Robust Visual Tracking via Constrained Multi-Kernel Correlation Filters

Bo Huang, Tingfa Xu, Shenwang Jiang, Yiwen Chen, and Yu Bai

***Abstract*—Discriminative Correlation Filter (DCF) based trackers are quite efficient in tracking objects by exploiting the circulant structure. The kernel trick further improves the performance of such trackers. The unwanted boundary effects, however, are difficult to solve in the kernelized correlation models. In this paper, we propose a novel Constrained Multi-Kernel Correlation tracking Filter (CMKCF), which applies spatial constraints to address this drawback. We build the multi-kernel models for multi-channel features with three different attributes, and then employ a spatial cropping operator on the semi-kernel matrix to address the boundary effects. For the constrained optimization solution, we develop an Alternating Direction Method of Multipliers (ADMM) based algorithm to learn our multi-kernel filters efficiently in the frequency domain. In particular, we suggest an adaptive updating mechanism by exploiting the feedback from high-confidence tracking results to avoid corruption in the model. Extensive experimental results demonstrate that the proposed method performs favorably on OTB-2013, OTB-2015, VOT-2016 and VOT-2018 dataset against several state-of-the-art methods.**

***Index Terms*—Discriminative Correlation Filter, spatial constraints, constrained optimization, adaptive updating.**

I. Introduction

VISUAL tracking has a plethora of practical applications in computer vision, including robotics [1], surveillance [2], video processing and biological image analysis [3]. The task of visual tracking is estimating the trajectory of a target in subsequent image frames, with an initial state (position and size) given in the first frame. Despite great progress has been made in the past decade, it is still a tough problem to design a generic tracker, since the target objects undergo significant appearance changes due to Occlusion (OCC), In-Plane or Out-of-Plane Rotation (IPR or OPR), Fast Motion (FM), Scale Variation (SV), Background Clutter (BC), etc.

Recently, Discriminative Correlation Filters (DCFs) have shown outstanding performance for visual object tracking thanks to their superior computation and fair robustness to photometric and geometric variations. Employing DCFs for visual tracking starts with MOSSE [4], which learns the Correlation Filters (CFs) using few samples in the frequency domain with an impressive speed of 669 FPS. Many recent works significantly advance the accuracy of DCF-based trackers from several aspects, such as feature representation [5], [6], nonlinear kernel [7], [8], scale estimation [9]–[11], prior probability [12], [13] and convolutional neural networks [14], [15]. Among them, the Kernel Trick [16] plays a vital role in improving the efficiency of trackers. The CSK method proposed by Heriques *et al.* [7] employs illumination intensity features and applies DCFs in a kernel space for the first time. The CSK method is further improved by using HOG features in the KCF tracking algorithm [8]. Danelljan *et al.* [5] exploit color attributes of target objects and learn an adaptive correlation filter by mapping multi-channel features into a Gaussian kernel space. To adaptively employ complementary features, the work MKCF in [17] extends KCF [8] to multi-kernel version to enhance the distinguishing ability of the model. Most recently, the MKCFup tracker [18] further improves the performance by taking advantage of the invariance-discriminative power spectrums of various features.

The standard formulation of DCFs uses circular correlation which allows to implement the learning of CFs efficiently by Fast Fourier Transform (FFT). However, the negative examples used for training the filters are implicitly generated through the application of a circular shift on the real-world examples. Due to the circularity, these negative examples are not realistic and are plagued by circular boundary effects, which dramatically hurt the tracking performance. To alleviate these unwanted boundary effects, some DCF-based trackers utilize the discriminative deep features (e.g. DeepSRDCF [19] and CCOT [20]) or deep tracking frameworks (e.g. MDNet [21] and FCNT [22]), but suffering from high complexity limits the real-time performance. There are also some hand-crafted feature based trackers [23]–[25], which can address this deficiency of DCFs and achieve efficient tracking performance. SRDCF [23] decreases the boundary effect by introducing a spatial regularization term to penalize the DCF coefficients depending on their spatial locations. Galoogahi *et al.* [24], [25] investigate the boundary effect problem by pre-multiplying the training images with a fixed

Manuscript received July 24, 2019; revised October 21, 2019 and December 6, 2019; accepted January 6, 2020. Date of publication January 10, 2020; date of current version October 23, 2020. This work was supported in part by the Major Science Instrument Program of the National Natural Science Foundation of China under Grant 61527802, in part by the General Program of National Nature Science Foundation of China under Grants 61371132 and 61471043, and in part by the International S&T Cooperation Program of China under Grant 2014DFR10960. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jingdong Wang. *(Corresponding author: Tingfa Xu.)*

B. Huang, S. Jiang, Y. Chen, and Y. Bai are with the School of Optics and Photonics, Image Engineering and Video Technology Lab, Beijing Institute of Technology, Beijing 1000811, China (e-mail: a1039377853@163.com; jiangwenj02@gmail.com; cyw951025@163.com; baiyu_bit@163.com).

T. Xu is with the Key Laboratory of Photoelectronic Imaging Technology and System, Ministry of Education of China, Beijing 100081 and Chongqing Innovation Center, Beijing Institute of Technology, Chongqing 401135, China (e-mail: ciom_xtf1@bit.edu.cn).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2020.2965482

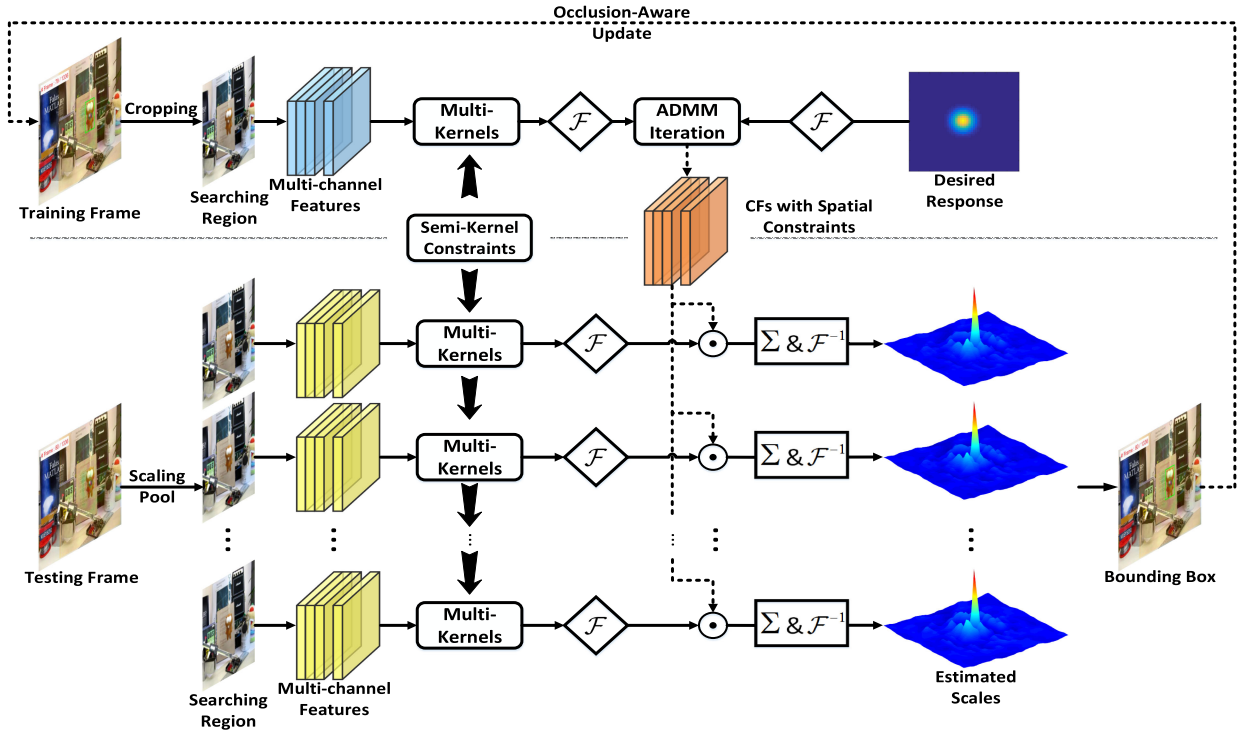


Fig. 1. A scheme of the proposed CMKCF for single object tracking. During the training stage, CMKCF applies a cropping operator on semi-kernels to alleviate the boundary effect. The multi-kernels are predefined, and the filters are then learned in the frequency domain via ADMM. In test stage, we extract the hand-craft features on multiple resolutions of the search area, and the multi-scale response maps are further obtained by element-wise multiplication of the features and filters in the frequency domain. Finally, The target is estimated on the highest peak of the best matching response map, and APCE is used to determine whether the target instance is reasonable enough for updating the filters.

masking matrix. CSR-DCF [26] handles this drawback through a preprocessing of foreground segmentation. The major disadvantage of such methods is that the spatial constraint breaks the circulant matrix structure, which makes it difficult to kernelize the model. How to effectively apply the spatial constraint to the kernel matrix remains an open problem.

In this work, we derive a Constrained Multi-Kernel Correlation Filter (CMKCF) based tracker that successfully employs the spatial constraint on the kernel model. In order to maintain the circulant property of the kernel matrix, we implement a spatial cropping operator on the semi-kernel matrix. The cropping matrix is a binary masking matrix that ensures that all real negative examples are densely extracted from the background. For the constrained optimization solution, we develop an efficient Alternating Direction Method of Multipliers (ADMM) [27] based algorithm to learn our multi-kernel CFs. More specifically, we suggest an adaptive updating mechanism to avoid the model corruption problem. In general DCF tracking approaches (e.g. KCF [8] and ECO [28]), they utilize a certain learning rate to update the training features in every frame or every several frames to make the model more adaptable. This may work in the scenes of a very short-term loss of a target. However, if the target is heavy occlusion or out-of-view for a while, their learning strategy will introduce inaccurate representations of the target, which will lead to irreversible errors. In order to increase the ability of algorithm to address the problem of target loss, we should be able to identify the reliable parts of tracking trajectory [29], [30]. In this work, we define the Average Peak-to-Correlation Energy (APCE) [31] to indicate the fluctuated degree of response maps. The confidence of tracking results is determined by the values of APCE and we update the filter only in frames with high-confidence. In summary, we have the following contributions in this paper:

- We propose a novel Constrained Multi-Kernel Correlation Filter (CMKCF) for visual tracking. The utilization of multi-kernels enhances the distinguishing ability of the model, and the implementation of spatial constraints solves the boundary effect very well.
- We develop an efficient Alternating Direction Method of Multipliers (ADMM) based algorithm for learning our multi-kernel filters in the frequency domain. Each subproblem has the closed form solution and our algorithm can empirically converge within very few iterations.
- We suggest an Average Peak-to-Correlation Energy (APCE) criterion to identify the reliable parts of the tracking trajectory, and we update the model adaptively in term of the feedback from high-confidence tracking results. Such adaptive updating mechanism avoids the model corruption, when the target is heavy occluded.

Fig. 1 presents a scheme of the proposed CMKCF for single object tracking. For evaluation purposes, we employ the public large-scale benchmarks, OTB-2013 [32], OTB-2015 [33], VOT-2016 [34] and VOT-2018 [40]. In order to make a fair comparison with existing state-of-the-art technologies, the organizers of the benchmarks recommend a set of common evaluation metrics, and maintain a large number of latest

algorithm results on the public dataset. In this paper, we use several representative metrics to evaluate the proposed tracker and demonstrate state-of-the-art performance. The rest of the paper is organized as follows: In Section II, we revisit the Kernelized Correlation Filters (KCFs). The Constrained Multi-Kernel Correlation Filter (CMKCF) is elaborated in detail in Section III (including learning CMKCFs, optimization algorithm, scale estimation and occlusion-aware update). The comparative experiments and quantitative evaluations are presented in Section IV. We reach the conclusions of the paper in Section V.

II. Revisit Kernelized Correlation Filters

Henriques *et al.* [8] propose to learn CFs in the spatial domain as solving the following kernel ridge regression problem,

$$\varepsilon(w) = \frac{1}{2}\sum_{i=1}^{L} ||y_i - f(x_i)||_2^2 + \frac{\lambda}{2}||w||_2^2 \tag{1}$$

Where L represents the size of the training signal x, and λ represents a regularization parameter ($\lambda \geq 0$). x_i indicates the sample after i cyclic shifts of x, and y_i is the i-th element of the predefined Gaussian shaped labels. The filter w is formulated by a linear combination of the samples, $w = \sum_{i=1}^{L} \alpha_i \varphi(x_i)$, where $\varphi(\cdot)$ is a nonlinear transformation. $f(x)$ is the linear classifier that has the form $f(x) = w^{\mathsf{T}}x$, where the transpose operator T is employed to ensure the operation is correlation not convolution. By using the Kernel Trick [16], we have,

$$\alpha = (K + \lambda I)^{-1} y \tag{2}$$

Where K is the kernel matrix, the elements in K are defined as, $K_{ij} = k(x_i, x_j) = \varphi(x_i)^{\mathsf{T}}\varphi(x_j)$. The solution of w is implicitly represented by the vector α. Fortunately, K is a circulant matrix that can be efficiently calculated in the Fourier domain, this allows one to express the objective in (2) equivalently as,

$$\alpha = \mathcal{F}^{-1}\left(\frac{\mathcal{F}(y)}{\mathcal{F}(k_{xx'}) + \lambda I}\right) \tag{3}$$

Where $\mathcal{F}$ and $\mathcal{F}^{-1}$ represent the Discrete Fourier transform (DFT) and Inverse Discrete Fourier transform (IDFT), respectively. $k_{xx'}$ denotes the first row of the circulant matrix K and the vector α contains all the α_i coefficients. In tracking, the classier response map for a single input z is obtained by computing the following formula,

$$\Phi(z) = \mathcal{F}^{-1}\left(\mathcal{F}^{*}(k_{zx_{\mathsf{model}}}) \circ \mathcal{F}(\alpha)\right) \tag{4}$$

Where x_{model} denotes the learned target appearance model and $\circ$ means the element-wise multiplication. The optimal target position is estimated on the highest peak of the response map. In the t-th frame, the update of the model is formulated as follows,

$$\begin{aligned} x_{\mathsf{model}}^{(t)} &= (1-\eta)x_{\mathsf{model}}^{(t-1)} + \eta x^{(t)} \\ \mathcal{F}(\alpha^{(t)}) &= (1-\eta)\mathcal{F}(\alpha^{(t-1)}) + \eta\mathcal{F}(\alpha) \end{aligned} \tag{5}$$

Where η is the online adaptation rate.

III. The Proposed CMKCF Method

A. Learning Constrained Multi-Kernel CFs

In the DCF formulation, the aim is to learn a convolution filter w from a set of training samples $\{(x_i, y_i)\}_{i=1}^{L}$. Each training sample x_i consists of a multi-dimensional feature map extracted from an image region. We apply three different attribute features, and tie them together in the same size. Kernel models are then established by concatenating every d-dimensional features together. Combining the features of all dimensions, and the constrained multi-kernel CFs can be expressed in the spatial domain as solving the following ridge regression problem,

$$\varepsilon(w) = \frac{1}{2}\sum_{i=1}^{L}\left\| y_i - \sum_{m=1}^{M} (w^m)^{\mathsf{T}}\mathbf{P}x_i \right\|_2^2 + \frac{\lambda}{2}\sum_{m=1}^{M}||w^m||_2^2 \tag{6}$$

Where M denotes the number of non-linear kernels, and w^m represents the m-th filter with a nonlinear transformation. $\mathbf{P}$ is the cropping operator (a binary matrix) which crops the mid D elements of signal x_i, where D is the size of the target. This cropping operation has been proved to solve the boundary effect well in [25]. Each solution of the filter w can be expanded as a linear combination of the inputs, $w^m = \sum_{i=1}^{L} \alpha_i^m \varphi^m(x_i)$. Therefore, we re-express each kernel classifier as,

$$f(x) = (w)^{\mathsf{T}}\mathbf{P}x = \sum_{i=1}^{L} \alpha_i (\varphi(x_i))^{\mathsf{T}}\varphi(\mathbf{P}x) = \bar{K}\alpha \tag{7}$$

Where $\bar{K}$ is the kernel matrix with a spatial constraint on the semi-kernel, and it is defined as follows,

$$\bar{K} = \begin{bmatrix} (\varphi(x_1))^{\mathsf{T}}\varphi(\mathbf{P}x) \\ (\varphi(x_2))^{\mathsf{T}}\varphi(\mathbf{P}x) \\ \vdots \\ (\varphi(x_L))^{\mathsf{T}}\varphi(\mathbf{P}x) \end{bmatrix} = \begin{bmatrix} k(x_1, \mathbf{P}x) \\ k(x_2, \mathbf{P}x) \\ \vdots \\ k(x_L, \mathbf{P}x) \end{bmatrix} \tag{8}$$

Where x_i represents the sample that applies a $(i-1)$-step discrete circular shift to the original signal x and $x_1 = x$. As $\mathbf{P}$ is a fixed binary matrix, we can figure out $\bar{K}$ is also a circulant matrix. We have,

$$\begin{aligned} f(x) &= \sum_{i=1}^{L} \alpha_i (\varphi(x_i))^{\mathsf{T}}\varphi(\mathbf{P}x) = \bar{K}\alpha \\ ||w||_2^2 &= (w)^{\mathsf{T}}w = \alpha^{\mathsf{T}}K\alpha \end{aligned} \tag{9}$$

As the solution of each kernel model is independent, (6) can be rewritten as,

$$\varepsilon(\alpha) = \frac{1}{2}\left\| y - \sum_{m=1}^{M} \bar{K}^m\alpha^m \right\|_2^2 + \frac{\lambda}{2}\sum_{m=1}^{M} (\alpha^m)^{\mathsf{T}}K^m\alpha^m \tag{10}$$

The difficulty of solving (10) is that the kernel matrix is difficult to obtain due to the large number of samples. Next, we will introduce the optimization algorithm, which avoids the direct solution of K and $\bar{K}$ in the spatial domain.

B. Optimization Algorithm

It is well-known that circular matrices in the spatial domain can be learned in the frequency domain, for computational efficiency. Because we use spatial constraints, we can't solve (10) directly in Fourier domain like KCF [8]. To this end, we introduce an auxiliary variable g, $\bar{K}g = \bar{K}\alpha$, and re-express (10) as,

$$\varepsilon(\alpha, g) = \frac{1}{2}\left\| y - \sum_{m=1}^{M} \bar{K}^m g^m \right\|_2^2 + \frac{\lambda}{2} \sum_{m=1}^{M} (\alpha^m)^{\mathsf{T}} K^m \alpha^m \tag{11}$$

The model in (11) is convex, the original problem can be split into two subproblems of solving the filter α and the auxiliary variable g by using the ADMM technique [27]. The Augmented Lagrangian form is formulated as,

$$\begin{aligned}\mathcal{L}(\alpha, g, \varsigma) = &\frac{1}{2}\left\| y - \sum_{m=1}^{M} \bar{K}^m \alpha^m \right\|_2^2 + \frac{\lambda}{2} \sum_{m=1}^{M} (\alpha^m)^{\mathsf{T}} K^m \alpha^m \\ &+ \sum_{m=1}^{M} (\varsigma^m)^{\mathsf{T}} (\bar{K}^m g^m - \bar{K}^m \alpha^m) \\ &+ \frac{\mu}{2} \sum_{m=1}^{M} ||\bar{K}^m g^m - \bar{K}^m \alpha^m||_2^2 \end{aligned} \tag{12}$$

Where μ is the penalty factor and ς is the $L \times 1$ Lagrange multiplier. We adopt the ADMM algorithm for alternatingly solving these two subproblems, g and α, and each subproblem has a closed form solution.

Subproblem g:

$$\begin{aligned} g = \arg\min_{g} \Bigg\{ &\frac{1}{2}\left\| y - \sum_{m=1}^{M} \bar{K}^m g^m \right\|_2^2 \\ &+ \sum_{m=1}^{M} (\varsigma^m)^{\mathsf{T}} (\bar{K}^m g^m - \bar{K}^m \alpha^m) \\ &+ \frac{\mu}{2} \sum_{m=1}^{M} ||\bar{K}^m g^m - \bar{K}^m \alpha^m||_2^2 \Bigg\} \end{aligned} \tag{13}$$

We find the minimum value by taking the derivative with respect to g,

$$g = \frac{\mu \bar{K}\alpha - \varsigma + y}{\bar{K} + \mu \bar{K}} \tag{14}$$

Where $\bar{K} = [(\sum_{j=1}^{L} (\varphi^1(x_j))^{\mathsf{T}} \varphi^1(\mathbf{P}x))^{\mathsf{T}}, \ldots, (\sum_{j=1}^{L} (\varphi^M(x_j))^{\mathsf{T}} \varphi^M(\mathbf{P}x))^{\mathsf{T}}]^{\mathsf{T}}$ is a $M \times L^2$ kernel matrix. $\alpha = [(\alpha^1)^{\mathsf{T}}, \ldots, (\alpha^M)^{\mathsf{T}}]^{\mathsf{T}}$ and $\varsigma = [(\varsigma^1)^{\mathsf{T}}, \ldots, (\varsigma^M)^{\mathsf{T}}]^{\mathsf{T}}$ are respectively the $M \times L$ over-complete representations of α and ς by concatenating their M kernel channels. Note that the kernel matrix $\bar{K}$ represents concatenating all possible cyclic shifts of $\bar{k}_{xx'} = [k^1(x_1, \mathbf{P}x)^{\mathsf{T}}, \ldots, k^M(x_1, \mathbf{P}x)^{\mathsf{T}}]^{\mathsf{T}}$, so $\bar{K}$ is a circulant matrix too. Therefore (14) can be computed in the Fourier domain, using,

$$C(u)v = \mathcal{F}^{-1}(\mathcal{F}^*(u) \circ \mathcal{F}(v)) \tag{15}$$

Where $C(u)$ represents concatenating all possible cyclic shifts of vector u. So we have,

$$\begin{aligned} g &= \frac{\mu C(\bar{k}_{xx'})\alpha - \varsigma + y}{C(\bar{k}_{xx'} + \mu \bar{k}_{xx'})} \\ &= \frac{\mu \mathcal{F}^{-1}(\mathcal{F}^*(\bar{k}_{xx'}) \circ \mathcal{F}(\alpha)) - \varsigma + y}{C(\bar{k}_{xx'} + \mu \bar{k}_{xx'})} \\ &= \mathcal{F}^{-1}\left(\frac{\mu \mathcal{F}^*(\bar{k}_{xx'}) \circ \mathcal{F}(\alpha) - \mathcal{F}(\varsigma) + \mathcal{F}(y)}{\mathcal{F}^*(\bar{k}_{xx'}) + \mu \mathcal{F}^*(\bar{k}_{xx'})} \right) \\ &\approx \mathcal{F}^{-1}\left(\frac{\mu \mathcal{F}^*(\bar{k}_{xx'}) \circ \mathcal{F}(\alpha) - \mathcal{F}(\varsigma) + \mathcal{F}(y)}{\mathcal{F}^*(\bar{k}_{xx'}) + \mu \mathcal{F}^*(\bar{k}_{xx'}) + \delta I_M} \right) \end{aligned} \tag{16}$$

Where δ is a small constant that prevents the denominator from being zero, and I_M is a $M \times M$ identity matrix.

Subproblem α:

$$\begin{aligned} \alpha &= \arg\min_{\alpha} \Bigg\{ \frac{\lambda}{2} \sum_{m=1}^{M} (\alpha^m)^{\mathsf{T}} K^m \alpha^m \\ &\qquad + \sum_{m=1}^{M} (\varsigma^m)^{\mathsf{T}} (\bar{K}^m g^m - \bar{K}^m \alpha^m) \\ &\qquad + \frac{\mu}{2} \sum_{m=1}^{M} ||\bar{K}^m g^m - \bar{K}^m \alpha^m||_2^2 \Bigg\} \\ &= \mathcal{F}^{-1}\left(\frac{\mu \mathcal{F}^*(\bar{k}_{xx'}) \circ \mathcal{F}(g) + \mathcal{F}(\varsigma)}{\mu \mathcal{F}^*(\bar{k}_{xx'}) + \lambda I_M} \right) \end{aligned} \tag{17}$$

Where $\bar{k}_{xx'} = [k^1(x_1, \mathbf{P}x)^{\mathsf{T}}, \ldots, k^M(x_1, \mathbf{P}x)^{\mathsf{T}}]^{\mathsf{T}}$ is the first row of the multi-kernel circulant matrix. We first extract multi-channel features with different attributes and apply the masking matrix $\mathbf{P}$ in the spatial domain via the LookUp Table (LUT). $\mathcal{F}(\bar{k}_{xx'})$ is then calculated by employing M independent Fast Fourier Transform (FFT) computations.

Since both α and g are solved in the frequency domain, we convert the original problem into the following subproblems:

$$\begin{cases} \mathcal{F}^*(g^{i+1}) = \dfrac{\mu \mathcal{F}^*(\bar{k}_{xx'}) \circ \mathcal{F}(\alpha^i) - \mathcal{F}(\varsigma^i) + \mathcal{F}(y)}{\mathcal{F}^*(\bar{k}_{xx'}) + \mu \mathcal{F}^*(\bar{k}_{xx'}) + \delta I_M} \\ \mathcal{F}^*(\alpha^{i+1}) = \dfrac{\mu \mathcal{F}^*(\bar{k}_{xx'}) \circ \mathcal{F}(g^{i+1}) + \mathcal{F}(\varsigma^i)}{\mu \mathcal{F}^*(\bar{k}_{xx'}) + \lambda I_M} \\ \mathcal{F}(\varsigma^{i+1}) = \mathcal{F}(\varsigma^i) + \mu \left(\mathcal{F}^*(\bar{k}_{xx'}) \circ \mathcal{F}(g^{i+1}) \right. \\ \qquad\qquad \left. - \mathcal{F}^*(\bar{k}_{xx'}) \circ \mathcal{F}(\alpha^{i+1}) \right) \end{cases} \tag{18}$$

Where μ is the stepsize parameter and it is updated as follow,

$$\mu^{i+1} = \min(\mu_{\max}, \beta \mu^i) \tag{19}$$

Where $\mu_{\max}$ denotes the maximum value of μ and β is the scale factor.

In tracking, the classier response map for a single input z is obtained by computing the following formula,

$$\Phi(z) = \mathcal{F}^{-1}\left(\mathcal{F}^*(\bar{k}_{z\mathbf{x}_{\text{model}}}) \circ \mathcal{F}(g) \right) \tag{20}$$

Where $\bar{k}_{zx_{\text{model}}} = [k^1(x_1, \bar{x}_{\text{model}})^{\mathsf{T}}, \ldots, k^M(x_1, \bar{x}_{\text{model}})^{\mathsf{T}}]^{\mathsf{T}}$ is the constrained kernel matrix. In the t-th frame, the update of

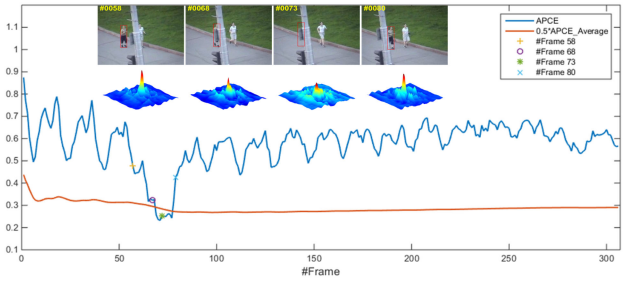


Fig. 2. Illustration of APCE criterion in sequence "*Jogging-1*" from OTB-2015, where the red bounding boxes represent the tracking results of the proposed CMKCF tracker. The blue curve represents the values of APCE, while the red one represents 0.5 times respective historical average values of APCE. $APCE = 0.4785$, $\Phi_{\max} = 0.9604$, when there is no occlusion in #Frame 58; $APCE = 0.3227$, $\Phi_{\max} = 0.6484$, when the target is partially occluded in #Frame 68; $APCE = 0.2528$, $\Phi_{\max} = 0.5082$, when the target is totally occluded in #Frame 73; $APCE = 0.4265$, $\Phi_{\max} = 0.8558$, when the target re-enters the field of view in #Frame 80.

the model is formulated as follows,

$$\begin{aligned} \bar{x}_{\text{model}}^{(t)} &= (1-\eta)\bar{x}_{\text{model}}^{(t-1)} + \eta \mathbf{P} x^{(t)} \\ \mathcal{F}(g^{(t)}) &= (1-\eta)\mathcal{F}(g^{(t-1)}) + \eta \mathcal{F}(g) \end{aligned} \tag{21}$$

C. Scale Estimation

In practical tracking applications, the target objects often undergo scale variation. A good scale adaption mechanism becomes necessary to enhance the tracking performance. Following SAMF [11], we apply the CFs on multiple resolutions of the search area to deal with the scale changes in videos. We fix the size of filters U_{T} and the scaling pool for the search area is defined as $\Theta = \{\theta_1, \theta_2, \ldots, \theta_S\}$, where S represents the number of scales. Intuitively, when a new frame comes out, we firstly crop the search region z with different scales $\{U_{\mathrm{T}}\theta_1, U_{\mathrm{T}}\theta_2, \ldots, U_{\mathrm{T}}\theta_S\}$ around the target center of the last frame. We then employ bilinear-interpolation to resize z into the fixed template size U_{T}. The final response map to find the proper target can be calculated by,

$$\Phi(z) = \underset{i=1,2,\ldots,S}{\arg\max} \left[\sum_{m=1}^{M} \mathcal{F}^{-1} \left(\mathcal{F}^{*}(\bar{k}^{m}_{z^{\theta_i}\mathbf{x}_{\text{model}}}) \circ \mathcal{F}(g^{m}) \right) \right] \tag{22}$$

Where z^{θ_i} refers to the search region with the size of $U_{\mathrm{T}}\theta_i$, which is resized to U_{T}. The target is located on the highest peak of the final response map.

D. Occlusion-Aware Update

Most DCF-based trackers update the CFs in every frame or every several frames to be more adaptable, without considering whether the tracking result is accurate or not. However, this may result in a deterministic failure when the target is severely occluded or completely missing in the current frame. In [31], Wang *et al.* utilize the Average Peak-to-Correlation Energy (APCE) to denote the fluctuation of the response map, and the necessity of model update depends on the feedback from confidence level of the tracking results. Inspired by them, we define APCE as,

$$APCE = \frac{(\Phi_{\max} - \Phi_{\min})^2}{ae^{B/L}} \tag{23}$$

Where $\Phi_{\max} = \max(\Phi(z))$, $\Phi_{\min} = \min(\Phi(z))$, and a is a constant used to control the ratio. B represents the area where the response value is greater than a threshold, and B is defined as,

$$B = B + 1 \quad \text{s.t.} \quad \sum_{j=1}^{L} (\Phi_j > 0.5 \times \Phi_{\max}) \tag{24}$$

Where Φ_j denotes the j-th element of $\Phi(z)$. As we can see in Fig. 1, when the target apparently appearing in the view scope in #Frame 58, the response map has a sharper peak and fewer noise, the area B will be smaller and APCE will become larger. When the target is occluded or missing in #Frame 68 and 73, the single peak of the response map becomes insignificant and a multimodal state occurs, the area B will increase and APCE will significantly decrease.

APCE indicates the fluctuated degree of response maps and the confidence level of the tracking results. We can figure out that the target representations are inaccurate when the APCE value significantly decreases, especially when APCE is less than 0.5 times its historical average. Therefore we discard these inaccurate target representations to avoid corruption in the model. The brief process of our CMKCF method is shown in Algorithm 1.

Algorithm 1: The Proposed CMKCF Tracker

Initial target bounding box $x^{(1)} = (\mathtt{x}^{(1)}, \mathtt{y}^{(1)}, \mathtt{w}^{(1)}, \mathtt{h}^{(1)})$ and other parameters;
Initial the target appearance model $\bar{x}_{\text{model}}^{(1)} = \mathbf{P}x^{(1)}$, and the filter $\mathcal{F}(g^{(1)})$;
for frame = 2, 3,..., until the last frame **do**
  Crop out the searching window from the entire frame;
  Build the target pyramid around $(\mathtt{x}^{(t-1)}, \mathtt{y}^{(t-1)})$ and extract the gray, CN and HOG features;
  Compute the correlation response map $\Phi(z)$ using $\bar{x}_{\text{model}}^{(t-1)}$ and $\mathcal{F}(g^{(t-1)})$;
  Estimate the optimal scale $s^{(t)}$, $x^{(t)} = (\mathtt{x}^{(t)}, \mathtt{y}^{(t)}, s^{(t)})$;
  Calculate the value of $APCE$ with formula (23);
  if $APCE > 0.5 * APCE\_Average$ **then**
    %% Update correlation filters via ADMM;
    while ADMM iteration **do**
      Update subproblem $\mathcal{F}(g^{i+1})$ using $\mathcal{F}(\alpha^i)$ and $\mathcal{F}(\varsigma^i)$, formula (16);
      Calculate the variable $\mathcal{F}(\alpha^{i+1})$ in formula (17);
      Compute the Lagrangians $\mathcal{F}(\varsigma^{i+1})$;
      Update the stepsize parameter μ;
    end while
    Update $\bar{x}_{\text{model}}^{(t)}$ and $\mathcal{F}(g^{(t)})$;
  end if
  Update $APCE\_Average$;
end for

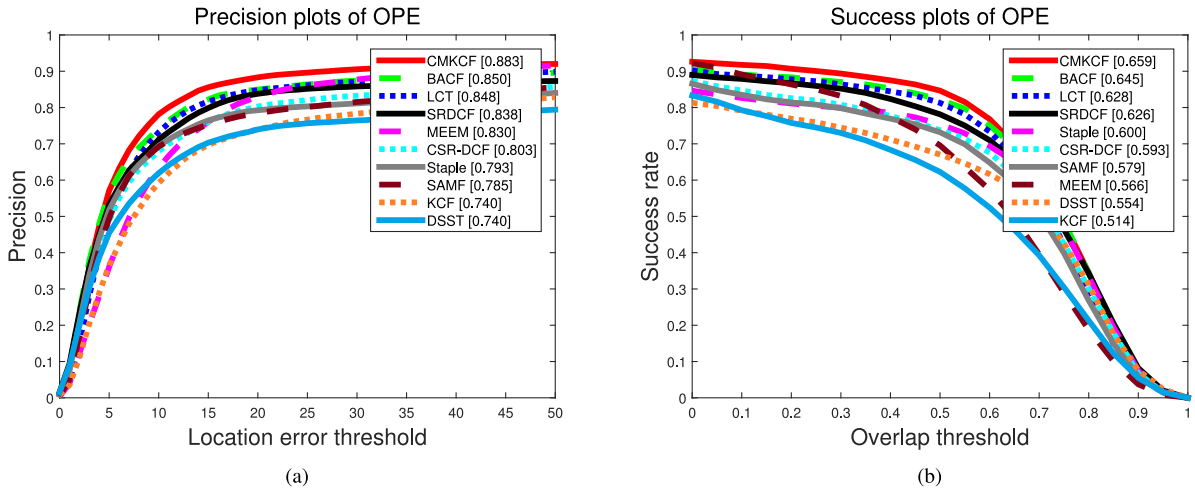


Fig. 3. The precision plot (a) and success plot (b) of OPE (one pass evaluation) on OTB-2013 dataset for 10 trackers. The legends show the precision scores and AUC scores for each tracker. Best viewed on color display.

IV. Experiment and Analysis

A. Implementation Details

The approach proposed in this paper is implemented in MATLAB R2014b on the Windows 7 x64 system with an Intel Core i5-4590 M 3.3 GHz processor and 8 GB DDR3 RAM. We crop a square searching region centered at the target, and the length of the region is set to $\sqrt{5\mathtt{wh}}$ ($\mathtt{w}$ and $\mathtt{h}$ represent the width and height of the target, respectively). We employ 1-channel gray, 10-channel CN [5], 31-channel HOG [35] features, and the total 42-channel features are then multiplied by a Hann window [4] to enhance the robustness. The dimension of the features used for each kernel, d, is set to 4. The regularization factor, λ, is set to 0.01, and the small constant, δ, is set to 10^{-4}. The scale adaptive scales parameters are set to 5 scale factors $\Theta = \{0.98, 0.99, 1.00, 1.01, 1.02\}$. The desired response is predefined by a 2D Gaussian function with bandwidth of $\sqrt{\mathtt{wh}}/16$. The online adaptation rate of CMKCF η is set to 0.02 for all experiments. For the ADMM optimization, the initial stepsize parameter μ_0, the maximum value $\mu_{\max}$ and the scale factor β are set to 2, 10 and 10^3, respectively. The threshold for APCE is set to 0.5 times its historical average, and the constant a is set to 2.0.

B. Comparisons on OTB Benchmarks

In order to evaluate our CMKCF tracker, we employ all the sequences on OTB-2013 benchmark [32] and OTB-2015 benchmark [33], which are classic benchmarks for visual tracking. On these two benchmarks, Precision and Success are the most popular metrics to evaluate the performance of the trackers. Precision metric indicates percentage of frames whose estimated locations lie in a given threshold distance to ground-truth centers, but it can hardly indicate the scale variations of the target object. Success metric measures the percentage of frames where the Intersection over Union (IoU) ratios of predicted and groundtruth bounding boxes are larger than a given threshold and the overall success performance is indicated by the Area Under the Curve (AUC) of success plots for all the thresholds. In this experiment, the error threshold used in the precision plot is set to 20 pixels, and the AUC scores are used to rank the trackers in the success plot.

In this section, we compare our approach with 9 state-of-the-art trackers from the literature: DSST [36], MEEM [37], SAMF [11], KCF [8], LCT [38], SRDCF [23], Staple [39], BACF [25] and CSR-DCF [26]. Fig. 3 and Fig. 4 show the overall performance of all the mentioned trackers in terms of the Precision and Success metric. It is worth noticing that the proposed method ranks first on both metrics on OTB-2013 dataset. On the success plot of OTB-2013, our approach provides an AUC score of 0.659, which outperforms the baseline KCF tracker by 14.5%. On OTB-2015, the average CMKCF performance on success plot is slightly lower than the first one in scores, but yields better performance in the average precision (center error). Further, our AUC score of 0.610 also leads to a significant gain of more than 10% compared to the baseline tracker. Among these comparative algorithms, SRDCF, BACF and CSR-DCF are designed to address the boundary effect. One can see that our CMKCF outperforms these three algorithms through the comprehensive comparison.

For further analyses on the tracking performance, we also demonstrate the advantages of our algorithm through the attribute-based comparison on sequences of the OTB-2013 dataset. The complete comparisons with 11 different attributes are illustrated in Fig. 5 and Fig. 6. Our CMKCF tracker achieves the best performance on all 11 attributes on Precision metric, and 10 attributes on Success metric, respectively. In case of Scale Variation (SV), SAMF, a milestone tracker for handling target size changes, achieves an AUC score of 50.7%. Our tracker provides a gain of 11.6% compared to SAMF, which is a significant improvement. On the success plots of Out-of-Plane Rotation (OPR), our algorithm exceeds the second by 1.8% in scores. The proposed CMKCF enhances the distinguishing ability of the model by utilizing multi-kernels with multi-attribute features, which addresses the OPR challenge well. Occlusion (OCC) and

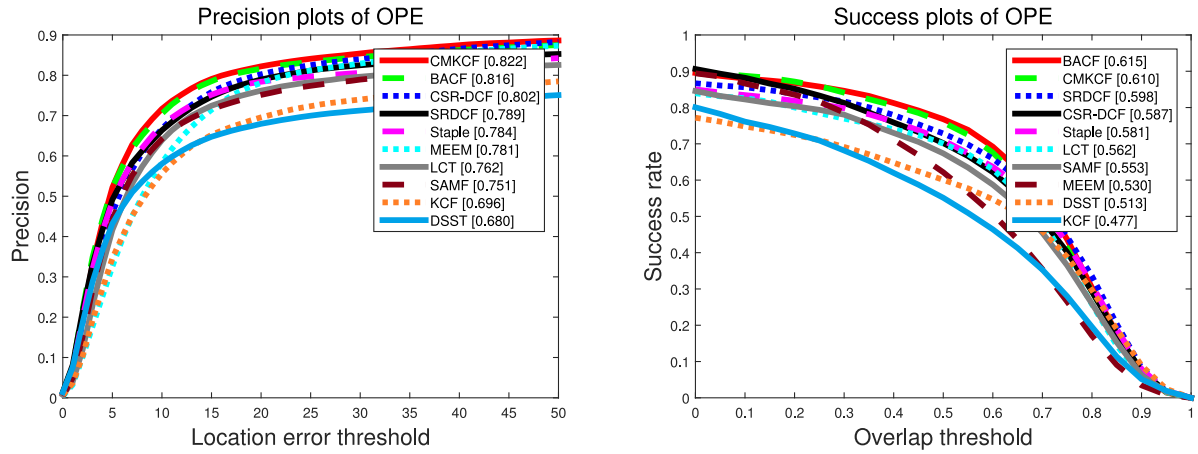


Fig. 4. The precision plot (a) and success plot (b) of OPE (one pass evaluation) on OTB-2015 dataset for 10 trackers. The legends show the precision scores and AUC scores for each tracker. Best viewed on color display.

Out-of-View (OV) may introduce inaccurate representations of the target and make the model gradually corrupt. The proposed CMKCF exploits an adaptive model updating mechanism via the feedback from high-confidence tracking results. So it's not surprising that our algorithm performs best in the both challenges.

C. Results on VOT Dataset

We compare our tracker with 5 top participants on VOT-2016 [34] and VOT-2018 [40] dataset, including KCF [8], DSST [36], SRDCF [23], CCOT [20], and Staple [39]. We conduct two sets of experiments: 1) the baseline evaluation in which trackers are reset with ground-truths when tracking failures occur; 2) the unsupervised evaluation where trackers are initialized with ground truth in the first frame. In this work, four primary metrics are used to analyze tracking performance: Accuracy (A), Robustness (R), Expected Average Overlap (EAO) and Area Under the Curve (AUC). The Accuracy (A) metric is the average overlap between the predicted and ground truth bounding boxes during successful tracking periods. The Robustness (R) metric measures how many times the tracker loses the target (fails) during tracking. The Expected Average Overlap (EAO) is an estimator of the average overlap a tracker is expected to attain for the baseline evaluation. Finally, AUC indicates the average overlap between the tracking box and the ground truth box under the unsupervised mechanism. The results of the mentioned metrics are shown in Table I and Table II. On VOT-2016, CMKCF obtains the second place of Robustness, EAO and AUC metric, and the third place of Accuracy metric. On VOT2018, CMKCF gets the best score, 0.4075, under the unsupervised evaluation. The more vivid A-R rank plots are shown in Fig. 7. The experimental results demonstrate that our tracker can not only achieve good accuracy but also appear very robust.

D. Visual Comparisons

For visual comparisons, we evaluate CMKCF with 6 state-of-the-art trackers including KCF [8], DSST [36], SAMF [11], LCT [38], SRDCF [23] and Staple [39]. KCF is the most typical kernelized correlation tracking filter. DSST and SAMF are designed to handle scale variations. LCT algorithm performs strong robustness in case of long term tracking, which can deal with the challenge of heavy occlusion. SRDCF can address boundary effects in the standard DCF trackers. Staple tracker applies more comprehensive features, which combines HOG with color names [5]. Considering clarity and representation, we mainly focus on the performance of the most common challenge factors, namely Occlusion (OCC), In-Plane or Out-of-Plane Rotation (IPR or OPR), Fast Motion (FM), Scale Variation (SV) and Background Clutter (BC). We choose some example sequences from the OTB-2015 dataset to illustrate the superiority of our tracker.

1) Occlusion: Occlusion pollutes the target model, if no measures are taken to remove this interference, it will lead to irreversible errors. Fig. 8 shows the situation in which the targets suffer partial or short-term complete occlusions. Among these comparison algorithms, only SRDCF can deal with the

TABLE I
VOT-2016 PERFORMANCE RESULTS. RED FONTS INDICATE THE BEST PERFORMANCE, THE BLUE FONTS INDICATE THE SECOND BEST ONES AND THE GREEN FONTS INDICATE THIRD ONES

Tracker	Baseline			Unsupervised
	A-R rank		EAO	Overlap
	Accuracy	Robustness	EAO	AUC
CMKCF	0.5325	18.1180	0.3011	0.4524
Staple	0.5433	23.8950	0.2952	0.3895
CCOT	0.5332	16.5817	0.3310	0.4701
SRDCF	0.5285	28.3167	0.2471	0.3980
DSST	0.5273	44.8138	0.1814	0.3267
KCF	0.4888	38.0820	0.1924	0.3023

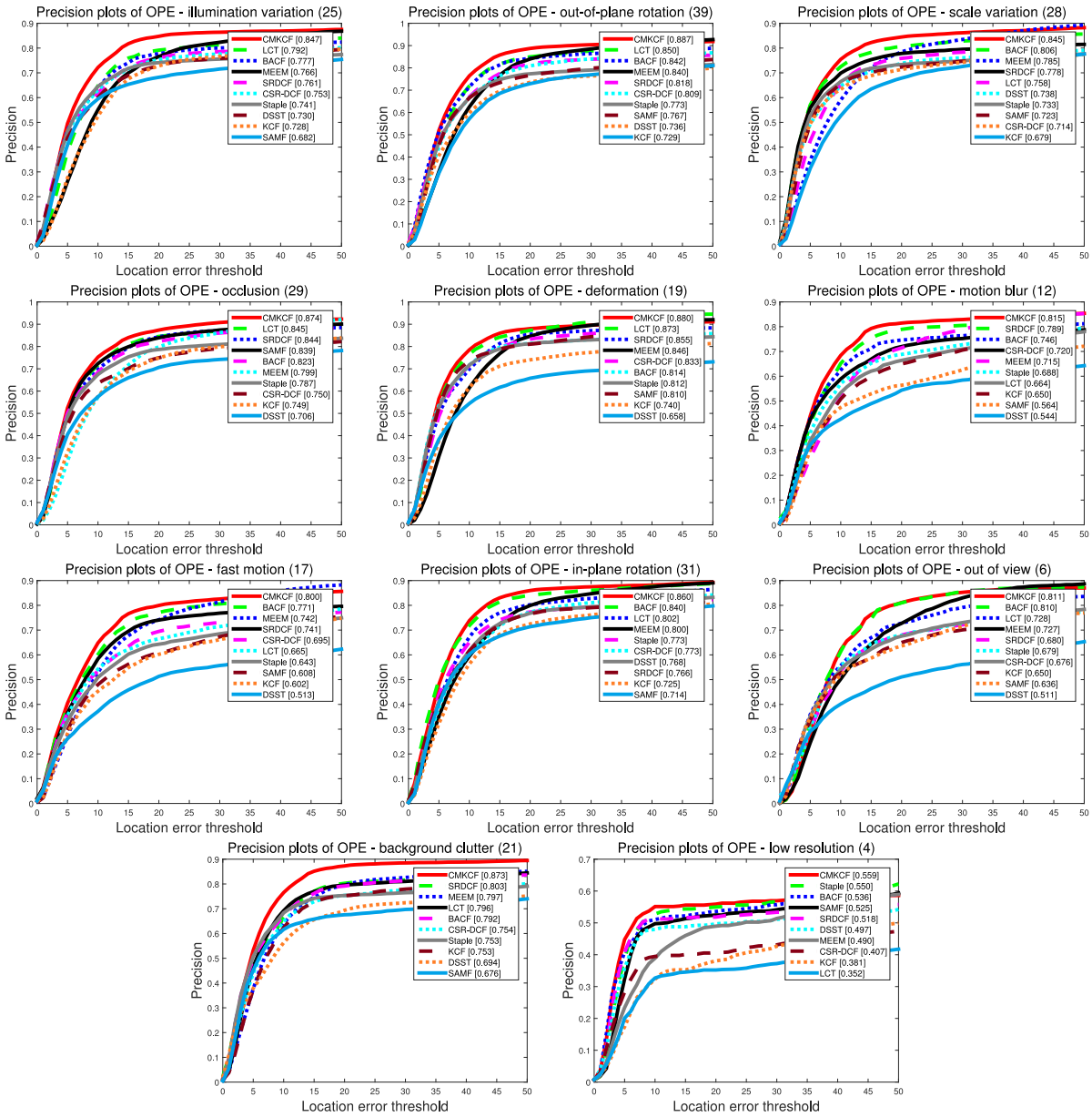


Fig. 5. Attribute-based analysis of our approach on the OTB-2013 dataset with all videos. Precision plots are shown for 11 attributes: IV (illumination variation), SV (scale variation), OCC (occlusion), DEF (deformation), MB (motion blur), FM (fast motion), IPR (in-plane rotation), OPR (out-of-plane rotation), OV (out-of-view), BC (background clutters), LR (low resolution). Attributes are displayed in each plot title, and the number of videos is appended to the end of each title.

challenge of occlusion in "*Jogging-2*" video sequences, even the LCT tracker designed for long-term tracking fails. Unfortunately, SRDCF fails in the other two video sequences. Our algorithm can deal with the challenge of occlusion in these three video sequences at the same time. The reason why our algorithm exceeds the other algorithms is that we use a feedback update criterion. When the occlusion occurs, the criterion will discard the inaccurate target representations to avoid introducing errors.

2) Rotation (In-Plane or Out-of-Plane): The challenge of rotation is caused by the movement of the target or the change of the viewpoint, and this challenge makes it difficult to model the appearance of the target. In rotation test on "*Dudek*" sequences in Fig. 9, none of the trackers lose their targets, but some trackers suffer from significant scale drift caused by the target rotation in or out of the image plane. Our algorithm tracks the target closely and keep a high degree of overlap, which demonstrates that our algorithm can address the rotation challenge well.

3) Fast Motion: Fast motion blurs the target, and we require a wider searching range to ensure that the target can be captured again. Video sequences in Fig. 10 are used to test the performance of these trackers to handle fast moving targets. On these sequences, only SRDCF and our CMKCF can track the target when fast motion occurs. Our tracking algorithm uses a large searching window and high-confidence updating mechanism, which ensure that our target will not be lost easily when it moves quickly.

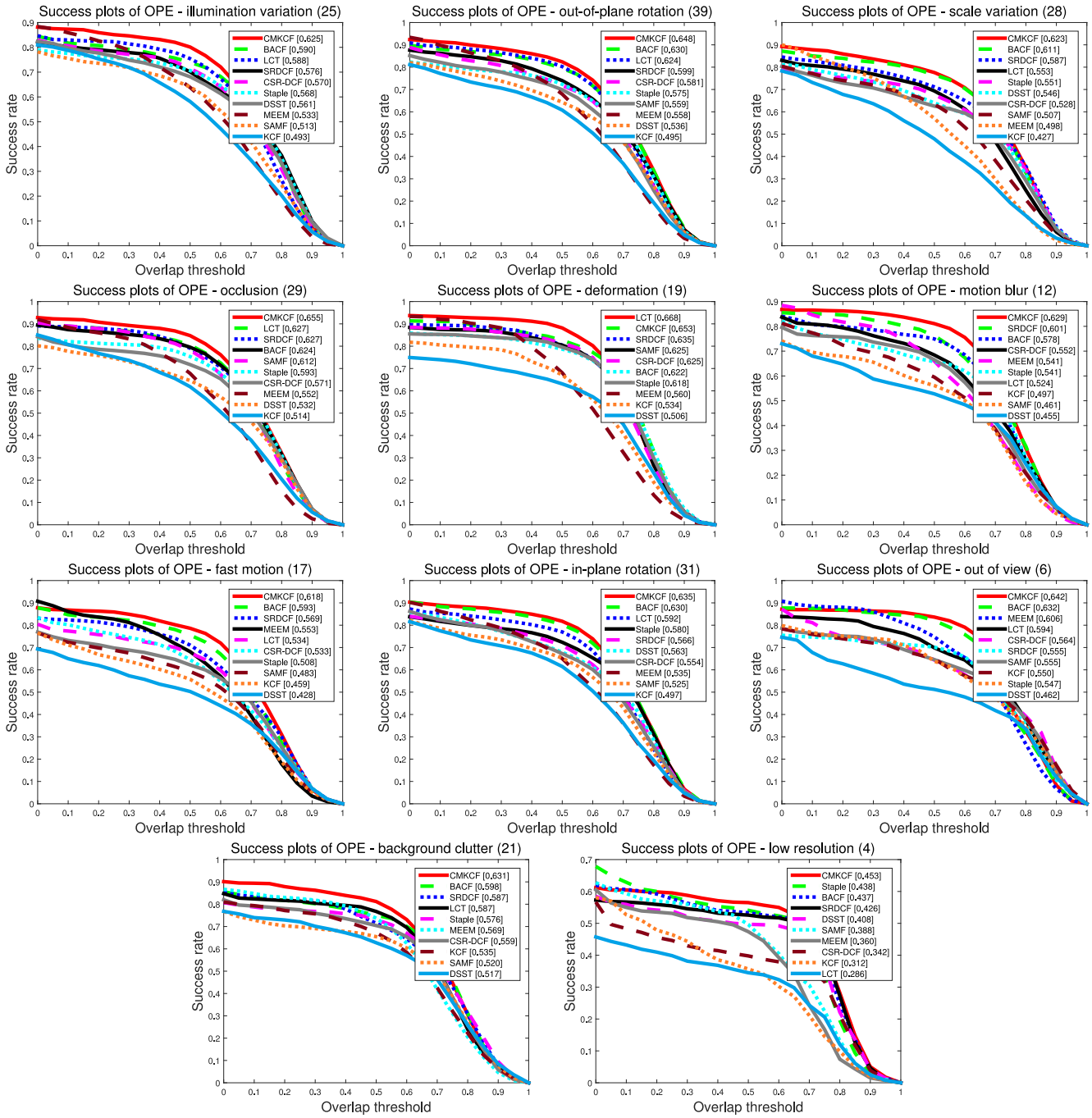
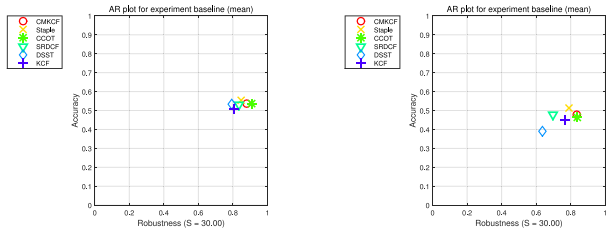


Fig. 6. Attribute-based analysis of our approach on the OTB-2013 dataset with all videos. Success plots are shown for 11 attributes: IV (illumination variation), SV (scale variation), OCC (occlusion), DEF (deformation), MB (motion blur), FM (fast motion), IPR (in-plane rotation), OPR (out-of-plane rotation), OV (out-of-view), BC (background clutters), LR (low resolution). Attributes are displayed in each plot title, and the number of videos is appended to the end of each title.

4) Scale Variation: The target size usually changes during tracking. Therefore, the tracker must adjust the bounding box according to the target size, otherwise the tracker may fail because of the lack of complete target information or the acquisition of redundant background information. Fig. 11 presents the tracking results in there video sequences with scale variations. Among the existing methods, two DCF based trackers, DSST and SAMF, are designed to handle scale variations. However, DSST and SAMF cannot adapt to scale variation in these sequences. Although our algorithm uses almost the same scale adaptive strategy as the SAMF algorithm, our CMKCF outperforms SAMF in capturing the target with different scales.

5) Background Clutter: The target undergoes the BC challenge in Fig. 12, the bounding box may drift onto the background, since distinguishing the target object from the background becomes very difficult through a rather simple model. One can see that the proposed CMKCF outperforms the other algorithms in handling this problem. The superiority of our



(a) (b)

Fig. 7. Accuracy-robustness rank plots for VOT-2016 (a) and VOT-2018 (b) tracking challenges.

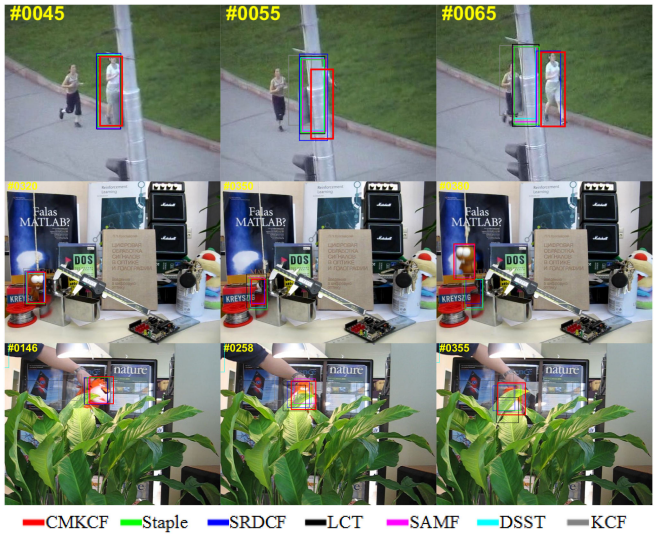


Fig. 8. Visual comparison with 6 state-of-the-art trackers in terms of Occlusion (OCC) challenges in the "*Jogging-2*," "*Lemming*" and "*Tiger2*" video sequences. Some representative frames are shown in the figures.

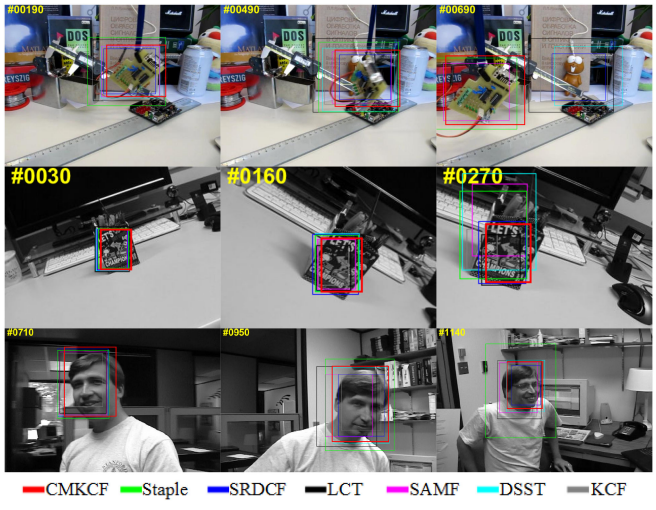


Fig. 9. Visual comparison with 6 state-of-the-art trackers in terms of In-Plane or Out-of-Plane Rotation (IPR or OPR) challenges in the "*Board*," "*Vase*" and "*Dudek*" video sequences. Some representative frames are shown in the figures.

TABLE II
VOT-2018 PERFORMANCE RESULTS. RED FONTS INDICATE THE BEST PERFORMANCE, THE BLUE FONTS INDICATE THE SECOND BEST ONES AND THE GREEN FONTS INDICATE THIRD ONES

Tracker	Baseline			Unsupervised
	A-R rank		EAO	Overlap
	Accuracy	Robustness	EAO	AUC
CMKCF	0.5015	23.1898	0.2462	0.4075
Staple	0.5225	44.0194	0.1688	0.3327
CCOT	0.4851	20.4138	0.2674	0.3909
SRDCF	0.4767	64.1136	0.1179	0.2445
DSST	0.3913	95.5587	0.0793	0.1722
KCF	0.4445	50.0994	0.1351	0.2671

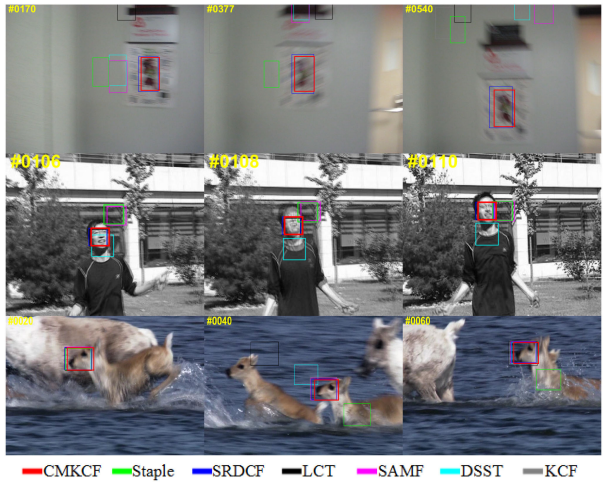


Fig. 10. Visual comparison with 6 state-of-the-art trackers in terms of Fast Motion (FM) challenges in the "*BlurOwl*," "*Jumping*" and "*Deer*" video sequences. Some representative frames are shown in the figures.

Fig. 11. Visual comparison with 6 state-of-the-art trackers in terms of Scale Variation (SV) challenges in the "*Freeman3*," "*Walking2*" and "*Singer1*" video sequences. Some representative frames are shown in the figures.

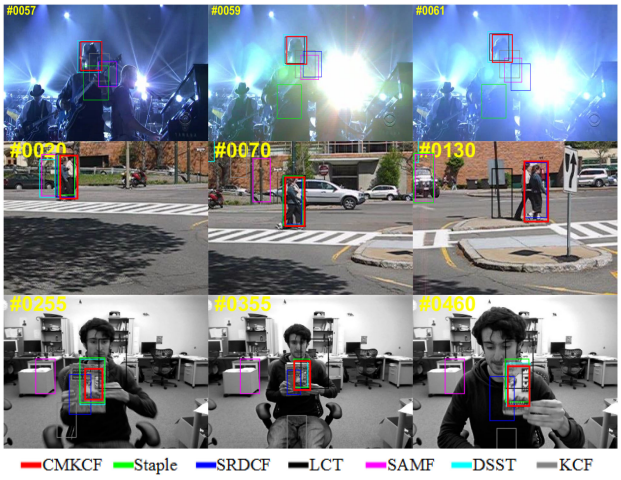


Fig. 12. Visual comparison with 6 state-of-the-art trackers in terms of Background Clutter (BC) challenges in the "*Shaking*," "*Couple*" and "*ClifBar*" video sequences. Some representative frames are shown in the figures.

algorithm could be attributed to the usage of the multi-attribute features, which enhances the robustness of target appearance modeling.

E. Comparisons With Deep Trackers

In this part, we compare our tracker against several representive deep trackers, including SiamFC [41], CF2 [6], HDT [42], CNN-SVM [43], DeepSRDCF [19], SiamRPN [44], SiamDW [45] and ATOM [46]. The OTB benchmarks are employed to evaluate the trackers in terms of Precision metric and Success metric. As shown in Table II, although our AUC score (65.9%) on OTB-2013 benchmark is slightly lower than SiamDW (66.2%), our method outperforms SiamFC (60.7%), CF2 (60.5%), HDT (60.3%), CNN-SVM (59.7%), DeepSRDCF (64.1%), SiamRPN (65.8%) and ATOM (64.3%). In addition, our average precision (88.3%) on OTB-2013 benchmark is also very competitive with SiamDW (93.2%). Notwithstanding the CMKCF tracker does well in the OTB-2013 dataset, our results on the OTB-2015 dataset are not optimistic. The reason may be that the distinguishing ability of our tracker is not comparable to that of deep features. The scenes of OTB-2015 videos are more complex, so traditional features can't contain enough information to identify targets.

F. Comparisons With Kernelized Trackers

Here, we compare our tracker against several state-of-the-art kernelized trackers on OTB-2013, including KCF [8], CN [5], MKCF [17] and MKCFup [18]. As shown in Fig. 13, Our average precision score (88.3%) and AUC score (65.9%) outperform that of MKCFup, and win the first place.

G. Ablation Studies

In this section, we analyze the proposed method on the OTB-2013 benchmark to study the contributions of Spatial Constraints (SC) and Different Attribute Features (DAF). CMKCF is

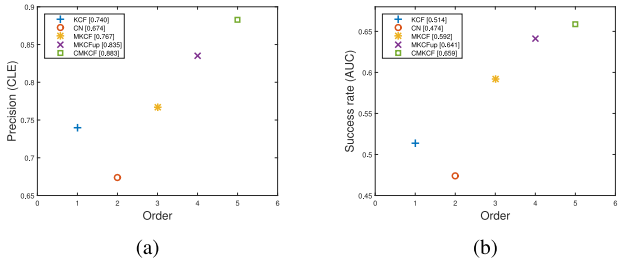


Fig. 13. Comparisons between CMKCF and 5 kernelized trackers on OTB-2013 dataset. (a) Precision results via OPE (one pass evaluation). (b) Results of the Area Under the success Curve (AUC).

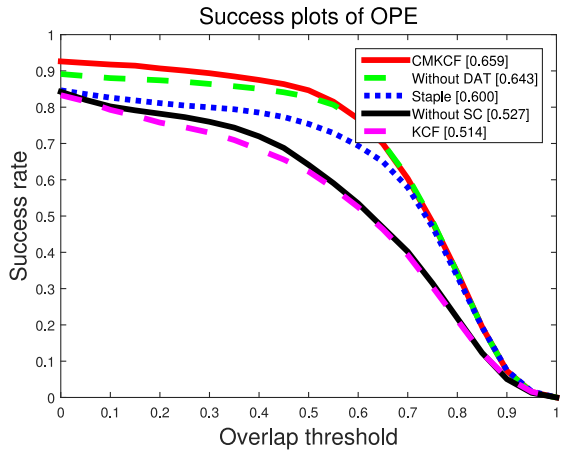


Fig. 14. Ablation study of CMKCF on OTB-2013 dataset. The AUC scores are used to rank the trackers.

our final method. KCF is one baseline tracker. Staple is another baseline tracker, which equips the same multi-attribute features as CMKCF. Without SC is the same method as CMKCF except it does not apply spatial constraints to the semi-kernel matrix, and its filter is directly solved by formula (3). Without DAF indicates the method that is the same as CMKCF except for only extracting HOG features. To make a fair comparison, CMKCF, Without SC and Without DAF use the same default parameters, so Without SC and Without DAF may not be optimal. Fig. 14 shows the comparisons with the 5 trackers mentioned above on OTB-2013 dataset, using success metric over all 51 videos. A significant decrease in the results of Without SC tracker indicates that the spatial constraint is critical to improve the performance of the tracker. The slight decrease in the AUC score of Without DAF shows that rich features can also improve the performance of tracker to a certain extent. In addition, our CMKCF exceeds Staple, which means that the performance gain of our tracker does not come from feature fusion.

H. Speed Analyses

Another important property of a tracker is its efficiency. For the optimal solution of CMKCF, we split the original problem into two sub-problems to calculate the updated filter more efficiently by ADMM. The computation of subproblem $\mathcal{F}^*(g)$ in formula (16) is bounded by $\mathcal{O}(ML\log(L)+ML)$, where

TABLE III
COMPARISONS WITH DEEP TRACKERS ON OTB BENCHMARKS IN TERMS OF PRECISION METRIC AND SUCCESS METRIC. RED FONTS INDICATE THE BEST PERFORMANCE, THE BLUE FONTS INDICATE THE SECOND BEST ONES AND THE GREEN FONTS INDICATE THIRD ONES

		SiamFC	CF2	HDT	CNN-SVM	DeepSRDCF	SiamRPN	SiamDW	ATOM	CMKCF
OTB-2013	Precision (%)	80.9	89.1	88.9	85.2	84.9	88.4	93.2	85.6	88.3
	Success (%)	60.7	60.5	60.3	59.7	64.1	65.8	66.2	64.3	65.9
OTB-2015	Precision (%)	77.1	83.7	84.8	81.4	85.1	85.1	92.3	87.0	82.2
	Success (%)	58.2	56.2	56.4	55.4	63.5	63.7	66.5	65.9	61.0
Average Scores (%)		69.2	72.38	72.6	70.4	74.4	75.8	79.6	75.7	74.4

TABLE IV
COMPARISON OF EXECUTION SPEED IN MATLAB R2014b SOFTWARE

Method	KCF	Staple	SRDCF	CMKCF
FPS	113.42	42.88	3.55	18.13

$L\log(L)$ is the cost of computing the FFT of the signal $\mathcal{F}^*(\bar{k}_{xx'})$ with the length of L. The cost of computing subproblem $\mathcal{F}^*(\alpha)$ using formula (17) is $\mathcal{O}(ML\log(L)+ML)$, which can be approximately reduced to $\mathcal{O}(ML)$, as $\mathcal{F}^*(\bar{k}_{xx'})$ has been calculated. In tracking, the amount of computation in formula (22) is $\mathcal{O}(SML\log(L))$. In order to further elaborate the tracking speed of CMKCF, we compare our method with 3 hand-crafted feature-based algorithms: KCF [8], SRDCF [23] and Staple [39]. KCF is the most basic algorithm, SRDCF is used to solve the boundary effect, and Staple utilizes the same features as CMKCF. The average FPS on OTB-2015 is shown in Table III. The execution speed of our method in this experiments has an average result of 18.13 FPS, which is more efficient than most deep learning based tracking methods.

V. CONCLUSION

In this paper, we employ constrained multi-kernel CFs for visual tracking. Our tracker not only enhances the ability of coding the target appearance, but also solves the boundary effect well. Furthermore, the solution of our model is optimized via the ADMM technique so that the filter can be calculated efficiently in the frequency domain. To avoid the model corruption problem, we suggest an Average Peak-to-Correlation Energy (APCE) criterion to identify the reliable parts of tracking trajectory. The experimental results demonstrate that the proposed tracker performs superiorly against several state-of-the-art algorithms on OTB-2013, OTB-2015, VOT-2016 and VOT-2018 dataset.

REFERENCES

[1] B. Ma *et al.*, "Visual tracking using strong classifier and structural local sparse descriptors," *IEEE Trans. Multimedia*, vol. 17, no. 10, pp. 1818–1828, Oct. 2015.

[2] W. Ruan *et al.*, "Multi-correlation filters with triangle-structure constraints for object tracking," *IEEE Trans. Multimedia*, vol. 21, no. 5, pp. 1122–1134, May 2019.

[3] Y. Liu *et al.*, "Context-aware three-dimensional mean-shift with occlusion handling for robust object tracking in RGB-D videos," *IEEE Trans. Multimedia*, vol. 21, no. 3, pp. 664–677, Mar. 2019.

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

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

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

[7] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 702–715.

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

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

[10] F. Li *et al.*, "Integrating boundary and center correlation filters for visual tracking with aspect ratio variation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2001–2009.

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

[12] C. Ma, Z. Miao, X.-P. Zhang, and M. Li, "A saliency prior context model for real-time object tracking," *IEEE Trans. Multimedia*, vol. 19, no. 11, pp. 2415–2424, Nov. 2017.

[13] N. Liang, G. Wu, W. Kang, Z. Wang, and D. D. Feng, "Real-time long-term tracking with prediction-detection-correction," *IEEE Trans. Multimedia*, vol. 20, no. 9, pp. 2289–2302, Sep. 2018.

[14] K. Chen and W. Tao, "Learning linear regression via single-convolutional layer for visual object tracking," *IEEE Trans. Multimedia*, vol. 21, no. 1, pp. 86–97, Jan. 2019.

[15] Q. Wang, C. Yuan, J. Wang, and W. Zeng, "Learning attentional recurrent neural network for visual tracking," *IEEE Trans. Multimedia*, vol. 21, no. 4, pp. 930–942, Apr. 2019.

[16] B. Schölkopf, *et al.*, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, Beyond*. Cambridge, MA, USA: MIT Press, 2002.

[17] M. Tang and J. Feng, "Multi-kernel correlation filter for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3038–3046.

[18] M. Tang, B. Yu, F. Zhang, and J. Wang, "High-speed tracking with multi-kernel correlation filters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4874–4883.

[19] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2015, pp. 58–66.

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

[21] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4293–4302.

[22] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3119–3127.

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

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

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

[26] A. Lukezic, T. Vojir, L. Č. Zajc, J. Matas, and M. Kristan, "Discriminative correlation filter with channel and spatial reliability," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6309–6318.

[27] S. Boyd *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[28] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6638–6646.

[29] X. Dong *et al.*, "Occlusion-aware real-time object tracking," *IEEE Trans. Multimedia*, vol. 19, no. 4, pp. 763–771, Apr. 2017.

[30] R. Yao, S. Xia, Z. Zhang, and Y. Zhang, "Real-time correlation filter tracking by efficient dense belief propagation with structure preserving," *IEEE Trans. Multimedia*, vol. 19, no. 4, pp. 772–784, Apr. 2017.

[31] M. Wang, Y. Liu, and Z. Huang, "Large margin object tracking with circulant feature maps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4021–4029.

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

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

[34] M. Kristan *et al.*, "The visual object tracking VOT 2016 challenge results," Springer, Oct. 2016. [Online]. Available: http://www.springer.com/gp/book/9783319488806

[35] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 2005, vol. 1, pp. 886–893.

[36] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf.*, Nottingham, U.K., Sep. 2014. [Online]. Available: http://www.bmva.org/bmvc/2014/papers/paper038/index.html

[37] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: Robust tracking via multiple experts using entropy minimization," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 188–203.

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

[39] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. Torr, "Staple: Complementary learners for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1401–1409.

[40] M. Kristan *et al.*, "The sixth visual object tracking VOT 2018 challenge results," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 3–53.

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

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

[43] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *Proc. Int. Conf. March. Learn.*, 2015, pp. 597–606.

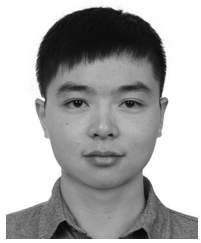
[44] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8971–8980.

[45] Z. Zhang and H. Peng, "Deeper and wider siamese networks for real-time visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4591–4600.

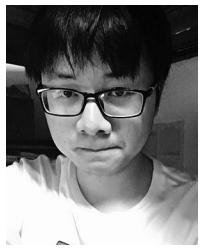
[46] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ATOM: Accurate tracking by overlap maximization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4660–4669.

Bo Huang is currently working toward the Ph.D. degree with the School of Optoelectronics, Beijing Institute of Technology, Beijing, China. His research interests include computer vision and real-time image/video processing.

Tingfa Xu received the Ph.D. degree from the Changchun Institute of Optics, Fine Mechanics and Physics, Changchun, China, in 2004. He is currently a Professor with the School of Optoelectronics, Beijing Institute of Technology, Beijing, China. His research interests include optoelectronic imaging and detection and hyper-spectral remote sensing image processing.



Shenwang Jiang received the B.E. degree from the School of Optoelectronics, Beijing Institute of Technology, Beijing, China, in 2014. He is currently working toward the Ph.D. degree in optical engineering with the Beijing Institute of Technology, Beijing, China. His research interests include image classification, object detection, and image processing.



Yiwen Chen is currently working toward the M.E. degree with the School of Optoelectronics, Beijing Institute of Technology, Beijing, China. His research interests include computer vision and real-time image/video processing.

Yu Bai is currently working toward the M.E. degree with the School of Optoelectronics, Beijing Institute of Technology, Beijing, China. Her research interests include computer vision and real-time image/video processing.