Visual Tracking via Spatially Aligned Correlation Filters Network

Mengdan Zhang[1(✉)], Qiang Wang[1], Junliang Xing[1], Jin Gao[1], Peixi Peng[1], Weiming Hu[1], and Steve Maybank[2]

[1] CAS Center for Excellence in Brain Science and Intelligence Technology, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, University of Chinese Academy of Sciences, Beijing, China
{mengdan.zhang,qiang.wang,jin.gao,wmhu}@nlpr.ia.ac.cn, pxpeng@pku.edu.cn

[2] Birkbeck College, University of London, London, UK
sjmaybank@dcs.bbk.ac.uk

Abstract. Correlation filters based trackers rely on a periodic assumption of the search sample to efficiently distinguish the target from the background. This assumption however yields undesired boundary effects and restricts aspect ratios of search samples. To handle these issues, an end-to-end deep architecture is proposed to incorporate geometric transformations into a correlation filters based network. This architecture introduces a novel spatial alignment module, which provides continuous feedback for transforming the target from the border to the center with a normalized aspect ratio. It enables correlation filters to work on well-aligned samples for better tracking. The whole architecture not only learns a generic relationship between object geometric transformations and object appearances, but also learns robust representations coupled to correlation filters in case of various geometric transformations. This lightweight architecture permits real-time speed. Experiments show our tracker effectively handles boundary effects and aspect ratio variations, achieving state-of-the-art tracking results on recent benchmarks.

Keywords: Visual tracking · Spatial transformer network
Deep learning · Correlation filters network

1 Introduction

Generic visual tracking aims to estimate the trajectory of a target in a video, given only its initial location. It has been widely applied, for example to

M. Zhang and Q. Wang—Contributed equally to this work.

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-01219-9_29) contains supplementary material, which is available to authorized users.

V. Ferrari et al. (Eds.): ECCV 2018, LNCS 11207, pp. 484–500, 2018.
https://doi.org/10.1007/978-3-030-01219-9_29

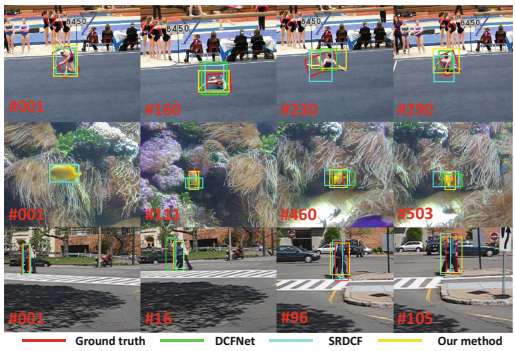

Fig. 1. Example videos (*Gymnastic3*, *Fish3* and *Pedestrian1*) in VOT2015 benchmark. General correlation filters (CF) based trackers such as DCFNet [33] and SRDCF [9] suffer performance decline in case of aspect ratio variations. DCFNet fails in case of fast motions because of the boundary effect.

video surveillance [1,13], and event recognition [27]. Visual tracking is challenging because the tracking scene contains complex motion patterns such as in-plane/out-of-plane rotation, deformation, and camera motion. A tracker has limited online samples with which to learn to adapt to these motion patterns.

The visual tracking of translating objects has been successfully tackled by recent correlation filters (CF) based approaches [10,18]. In these approaches, a circular window is moved over the search sample, leading to a dense and accurate estimation of the object translation. This circular sliding window operation assumes a periodic extension of the search sample, which enables efficient detection using the Fast Fourier transform, it however yields undesired boundary effects and restricts the aspect ratio of the search sample. Therefore, in cases of fast motions, rotations, and deformations which are common in practice, the performance of CF based trackers often drops significantly. As shown in Fig. 1, aspect ratio variation occurs frequently in the videos *Gymnastic3* and *Fish3*, and fast motion occurs frequently in the video *Pedestrian1*. Translation based CF trackers often fail on these challenging scenarios.

To address the above issues, spatially regularized CF based trackers [6,7,9,11] introduce a spatial regularization component within the CF to ensure that a CF tracker can work on a large image region effectively and can thus handle fast motions by reducing the boundary effect. The major disadvantage of these methods is that the regularized objective function is costly to optimize, even in the Fourier domain. CF with limited boundaries (CFLB) [16] and background-aware CF (BACF) [15] propose to exploit a masking matrix to allow search samples larger than the filter. However, BACF does not have a closed-form solution, which makes it difficult to be integrated into a deep neural network to boost the tracking performance. Many CF based trackers [8,15,16,26,44] ignore the aspect ratio variation, and the scale variation is handled by searching on several scale layers or learning a scale CF. Recently, the IBCCF tracker [25] addresses

aspect ratio variation by integrating 1D Boundary and 2D Center CFs where boundary and center filters are enforced by a nearly orthogonal regularization term. However, this integration has a high computation cost, which rules out real-time applications.

In this paper, we propose a novel end-to-end learnable spatially aligned CF based network to handle complex motion patterns of the target. A spatial alignment module (SAM) is incorporated into a differentiable CF based network to provide spatial alignment capabilities and reduce the CF's search space of the object motion. To be specific, conditioned on the consecutive frame regions (former target region and latter search region), SAM performs translation, aspect ratio variation and cropping on the search frame. This allows the network not only to select a region of an image that is most relevant to the target, but also to transform this region to a canonical pose to simplify the localization and recognition in the following CF layer. Once the CF layer obtains the transformed image from the SAM, it generates a Gaussian response map reflecting the object's position, scale and aspect ratio. Therefore, to generate this kind of the Gaussian response, our feature learning coupled to the CF layer is restricted to be positively adaptive to object geometric variations, which further boosts the capability of our network to handle complex object motion patterns. It should be noted that both the SAM and the CF layer can be trained with the standard back-propagation algorithm, allowing for end-to-end training of the whole tracking network on the ILSVRC2015 [12] dataset. After the whole network training on ILSVRC2015, both the SAM and the cascade CF tracking are learned in a data driven manner to be robust to general transformations existed in the training sample pairs.

In the online tracking process, the weights from the feature extraction layers and the SAM are frozen, while the coefficients of the CF layer are updated continuously to learn video-specific tracking cues. The SAM brings our tracker's attention to the target area according to its knowledge of various motion patterns learnt off-line and guides our CF to estimate the object motion more adaptively and accurately. Moreover, the light-weight network architecture and the fast calculation of the CF layer allow efficient tracking at a real-time speed. We conduct experiments on large benchmarks [22,41,42], and the results demonstrate that our algorithm performs competitively against state-of-the-art methods.

To sum up, the contributions of this work are three folds:

- We introduce a differentiable SAM in CF based tracking to address the challenging issues including boundary effects and aspect ratio variations in the previous CF based trackers, enabling better learnability for complex object motion patterns.
- We propose to learn discriminative convolutional features coupled to the spatially aligned CF to generate a Gaussian response map reflecting object's position, scale and aspect ratio, which allows accurate object localization.
- The proposed deep architecture for spatially aligned CF tracking is trained off-line from end to end. The spatial alignment and the CF based localization

are conducted in a mutual reinforced way, which ensures an accurate motion estimation inferred from the consistently optimized network. Our network also permits real-time tracking.

2 Related Work

Correlation Filter Based Trackers. The CF based trackers [8,26] are very popular due to their promising performance and computational efficiency. Since Bolme et al. [3] introduced the CF into the visual tracking field, several extensions have been proposed to improve the tracking performance. The examples include kernelized correlation filters [18,36], multiple dimensional features [10], context learning [15,28], scale estimation [8,26], re-detection [30], short-term and long-term memory [20], spatial regularization [9] and deep learning based CFs [6,29,32,38]. In this paper, we demonstrate that feature extraction, spatial alignment, CF based appearance modeling can be integrated into one network for end-to-end prediction and optimization, so that motion patterns of the object such as fast motions and aspect ratio variations are handled well by the CF based trackers.

Deep Learning Based Trackers. Recent works based on online deep learning trackers have shown high performance [31,35,40]. Despite the high performances, these trackers require frequent fine-tuning to adapt to object appearance changes. This fine-tuning is slow and prohibits real-time tracking. Furthermore, Siamese networks have received growing attention due to its two stream identical structure. These include tracking by object verification [37], tracking by correlation [2] and tracking by location axis prediction [17]. Although our spatial alignment module has a similar network architecture as [17], it permits back-propagation and is learnt with the CF in a mutual reinforced way. It provides the CF with an approximately aligned target to simplify the localization and recognition conducted in the CF layer. The CF layer is updated online to refine the alignment provided by the spatial alignment module for tracking accuracy. Moreover, to avoid over-fitting the network to tracking datasets, we train our network on ILSVRC2015 dataset instead of the ALOV300++ dataset.

Spatial Transformer Network. The spatial transformer network (STN) [21] has demonstrated excellent performances in selecting regions of interests automatically. It is used in face detection [4] to map the detected facial landmarks to their canonical positions to better normalize the face patterns. Dominant human proposals are extracted by STN in regional multi-person pose estimation [14]. For the first time, we introduce an STN into visual tracking. In order to well fit the characteristic of visual tracking, we modify a general STN from a single-input module to a two-stream module. Therefore, our two-stream module is called a spatial alignment module, which transforms the target object more purposefully for visual tracking.

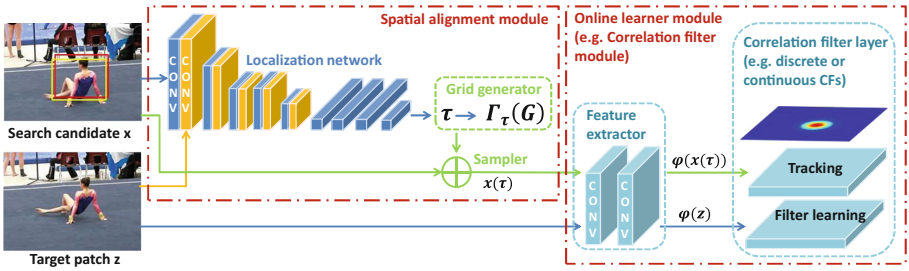

Fig. 2. Pipeline of our algorithm. Note that the red bounding box in the search patch $\mathbf{x}$ represents the initial candidate target position and the yellow one represents the aligned position provided by our SAM. Our SAM is generic and the CF module can be replaced by other online tracking learners. (Color figure online)

3 Spatially Aligned Correlation Filters Network

3.1 Overview

The architecture of the proposed spatially aligned CF based network (SACFNet) is shown in Fig. 2 to handle complex motion patterns of the target. It contains two components: a novel spatial alignment module (SAM) and a correlation filter (CF) module. The SAM contains a localization network, a grid generator and a sampler. The CF module contains a feature extractor and a CF based appearance modeling and tracking layer. The SAM brings the target into a CF's attention in the form of a canonical pose (centralized with the fixed scale and aspect ratio). Since this module is differentiable, the spatial alignment and CF based localization are optimized in a mutual reinforced way, which ensures accurate motion estimations inferred from the consistently optimized network.

Denote a training sample as $\mathbf{x}$ which contains a target object drifting away from the center of this sample with different scale and aspect ratio from the canonical one. Let $\tau^\diamond$ be the expected transformation according to which the target object in $\mathbf{x}$ can be transformed to the center with the canonical scale and aspect ratio. In this paper, we just consider the object translations, scale and aspect ratio variations. Thus, $\tau^\diamond$ has four parameters including translations and scales along the horizontal and vertical axes, denoted $\tau^\diamond = \{dx, dy, dsx, dsy\}$. $\mathbf{y}(\tau^\diamond)$ is a canonical Gaussian correlation response based on the expected transformation $\tau^\diamond$. $\{\varphi^l(\cdot)\}_{l=1}^D$ denotes the D-dimensional representations obtained from the feature extractor coupled to the CF layer. The multi-channel CF is denoted as $\{\mathbf{w}^l\}_{l=1}^D$. Then, learning an SACFNet in the spatial domain is formulated by minimizing the objective function:

$$\begin{aligned} \epsilon(\theta_1, \theta_2) = & \frac{1}{2} \| \sum_{l=1}^{D} \mathbf{w}_{\theta_2}^l \star \varphi_{\theta_2}^l(\mathbf{x}(\tau_{\theta_1})) - \mathbf{y}(\tau^\diamond) \|_2^2 + \lambda \sum_{l=1}^{D} \| \mathbf{w}_{\theta_2}^l \|_2^2, \\ & \text{s.t.} \quad \mathbf{x}(\tau_{\theta_1}) = \mathbf{x} \circ \tau_{\theta_1}, \mathbf{y}(\tau^\diamond) = \mathbf{y} \circ \tau^\diamond, \end{aligned} \tag{1}$$

where $\star$ denotes a circular correlation operator, $\circ$ denotes that the image is transformed according to the transformation parameters via the grid generator and the sampler as in STN [21] and the constant $\lambda \geq 0$ is the weight of the regularization term. Note that $\mathbf{y}$ is the Gaussian correlation response whose mean, variance and magnitude are related to the object position, scale and aspect ratio in the sample $\mathbf{x}$. We learn parameters of the SAM denoted as θ_1 to generate an estimate of the object transformation denoted as τ_{θ_1}. This estimate τ_{θ_1} is expected to be equal to the true transformation $\tau^{\diamond}$. At the same time, we learn parameters of the feature extractor θ_2 to generate $\{\varphi^l(\cdot)\}_{l=1}^D$ and $\{\mathbf{w}^l\}_{l=1}^D$.

We find that it is difficult to directly learn these two twisted parameters in Eq. (1). Traditional image alignment algorithms such as [34,43] usually learn parameters of image transformations and object appearance models using the iterative optimization strategy. Therefore, in the training stage of out network, for a easy convergence, we divide the off-line training process of SACFNet into three steps: (1) pre-training the SAM, (2) boosting the feature learning in the CF module based on the pre-trained SAM, and (3) end-to-end fine-tuning for a global optimization. In the tracking stage, object localization is carried out directly with one pass based on our pre-learnt deep neural network. No network fine-tuning is carried out in the tracking stage. More details will be shown in the following three subsections.

3.2 Spatial Alignment Module

Because the parameters are twisted together in the optimization problem in Eq. (1), it is straightforward to first fix the feature extractor θ_2 and learn the SAM based on the subproblem:

$$\begin{aligned} \epsilon_1(\theta_1) &= \frac{1}{2}\|\sum_{l=1}^{D} \mathbf{w}^l \star \varphi^l(\mathbf{x}(\tau_{\theta_1})) - \mathbf{y}(\tau^{\diamond})\|_2^2, \\ \text{s.t.} \quad & \mathbf{x}(\tau_{\theta_1}) = \mathbf{x} \circ \tau_{\theta_1}, \mathbf{y}(\tau^{\diamond}) = \mathbf{y} \circ \tau^{\diamond}. \end{aligned} \tag{2}$$

Because in the beginning of the training process of the SACFNet, parameters in the feature extractor θ_2 are randomly initialized. Thus, the corresponding correlation filter $\{\mathbf{w}^l\}_{l=1}^D$ has a poor tracking performance. It can not provide a reliable supervision to the SAM, which affects the quality of the learning process of this module. Meanwhile, since 3D object movements such as deformations and out-of-plane rotations usually occur in visual tracking, learning 2D transformations based on the image matching loss as in Eq. (3) has limitations to handle 3D movements and has a large modeling error:

$$\epsilon_1(\theta_1) = \|\mathbf{x}(\tau_{\theta_1}) - \mathbf{x}(\tau^{\diamond})\|_2. \tag{3}$$

Therefore, our SAM focuses on regressing the target bounding boxes to integrally contain the target instead of a detailed image matching:

$$\epsilon_1(\theta_1) = \|\tau_{\theta_1} - \tau^{\diamond}\|_1. \tag{4}$$

2D affine transform is sufficient to model the target global transform and this loss is also exploited in GOTURN [17]. Compared to the particle filtering based tracking methods [24,31] which generate transformed sample candidates based on the random sampling on a Gaussian distribution, our SAM learns to directly estimate the correct transform and generate a sample containing the centralized object with the proper scale and aspect ratio.

Network Architecture. We exploit a two-stream (Siamese) architecture for the localization network of the SAM to estimate the target transformation. The target patch in the preceding frame $t-1$ and the search patch in the consecutive frame t are fed into this module as inputs. In this way, the object in the search patch is not only brought into attention, but also aligned with the object in the target patch, which is more favorable for visual tracking. Each stream contains the first five convolutional layers of the CaffeNet [23]. Features from two streams are then combined and fed into following three fully connected layers, which finally output the transformation parameters. Specifically, the number of feature channels in each fully connected layer is set to 4096 and the number of the transformation parameters is set to 4. The predicted transformation parameters are used to create a sampling grid to select a target region from the whole image, namely the grid generator and sampler in STN [21]. In this stage, the selected target region is not exploited for the optimization in Eq. (4).

3.3 Feature Learning for Correlation Filters

After the first stage training of the SAM, we freeze this module and carry out feature learning coupled to the CF layer:

$$\begin{aligned} \epsilon_2(\theta_2) = & \frac{1}{2}\| \sum_{l=1}^{D} \mathbf{w}_{\theta_2}^l \star \varphi_{\theta_2}^l(\mathbf{x}(\tau_{\theta_1^\diamond})) - \mathbf{y}(\tau_{\theta_1^\diamond})\|_2^2 + \lambda \sum_{l=1}^{D} \|\mathbf{w}_{\theta_2}^l\|_2^2, \\ & \text{s.t.} \quad \mathbf{x}(\tau_{\theta_1^\diamond}) = \mathbf{x} \circ \tau_{\theta_1^\diamond}, \mathbf{y}(\tau_{\theta_1^\diamond}) = \mathbf{y} \circ \tau_{\theta_1^\diamond}, \end{aligned} \tag{5}$$

where the transformation $\tau_{\theta_1^\diamond}$ is estimated by the pre-trained SAM. Notably, $\mathbf{y}(\tau_{\theta_1^\diamond})$ is a Gaussian response in the joint scale-displacement space corresponding to the augmented sample $\mathbf{x}(\tau_{\theta_1^\diamond})$. Compared to the canonical Gaussian response $\mathbf{y}(\tau^\diamond)$, its center $\mu(\tau_{\theta_1^\diamond})$, variance $\Sigma(\tau_{\theta_1^\diamond})$ and magnitude changes according to the Euclidean distance between the object state (position, scale and aspect ratio) in $\mathbf{x}(\tau_{\theta_1^\diamond})$ and the object state in the canonical image patch. The object in the canonical image patch is centralized with the fixed scale and aspect ratio. Therefore, compared to a general CFNet [33,38] whose training samples contain objects with a canonical pose and the Gaussian response is unique, our CF based appearance modeling considers object motion variations and is context-aware.

Network Architecture. Similar to [33], our CF module consists two branches: a filter learning branch and a tracking branch. Both branches exploit the same feature extractor which contains two convolutional layers with kernels whose sizes are $3 \times 3 \times 3 \times 96$ and $3 \times 3 \times 96 \times 32$. Specifically, a target patch $\mathbf{z}$ is fed into the filter learning branch to learn the parameters in the CF layer:

$$\hat{\mathbf{w}}_{\theta_2}^{l} = \frac{\hat{\mathbf{y}}^{*} \odot \hat{\varphi}_{\theta_2}^{l}(\mathbf{z})}{\sum_{k=1}^{D} \hat{\varphi}_{\theta_2}^{k}(\mathbf{z}) \odot (\hat{\varphi}_{\theta_2}^{k}(\mathbf{z}))^{*} + \lambda}, \tag{6}$$

where $\hat{\mathbf{y}}$ denotes the discrete Fourier transform of $\mathbf{y}$, *i.e.*, $\mathcal{F}(\mathbf{y})$, $\mathbf{y}^{*}$ represents the complex conjugate of $\mathbf{y}$, and $\odot$ denotes the Hadamard product. Note that for CF based appearance modeling, the object in the target patch $\mathbf{z}$ is centralized with the fixed scale and aspect ratio. Thus, its corresponding response $\mathbf{y}$ has a canonical form. The other tracking branch works on a search patch selected by the SAM from the whole image. The correlation response between the learnt CF in Eq. (6) and this search patch is calculated in the CF layer. Then, the CF module is trained by minimizing the difference between this real correlation response $g_{\theta_2}(\mathbf{x}(\tau_{\theta_1^{\diamond}}))$ and the expected Gaussian-shaped response $\mathbf{y}(\tau_{\theta_1^{\diamond}})$:

$$\epsilon_2(\theta_2) = \|g_{\theta_2}(\mathbf{x}(\tau_{\theta_1^{\diamond}})) - \mathbf{y}(\tau_{\theta_1^{\diamond}})\|_2^2 + \gamma\|\theta_2\|_2^2, \tag{7}$$

$$g_{\theta_2}(\mathbf{x}(\tau_{\theta_1^{\diamond}})) = \mathcal{F}^{-1}\Big(\sum_{l=1}^{D} \hat{\mathbf{w}}_{\theta_2}^{l*} \odot \hat{\varphi}_{\theta_2}^{l}\left(\mathbf{x}(\tau_{\theta_1^{\diamond}})\right)\Big), \tag{8}$$

where the constant $\gamma \geq 0$ is the relative weight of the regularization term. Therefore, effective feature learning is achieved by training the CF module under the guidance of the SAM.

The training process of the CF module is explained as follows. For explanation clarity, we omit the subscript θ_2 in the following equations. Since the operations in the forward pass only contain Hadamard product and division, we can calculate the derivative per-element:

$$\frac{\partial \epsilon_2}{\partial \hat{g}_{uv}^{*}(\mathbf{x}(\tau_{\theta_1^{\diamond}}))} = \left(\mathcal{F}\left(\frac{\partial \epsilon_2}{\partial g(\mathbf{x}(\tau_{\theta_1^{\diamond}}))}\right)\right)_{uv}. \tag{9}$$

For the back-propagation of the tracking branch,

$$\frac{\partial \epsilon_2}{\partial (\hat{\varphi}_{uv}^{l}(\mathbf{x}(\tau_{\theta_1^{\diamond}})))^{*}} = \frac{\partial \epsilon_2}{\partial \hat{g}_{uv}^{*}(\mathbf{x}(\tau_{\theta_1^{\diamond}}))}(\hat{\mathbf{w}}_{uv}^{l}), \tag{10}$$

$$\frac{\partial \epsilon_2}{\partial \varphi^{l}(\mathbf{x}(\tau_{\theta_1^{\diamond}}))} = \mathcal{F}^{-1}\left(\frac{\partial \epsilon_2}{\partial (\hat{\varphi}^{l}(\mathbf{x}(\tau_{\theta_1^{\diamond}})))^{*}}\right). \tag{11}$$

For the back-propagation of the filter learning branch, we treat $\hat{\varphi}_{uv}^{l}(\mathbf{z})$ and $(\hat{\varphi}_{uv}^{l}(\mathbf{z}))^{*}$ as independent variables.

$$\frac{\partial \epsilon_2}{\partial \hat{\varphi}_{uv}^{l}(\mathbf{z})} = \frac{\partial \epsilon_2}{\partial \hat{g}_{uv}^{*}(\mathbf{x}(\tau_{\theta_1^{\diamond}}))} \Gamma_1, \tag{12}$$

$$\Gamma_1 = \frac{(\hat{\varphi}_{uv}^{l}(\mathbf{x}(\tau_{\theta_1^{\diamond}})))^{*}\hat{\mathbf{y}}_{uv}^{*}(\tau_{\theta_1^{\diamond}}) - \hat{g}_{uv}^{*}(\mathbf{x}(\tau_{\theta_1^{\diamond}}))(\hat{\varphi}_{uv}^{l}(\mathbf{z}))^{*}}{\sum_{k=1}^{D} \hat{\varphi}_{uv}^{k}(\mathbf{z})(\hat{\varphi}_{uv}^{k}(\mathbf{z}))^{*} + \lambda}, \tag{13}$$

$$\frac{\partial \epsilon_2}{\partial (\hat{\varphi}_{uv}^{l}(\mathbf{z}))^{*}} = \frac{\partial \epsilon_2}{\partial \hat{g}_{uv}^{*}(\mathbf{x}(\tau_{\theta_1^{\diamond}}))} \Gamma_2, \tag{14}$$

$$\Gamma_2 = \frac{-\hat{g}^*_{uv}(\mathbf{x}(\tau_{\theta_1^\diamond}))\hat{\varphi}^l_{uv}(\mathbf{z})}{\sum_{k=1}^{D}\hat{\varphi}^k_{uv}(\mathbf{z})(\hat{\varphi}^k_{uv}(\mathbf{z}))^* + \lambda}, \tag{15}$$

$$\frac{\partial \epsilon_2}{\partial \varphi^l(\mathbf{z})} = \mathcal{F}^{-1}\left(\frac{\partial \epsilon_2}{\partial(\hat{\varphi}^l(\mathbf{z}))^*} + \left(\frac{\partial \epsilon_2}{\partial \hat{\varphi}^l(\mathbf{z})}\right)^*\right). \tag{16}$$

3.4 Model Training and Online Tracking

Model Training. We design a three-step procedure to train the proposed deep architecture for visual tracking: (1) pre-training the SAM (Sect. 3.2), (2) pre-training the CF module based on the pre-trained SAM (Sect. 3.3), and (3) fine-tuning the whole network to make the spatial alignment and the CF based localization optimized in a mutual reinforced way:

$$\begin{aligned}\epsilon(\theta_1,\theta_2) =& \frac{1}{2}\|\sum_{l=1}^{D}\mathbf{w}^l_{\theta_2} \star \varphi^l_{\theta_2}(\mathbf{x}(\tau_{\theta_1})) - \mathbf{y}(\tau^\diamond)\|_2^2 + \lambda\sum_{l=1}^{D}\|\mathbf{w}^l_{\theta_2}\|_2^2 + \|\tau_{\theta_1} - \tau^\diamond\|_1, \\ &\text{s.t.} \quad \mathbf{x}(\tau_{\theta_1}) = \mathbf{x}\circ\tau_{\theta_1}, \mathbf{y}(\tau^\diamond) = \mathbf{y}\circ\tau^\diamond,\end{aligned} \tag{17}$$

We maintain the loss from Eq. (4) for a better convergence as many STN based methods have done [4]. All the training stages are carried out on the ILSVRC2015 dataset, because it contains different scenes and objects from the canonical tracking benchmarks. A deep model can be safely trained on it without the risk of over-fitting to the domain of tracking videos. Pairs of search and target patches are extracted from this video dataset. Specifically, a target patch is generated for each frame by cropping an image region from an object bounding box. For each search patch, we randomly sample a set of source patches from the consecutive frame. The source patches are generated by randomly perturbing the bounding box to mimic motion changes (*e.g.*, translations, scale and aspect ratio variations) between frames. We follow the practice in GOTURN, assuming that the motion between frames follows a Laplace distribution.

Online Tracking. In the online tracking process, the feature extractor and the SAM are frozen. The CF layer is updated following the common practice in CF based trackers:

$$\hat{\mathbf{w}}^l_t = (1-\alpha)\cdot\hat{\mathbf{w}}^l_{t-1} + \alpha\cdot\hat{\mathbf{w}}^l, \tag{18}$$

where $\alpha = 0.01$ is the update rate. The computation cost of this online adaptation strategy is cheap compared to online network fine-tuning, and it is effective for a CF to adapt to object appearance changes quickly. When a new frame comes, we extract a search patch from the center location predicted in the previous frame. The SAM works on this patch and the target patch from the previous frame, and provides an initial estimation of object translation, scale and aspect ratio. The grid generator and sampler extract an aligned image patch in this new frame. For a more accurate scale estimation, based on this aligned image patch, we extract another two image patches using the scale factors

$\{a^s|a = 1.0275, s = \{-1, 1\}\}$ similarly to [33] for fine-grained alignment. These image patches are fed into the CF module for object localization. The final target scale is estimated based on the scale factors and the transformation parameters from the SAM.

Issue of General Object Movements. SAM is motivated to solve issues of the fixed target aspect ratio and the boundary effect in CF based appearance modeling and tracking. As the learning of general transformations such as deformations and out-of-plane rotations is very difficult even with accurate sample annotations, it is thus infeasible in the tracking problem to learn all these transformations in a single model without sample annotations. Nevertheless, our algorithm can well handle general transformations: (1) SAM focuses on regressing the target bounding boxes to integrally contain the target instead of a detailed target matching as explained in Sect. 3.2. SAM is trained in a data driven manner to be robust to deformations and out-of-plane rotations existed in the training sample pairs; and (2) the following processing step of cascade CF tracking is also very robust to these transformations owning to its data driven learning. As the objective of visual tracking is to estimate the target bounding boxes, we find our current design of SAM is effective and provide more accurate object locations than its counterparts.

4 Experiments

4.1 Experimental Setups

Implementation Details. Because our SAM is generic, apart from the canonical CF formulation, it is straightforward to introduce SAM into other online learners. Thus, in our experiments, we provide two versions of our SACFNet: (1) SACF$^{(D)}$ exploits a canonical discrete CF module as explained in Sect. 3.3; (2) SACF$^{(C)}$ exploits a continuous CF module which is same as ECO[1]. In the pre-training process of the SAM, we extract a target patch of 2^2 times the size of the target bounding box and then resize it to 227×227. The parameters of the convolutional layers are frozen and taken from the CaffeNet. We train three fully connected layers where the learning rate is $1e{-}5$, and the batch size is 50. In the pre-training process of the CF module, following the canonical CF setting, the padding size is 2 and the input size of the feature extractor is 125×125. The regularization weight λ is set to $1e{-}4$ and the Gaussian spatial bandwidth is set to 0.1. We train this CF module with a learning rate exponentially decaying from $1e{-}4$ to $1e{-}5$ and a batch size of 32. In the end-to-end training process, the two modules are learnt in a mutual reinforce manner with a learning rate of $1e{-}5$ and a batch size of 32. Our experiments are performed with the MatConvNet toolbox [39] on a PC with an i7 3.4 GHz CPU and a GeForce GTX Titan Black GPU. The mean speed of SACF$^{(D)}$ on OTB2015 dataset is 23 frames per second.

[1] https://github.com/martin-danelljan/ECO.

Benchmark Datasets and Evaluation Metrics. OTB [41,42] is a standard benchmark which contains 100 fully annotated targets with 11 different attributes. We follow the protocol of OTB and report results based on success plots and precision plots. The success plots show the percentage of frames in which the overlap score exceeds a threshold. In these plots, the trackers are ranked using the area under the curve (AUC) displayed in the legend. The precision plots show the percentage of frames where the center location error is below a threshold. A threshold of 20 pixels is exploited to rank trackers. The VOT dataset [22] comprises 60 videos showing various objects in challenging backgrounds. Trackers are evaluated in terms of accuracy and robustness. The accuracy score is based on the overlap with ground truth, while the robustness is determined by the failure rate. We use the expected average overlap (EAO) measure to analyze the overall tracking performance.

4.2 Ablation Studies

Our SACF$^{(D)}$ is learnt off-line in three steps as discussed in Sect. 3.4. In this section, we conduct ablation analysis on three datasets to validate the effectiveness of the proposed training steps, as shown in Table 1.

First, our SAM learned in the first training step is compared with GOTURN to show the effect of the training dataset and the tracking performance. SAM has a lower tracking performance than GOTURN on VOT2015 and OTB2013, because the annotations of bounding boxes in ILSVRC2015 are quite looser than ALOV300++ which is the training dataset of GOTURN, and there are video overlaps between ALOV300++ and VOT2015/OTB2013/OTB2015. The loose annotations make SAM tend to contain the whole object as shown in the video *Gymnastic3* in Fig. 1, and provide a coarse prediction which requires further precise localization from the CF module. Both SAM and GOTURN suffer easy tracking drifts because of the error accumulation and perform poorly on OTB2015 dataset which has a lower overlap ratio of videos with ALOV300++. Therefore, it is very difficult to precisely learn complex geometric transformations under a single supervision of the regression loss in Eq. (4).

Second, to verify the superiority of the training strategy in the second step, our CF module which is trained in the second step under the guidance of the SAM (denoted by CF-Aug) is compared with its baseline namely DCFNet tracker. Specifically, CF-Aug and DCFNet have the same tracking process and differ in the training strategy. In the training stage, the input search patch of CF-Aug outputted by SAM contains a target drifting from the center with the aspect ratio variation. It is expected to generate a Gaussian response whose center, variance, and magnitude vary correspondingly. Contrastively, DCFNet works on a canonical search patch and generates a canonical response. As shown in Table 1, with data augmentation and the appearance modeling related to object scale and aspect ratio variations, our learnt CF-Aug performs favorably against DCFNet. Third, the integration of the SAM and the CF-Aug learned from the second training step is named SACF$^{(D)}$-iter1. In the tracking process, this tracker exploits the SAM to first coarsely localize the target to reduce a CF's search

Table 1. An illustration of the effectiveness of each training stage on VOT2015, OTB2013, and OTB2015. Red, blue and green fonts indicate the 1st, 2nd, and 3rd performance respectively.

Stage	Tracker	VOT2015 A	VOT2015 R	VOT2015 EAO	OTB2013 AUC	OTB2015 AUC
1	GOTURN [17]	0.48	2.02	0.203	0.457	0.115
	SAM	0.43	3.24	0.158	0.297	0.132
2	DCFNet [33]	*0.53*	1.68	0.217	0.622	0.580
	CF-Aug	**0.55**	1.67	0.225	0.628	0.600
	SAM-DCFNet	0.52	1.19	0.280	0.639	0.610
	SACF$^{(D)}$-iter1	0.52	*1.16*	*0.287*	*0.648*	*0.612*
3	SACF$^{(D)}$	0.51	**1.00**	**0.324**	**0.664**	**0.633**
-	ECO [6]	0.57	1.29	0.326	0.709	0.688
-	SACF$^{(C)}$	0.57	1.07	0.343	0.713	0.693

space and then achieves the fine-grained localization based on a CF. The direct combination of SAM and DCFNet is named SAM-DCFNet. Because CF-aug is learnt coupled to SAM, SACF$^{(D)}$-iter1 shows a better performance.

Moreover, the effectiveness of the end-to-end fine-tuning is evaluated by comparing the fine-tuned SACF$^{(D)}$ in the third training step and SACF$^{(D)}$-iter1. SACF$^{(D)}$ outperforms SACF$^{(D)}$-iter1 on all three benchmark datasets because the SAM and the CF module are learnt in a reinforced way. Conclusively, SAM estimates the global transform of a target in two consecutive frames and thus provides a coarse target localization. Only based on coarse estimations, background noise is gradually introduced into the target template leading to tracking drifts. CFs work well in local fine-grained search spaces of translations and scales, but cannot well handle aspect ratio variations and large motions, suffering tracking misalignment and drifts. By combining two complementary components, the target template exploited by SAM is more precise and the search space of CFs can be narrowed to local refinement. SACF$^{(D)}$ is superior to SAM and CF-Aug on three datasets. SACF$^{(C)}$ also outperforms baseline ECO as shown in Table 1 and Fig. 5. Note that because object annotations in VOT benchmarks change aspect ratios more frequently than in the OTB benchmarks, SACF$^{(C)}$ obtains more significant improvements in VOT benchmarks. The results also prove the generalization capability of our SAM. Especially, according to the robustness measure in VOT2015, the incorporation of a SAM does not degrade the robustness of SACF$^{(D)}$ and SACF$^{(C)}$.

4.3 Comparisons with the State-of-the-Arts

OTB Dataset. We compare our two versions of SACFNet (SACF$^{(D)}$ and SACF$^{(C)}$) against recent state-of-the-art trackers including BACF [15], ECO [6], SINT_flow [37], STAPLE_CA (CACF) [28], CFNet [38], ACFN [5], IBCCF [25], SiamFC_3s [2], SAMF [26], SRDCF [9], and CNN-SVM [19]. Figure 3 illustrates precision and success plots on OTB2013 and OTB2015.

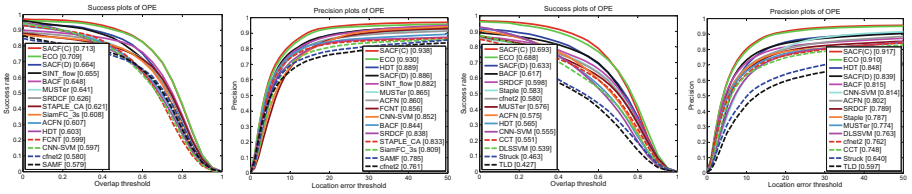

Fig. 3. Success plots and precision plots showing a comparison with recent state-of-the-art methods on OTB2013 and OTB2015.

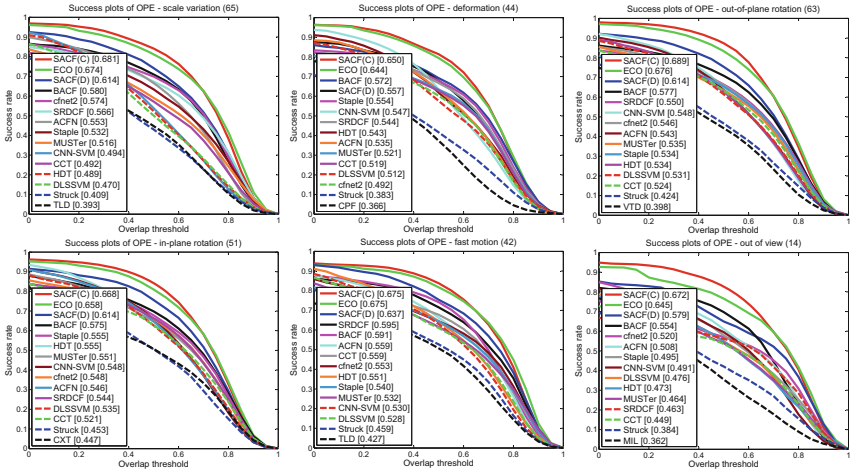

Fig. 4. Attribute-based analysis on the OTB2015 dataset.

From Fig. 3 we can draw three conclusions. First, SACF$^{(D)}$ outperforms most CF based trackers with a scale estimation (*e.g.*, SiamFC_3s and SAMF). SACF$^{(D)}$ is superior to IBCCF (AUC scores of 0.660 and 0.630 on OTB2013 and OTB2015) which considers the aspect ratio variation issue, and is more efficient than IBCCF. SACF$^{(D)}$ significantly outperforms ACFN, although ACFN introduces an attentional CF network to handle the target drift, blurriness, occlusion, scale changes, and flexible aspect ratio. SACF$^{(C)}$ also outperforms ECO benefiting from the consideration of object aspect ratio variations. Conclusively, SACFNet provides an effective and efficient way to tackle issues of the object scale and aspect ratio variations.

Second, SACF$^{(D)}$ provides a competitive tracking performance against BACF and SRDCF which solve the boundary effect problem. In contrast to SINT_flow where the Siamese tracking network and the optical flow method are isolated to each other, our SAM and CF module cooperate with each other and are learnt in a mutual reinforced way. Conclusively, compared to recent CF based trackers designed for handling boundary effects and Siamese network based trackers con-

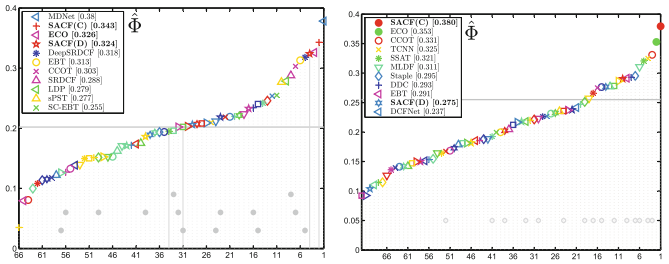

Fig. 5. EAO ranking with trackers in VOT2015 (left) and VOT2016 (right).

sidering object motions, SACF$^{(D)}$ provides a new strategy to benefit from the motion information while reducing boundary effects.

Third, SACF$^{(D)}$ outperforms traditional CFs based trackers (*e.g.*, CFNet, STAPLE_CA and HDT) and Siamese network based trackers (*e.g.*, SINT_flow, SiamFC_3s) on both datasets. Our feature learning coupled to the CF layer and the guidance of the SAM enhance the performance of a CF based tracker. Moreover, benefited from the integration of a CF layer, compared to other Siamese networks, our SACF$^{(D)}$ can online update the object appearance modeling efficiently without fine-tuning the network.

Attribute Based Analysis Related to Object Complex Motions. SACF$^{(D)}$ is evaluated on attributes to show its capability of tackling issues of aspect ratio variation and boundary effects on OTB2015 dataset, as shown in Fig. 4. Specifically, in cases of scale variation, deformation, and in-plane/out-of-plane rotation, the target scale and aspect ratio changes. In cases of fast motion and out-of-view, the boundary effects degrades tracking performance easily. We copy the AUC scores of IBCCF from its paper (scale variation: 0.610, occlusion: 0.600, out-of-plane rotation: 0.597, in-plane rotation: 0.589). SACF$^{(D)}$ is superior to IBCCF in all these cases related to the aspect ratio variation. SACF$^{(D)}$ outperforms its baseline tracker CFNet by large margins in cases of all the attributes. Our SAM learns useful motion patterns from the external dataset and simplify the localization and recognition in the following CF module.

VOT Dataset. We show the comparative results on VOT dataset in Fig. 5. SACF$^{(D)}$ and SACF$^{(C)}$ significantly exceed the VOT2015 *published sota bound* (grey line) and outperforms C-COT [11], DeepSRDCF [7] and EBT [45]. SACF$^{(C)}$ ranks first in VOT2016 dataset and outperforms ECO. The experimental results show the effectiveness of feature learning and the SAM.

5 Conclusion

We propose a novel visual tracking network that tackles the issues of boundary effects and aspect ratio variations in CF based trackers. The proposed deep

architecture enables feature learning, spatial alignment and CF based appearance modeling to be carried out simultaneously from end-to-end. Therefore, the spatial alignment and CF based localization are conducted in a mutual reinforced way, which ensures an accurate motion estimation inferred from the consistently optimized network.

Acknowledgements. This work is supported by the Natural Science Foundation of China (Grant No. 61751212, 61472421, 61602478), the NSFC-general technology collaborative Fund for basic re-search (Grant No. U1636218), the Key Research Program of Frontier Sciences, CAS, Grant No. QYZDJ-SSW-JSC040, and the CAS External cooperation key project.

References

1. Andriluka, M., Roth, S., Schiele, B.: People-tracking-by-detection and people-detection-by-tracking. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
2. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.S.: Fully-convolutional siamese networks for object tracking. In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9914, pp. 850–865. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48881-3_56
3. Bolme, D., Beveridge, J., Draper, B., Lui, Y.: Visual object tracking using adaptive correlation filters. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 2544–2550 (2010)
4. Chen, D., Hua, G., Wen, F., Sun, J.: Supervised transformer network for efficient face detection. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9909, pp. 122–138. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46454-1_8
5. Choi, J., Chang, H., Yun, S., Fischer, T., Demiris, Y., Choi, J.: Attentional correlation filter network for adaptive visual tracking, pp. 4828–4837 (2017)
6. Danelljan, M., Bhat, G., Khan, F., Felsberg, M.: ECO: efficient convolution operators for tracking. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 6931–6939 (2017)
7. Danelljan, M., Häger, G., Khan, F., Felsberg, M.: Convolutional features for correlation filter based visual tracking. In: Proceedings of IEEE International Conference on Computer Vision Workshops, pp. 58–66 (2015)
8. Danelljan, M., Häger, G., Khan, F., Felsberg, M.: Accurate scale estimation for robust visual tracking. In: Proceedings of British Machine Vision Conference, pp. 65.1–65.11 (2014)
9. Danelljan, M., Häger, G., Khan, F., Felsberg, M.: Learning spatially regularized correlation filters for visual tracking. In: Proceedings of IEEE International Conference on Computer Vision, pp. 4310–4318 (2015)
10. Danelljan, M., Khan, F., Felsberg, M., van de Weijer, J.: Adaptive color attributes for real-time visual tracking. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1090–1097 (2014)
11. Danelljan, M., Robinson, A., Shahbaz Khan, F., Felsberg, M.: Beyond correlation filters: learning continuous convolution operators for visual tracking. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9909, pp. 472–488. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46454-1_29

12. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Li, F.: Imagenet: a large-scale hierarchical image database. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)
13. Emami, A., Dadgostar, F., Bigdeli, A., Lovell, B.: Role of spatiotemporal oriented energy features for robust visual tracking in video surveillance. In: Proceedings of International Conference on Advanced Video and Signal-Based Surveillance, pp. 349–354 (2012)
14. Fang, H., Xie, S., Lu, C.: RMPE: regional multi-person pose estimation. arXiv preprint arXiv:1612.00137 (2016)
15. Hamed, K., Ashton, F., Simon, L.: Learning background-aware correlation filters for visual tracking. In: Proceedings of IEEE International Conference on Computer Vision, pp. 1144–1152 (2017)
16. Hamed, K., Terence, S., Simon, L.: Correlation filters with limited boundaries. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 4630–4638 (2015)
17. Held, D., Thrun, S., Savarese, S.: Learning to track at 100 FPS with deep regression networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 749–765. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_45
18. Henriques, J., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Trans Pattern Anal. Mach. Intell. **37**(3), 583–596 (2015)
19. Hong, S., You, T., Kwak, S., Han, B.: Online tracking by learning discriminative saliency map with convolutional neural network. In: Proceedings of International Conference on Machine Learning, pp. 597–606 (2015)
20. Hong, Z., Chen, Z., Wang, C., Mei, X., Prokhorov, D., Tao, D.: Multi-store tracker (MUSTer): a cognitive psychology inspired approach to object tracking. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 749–758 (2015)
21. Jaderberg, M., Simonyan, K., Zisserman, A.: Spatial transformer networks. In: Proceedings of Neural Information Processing Systems, pp. 2017–2025 (2015)
22. Kristan, M., et al.: The visual object tracking VOT2015 challenge results. In: Proceedings of IEEE International Conference on Computer Vision Workshops, pp. 1–23 (2015)
23. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. In: Proceedings of Neural Information Processing Systems, pp. 1097–1105 (2012)
24. Kwon, J., Lee, K.: Visual tracking decomposition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1269–1276 (2010)
25. Li, F., Yao, Y., Li, P., Zhang, D., Zuo, W., Yang, M.: Integrating boundary and center correlation filters for visual tracking with aspect ratio variation. arXiv preprint arXiv:1710.02039 (2017)
26. Li, Y., Zhu, J.: A scale adaptive kernel correlation filter tracker with feature integration. In: Agapito, L., Bronstein, M.M., Rother, C. (eds.) ECCV 2014. LNCS, vol. 8926, pp. 254–265. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16181-5_18
27. Liu, L., Xing, J., Ai, H., Ruan, X.: Hand posture recognition using finger geometric feature. In: Proceedings of IEEE International Conference on Pattern Recognition, pp. 565–568 (2012)

28. Mueller, M., Neil, S., Bernard, G.: Context-aware correlation filter tracking. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1387–1395 (2017)
29. Ma, C., Huang, J., Yang, X., Yang, M.: Hierarchical convolutional features for visual tracking. In: Proceedings of IEEE International Conference on Computer Vision, pp. 3074–3082 (2015)
30. Ma, C., Yang, X., Zhang, C., Yang, M.: Long-term correlation tracking. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 5388–5396 (2015)
31. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking, pp. 4293–4302 (2016)
32. Qi, Y., et al.: Hedged deep tracking. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 4303–4311 (2016)
33. Wang, Q., Gao, J., Xing, J., Zhang, M., Hu, W.: DCFNet: discriminant correlation filters network for visual tracking. arXiv preprint arXiv:1704.04057 (2017)
34. Song, W., Zhu, J., Li, Y., Chen, C.: Image alignment by online robust PCA via stochastic gradient descent. IEEE Trans Circuits Syst. Video Technol. **26**(7), 1241–1250 (2016)
35. Song, Y., Ma, C., Gong, L., Zhang, J., Lau, R., Yang, M.: CREST: convolutional residual learning for visual tracking, pp. 2574–2583 (2017)
36. Tang, M., Feng, J.: Multi-kernel correlation filter for visual tracking. In: Proceedings of IEEE International Conference on Computer Vision, pp. 3038–3046 (2015)
37. Tao, R., Gavves, E., Smeulders, A.: Siamese instance search for tracking. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1420–1429 (2016)
38. Valmadre, J., Bertinetto, L., Henriques, J., Vedaldi, A., Torr, P.: End-to-end representation learning for correlation filter based tracking. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 5000–5008 (2017)
39. Vedaldi, A., Lenc, K.: MatConvNet: convolutional neural networks for Matlab. In: ACM MM (2015)
40. Wang, L., Ouyang, W., Wang, X., Lu, H.: Visual tracking with fully convolutional networks. In: Proceedings of IEEE International Conference on Computer Vision, pp. 3119–3127 (2015)
41. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: a benchmark. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 2411–2418 (2013)
42. Wu, Y., Lim, J., Yang, M.: Object tracking benchmark. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1834–1848 (2015)
43. Wu, Y., Shen, B., Ling, H.: Online robust image alignment via iterative convex optimization. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1808–1814 (2012)
44. Zhang, M., Xing, J., Gao, J., Hu, W.: Robust visual tracking using joint scale-spatial correlation filters. In: Proceedings of IEEE International Conference on Image Processing, pp. 1468–1472 (2015)
45. Zhu, G., Porikli, F., Li, H.: Tracking randomly moving objects on edge box proposals. arXiv preprint arXiv:1507.08085 (2015)