Visual Tracking via Adaptive Spatially-Regularized Correlation Filters

Kenan Dai[1], Dong Wang[1]*, Huchuan Lu[1,2], Chong Sun[1,3], Jianhua Li[1]
[1] School of Information and Communication Engineering, Dalian University of Technology, China
[2]Peng Cheng Laboratory [3] Tencent Youtu Lab
dkn2014@mail.dlut.edu.cn, {wdice,lhchuan}@dlut.edu.cn, waynecool@mail.dlut.edu.cn, jianhual@dlut.edu.cn

Abstract

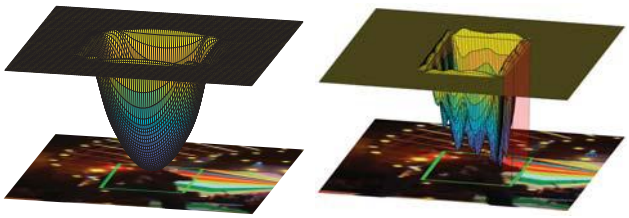
In this work, we propose a novel adaptive spatially-regularized correlation filters (ASRCF) model to simultaneously optimize the filter coefficients and the spatial regularization weight. First, this adaptive spatial regularization scheme could learn an effective spatial weight for a specific object and its appearance variations, and therefore result in more reliable filter coefficients during the tracking process. Second, our ASRCF model can be effectively optimized based on the alternating direction method of multipliers, where each subproblem has the closed-from solution. Third, our tracker applies two kinds of CF models to estimate the location and scale respectively. The location CF model exploits ensembles of shallow and deep features to determine the optimal position accurately. The scale CF model works on multi-scale shallow features to estimate the optimal scale efficiently. Extensive experiments on five recent benchmarks show that our tracker performs favorably against many state-of-the-art algorithms, with real-time performance of 28fps.

1. Introduction

Visual tracking [36, 25, 24] is a fundamental computer vision problem and has many realistic applications including video surveillance, behavior analysis, to name a few. Although many efforts have been done, it is still a tough task to design a robust and efficient tracker due to the difficulties from both foreground and background variations.

Recently, tracking algorithms based on correlation filters (CF) have achieved top-ranked performance and drawn increasing attentions. Usually, the CF-based trackers [18, 12, 11, 15, 8] exploit large numbers of cyclically shifted samples for learning, and convert the correlation operations in the spatial domain to the element-wise multiplications in the frequency domain, thereby reducing the computation complexity and improving the tracking speed significantly.

*Corresponding Author: Dr. Wang



(a) SRDCF [11] (b) ASRCF

Figure 1. The visualization of different spatial regularizations for the (a) SRDCF [11] and (b) ASRCF methods. For SRDCF, the spatial regularization has a negative Gaussian shape, which is almost equal for different objects and fixed during the tracking process. By contrast, our ASRCF method attempts to learn an adaptive spatial regularization, which is flexible for different objects in different time. As shown in (b), the ASRCF model has learned an effective spatial regularization that provides a higher penalty on the noise part and a lower penalty on the reliable part.

However, there exist two major imperfections of the earlier CF-based methods. First, the circulant shifted sampling process always suffers from periodic repetitions on boundary positions and makes the CF model be trained with a portion of unreal samples. This dilemma has been alleviated to some extent with additional pre-defined spatial constraints on filter coefficients [11, 15]. But these constraints are usually fixed for different objects and not changed during the tracking process, which cannot fully exploit the diversity information of different objects in different time. Second, the object localization and scale estimation are usually conducted on the same feature space, which requires extracting multi-scale feature maps during the tracking process. This strategy significantly increases the computational load and decreases the tracking speeds when the tracker exploits some powerful and complicated features (such as features extracted from deep networks). That is why the top-ranked CF trackers often runs very slow (e.g., DeepSRDCF [10], C-COT [13], DRT [32] and RPCF [33]).

In this work, we develop a robust and efficient CF-based tracker with two major efforts: adaptive spatial regularization and efficient scale estimation. The contributions of this

work can be summarized as follows.

First, this work proposes a novel adaptive spatially-regularized correlation filters (ASRCF) model, which could effectively estimate an object-aware spatial regularization (see Figure 1) and obtain more reliable filter coefficients during the tracking process. Our ASRCF is a general CF model and the well-known KCF, SRDCF and BACF algorithms are all its special cases.

Second, our ASRCF model can be effectively optimized via the alternating direction method of multipliers (ADMM), where each subproblem has the analytic solution.

Third, our tracker effectively and efficiently estimates both location and scale with two CF models: one exploits complicated features for accurate localization; and the other exploits shallow features for fast scale estimation.

Overall, our tracker achieves very remarkable performance with a real-time speed on the OTB2015, TC128, VOT2016, VOT2017 and LaSOT benchmarks.

2. Related Work

The trackers based on correlation filters (CF) have achieved great success in recent years. We briefly introduce some relevant ones to highlight our motivations. The MOSSE [3] method is the earliest CF-based tracker, which uses only grayscale samples to train the filter. The CSK [17] tracker introduces kernel trick into the CF formula. By exploiting circulant shifted samples, the filter coefficients can be efficiently optimized in the frequency domain. Based on CSK [17], the KCF [18] method exploits multi-channel HOG [7] features to enhance the feature representation ability and improves the tracking performance significantly. Similarly, the color naming features are introduced to achieve robust tracking in color videos [12]. The DSST [9], SAMF [26] and IBCCF [23] trackers address the scale adaptation problem using multi-scale searching strategies.

The traditional CF methods rely on a periodic assumption of the training and detection samples, which produces unexpected boundary effects and makes the tracker be trained and applied on a portion of unreal samples. To address this issue, Danelljan *et al.* [11] introduce a spatial regularization term in the CF formulae to penalize the filter coefficients near the boundary regions. Galoogahi *et al.* [15] directly multiply the filter with a binary matrix to generate real positive and negative samples for model training. The aforementioned two spatial constraints are widely used in subsequent research works [8, 13, 22, 32]. These spatial constraints are usually fixed for different objects and not changed during the tracking process; thus, they cannot fully exploit the diversity information of different objects in different frames. In this wok, we propose a novel adaptive spatial regularization term to make the tracker learn more reliable filter coefficients during the tracking process.

Recently, many researchers have attempted to combine the CF model with deep visual features, making the CF-based trackers achieve state-of-the-art performance [29, 8, 13, 22, 32]. Ma *et al.* [29] exploit three layers of CNN features pre-trained on the classification to generate feature maps for training CF models. Danelljan *et al.* [13] use the continuous convolution filters for combinations of feature maps with different spatial resolutions. However, these CF-based trackers no longer have the speed advantage due to the complicated deep features. Particularly, their scale estimation strategies require extracting multi-scale deep features, which is extremely expensive and makes the tracker very slow. In this work, we exploit two kinds of CF models to estimate the location and scale separately. The accurate object localization is obtained based on one CF model only with single scale robust deep features; while the efficient scale estimation is conducted with the other CF model with multi-scale shallow features.

3. Adaptive Spatially-Regularized Correlation Filters (ASRCF)

3.1. Objective Function of Our ASRCF Model

Original Correlation Filters (CF): The original multi-channel CF model in the spatial domain aims to minimize the following objective function [18]:

$$E(\mathbf{H}) = \frac{1}{2}\left\|\mathbf{y} - \sum_{k=1}^{K} \mathbf{x}_k * \mathbf{h}_k\right\|_2^2 + \frac{\lambda}{2}\sum_{k=1}^{K}\|\mathbf{h}_k\|_2^2, \quad (1)$$

where $\mathbf{x}_k \in \mathbb{R}^{T\times 1}$ and $\mathbf{h}_k \in \mathbb{R}^{T\times 1}$ denote the k-th channel of the vectorized image and filter respectively, and K is the total channel number. The vector $\mathbf{y} \in \mathbb{R}^{T\times 1}$ is the desired response (i.e., the Gaussian-shaped ground truth), $*$ denotes the spatial correlation operator and λ is a regularization constant. $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_K]$ is the matrix representing the filters from all K channels.

The original CF model suffers from periodic repetitions on boundary positions caused by circulant shifted samples, which inevitably degrades the tracking performance. To solve this problem, several spatial constraints have been introduced to alleviate the unexpected boundary effects. The representative methods include spatially regularized discriminative correlation filters (SRDCF) [11] and background-aware correlation filters (BACF) [15]. Their basic ideas are presented as follows.

SRDCF: The SRDCF method [11] introduces a spatial regularization to penalize the filter coefficients with respect to their spatial locations and modifies the objective function as

$$E(\mathbf{H}) = \frac{1}{2}\left\|\mathbf{y} - \sum_{k=1}^{K} \mathbf{x}_k * \mathbf{h}_k\right\|_2^2 + \frac{\lambda}{2}\sum_{k=1}^{K}\|\widetilde{\mathbf{w}} \odot \mathbf{h}_k\|_2^2, \quad (2)$$

where $\widetilde{\mathbf{w}}$ is a negative Gaussian-shaped spatial weight vector to make the learned filters have a high response around the center of the tracked object.

BACF: The BACF method [15] proposes a background-aware CF and introduces the following objective function:

$$E(\mathbf{H}) = \frac{1}{2}\left\|\mathbf{y} - \sum_{k=1}^{K}\mathbf{x}_k * \left(\mathbf{P}^\top \mathbf{h}_k\right)\right\|_2^2 + \frac{\lambda}{2}\sum_{k=1}^{K}\|\mathbf{h}_k\|_2^2, \tag{3}$$

where $\mathbf{P} \in \mathbb{R}^{T\times T}$ is a diagonal binary matrix to make the correlation operator directly apply on the true foreground and background samples.

The constraints on equations (2) and (3) are fixed during the tracking process and identical for different objects, which cannot well reflect the characteristics and appearance variations of a specific object. Thus, it is reasonable to introduce an adaptive spatial regularization into the CF model.

Our Objective Function: Motivated by the discussions above, we propose a novel adaptive spatially-regularized correlation filters (ASRCF) method to learn effective multi-channel CFs. Our objective function is defined as follows:

$$\begin{aligned} E(\mathbf{H},\mathbf{w}) =& \frac{1}{2}\left\|\mathbf{y} - \sum_{k=1}^{K}\mathbf{x}_k * (\mathbf{P}^\top \mathbf{h}_k)\right\|_2^2 \\ &+ \frac{\lambda_1}{2}\sum_{k=1}^{K}\|\mathbf{w}\odot\mathbf{h}_k\|_2^2 + \frac{\lambda_2}{2}\|\mathbf{w}-\mathbf{w}^r\|_2^2 \end{aligned}. \tag{4}$$

In equation (4), the first term is the ridge regression term that convolves the training data $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_K]$ with the filter $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_K]$ to fit the Gaussian-distributed ground truth $\mathbf{y}$. The second term is a regularization term introducing an adaptive spatial regularization on the filter $\mathbf{H}$, where the spatial weight $\mathbf{w}$ requires to be optimized. The third term attempts to make the adaptive spatial weight $\mathbf{w}$ be similar to a reference weight $\mathbf{w}^r$. This constraint introduces a priori information on $\mathbf{w}$ and avoids model degradation[1]. λ_1 and λ_2 are the regularization parameters of the second and third terms, respectively.

We note that the proposed ASRCF is a general CF model and the well-known KCF, SRDCF and BACF algorithms are all special cases of our model (shown in Table 1).

Table 1. The generalization ability of our ASRCF model.

Method	$\mathbf{P}$	$\mathbf{w}$
KCF	$\mathbf{P}=\mathbf{I}$	$\mathbf{w}=\mathbf{1}, \lambda_2 = 0$
SRDCF	$\mathbf{P}=\mathbf{I}$	$\mathbf{w}=\widetilde{\mathbf{w}}, \lambda_2 = 0$
BACF	-	$\mathbf{w}=\mathbf{1}, \lambda_2 = 0$

3.2. Optimization of Our ASRCF Model

Inspired by previous works [11, 15], correlation filters are usually learned in the frequency domain for efficient training and testing. Thus, we express the objective function (4) in the frequency domain (using Parseval's theorem), and convert it into the equality constrained optimization form:

$$\begin{aligned} E(\mathbf{H},\widehat{\mathbf{G}},\mathbf{w}) =& \tfrac{1}{2}\left\|\widehat{\mathbf{y}} - \sum_{k=1}^{K}\widehat{\mathbf{x}}_k \odot \widehat{\mathbf{g}}_k\right\|_2^2 \\ &+ \tfrac{\lambda_1}{2}\sum_{k=1}^{K}\|\mathbf{w}\odot\mathbf{h}_k\|_2^2 + \tfrac{\lambda_2}{2}\sum_{k=1}^{K}\|\mathbf{w}-\mathbf{w}_r\|_2^2 \\ s.t.,\ \widehat{\mathbf{g}}_k =& \sqrt{T}\mathbf{F}\mathbf{P}^\top\mathbf{h}_k,\ k=1,...,K \end{aligned}, \tag{5}$$

where $\widehat{\mathbf{G}} = [\widehat{\mathbf{g}}_1, \widehat{\mathbf{g}}_2, ..., \widehat{\mathbf{g}}_K]$ ($\widehat{\mathbf{g}}_k = \sqrt{T}\mathbf{F}\mathbf{P}^\top\mathbf{h}_k, k = 1, ..., K$) is an auxiliary variable matrix. In equation (5), the symbol $\widehat{\ }$ denotes the discrete Fourier transform form of a given signal, and $\mathbf{F}$ is the orthonormal $T\times T$ matrix of complex basis vectors to map any T dimensional vectorized signal into the Fourier domain (such as $\widehat{\mathbf{a}} = \sqrt{T}\mathbf{F}\mathbf{a}$, $\mathbf{a}\in\mathbb{R}^{T\times 1}$). The model in equation (5) is bi-convex, and can be minimized to obtain a local optimal solution using the alternating direction method of multipliers (ADMM) [4]. The augmented Lagrangian form of equation (5) can be formulated as

$$\begin{aligned} &L(\mathbf{H},\widehat{\mathbf{G}},\mathbf{w},\widehat{\mathbf{V}}) \\ &= E(\mathbf{H},\widehat{\mathbf{G}},\mathbf{w}) + \sum_{k=1}^{K}\widehat{\mathbf{v}}_k^\top(\widehat{\mathbf{g}}_k - \sqrt{T}\mathbf{F}\mathbf{P}^\top\mathbf{h}_k) \\ &\quad + \tfrac{\mu}{2}\sum_{k=1}^{K}\left\|\widehat{\mathbf{g}}_k - \sqrt{T}\mathbf{F}\mathbf{P}^\top\mathbf{h}_k\right\|_2^2 \end{aligned}, \tag{6}$$

where $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_K] \in \mathbb{R}^{T\times K}$ is the Lagrange multiplier, and $\widehat{\mathbf{V}} = [\widehat{\mathbf{v}}_1, \widehat{\mathbf{v}}_2, ..., \widehat{\mathbf{v}}_K] \in \mathbb{R}^{T\times K}$ is the corresponding Fourier transform. By introducing $\mathbf{s}_k = \frac{1}{\mu}\mathbf{v}_k$ ($k = 1, 2, ..., K$), the optimization of equation (6) is equivalent to solving equation (7) .

$$\begin{aligned} L(\mathbf{H},\widehat{\mathbf{G}},\mathbf{w},\widehat{\mathbf{S}}) =& \tfrac{1}{2}\left\|\widehat{\mathbf{y}} - \sum_{k=1}^{K}\widehat{\mathbf{x}}_k \odot \widehat{\mathbf{g}}_k\right\|_2^2 \\ &+ \tfrac{\lambda_1}{2}\sum_{k=1}^{K}\|\mathbf{w}\odot\mathbf{h}_k\|_2^2 + \tfrac{\lambda_2}{2}\|\mathbf{w}-\mathbf{w}_r\|_2^2 \\ &+ \tfrac{\mu}{2}\sum_{k=1}^{K}\left\|\widehat{\mathbf{g}}_k - \sqrt{T}\mathbf{F}\mathbf{P}^\top\mathbf{h}_k + \widehat{\mathbf{s}}_k\right\|_2^2 \end{aligned}, \tag{7}$$

where $\widehat{\mathbf{S}} = [\widehat{\mathbf{s}}_1, \widehat{\mathbf{s}}_2, ..., \widehat{\mathbf{s}}_K] \in \mathbb{R}^{T\times K}$.

Then, the ADMM algorithm is adopted by alternately solving the following subproblems:

Subproblem H: If $\widehat{\mathbf{G}}$, $\mathbf{w}$ and $\widehat{\mathbf{S}}$ are given, the optimal $\mathbf{H}^*$ can be obtained as

$$\begin{aligned} \mathbf{h}_k^* &= \arg\min_{\mathbf{h}_k}\left\{\begin{array}{l} \frac{\lambda_1}{2}\|\mathbf{w}\odot\mathbf{h}_k\|_2^2 + \\ \frac{\mu}{2}\left\|\widehat{\mathbf{g}}_k - \sqrt{T}\mathbf{F}\mathbf{P}^\top\mathbf{h}_k + \widehat{\mathbf{s}}_k\right\|_2^2 \end{array}\right\} \\ &= \left[\lambda_1\mathbf{W}^\top\mathbf{W} + \mu T\mathbf{P}^\top\mathbf{P}\right]^{-1}\mu T\mathbf{P}\left(\mathbf{s}_k + \mathbf{g}_k\right) \\ &= \frac{\mu T\mathbf{p}\odot(\mathbf{s}_k + \mathbf{g}_k)}{\lambda_1(\mathbf{w}\odot\mathbf{w}) + \mu T\mathbf{p}} \end{aligned}, \tag{8}$$

[1] If there is no third term, the solution of $\mathbf{w}$ will be degraded, i.e., $\mathbf{w} = \mathbf{0}$.

where $\mathbf{W} = diag(\mathbf{w}) \in \mathbb{R}^{T\times T}$ represents the diagonal matrix and $\mathbf{p} = [P_{11}, P_{22}, ..., P_{TT}]^\top$ is the column vector composed by the diagonal elements of the cropping matrix $\mathbf{P}$ (For $\mathbf{P}$, we also have $\mathbf{P}^\top\mathbf{P} = \mathbf{P}$). Equation (8) shows that the solution of $\mathbf{h}_k$ merely requires the element-wise multiplication and the inverse fast Fourier transform (i.e., $\mathbf{s}_k = \frac{1}{\sqrt{T}}\mathbf{F}^\top\widehat{\mathbf{s}}_k$ and $\mathbf{g}_k = \frac{1}{\sqrt{T}}\mathbf{F}^\top\widehat{\mathbf{g}}_k$). Thus, the computation complexities of solving $\mathbf{h}_k$ and all $\mathbf{H}$ are $O(T\log T)$ and $O(KT\log T)$ respectively.

Subproblem $\hat{\mathbf{G}}$: If other variables in equation (7) are fixed, the optimal $\hat{\mathbf{G}}^*$ can be estimated by solving the optimization problem (9).

$$\widehat{\mathbf{G}}^* = \arg\min_{\widehat{\mathbf{G}}} \left\{ \begin{array}{l} \frac{1}{2}\left\|\widehat{\mathbf{y}} - \sum_{k=1}^{K}\widehat{\mathbf{x}}_k \odot \widehat{\mathbf{g}}_k\right\|_2^2 + \\ \frac{\mu}{2}\sum_{k=1}^{K}\left\|\widehat{\mathbf{g}}_k - \sqrt{T}\mathbf{F}\mathbf{P}^\top\mathbf{h}_k + \widehat{\mathbf{s}}_k\right\|_2^2 \end{array} \right\}. \tag{9}$$

However, it is difficult to optimize the problem (9) due to its high computation complexity. Thus, we consider processing on all channels of each pixel, and reformulate the optimization problem (9) as

$$\mathcal{V}_j^*(\widehat{\mathbf{G}}) = \arg\min_{\mathcal{V}_j(\widehat{\mathbf{G}})} \left\{ \begin{array}{l} \frac{1}{2}\left\|\widehat{\mathbf{y}}_j - \mathcal{V}_j(\widehat{\mathbf{X}})^\top\mathcal{V}_j(\widehat{\mathbf{G}})\right\|_2^2 + \\ \frac{\mu}{2}\sum_{k=1}^{K}\left\|\mathcal{V}_j(\widehat{\mathbf{G}}) + \mathcal{V}_j(\widehat{\mathbf{M}})\right\|_2^2 \end{array} \right\}, \tag{10}$$

$$\mathcal{V}_j(\hat{\mathbf{M}}) = \mathcal{V}_j(\hat{\mathbf{S}}) - \mathcal{V}_j(\sqrt{T}\mathbf{F}\mathbf{P}^\top\mathbf{H}), \tag{11}$$

where $\mathcal{V}_j(\hat{\mathbf{g}}) \in \mathbb{R}^{K\times 1}$ denotes the values of all channels of filter $\hat{\mathbf{g}}$ on pixel j. Then, the analytical solution of equation (10) can be obtained as

$$\begin{aligned}\mathcal{V}_j^*(\widehat{\mathbf{G}}) = \frac{1}{\mu T}\left(\mathbf{I} - \frac{\mathcal{V}_j(\widehat{\mathbf{X}})\mathcal{V}_j(\widehat{\mathbf{X}})^\top}{\mu T + \mathcal{V}_j(\widehat{\mathbf{X}})^\top\mathcal{V}_j(\widehat{\mathbf{X}})}\right) \\ \left(\widehat{\mathbf{y}}_j\mathcal{V}_j(\widehat{\mathbf{X}}) + \mu V_j(\sqrt{T}\mathbf{F}\mathbf{P}^\top\mathbf{H}) - \mu V_j(\widehat{\mathbf{S}})\right)\end{aligned}. \tag{12}$$

The derivation of equation (12) uses the Sherman Morrsion formula: $\left(\mathbf{A} + \mathbf{u}\mathbf{v}^\top\right)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^\top\mathbf{A}^{-1}}{1+\mathbf{v}^\top\mathbf{A}^{-1}\mathbf{u}}$ (here $\mathbf{u}$ and $\mathbf{v}$ are two column vectors and $\mathbf{u}\mathbf{v}^\top$ is a rank-one matrix).

Solving $\mathbf{w}$: If $\mathbf{H}$, $\widehat{\mathbf{G}}$ and $\widehat{\mathbf{S}}$ are fixed, the closed-form solution regrading $\mathbf{w}$ can be determined as

$$\begin{aligned}\mathbf{w}^* &= \arg\min_{\mathbf{w}}\left\{\frac{\lambda_1}{2}\sum_{k=1}^{K}\|\mathbf{N}_k\mathbf{w}\|_2^2 + \frac{\lambda_2}{2}\|\mathbf{w} - \mathbf{w}^r\|_2^2\right\} \\ &= (\lambda_1\sum_{k=1}^{K}\mathbf{N}_k^\top\mathbf{N}_k + \lambda_2\mathbf{I})^{-\mathbf{1}}\lambda_2\mathbf{w}^r \\ &= \frac{\lambda_2\mathbf{w}^r}{\lambda_1\sum_{k=1}^{K}\mathbf{h}_k\odot\mathbf{h}_k + \lambda_2\mathbf{1}}\end{aligned}, \tag{13}$$

where $\mathbf{N}_k = \text{diag}(\mathbf{h}_k) \in \mathbb{R}^{T\times T}$. In practice, we utilize an additional ADMM solver to obtain the weight $\mathbf{w}^*$ for better convergence. Some representative examples of the learned weights are shown in Figure 2. From this figure, we can see that the adaptive spatial regularization learning works well in introducing large penalties on some unreliable regions, thereby encouraging the learned filters to focus more on the reliable regions of the tracked object in the next iteration.

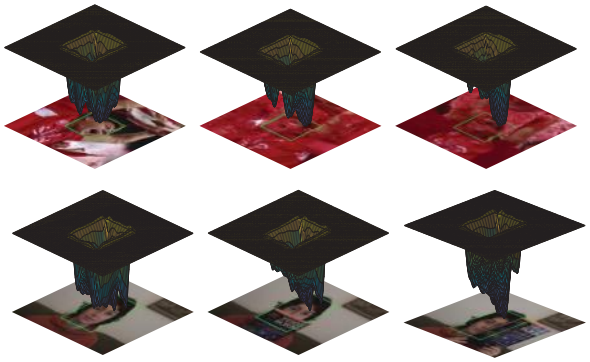


Figure 2. The visualization of the adaptive spatial regularization. For each pixel, a larger value of the adaptive spatial regularization will give a greater learning penalty of the filter at this pixel. Better viewed in color and zoom in for details.

Lagrangian Multiplier Update: We update Lagrangian multipliers as

$$\widehat{\mathbf{S}}^{i+1} = \widehat{\mathbf{S}}^i + \widehat{\mathbf{G}}^{i+1} - \widehat{\mathbf{H}}^{i+1}, \tag{14}$$

where $\widehat{\mathbf{S}}^i$ denotes the Fourier transform of the Lagrangian in the previous state, $\hat{\mathbf{G}}^{(i+1)}$ and $\hat{\mathbf{H}}^{(i+1)}$ are the current solutions to the two subproblems above at iteration $i+1$. The regularization constant μ is commonly set as $\mu^{(i+1)} = \min(\mu_{\max}, \beta\mu^{(i)})$ [4].

Thus, the optimization process can be conducted by iteratively applying the four steps above, including (1) solving $\mathbf{H}$, (2) solving $\widehat{\mathbf{G}}$, (3) solving $\mathbf{w}$ and (4) updating Lagrangian multipliers. After convergence, the optimal filter parameter $\mathbf{H}^*$ (with its Fourier transform $\widehat{\mathbf{G}}^*$) and spatial regularization weight $\mathbf{w}^*$ can be obtained.

4. Object Localization and Scale Estimation

4.1. Object Localization

For tracking, the location of the tracked object can be determined in the Fourier domain as

$$\widehat{\mathbf{r}} = \sum_{k=1}^{K}\widehat{\mathbf{x}}_k \odot \widehat{\mathbf{g}}_k, \tag{15}$$

where $\mathbf{r}$ and $\widehat{\mathbf{r}}$ denote the response map and its Fourier transform. In this work, we adopt ensembles of deep and shallow features for object localization (see implementation details in Section 5). After obtaining the response map, the optimal location can be obtained based on the maximum response.

4.2. Model Update

Similar to other CF-based trackers [18, 11, 15], we train our filters with an online adaptive template scheme. The adaptive method of the template model is as follow:

$$\widehat{\mathbf{X}}_{model}^{new} = (1-\eta)\widehat{\mathbf{X}}_{model}^{old} + \eta\widehat{\mathbf{X}}^{*}, \tag{16}$$

where $\hat{\mathbf{X}}_{model}^{(new)}$ represents the newly updated template model, $\hat{\mathbf{X}}_{model}^{(old)}$ is the old template model and $\widehat{\mathbf{X}}^{*}$ denotes the current observation (η is the online learning rate). In the meanwhile, we update the reference weight as $\mathbf{w}^{r} \leftarrow \mathbf{w}^{*}$. Similar to [11], the reference weight $\mathbf{w}^{r}$ is initialized with a negative Gaussian shape in the first frame. We note that the aforementioned update schemes make our model effectively adapt to the appearance variations of the tracked object and introduce a more reasonable priori for adaptive spatial regularization during the tracking process.

4.3. Scale Estimation

For scale estimation, the previous CF-based trackers [18, 11, 15] usually apply the learned filter on multiple resolutions of the searching area to estimate scale changes, and then select the optimal scale with the maximum response. This manner leads to two imperfections for the CF-based model with deep features: (1) it is very time-consuming to extract multi-scale deep visual features; and (2) it is difficult to estimate the accurate scale based on deep CNN features since the pooling layers make feature descriptions loss some detailed information.

In this work, we attempt to learn two CF models (one location CF is for object localization and the other scale CF is for scale estimation). The location CF model for object localization is trained on ensembles of deep and shallow features. Although the extraction process of this CF model is time-consuming, it merely requires to be extracted on one scale search region during the tracking process. The scale CF model for scale estimation is trained on efficient shallow features (HOG features in this work). During the tracking process, we apply this CF model on five scale search regions and obtain their related response maps. Then, the best scale is determined based on the scale corresponding to the maximum score of five response maps. The effectiveness of our designed scale estimation scheme is verified in Section 5.2.

In every frame, the overall framework (Figure 4.3) first estimates the position using the location CF model with complicated features, and then applies the scale CF model to refine the scale based on five scale HOG feature maps.

5. Experiments

Our tracker is implemented based on the MATLAB2017a platform with the MatConvNet toolbox, and runs on a PC machine with an Intel i7 8700 CPU, 32GB RAM and a single NVIDIA GTX 1080Ti GPU, 11G memory. The tracking speed of our tracker is 28*fps* approximately, which makes our tracker meet the real-time requirement. For localization, we exploit an ensemble of deep (Norm1 from VGG-M, Conv4-3 from VGG-16) and handcrafted (HOG) features for object representation. Besides, we merely use five-scale HOG features for scale estimation. The regularization parameters λ_1 and λ_2 are empirically chosen as λ_1= 0.2 and λ_2= 0.001, respectively. We set the learning rates of our ASRCF model as $\eta = 0.0186$, and use three-step iterations for the ADMM optimization process. The penalty factor μ of ADMM is initially set to 1 and then updated by $\mu^{(i+1)} = \min(\mu_{\max}, \beta\mu^{(i)})$, where $\beta = 10$ and $\mu_{\max} = 10^4$. Our project is available on the website: `http://github.com/Daikenan/ASRCF`.

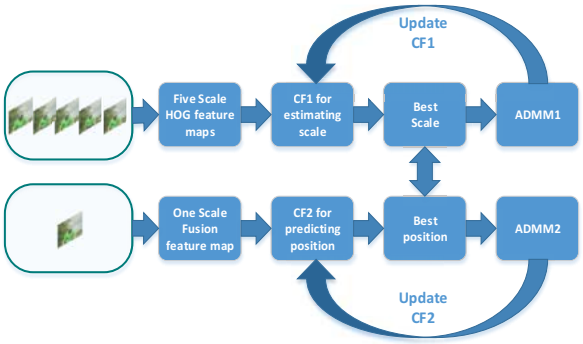


Figure 3. The tracking framework of location and scale CF models.

In this section, we demonstrate the effectiveness of our tracker on the OTB2015 [35], TC128 [27] VOT2016 [19], VOT2017 [20] and LaSOT [14] datasets.

5.1. Quantitative Evaluation

OTB2015 Dataset. The OTB2015 [35] dataset is one of the most popular tracking benchmarks which consists of 100 challenging image sequences with 11 different attributes, such as illumination variation (IV), scale variation (SV), occlusion (OCC), deformation (DEF), motion blur (MB), fast motion (FM), in-plane Rotation (IPR), out-of-plane rotation (OPR), out-of-view (OV), background clutters (BC) and low resolution (LR). The one pass evaluation (OPE) is employed to evaluate different trackers based on two criteria: distance precision and overlap success.

We compare our tracker against recent state-of-the-art trackers including ECO [8], MDNet [30], LSART [31], C-COT [13], DaSiamRPN [39], SiamRPN [21], DeepSRD-CF [10], ACT [5], BACF [15], StructSiam [38], CF2 [29], SRDCF [11], SiamFC [2], Staple [1], CFNet [1] and KCF [18]. Figure 4 reports both precision and success plots of different trackers in terms of the OPE rule. Overall, the proposed tracking algorithm achieves almost the best result with an AUC score of 0.692 and a distance precision rate of 0.922. In Table 2, we summarize both accuracies and speeds of top-5 trackers on OTB2015. Among these top-

ranked methods, our tracker achieves almost the best accuracy and the fastest speed (the only tracker with real-time performance).

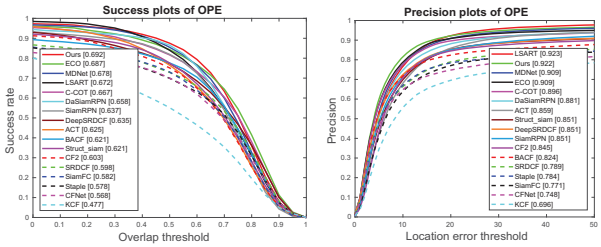


Figure 4. Precision and success plots on OTB2015 [35]. The legend contains the average distance precision score at 20 pixels and the area-under-the-curve (AUC) score for each tracker.

Table 2. Accuracy and speed comparisons of top-5 trackers on the OTB2015 dataset. The best two results are shown in **red** and **blue** fonts, respectively.

	C-COT	LSART	MDNet	ECO	Ours
Success	0.667	0.672	0.678	**0.687**	**0.692**
Precision	0.896	**0.923**	0.909	0.909	**0.922**
GPU/CPU	CPU	GPU	GPU	GPU	GPU
FPS	0.7	1.3	1.7	**17.9**	**28.0**

Figure 6 illustrates overlap success plots of different trackers with 6 attributes (such as background clutter, deformation, occlusion, scale variation and so on). We can see that our tracker achieves almost the best performance in these attributes. First, our tracker performs well under background clutter, deformation and occlusion conditions, and obtains 1.6%, 1.6% and 0.8% gain respectively than the second best tracker (ECO [8]). This is mainly owed to the proposed adaptive spatial regularization, which makes the learned filter focus on the reliable features of the tracked object and alleviate the effects of unexpected noises within the object region. In addition, our tracker works well in handling scale variation based on the designed scale estimation scheme using multi-scale shallow features.

TC128 Dataset. We perform comparisons on the TC128 [27] dataset, which consists of 128 challenging color sequences. We compare our tracker with 8 state-of-the-art trackers including ECO [8], C-COT [13], SRDCF [11], SRDCFdecon, DeepSRDCF [10], MCCT [34], BACF [15], MCPF [37] and 32 more default trackers in TC128. The results of top 15 trackers are reported in Figure 7, from which we can see that the proposed tracker performs the best in terms of both precision and success criterion.

VOT2016 Dataset. We also perform comparisons on the VOT2016 dataset [19] which contains 60 challenging sequences. During the test phase, the tracker will be reset if there is no overlap between prediction and groundtruth. The expected average overlap (EAO) considering both bounding box overlap (accuracy) and reset times (robustness) serves as the major evaluation metric on VOT2016. In Table 3(a),

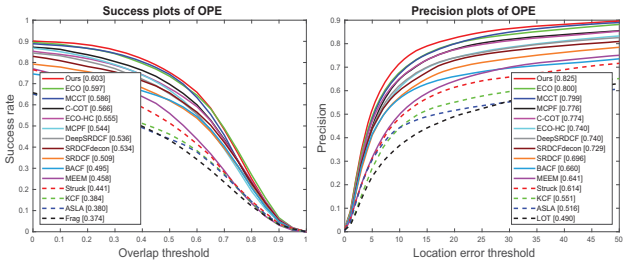


Figure 7. Performance evaluation on the TC128 dataset in terms of success and precision plots.

we compare our method with top-10 trackers including C-COT, TCNN, SSAT, MLDF, Staple, DDC, EBT, SRBT, STAPLE+ and DNT. Table 3 shows that our tracker achieves the best in terms of EAO and R scores, furthermore, our tracker is much faster than the second best tracker (C-COT).

VOT2017 Dataset. The VOT2017 [20] dataset contains 60 challenging sequences (replacing some simple sequences with more difficult ones in VOT2016) and has more accurate groundtruth. The evaluation criteria in VOT2017 is same as that in VOT2016. In Table 3(b), we compare our method with top-10 trackers in the VOT2017 [20] official report. The compared trackers include LSART [31], ECO [8], CFCF [16], GNet, MCCT [34], C-COT [13], C-SRDCF [28], SiamDCF, MCPF [37] and CRT [6]. Table 3(b) shows that our tracker achieves the best performance in terms of EAO while maintaining very competitive A and R scores. As presented before, our tracker is much faster than the second best tracker (LSART).

LaSOT Dataset. The LaSOT [14] dataset is a recent large-scale dataset with 1,400 sequences and more than 3.5M frames in total (the average frame length is more than 2,500 frames). We also follow the one-pass evaluation to compare different trackers based on three criteria (precision, normalized precision and success). The success and precision plots are reported to compare the proposed trackers with 34 trackers reported in [14]. We refer the readers to [14] for more detailed descriptions. Figure 8 shows that our tracker also achieves very competitive results, especially better than all CF-based methods (e.g., ECO [8] and STRCF [22]).

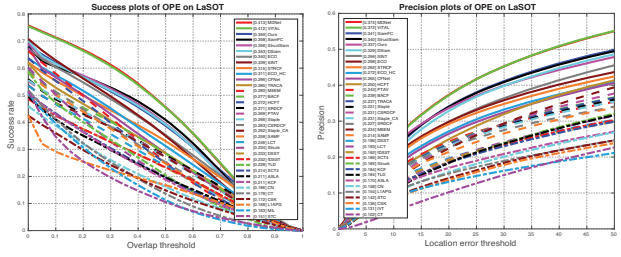


Figure 8. Performance evaluation on the LaSOT dataset in terms of success and precision plots.

Figure 5. Qualitative evaluation of our tracker and related algorithms on the *Bird1*, *Biker*, *Freeman4*, *Human3* and *Singer2* sequences.

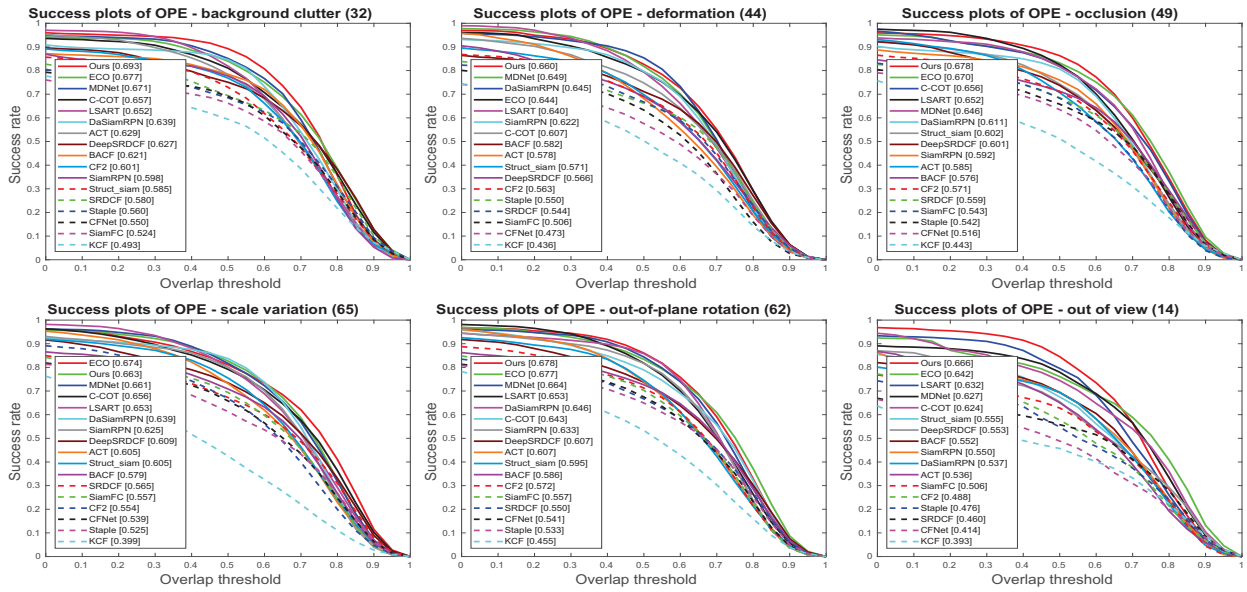


Figure 6. Evaluation of different trackers with 6 attributes on the OTB-2015 [35] dataset. The legend contains the average distance precision score at 20 pixels and the area-under-the-curve score for each tracker.

Table 3. Performance evaluation on the VOT2016 [19] and VOT2017 [20] datasets. In this table, we compare our method with top-10 trackers in the VOT2016 [19] and VOT2017 [20] official reports. The results are presented in terms of expected average overlap (EAO), accuracy rank (A) and robustness rank (R). The best three results are shown in **red**, **blue** and **green** colors, respectively.

(a) VOT 2016

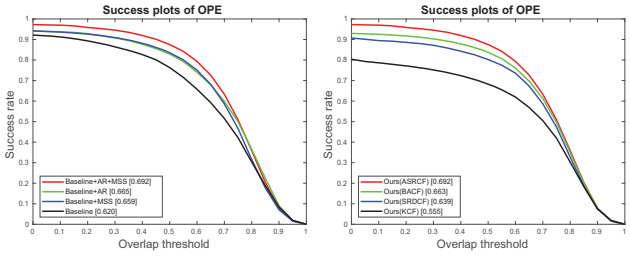
	DNT	STAPLE+	SRBT	EBT	DDC	Staple	MLDF	SSAT	TCNN	C-COT	Ours
EAO	0.278	0.286	0.290	0.291	0.293	0.295	0.311	0.321	**0.325**	**0.331**	**0.391**
A	0.515	**0.557**	0.496	0.465	0.541	0.544	0.490	**0.577**	0.554	0.539	**0.563**
R	0.329	0.368	0.350	0.252	0.345	0.378	**0.233**	0.291	0.268	**0.238**	**0.187**

(b) VOT 2017

	MCPF	SiamDCF	CSRDCF	C-COT	MCCT	GNet	ECO	CFCF	CFWCR	LSART	Ours
EAO	0.248	0.249	0.256	0.267	0.270	0.274	0.280	0.286	**0.303**	**0.323**	**0.328**
A	**0.510**	0.500	0.491	0.494	**0.525**	0.502	0.483	**0.509**	0.484	0.493	0.494
R	0.427	0.473	0.356	0.318	0.323	0.276	0.276	0.281	**0.267**	**0.218**	**0.234**

5.2. Ablation Studies

Effectiveness of Different Components. We conduct the ablation studies to verify the effectiveness of key components in our tracker, and report the comparison results in Figure 9(a). The basic notions are as follows. (1) 'Baseline' denotes the method that does not exploit the adaptive spatial regularization and the designed scale estimation scheme. (2) 'Baseline+AR' means the baseline method with adding the adaptive spatial regularization. (3) 'Baseline+MSS' stands for the baseline method replacing multi-scale estimation on the original feature space with multi-scale estimation on shallow features. (4) 'Baseline+AR+MSS' is our final tracker that combines the baseline method with both adaptive spatial regularization and multi-scale estimation on shallow features. From Figure 9(a), we can see that both adaptive spatial regularization and designed scale estimation scheme contribute to the substantial improvement over the baseline method. Besides, our final tracker improves the baseline method by 7.1% and 6.5% in terms of success and precision criterion, respectively.



(a) (b)

Figure 9. Ablation analysis using the OTB-2015 [35] Dataset.

Different Variants of Our ASRCF Model. In Table 1, we point out the well-known KCF [18], SRDCF [11] and BACF [15] methods are all special cases of our ASRCF model. To show the effectiveness of ASRCF, we compare it with those special cases with the same feature extraction and scale estimation scheme[2]. From Figure 9(b), we can see that the SRDCF and BACF methods performs better than the original KCF algorithm, which can be attributed to the adopted spatial constraints on filter coefficients. Our ASRCF model provides an adaptive spatial regularization, which makes the tracker achieve the best results.

6. Conclusions

In this work, we attempt to introduce an adaptive spatial regularization into the objective function of correlation filters (denoted as ASRCF). Compared with previous works using fixed spatial constraints, this regularization could be effectively learned with respect to a specific object being tracked and updated to consider the appearance variations during the tracking process. Our ASRCF model is effectively optimized using the ADMM algorithm, which can learn the reliable filter coefficients and therefore make our tracker robust. To speed up our tracker, we exploit two CF models to estimate the location and scale separately. One CF model with complicated features is responsible for accurate localization. The other CF model with multi-scale shallow features is aimed to accelerate scale estimation. Extensive experimental results show that our ASRCF tracker performs significantly better than many state-of-the-art tracking algorithms, with a real-time speed of 28*fps*.

Acknowledgement. This paper is supported in part by National Natural Science Foundation of China Nos. 61872056, 61771088, 61751212, 61725202 and 61829102, and in part by the Fundamental Research Funds for the Central Universities under Grant No. DUT18JC30. This work is also sponsored by CCF-Tencent Open Research Fund.

[2] In the original source codes, the adopted features and scale estimation manners of KCF, SRDCF, BACF and our methods are different; thus, direct comparisons among them are not fair. Here, we adopt the same feature extraction and scale estimation scheme for fair evaluation.

References

[1] Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip H. S. Torr. Staple: Complementary learners for real-time tracking. In *CVPR*, 2016.

[2] Luca Bertinetto, Jack Valmadre, João F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *ECCVW*, 2016.

[3] David S. Bolme, J. Ross Beveridge, Bruce A. Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *CVPR*, 2010.

[4] Stephen P. Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[5] Boyu Chen, Dong Wang, Peixia Li, Shuang Wang, and Huchuan Lu. Real-time 'Actor-Critic' tracking. In *ECCV*, 2018.

[6] Kai Chen and Wenbing Tao. Convolutional regression for visual tracking. *IEEE Transcations on Image Processing*, 27(7):3611–3620, 2018.

[7] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[8] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ECO: efficient convolution operators for tracking. In *CVPR*, 2017.

[9] Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, and Michael Felsberg. Accurate scale estimation for robust visual tracking. In *BMVC*, 2014.

[10] Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, and Michael Felsberg. Convolutional features for correlation filter based visual tracking. In *ICCVW*, 2015.

[11] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Learning spatially regularized correlation filters for visual tracking. In *ICCV*, 2015.

[12] Martin Danelljan, Fahad Shahbaz Khan, Michael Felsberg, and Joost van de Weijer. Adaptive color attributes for real-time visual tracking. In *CVPR*, 2014.

[13] Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *ECCV*, 2016.

[14] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. LaSOT: A high-quality benchmark for large-scale single object tracking. *CoRR*, abs/1809.07845, 2018.

[15] Hamed Kiani Galoogahi, Ashton Fagg, and Simon Lucey. Learning background-aware correlation filters for visual tracking. In *ICCV*, 2017.

[16] Erhan Gundogdu and A. Aydin Alatan. Good features to correlate for visual tracking. *IEEE Transcations on Image Processing*, 27(5):2526–2540, 2018.

[17] João F. Henriques, Rui Caseiro, Pedro Martins, and Jorge P. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *ECCV*, 2012.

[18] Joao F. Henriques, Caseiro Rui, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596, 2015.

[19] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman P. Pflugfelder, Luka Cehovin, Tomás Vojír, Gustav Häger, Alan Lukezic, and Gustavo Fernández. The visual object tracking VOT2016 challenge results. In *ECCVW*, 2016.

[20] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman P. Pflugfelder, Luka Cehovin Zajc, Tomas Vojir, and Gustav Häger. The visual object tracking VOT2017 challenge results. In *ICCVW*, 2017.

[21] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, 2018.

[22] Feng Li, Cheng Tian, Wangmeng Zuo, Lei Zhang, and Ming-Hsuan Yang. Learning spatial-temporal regularized correlation filters for visual tracking. In *CVPR*, 2018.

[23] Feng Li, Yingjie Yao, Peihua Li, David Zhang, Wangmeng Zuo, and Ming-Hsuan Yang. Integrating boundary and center correlation filters for visual tracking with aspect ratio variation. In *ICCVW*, 2017.

[24] Peixia Li, Dong Wang, Lijun Wang, and Huchuan Lu. Deep visual tracking: Review and experimental comparison. *Pattern Recognition*, 76:323–338, 2018.

[25] Xi Li, Weiming Hu, Chunhua Shen, Zhongfei Zhang, Anthony R. Dick, and Anton van den Hengel. A survey of appearance models in visual object tracking. *ACM Transactions on Intelligent Systems and Technology*, 4(4):58:1–58:48, 2013.

[26] Yang Li and Jianke Zhu. A scale adaptive kernel correlation filter tracker with feature integration. In *ECCVW*, 2014.

[27] Pengpeng Liang, Erik Blasch, and Haibin Ling. Encoding color information for visual tracking: Algorithms and benchmark. *IEEE Transcations on Image Processing*, 24(12):5630–5644, 2015.

[28] Alan Lukezic, Tomas Vojir, Luka Cehovin Zajc, Jiri Matas, and Matej Kristan. Discriminative correlation filter with channel and spatial reliability. In *CVPR*, 2017.

[29] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang. Hierarchical convolutional features for visual tracking. In *ICCV*, 2015.

[30] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, 2016.

[31] Chong Sun, Huchuan Lu, and Ming-Hsuan Yang. Learning spatial-aware regressions for visual tracking. In *CVPR*, 2018.

[32] Chong Sun, Dong Wang, Huchuan Lu, and Ming-Hsuan Yang. Correlation tracking via joint discrimination and reliability learning. In *CVPR*, 2018.

[33] Yuxuan Sun, Chong Sun, Dong Wang, You He, and Huchuan Lu. Roi pooled correlation filters for visual tracking. In *CVPR*, 2019.

[34] Ning Wang, Wengang Zhou, Qi Tian, Richang Hong, Meng Wang, and Houqiang Li. Multi-cue correlation filters for robust visual tracking. In *CVPR*, 2018.

[35] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1834–1848, 2015.

[36] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4):13, 2006.

[37] Tianzhu Zhang, Changsheng Xu, and Ming-Hsuan Yang. Multi-task correlation particle filter for robust object tracking. In *CVPR*, 2017.

[38] Yunhua Zhang, Lijun Wang, Jinqing Qi, Dong Wang, Mengyang Feng, and Huchuan Lu. Structured siamese network for real-time visual tracking. In *ECCV*, 2018.

[39] Zheng Zhu, Qiang Wang, Li Bo, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *ECCV*, 2018.