.e.* HCF).

5.2. Internal Analysis of the proposed approach

5.2.1 Impacts of Boundary CFs and near-orthogonality

Here we investigate the impact of boundary CFs and near-orthogonality property on the proposed IBCCF approach. To achieve this, we make four different variants of IBCCF: the tracker only with 1D boundary CFs (BCFs), the tracker only with the center CF (i.e., HCF), the IBCCF tracker without orthogonality constraints denoted by IBCCF (w/o constraint) and full IBCCF model. Table 2 summarizes the mean OP and AUC score of four methods on OTB-2015.

From Table 2, one can see that both BCFs tracker and orthogonality constraint are key parts of the proposed IBCCF method, and they can bring significant improvements over the center CF tracker. Detailed analysis on the results can be found in the supplementary material.

5.2.2 Time Analysis

Here we analyze the average time cost of IBCCF for each stage on OTB-2015 dataset. The results are shown in Table 3. One can clearly see that all subproblems including $\mathbf{g}$ and $\mathbf{u_k}$ can be solved rapidly, validating the efficiency of the ADMM solution. Overall, the average running time of IBCCF and IBCCF (w/o constraint) is about 1.25 and 2.19 fps on OTB-2015 dataset, respectively.
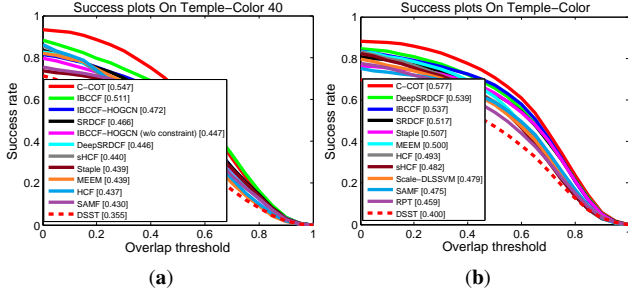
5.3. Temple-Color dataset

In this section, we perform comparative experiments on Temple-Color dataset which contains 128 color sequences. Different from the OTB dataset, it contain more video sequences with aspect ratio changes. Hence, to better exploit the potential of IBCCF, we also choose a subset of 40 sequences with the largest standard deviations of aspect ratio

Scale Variations: In the case of scale variations, target size continuously changes during tracking. It is worth noting that most of videos in this attribute only undergo scale changes rather than aspect ratio variation. Despite in such setting, as illustrated in the upper left of Fig. 7, IBCCF still performs favorably among the compared trackers and is superior to the sHCF method by 9.5%, demonstrating its effectiveness on handling target size variations.

In-plane/Out-of-plane Rotation: In this case, targets encounter with rotation due to fast motion or viewpoint changes, which often cause aspect ratio variation of targets. As shown in the upper right and lower left of Fig. 7, our IBCCF is robust to such kinds of variations and outperforms most of other trackers. Specifically, IBCCF achieves remarkable improvements over its counterparts HCF, i.e., 2.5% and 6.4% gains in case of in-plane and out-of-plane rotation, respectively. It indicates our IBCCF can deal with aspect ratio variation caused by rotations.

	Time Cost(ms)
CCF Feature Extraction	95
BCFs Feature Extraction	141
CCF Prediction	26
BCFs Prediction	40
Subproblem $\mathbf{w}$	51
Subproblem $\mathbf{g}$	37
Subproblem $\mathbf{w_k}$	40
Subproblem $\mathbf{u_k}$	4

Table 3: Time cost of IBCCF for each stage on OTB-2015 dataset. Note that all the time listed above is measured in milliseconds.

variation from Temple-Color dataset and compare IBCCF with other methods. Note that the sequences in the subset are not overlapped with other datasets. In addition, for validating the effectiveness of IBCCF with hand-crafted features, we implement two variants of IBCCF with HOG and color name [10] features (*i.e.* IBCCF-HOGCN, IBCCF-HOGCN (w/o constraint)).

Fig. 8 illustrates the comparison of overlap success plots for different trackers on two datasets. From Fig. 8(a), one can see that IBCCF ranks the second among all trackers, demonstrating the effectiveness of IBCCF on handling aspect ratio variation again. Furthermore, IBCCF-HOGCN also performs favorably against other methods and surpasses all of its counterparts (*i.e.* IBCCF-HOGCN (w/o constraint), DSST and SAMF). This validates the superiority of IBCCF under hand-crafted feature setting. As shown in Fig. 8(b), IBCCF is among the top three best-performed trackers and outperforms its counterpart HCF by 4.4%.

Figure 8: Comparison of the overlap success plots on two datasets. (a) the subset of 40 sequences from Temple-Color dataset. (b) the complete Temple-Color dataset.

5.4. The VOT benchmarks

Finally, we conduct experiments on Visual Object Tracking (VOT) benchmark [5], which consist of 60 challenging videos from real-life datasets. In VOT benchmark, a tracker is initialized at the first frame and reset again when it drifts the target. The performance is measured in terms of accuracy, robustness and expected average overlap (EAO). The accuracy computes the average overlap ratio between the estimated positions and ground truth. The robustness score evaluates the average number of tracking failures. And the

EAO metrics measures the average no-reset overlap of a tracker run on several short-term sequences.

VOT-2016 results. We compare IBCCF with several state-of-the-art trackers, including MDNet [26] (VOT-2015 winner), TCNN [25] (VOT-2016 winner) and part-based trackers such as DPT, GGTv2, SMPR and SHCT. All the results are obtained from VOT-2016 challenge website[2]. Table 4 lists the results on VOT-2016 dataset. One can note that IBCCF outperforms HCF method in terms of all three metrics. In addition, IBCCF also performs favorably against the part-based trackers, validating the superiority of boundary tracking on handling aspect ratio variation.

VOT-2017 results. At the time of writing, the results of VOT-2017 challenge were not available. Hence, we only report our results on the three metrics. In particular, the EAO, accuracy and robustness scores of IBCCF on VOT-2017 dataset are 0.209, 0.48 and 1.57, respectively.

6. Conclusion

In this work, we propose a tracking framework by integrating boundary and center correlation filters (IBCCF) to address the aspect ratio variation problem. Besides tracking the target center, a family of 1D boundary CFs is introduced to localize the left, right, top and bottom boundaries, thus can adapt to the target scale and aspect ratio changes flexibly. Furthermore, we analyze the near-orthogonality property between the center and boundary CFs, and impose an extra orthogonality constraint on the IBCCF model for improving the performance. An ADMM algorithm is also developed to solve the proposed model. We perform both qualitative and quantitative evaluation on four challenging benchmarks, and the results show that the proposed IBCCF approach perform favorably against several state-of-the-art trackers. Since we only employ the basic HCF model as the center CF tracker, in the future, we will incorporate with spatial regularization and continuous convolution to further improve our IBCCF.

7. Acknowledgements

References

[1] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. Torr. Staple: Complementary learners for real-time tracking. In *CVPR*, 2016.

[2] A. Bibi and B. Ghanem. Multi-template scale-adaptive kernelized correlation filters. In *ICCVW*, 2015.

---

[2]http://www.votchallenge.net/vot2016/

	DSST [7]	sKCF	Struck [13]	CCOT [11]	SRDCF [9]	DeepSRDCF [8]	Staple [1]	MDNet_N [26]	TCNN [25]	DPT	SMPR	SHCT	HCF [23]	IBCCF
EAO	0.181	0.153	0.142	0.331	0.247	0.276	0.295	0.257	0.325	0.236	0.147	0.266	0.220	0.266
Accuracy	0.5	0.44	0.44	0.52	0.52	0.51	0.54	0.53	0.54	0.48	0.44	0.54	0.47	0.51
Robustness	2.72	2.75	1.5	0.85	1.5	1.17	1.35	1.2	0.96	1.75	2.78	1.42	1.38	1.22

Table 4: Comparison of different state-of-the-art trackers on VOT-2016 dataset.

[3] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In *CVPR*, 2010.

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[5] L. Čehovin, A. Leonardis, and M. Kristan. Visual object tracking performance measures revisited. *TIP*, 25(3):1261–1274, 2016.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[7] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg. Discriminative scale space tracking. *TPAMI*, PP(99):1–1, 2016.

[8] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Convolutional features for correlation filter based visual tracking. In *ICCVW*, 2015.

[9] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Learning spatially regularized correlation filters for visual tracking. In *ICCV*, 2015.

[10] M. Danelljan, F. S. Khan, M. Felsberg, and J. V. D. Weijer. Adaptive color attributes for real-time visual tracking. In *CVPR*, 2014.

[11] M. Danelljan, A. Robinson, F. Shahbaz Khan, and M. Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *ECCV*, 2016.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[13] S. Hare, A. Saffari, and P. H. Torr. Struck: Structured output tracking with kernels. In *ICCV*, 2011.

[14] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *TPAMI*, 37(3):583–596, 2015.

[15] H. Kiani Galoogahi, T. Sim, and S. Lucey. Multi-channel correlation filters. In *ICCV*, 2013.

[16] H. Kiani Galoogahi, T. Sim, and S. Lucey. Correlation filters with limited boundaries. In *CVPR*, 2015.

[17] Y. Li and J. Zhu. A scale adaptive kernel correlation filter tracker with feature integration. In *ECCVW*, 2014.

[18] Y. Li, J. Zhu, and S. C. H. Hoi. Reliable patch trackers: Robust visual tracking by exploiting reliable patches. In *CVPR*, 2015.

[19] P. Liang, E. Blasch, and H. Ling. Encoding color information for visual tracking: Algorithms and benchmark. *TIP*, 24(12):5630–5644, 2015.

[20] S. Liu, T. Zhang, X. Cao, and C. Xu. Structural correlation filter for robust visual tracking. In *CVPR*, 2016.

[21] T. Liu, G. Wang, and Q. Yang. Real-time part-based visual tracking via adaptive correlation filters. In *CVPR*, 2015.

[22] A. Lukežič, T. Vojíř, L. Čehovin, J. Matas, and M. Kristan. Discriminative correlation filter with channel and spatial reliability. *arXiv preprint arXiv:1611.08461*, 2016.

[23] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang. Hierarchical convolutional features for visual tracking. In *ICCV*, 2015.

[24] C. Ma, X. Yang, C. Zhang, and M. H. Yang. Long-term correlation tracking. In *CVPR*, 2015.

[25] H. Nam, M. Baek, and B. Han. Modeling and propagating cnns in a tree structure for visual tracking. *arXiv preprint arXiv:1608.07242*, 2016.

[26] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, June 2016.

[27] J. Ning, J. Yang, S. Jiang, L. Zhang, and M. H. Yang. Object tracking via dual linear structured svm and explicit feature map. In *CVPR*, 2016.

[28] M. Pedersen, K. Baxter, B. Templeton, and D. Theobald. The matrix cookbook. *Technical University of Denmark*, 7:15, 2008.

[29] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M. H. Yang. Hedged deep tracking. In *CVPR*, 2016.

[30] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[31] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual tracking: An experimental survey. *TPAMI*, 36(7):1442–1468, 2014.

[32] Y. Sui, Z. Zhang, G. Wang, Y. Tang, and L. Zhang. Real-time visual tracking: Promoting the robustness of correlation filter learning. In *ECCV*, 2016.

[33] M. Tang and J. Feng. Multi-kernel correlation filter for visual tracking. In *ICCV*, 2015.

[34] R. Tao, E. Gavves, and A. W. Smeulders. Siamese instance search for tracking. In *CVPR*, 2016.

[35] Y. Wu, J. Lim, and M.-H. Yang. Object tracking benchmark. *TPAMI*, 37(9):1834–1848, 2015.

[36] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song. Recent advances and trends in visual tracking: A review. *Neurocomputing*, 74(18):3823–3831, 2011.

[37] J. Zhang, S. Ma, and S. Sclaroff. Meem: robust tracking via multiple experts using entropy minimization. In *ECCV*, 2014.

[38] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang. Fast visual tracking via dense spatio-temporal context learning. In *ECCV*, 2014.

[39] L. Zhang, W. Wu, T. Chen, N. Strobel, and D. Comaniciu. Robust object tracking using semi-supervised appearance dictionary learning. *Pattern Recognition Letters*, 62(C):17–23, 2015.

[40] W. Zuo, X. Wu, L. Lin, L. Zhang, and M.-H. Yang. Learning support correlation filters for visual tracking. *arXiv preprint arXiv:1601.06032*, 2016.