Correlation Tracking via Joint Discrimination and Reliability Learning

Chong Sun[1], Dong Wang[1]*, Huchuan Lu[1], Ming-Hsuan Yang[2]
[1]School of Information and Communication Engineering, Dalian University of Technology, China
[2]Electrical Engineering and Computer Science, University of California, Merced, USA
waynecool@mail.dlut.edu.cn, {wdice,lhchuan}@dlut.edu.cn, mhyang@ucmerced.edu

Abstract

For visual tracking, an ideal filter learned by the correlation filter (CF) method should take both discrimination and reliability information. However, existing attempts usually focus on the former one while pay less attention to reliability learning. This may make the learned filter be dominated by the unexpected salient regions on the feature map, thereby resulting in model degradation. To address this issue, we propose a novel CF-based optimization problem to jointly model the discrimination and reliability information. First, we treat the filter as the element-wise product of a base filter and a reliability term. The base filter is aimed to learn the discrimination information between the target and backgrounds, and the reliability term encourages the final filter to focus on more reliable regions. Second, we introduce a local response consistency regular term to emphasize equal contributions of different regions and avoid the tracker being dominated by unreliable regions. The proposed optimization problem can be solved using the alternating direction method and speeded up in the Fourier domain. We conduct extensive experiments on the OTB-2013, OTB-2015 and VOT-2016 datasets to evaluate the proposed tracker. Experimental results show that our tracker performs favorably against other state-of-the-art trackers.

1. Introduction

Visual tracking is a hot topic for its wide applications in many computer vision tasks, such as video surveillance, behaviour analysis, augmented reality, to name a few. Though many trackers [12, 26, 20, 25, 16] have been proposed to address this task, designing a robust visual tracking system is still challenging as the tracked target may undergo large deformations, rotations and other challenges.

Numerous recent studies apply the correlation filter (CF) for robust visual tracking. With low computational load, the CF-based tracker can exploit large numbers of cyclically shifted samples for learning, thus showing superior performance. However, as the correlation filter takes the entire image as the positive sample and the cyclically shifted images as negative ones, the learned filters are likely to be influenced by the background regions. Existing methods (*e.g.* [8, 10, 5]) address this problem by incorporating a spatial regularization on the filter, so that the learned filter weights focus on the central part of the target object. In [14], the authors prove that the correlation filter method can be used to simulate the conventional ridge regression method. By multiplying the filter with a binary mask, the tracker is able to generate the real training samples having the same size as the target object, and thus better suppressing the background regions. However, this method has two limitations: first, it exploits the augmented Lagrangian method for model learning, which limits the model extension; second, even though the background region outside the bounding box is suppressed, the tracker may also be influenced by the background region inside the bounding box.

With the great success of deep convolutional neural network (CNN) in object detection and classification, more and more CF based trackers resort to the pre-trained CNN model for robust target representation [29, 28, 5]. Since most of CNN models are pre-trained with respect to the task

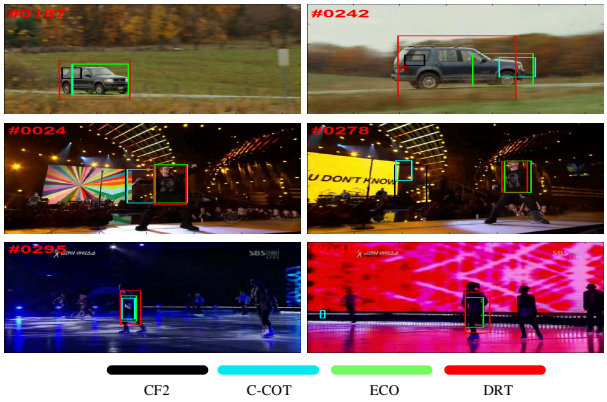


Figure 1. Example tracking results of different methods on the OTB dataset. Our tracker (DRT) has comparable or better results compared with the existing best tracker ECO.

*Corresponding Author

of object classification or detection, they tend to retain the features useful for distinguishing different categories of objects, and lose much information for instance level classification. Thus, the responses of the feature map are usually sparsely and non-uniformly distributed, which makes the learned filter weights inevitably highlight the high response regions. In this case, the tracking results are dominated by such high response regions, while these regions are in fact not always reliable (see Figure 3 for an example).

In this paper, we present a novel CF-based optimization problem to clearly learn the discrimination and reliability information, and then develop an effective tracking method (denoted as DRT). The concept of the base filter is proposed to focus on discrimination learning. To do this, we introduce the local response consistency constraint into the traditional CF framework. This constraint ensures that the responses generated by different sub-regions of the base filter have small difference, thereby emphasizing the similar importance of each sub-region. The reliability weight map is also considered in our formula. It is online jointly learned with the base filter and is aimed at learning the reliability information. The base filter and reliability term are jointly optimized by the alternating direction method, and their element-wise product produces effective filter weights for the tracking task. Finally, we conduct extensive experiments on three benchmark datasets to demonstrate the effectiveness of our method (see Figure 1 and Section 6).

Our contributions are four folds:

- Our work is the first attempt to jointly model both discrimination and reliability information using the correlation filter framework. We treat an ideal filter as the element-wise product of a base filter and a reliability term and propose a novel optimization problem with insightful constraints.
- The local response consistency constraint is introduced to ensure that different sub-regions of the base filter have similar importance. Thus, the base filter will highlight the entire target even though the feature maps may be dominated by some specific regions.
- The reliability weight map is exploited to depict the importance of each sub-region in the filter (i.e. reliability learning) and is online jointly learned with the base filter. Being insusceptible to the response distributions of the feature map, it can better reflect the real tracking performance for different sub-regions.
- Our tracker achieves remarkable tracking performance on the OTB-2013, OTB-2015 and VOT-2016 benchmarks. Our tracker has the best results on all the reported datasets.

2. Relate Work

Correlation filters (CF) have shown great success in visual tracking for their efficient learning process. In this section, we briefly introduce the CF-based trackers that are closely related to our work.

The early CF-based trackers exploit a single feature channel as input, and thus usually have very impressive tracking speed. The MOSSE tracker [3] exploits the adaptive correlation filter, which optimizes the sum of squared error. Henriques *et al*. [11] introduce the kernel trick into the correlation filter formula. By exploiting the property of the circulant matrix, they provide an efficient solver in the Fourier domain. The KCF [12] tracker further extends the method [11], and shows improved performance can be achieved when muti-channel feature maps are used. Motivated by the effectiveness of the multi-channel correlation filter methods and the convolution neural network, several methods are proposed to combine them both. Deeply investing the representation property of different convolution layers in the CNN model, Ma *et al*. [21] propose to combine feature maps generated by three layers of convolution filters, and introduce a coarse-to-fine searching strategy for target localization. Danelljan *et al*. [10] propose to use the continuous convolution filter for combinations of feature maps with different spatial resolutions. As fewer model parameters are used in the model, the tracker [10] is insusceptible to the over-fitting problem, and thus has superior performance than [21]. Another research hotspot for the CF-based methods is how to suppress the boundary effects. Typical methods include the trackers [8] and [14]. In the SRDCF tracker [8], a spatial regular term is exploited to penalize the filter coefficients near the boundary regions. Different from [8], the BACF tracker [14] directly multiplies the filter with a binary matrix. This tracker can generate real positive and negative samples for training while at the same time share the computation efficiency of the original CF method. Compared to our method, these trackers have not attempted to suppress the background regions inside the target bounding box, and their learned filter weights tend to be dominated by the salient regions in the feature map.

Patch-based correlation filters have also been widely exploited [19, 18]. Liu *et al*. [19] propose an ensemble of part trackers based on the KCF method, and use the peak-to-sidelobe ratio and the smooth constraint of confidence map for combinations of different base trackers. In the method [18], the authors attempt to learn the filter coefficients of different patches simultaneously under the assumption that the motions of sub-patches are similar. Li *et al*. [17] detect the reliable patches in the image, and propose to use the Hough voting-like strategy to estimate the target states based on the sub-patches. Most of the previous patch-based methods intend to address the problems of deformation and partial occlusion explicitly. Different from them, our method is aimed to suppress the influence of the non-uniform energy distribution of the feature maps and conduct a joint learning of both discrimination and reliability.

3. Proposed Method

3.1. Correlation Filter for Visual Tracking

We first briefly revisit the conventional correlation filter (CF) formula. Let $\mathbf{y}=[y_1, y_2, ..., y_K]^\top \in \mathbb{R}^{K\times 1}$ denote gaussian shaped response, and $\mathbf{x}_d \in \mathbb{R}^{K\times 1}$ be the input vector (in the two-dimensional case, it should be a feature map) for the d-th channel, then the correlation filter learns the optimal $\mathbf{w}$ by optimizing the following formula:

$$\dot{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_{k=1}^{K} \left(y_k - \sum_{d=1}^{D} \mathbf{x}_{k,d}^\top \mathbf{w}_d \right)_2^2 + \lambda \left\|\mathbf{w}\right\|_2^2, \quad (1)$$

where $\mathbf{x}_{k,d}$ is the k-step circular shift of the input vector $\mathbf{x}_d$, y_k is the k-th element of $\mathbf{y}$, $\mathbf{w}=\left[\mathbf{w}_1^\top, \mathbf{w}_2^\top, ..., \mathbf{w}_D^\top\right]^\top$ where $\mathbf{w}_d \in \mathbb{R}^{K\times 1}$ stands for the filter of the d-th channel. For circular matrix in CF, K stands for both the dimension of features and the number of training samples. An analytical solution can be found to efficiently solve the optimization problem (1) in the Fourier domain.

3.2. Joint Discrimination and Reliability Modeling

Different from the previous CF-based methods, we treat the filter weight $\mathbf{w}_d$ of the d-th feature channel as the element-wise product of a base filter $\mathbf{h}_d$ and a reliability weight map $\mathbf{v}_d$,

$$\mathbf{w}_d = \mathbf{h}_d \odot \mathbf{v}_d, \quad (2)$$

where $\odot$ is the hadamard product, $\mathbf{h}_d \in \mathbb{R}^{K\times 1}$ is used to denote the base filter, $\mathbf{v}_d \in \mathbb{R}^{K\times 1}$ is the reliability weight for each target region, the values of $\mathbf{v}_d$ corresponding to the non-target region are set to zeros (illustrated in Figure 2).

To learn a compact reliability map, we divide the target region into M patches, and use a variable $\beta_m, m \in \{1, ..., M\}$ to denote the reliability for each patch (β_m is shared across the channels), thus $\mathbf{v}_d$ can be written as

$$\mathbf{v}_d = \sum_{m=1}^{M} \beta_m \mathbf{p}_d^m, \quad (3)$$

where $\mathbf{p}_d^m \in \mathbb{R}^{K\times 1}$ is a binary mask (see Figure 2) which crops the filter region for the m-th patch.

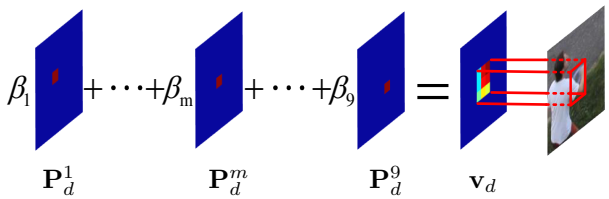


Figure 2. Illustration showing how we compute the reliability map $\mathbf{v}_d$. The computed reliability map only has non-zeros values corresponding to the target region, thus the real positive and negative samples can be generated when we circularly shift the input image.

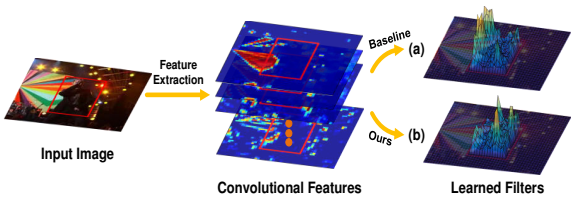


Figure 3. Example showing that our learned filter coefficients are insusceptible to the response distribution of the feature map. In (a) and (b), we compute the square sum of filter coefficients across the channel dimension, and obtain a spatial energy distribution for the learned filter. (a) The baseline method, which does not consider the local consistency regular term and set $\beta_m, m = \{1, ..., M\}$ as 1. (b) The proposed joint learning formula. Compared to our method, the baseline method learns large coefficients on the background (*i.e.* the left-side region in the bounding box).

Based on the previous descriptions, we attempt to jointly learn the base filter $\mathbf{h} = \left[\mathbf{h}_1^\top, ..., \mathbf{h}_D^\top\right]^\top \in \mathbb{R}^{KD\times 1}$ and the reliability vector $\boldsymbol{\beta} = [\beta_1, ..., \beta_M]^\top$ by using the following optimization problem:

$$\begin{aligned} \left[\dot{\mathbf{h}}, \dot{\boldsymbol{\beta}}\right] &= \arg\min_{\mathbf{h},\boldsymbol{\beta}} f\left(\mathbf{h}, \boldsymbol{\beta}; \mathbf{X}\right) \\ s.t.\ \ \theta_{\min} &\le \beta_m \le \theta_{\max},\ \forall m \end{aligned}, \quad (4)$$

where the objective function $f\left(\mathbf{h}, \boldsymbol{\beta}; \mathbf{X}\right)$ is defined as

$$f\left(\mathbf{h}, \boldsymbol{\beta}; \mathbf{X}\right) = f_1\left(\mathbf{h}, \boldsymbol{\beta}; \mathbf{X}\right) + \eta f_2\left(\mathbf{h}; \mathbf{X}\right) + \gamma \left\|\mathbf{h}\right\|_2^2. \quad (5)$$

In this equation, the first term is the data term with respect to the classification error of training samples, the second term is a regularization term to introduce the local response consistency constraint on the filter coefficient vector $\mathbf{h}$, and the last one is a squared ℓ_2-norm regularization to avoid model degradation. In the optimization problem (4), we also add some constraints on the learned reliability coefficients $\beta_1, ..., \beta_M$. These constraints prevent all reliability weights being assigned to a small region of the target especially when the number of training samples is limited, and encourage our model to obtain an accurate weight map. We note that the optimization problem (5) encourages learning more reliable correlation filters (see Figure 4 for example).

Data Term. The data term $f_1(\mathbf{h}, \boldsymbol{\beta}; \mathbf{X})$ is indeed a loss function which ensures that the learned filter has a Gaussian shaped response with respect to the circulant sample matrix. By introducing our basic assumption in equation (3) into the standard CF model, $f_1\left(\mathbf{h}, \boldsymbol{\beta}; \mathbf{X}\right)$ can be rewritten as

$$\begin{aligned} & f_1\left(\mathbf{h}, \boldsymbol{\beta}; \mathbf{X}\right) \\ &= \sum_{k=1}^{K} \left(y_k - \sum_{d=1}^{D} \mathbf{x}_{k,d}^\top \left(\mathbf{v}_d \odot \mathbf{h}_d\right) \right)^2 \\ &= \sum_{k=1}^{K} \left(y_k - \sum_{d=1}^{D} \mathbf{x}_{k,d}^\top \mathbf{V}_d \mathbf{h}_d \right)^2 \\ &= \left\| \mathbf{y} - \sum_{d=1}^{D} \mathbf{X}_d^\top \mathbf{V}_d \mathbf{h}_d \right\|_2^2 \\ &= \left\| \mathbf{y} - \mathbf{X}^\top \mathbf{V} \mathbf{h} \right\|_2^2 \end{aligned}, \quad (6)$$

where $\mathbf{V}_d = \operatorname{diag}(\mathbf{v}_d(1), \mathbf{v}_d(2), ..., \mathbf{v}_d(K)) \in \mathbb{R}^{K\times K}$ is a diagonal matrix, $\mathbf{X}_d = [\mathbf{x}_{1,d}, \mathbf{x}_{2,d}, ..., \mathbf{x}_{K,d}] \in \mathbb{R}^{K\times K}$ is the circulant matrix of the d-th channel, $\mathbf{X} = \left[\mathbf{X}_1^\top, \mathbf{X}_2^\top, ..., \mathbf{X}_D^\top\right]^\top \in \mathbb{R}^{KD\times K}$ stands for a contactated matrix of all circulant matrices from different channels, and $\mathbf{V} = \mathbf{V}_1 \oplus \mathbf{V}_2 \oplus \cdots \oplus \mathbf{V}_D \in \mathbb{R}^{DK\times DK}$ denotes a block diagonal matrix where $\mathbf{V}_d$ is the d-the diagonal block.

Local Response Consistency. The regularization term $f_2(\mathbf{h}; \mathbf{X})$ constrains that the base filter generates consistent responses for different fragments of the cyclically shifted sample. By this means, the base filter learns equal importance for each local region, and reliability information is separated from the base filter. The term $f_2(\mathbf{h}; \mathbf{X})$ can be defined as

$$\begin{aligned} & f_2(\mathbf{h}; \mathbf{X}) \\ &= \sum_{k=1}^{K} \sum_{m,n}^{M} \left(\sum_{d=1}^{D} (\mathbf{P}_d^m \mathbf{x}_{k,d})^\top \mathbf{h}_d - \sum_{d=1}^{D} (\mathbf{P}_d^n \mathbf{x}_{k,d})^\top \mathbf{h}_d \right)^2 \\ &= \sum_{m,n}^{M} \left\| \sum_{d=1}^{D} \mathbf{X}_d^\top \mathbf{P}_d^m \mathbf{h}_d - \sum_{d=1}^{D} \mathbf{X}_d^\top \mathbf{P}_d^n \mathbf{h}_d \right\|_2^2 \\ &= \sum_{m,n}^{M} \left\| \mathbf{X}^\top \mathbf{P}^m \mathbf{h} - \mathbf{X}^\top \mathbf{P}^n \mathbf{h} \right\|_2^2 \end{aligned}, \tag{7}$$

where $\mathbf{P}_d^m = \operatorname{diag}(\mathbf{p}_d^m(1), \mathbf{p}_d^m(2), ..., \mathbf{p}_d^m(K)) \in \mathbb{R}^{K\times K}$, $\mathbf{P}^m = \mathbf{P}_1^m \oplus \mathbf{P}_2^m \oplus \cdots \oplus \mathbf{P}_D^m \in \mathbb{R}^{DK\times DK}$. For each cyclically shifted sample $\mathbf{x}_{k,d}$, $(\mathbf{P}_d^m \mathbf{x}_{k,d})^\top \mathbf{h}_d$ is the response for the m-th fragment of $\mathbf{x}_{k,d}$.

3.3. Joint Discrimination and Reliability Learning

Based on the discussions above, the base filter and the reliability vector can be jointly learned by solving the optimization problem (4), which is a non-convex but differentiable problem for both $\mathbf{h}$ and $\boldsymbol{\beta}$. However, it can be converted into a convex differentiable problem if either $\mathbf{h}$ or $\boldsymbol{\beta}$ is known. Thus, in this work, we attempt to solve the optimal $\mathbf{h}$ and $\boldsymbol{\beta}$ via the alternating direction method.

Solving $\mathbf{h}$. To solve the optimal $\mathbf{h}$, we first compute the derivative of the objective function (5) , then by setting the derivative to be zero, we obtain the following normal equations:

$$\mathbf{A}\mathbf{h} = \mathbf{V}\mathbf{X}\mathbf{y}. \tag{8}$$

The matrix $\mathbf{A}$ is defined as

$$\begin{aligned} \mathbf{A} = \mathbf{g}(\mathbf{V}, \mathbf{X}) + 2\eta \sum_{m=1}^{M} M\mathbf{g}(\mathbf{P}^m, \mathbf{X}) \\ - 2\eta \mathbf{g}(\sum_{m=1}^{M} \mathbf{P}^m, \mathbf{X}) + \gamma \mathbf{I} \end{aligned}, \tag{9}$$

where $\mathbf{g}(\mathbf{\Lambda}, \mathbf{R}) = \mathbf{\Lambda}^\top \mathbf{R} \mathbf{R}^\top \mathbf{\Lambda}$, $\mathbf{R}$ is a circulant matrix and $\mathbf{\Lambda}$ is a diagonal matrix.

In this work, we exploit the conjugate gradient descent method due to its fast convergence speed. The update process can be performed via the following iterative steps [24]:

$$\begin{aligned} \alpha^{(i)} &= {\mathbf{r}^{(i)}}^\top \mathbf{r}^{(i)} / {\mathbf{u}^{(i)}}^\top \mathbf{A}\mathbf{u}^{(i)} \\ \mathbf{h}^{(i+1)} &= \mathbf{h}^{(i)} + \alpha^{(i)} \mathbf{u}^{(i)} \\ \mathbf{r}^{(i+1)} &= \mathbf{r}^{(i)} + \alpha^{(i)} \mathbf{A}\mathbf{u}^{(i)} \\ \mu^{(i+1)} &= \left\|\mathbf{r}^{(i+1)}\right\|_2^2 / \left\|\mathbf{r}^{(i)}\right\|_2^2 \\ \mathbf{u}^{(i+1)} &= -\mathbf{r}^{(i+1)} + \mu^{(i+1)} \mathbf{u}^{(i)} \end{aligned}, \tag{10}$$

where $\mathbf{u}^{(i)}$ denotes the search direction at the i-th iteration, $\mathbf{r}^{(i)}$ is the residual after the i-th iteration. Clearly, the computational load lies in the update of $\alpha^{(i)}$ and $\mathbf{r}^{(i+1)}$ since it requires to compute ${\mathbf{u}^{(i)}}^\top \mathbf{A}\mathbf{u}^{(i)}$ and $\mathbf{A}\mathbf{u}^{(i)}$ in each iteration. As shown in equation (9), the first three terms have the same form. For clarity, we take the first term as an example to show how we compute ${\mathbf{u}^{(i)}}^\top \mathbf{A}\mathbf{u}^{(i)}$ and $\mathbf{A}\mathbf{u}^{(i)}$ efficiently. Let $\mathbf{A}_1$ denote the first term of equation (9), then

$$\begin{aligned} {\mathbf{u}^{(i)}}^\top \mathbf{A}_1 \mathbf{u}^{(i)} &= {\mathbf{u}^{(i)}}^\top \mathbf{V}^\top \mathbf{X}\mathbf{X}^\top \mathbf{V}\mathbf{u}^{(i)} \\ &= \sum_{d=1}^{D} \left\| \mathbf{X}_d^\top \mathbf{V}_d \mathbf{u}_d^{(i)} \right\|_2^2 \\ &= \tfrac{1}{K} \sum_{d=1}^{D} \left\| \widehat{\mathbf{X}}_d^H \odot \mathcal{F}\left(\mathbf{V}_d \mathbf{u}_d^{(i)}\right) \right\|_2^2 \end{aligned}, \tag{11}$$

where $\widehat{\mathbf{X}}_d = \mathcal{F}(\mathbf{x}_d)$ is the Fourier transform of the base vector $\mathbf{x}_d$ (corresponding to the input image without shift), $\mathbf{u}_d^{(i)}$ is the subset of $\mathbf{u}^{(i)}$ corresponding to the d-th channel, $(\cdot)^H$ denotes the conjugate of a vector. Because $\mathbf{V}_d \mathbf{u}_d^{(i)}$ is a vector and $\mathbf{X}_d$ is the circulant matrix, the operation $\mathbf{X}_d^\top (\mathbf{V}_d \mathbf{u}_d^{(i)})$ can be viewed as a circular correlation process and can be efficiently computed in the Fourier domain.

Similarly, $\mathbf{A}_1 \mathbf{u}^{(i)}$ can be computed as

$$\mathbf{A}_1 \mathbf{u}^{(i)} = \mathbf{V}^\top \mathbf{X}\mathbf{X}^\top \mathbf{V}\mathbf{u}^{(i)} = \begin{bmatrix} \mathbf{V}_1^\top \mathbf{X}_1 \sum_{j=1}^{D} \mathbf{X}_j^\top \mathbf{V}_j \mathbf{u}_j^{(i)} \\ \mathbf{V}_2^\top \mathbf{X}_2 \sum_{j=1}^{D} \mathbf{X}_j^\top \mathbf{V}_j \mathbf{u}_j^{(i)} \\ ... \\ \mathbf{V}_D^\top \mathbf{X}_D \sum_{j=1}^{D} \mathbf{X}_j^\top \mathbf{V}_j \mathbf{u}_j^{(i)} \end{bmatrix}. \tag{12}$$

The d-th term $\mathbf{V}_d^\top \mathbf{X}_d \sum_{j=1}^{D} \mathbf{X}_j^\top \mathbf{V}_j \mathbf{u}_j^{(i)}$ can be computed as

$$\begin{aligned} & \mathbf{V}_d^\top \mathbf{X}_d \sum_{j=1}^{D} \mathbf{X}_j^\top \mathbf{V}_j \mathbf{u}_j^{(i)} \\ &= \mathbf{V}_d^\top \mathcal{F}^{-1} \left(\widehat{\mathbf{X}}_d \odot \sum_{j=1}^{D} \widehat{\mathbf{X}}_j^H \odot \mathcal{F}(\mathbf{V}_j \mathbf{u}_j^{(i)}) \right) \end{aligned}, \tag{13}$$

where $\widehat{\mathbf{X}}_j^H \odot \mathcal{F}(\mathbf{V}_j \mathbf{u}_j^{(i)})$ has been computed in equation (11) and can be directly used. The computational complexities of (11) and (12) are therefore $\mathcal{O}(DK \log K)$.

Solving $\boldsymbol{\beta}$. If the filter vector $\mathbf{h}$ is given, the reliability weight vector $\boldsymbol{\beta} = [\beta_1, \beta_2, ..., \beta_M]^\top$ can be obtained by solving the following optimization problem:

$$\dot{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left\| \mathbf{y} - \sum_{d=1}^{D} \mathbf{X}_d^\top \mathbf{V}_d \mathbf{h}_d \right\|_2^2 , \quad s.t. \quad \theta_{\min} \leq \beta_m \leq \theta_{\max}, \ \forall m \tag{14}$$

where the term $f_2(\mathbf{h}; \mathbf{X})$ is ignored as it does not include $\boldsymbol{\beta}$. With some derivations, the problem (14) can be converted as follows:

$$\dot{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \boldsymbol{\beta}^\top \mathbf{C}^\top \mathbf{C} \boldsymbol{\beta} - 2\boldsymbol{\beta}^\top \mathbf{C}^\top \mathbf{y} , \quad s.t. \quad \theta_{\min} < \beta_m < \theta_{\max}, \ \forall m \tag{15}$$

where $\mathbf{C} = \left[\mathbf{C}^1, ..., \mathbf{C}^M\right] \in \mathbb{R}^{K \times M}$, and $\mathbf{C}^m$ is computed as $\mathbf{C}^m = \mathcal{F}^{-1}(\sum_{d=1}^{D} \widehat{\mathbf{X}}_d^H \odot \mathcal{F}(\mathbf{P}_d^m \mathbf{h}_d))$, whose computational complexity is $\mathcal{O}(DK\log(K))$. This optimization problem is a convex quadratic programming method, which can be efficiently solved via standard quadratic programming.

3.4. Model Extension

We note the proposed model (Section 3.2) and its optimization method (Section 3.3) are defined and derived based on the training sample in one frame. Recent studies (like [10]) demonstrate that it is more effective to learn the correlation filter using a set of training samples from multiple frames. Thus, we can extend our optimization problem (4) to consider multi-frame information as follows:

$$\left[\dot{\mathbf{h}}, \dot{\boldsymbol{\beta}}\right] = \arg\min_{\mathbf{h}, \boldsymbol{\beta}} \sum_{t=1}^{T} \kappa_t f(\mathbf{h}, \boldsymbol{\beta}; \mathbf{X}^t) , \quad s.t. \quad \theta_{\min} < \beta_m < \theta_{\max}, \ \forall m \tag{16}$$

where $\mathbf{X}^t$ denotes the sample matrix in the t-th frame, κ_t is a temporal weight to indicate the contribution of the t-th frame. It is not difficult to prove that the previous derivations (in Section 3.2 and 3.3) can be applicable for solving the optimization problem (16).

In Figure 4, we provide examples showing that our tracker can accurately learn the reliability value for each patch region. In the first row, the left part of the frisbee is frequently occluded, our method learns a small reliability value for such regions. The example in the second row demonstrates that our method can accurately determine that the fast moving legs are not reliable. In the last example, the background regions are associated with small weights via the proposed model, thereby facilitating the further tracking process.

4. Model Update

Most correlation filter based tracking algorithms perform model update in each frame, which results in high computation load. Recently, the ECO method proves that the sparse update mechanism is a better choice for the CF based trackers. Following the ECO method, we also exploit the sparse update mechanism in our tracker. In the update frame, we use the conjugate gradient descent method to update the base filter coefficient vector $\mathbf{h}$ and then we update $\boldsymbol{\beta}$ based on the updated base filter by solving a quadratic programming problem. In each frame, we initialize the weight for the training frame as ω and weights of previous training samples are decayed as $(1-\omega)\kappa_t$. When the number of training samples exceeds the pre-defined value $T_{\max}$, we follow the ECO method and use the Gaussian Mixture Model (GMM) for sample fusion. We refer the readers to [5] for more details.

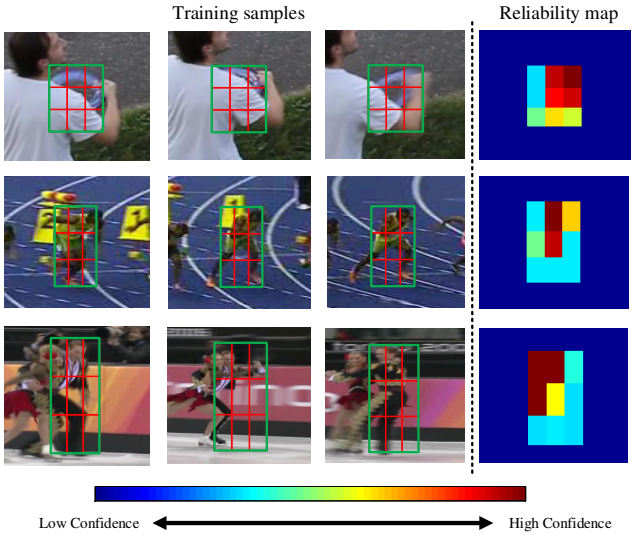


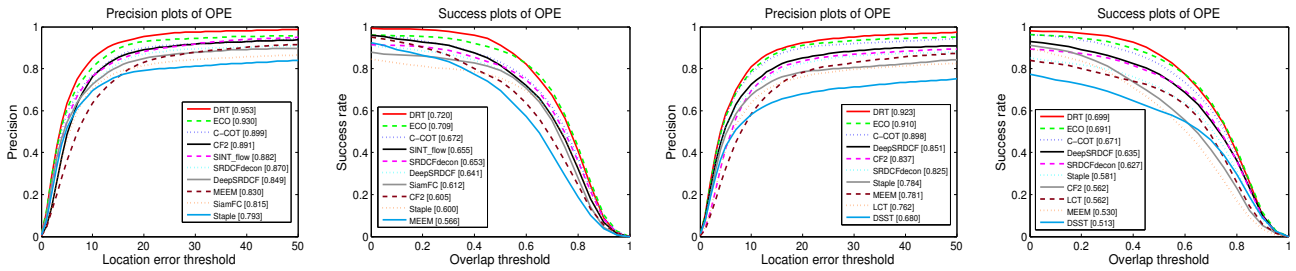
Figure 4. Illustration showing that reliable regions can be determined through the joint learning formula. We show three example training samples on the left three columns, and show the learned reliable weight maps on the fourth column.

5. Target Localization

In the detection process at the t-th frame, we use a multi-scale search strategy [10, 5] for joint target localization and scale estimation. We extract the ROI regions with different scales centred in the estimated position of last frame, and obtain the multi-channel feature map $\mathbf{x}_d^s, d = \{1, ..., D\}, s = \{1, ..., S\}$ for the ROI region, where s is the scale index. Then we compute the response for the target localization in scale s as

$$\mathbf{r}_s = \sum_{d=1}^{D} \mathcal{F}^{-1}(\mathcal{F}(\mathbf{w}_d) \odot (\mathcal{F}(\mathbf{x}_d^s))^H). \tag{17}$$

The target location and scale are then jointly determined by finding the maximum value in the S response maps. This joint estimation strategy shows better performance than the previous methods, which first estimate the target position and then refine the scale based on the estimated position.



OTB-2013 Dataset OTB-2015 Dataset

Figure 5. Precision and success plots of different trackers on the OTB-2013 and OTB-2015 datasets in terms of the OPE rule. This figure only shows the plots of top 10 trackers for clarity. In the legend behind of name of each tracker, we show the distance precision score at the threshold on 20 pixels and the area under curve (AUC) score.

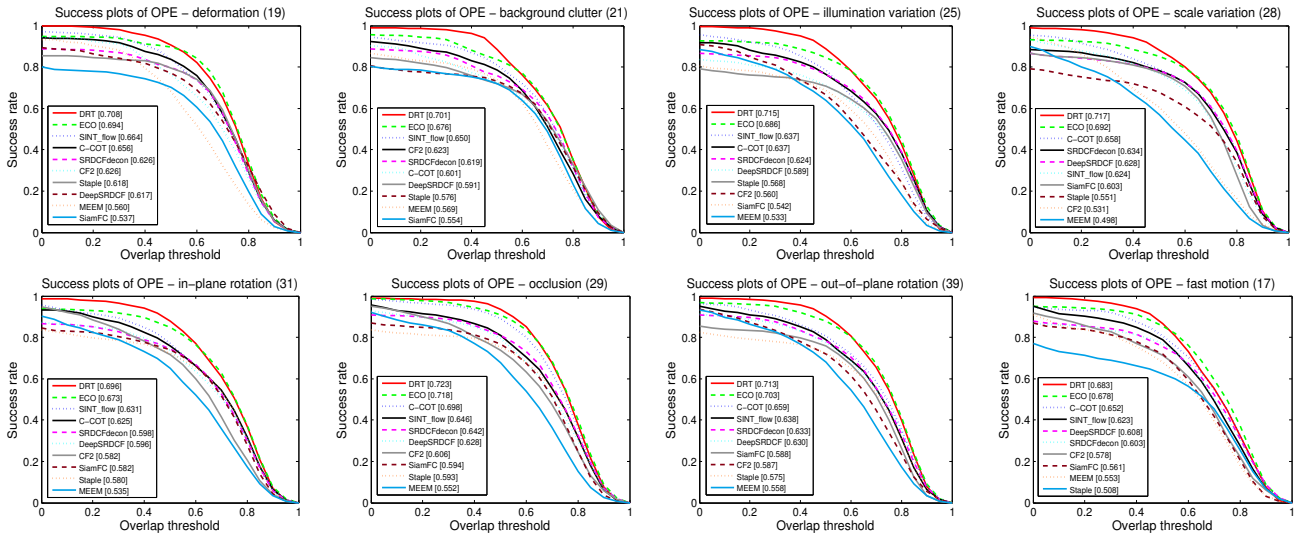


Figure 6. Performance evaluation on different attributes of the benchmark in terms of the OPE criterion. Merely top 10 trackers for each attributes are illustrated for clarity.

6. Experimental Results

We demonstrate the effectiveness of the proposed tracker on the OTB-2013 [30], OTB-2015 [31] and VOT-2016 [15] benchmark datasets. Since our method jointly considers both discrimination and reliability for tracking, we denote it as **DRT** for clarity.

6.1. Implementation Details

The proposed DRT method is mainly implemented in MATLAB and is partially accelerated with the Caffe toolkit [13]. Similar with the ECO method, we also exploit an ensemble of deep (Conv1 from VGG-M, Conv4-3 from VGG-16 [4]) and hand-crafted (HOG and Color Names) features for target representation. In our tracker, we use a relatively small learning rate ω (*i.e.* 0.011) for first 10 frames to avoid model degradation with limited training samples, and use a larger one (*i.e.* 0.02) in the following tracking process. The maximum number of training samples $T_{\max}$ and the number of fragments as set as 50 and 9 repectively. As to the online joint learning formula, the trade-off parameter η for the local consistency term is set as 1 by default and $\theta_{\min}$ and $\theta_{\max}$ are set as 0.5 and 1.5 respectively. One implementation of our tracker can be found in `https://github.com/cswaynecool/DRT`.

6.2. Performance Evaluation

OTB-2013 Dataset. The OTB-2013 dataset [30] is one of the most widely used dataset in visual tracking and contains 50 image sequences with various challenging factors. Using this dataset, we compare the proposed DRT method with the 29 default trackers in [30] and 9 more state-of-the-art trackers including ECO [5], C-COT [10], Staple [1], CF2 [21], DeepSRDCF [7], SRDCFdecon [9], SINT [27], SiamFC [2] and MEEM [32]. The one-pass evaluation (OPE) is employed to compare different trackers, based on two criteria (center location error and bounding box overlap ratio).

Figure 5 (a) reports the precision and success plots of different trackers based on the two criteria above, respectively. Among all compared trackers, the proposed DRT

Table 1. Performance evaluation of different state-of-the-art trackers in the VOT-2016 dataset. In this dataset, we compare our DRT method with the top 10 trackers. The best two results are marked in red and blue bold fonts, respectively.

	STAPLE+	SRBT	EBT	DDC	Staple	MLDF	SSAT	TCNN	C-COT	ECO	DRT
EAO	0.286	0.290	0.291	0.293	0.295	0.311	0.321	0.325	0.331	**0.374**	**0.442**
R	0.368	0.350	0.252	0.345	0.378	0.233	0.291	0.268	0.238	**0.200**	**0.140**
A	0.557	0.496	0.465	0.541	0.544	0.490	**0.577**	0.554	0.539	0.551	**0.569**

method obtains the best performance, which achieves the 95.3% distance precision rate at the threshold of 20 pixels and a 72.0% area-under-curve (AUC) score.

We note that it is very useful to evaluate the performance of trackers in various attributes. The OTB-2013 dataset is divided into 11 attributes, each of which corresponds to a challenging factor (*e.g.*, illumination, deformation and scale change). Figure 6 illustrates the overlap success plots of the top 10 algorithms on 8 attributes. We can see that our tracker achieves the best performance in all these attributes. Specially, the proposed method improves the second best tracker ECO by 1.4%, 2.5%, 2.9% and 2.5% in the attributes of deformation, background clutter, illumination variation and scale variation, respectively. These results validate that our method is effective in handling such challenges. When the object suffers from large deformations, parts of the target object will be not reliable. Thus, it is crucial to conduct accurate reliability learning in dealing with this case. Since our joint learning formula is insusceptible to the feature map response distributions, it can learn the reliability score for each region more accurately. Similarly, influenced by the cluttered background and abrupt illumination change, the feature map inevitably highlights the background or unreliable regions in the image. Existing CF-based algorithms learn large filter weights in such regions, thereby resulting in the tracking failure. In addition, these trackers usually assign most filter weights to the learned dominant regions and ignore certain parts of the target object, which leads to inferior scale estimation performance. By joint discrimination and reliability learning, the proposed DRT method is robust to numerous challenges and therefore achieves a remarkable performance in comparison with other ones.

OTB-2015 Dataset. The OTB-2015 dataset [31] is an extension of the OTB-2013 dataset, and contains 50 more video sequences. We also evaluate the performance of the proposed DRT method over all 100 videos in this dataset. In our experiment, we compare with 29 default trackers in [31] and other 9 state-of-the-art trackers including ECO [5], C-COT [10], Staple [1], CF2 [21], DeepSRDCF [7], SRDCFDecon [9], LCT [22], DSST [6] and MEEM [32].

Figure 5 (b) reports both precision and success plots of different trackers in terms of the OPE rule. Overall, our DRT method provides the best result with a distance precision rate of 92.3% and with an AUC score of 69.9%, which again achieves a substantial improvement of several outstanding trackers (e.g., ECO, C-COT and DeepSRDCF).

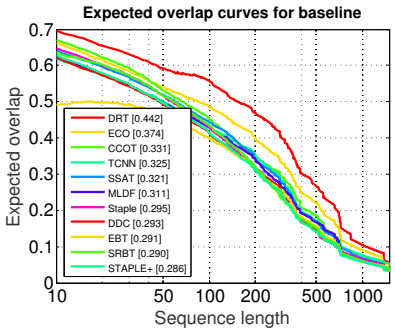
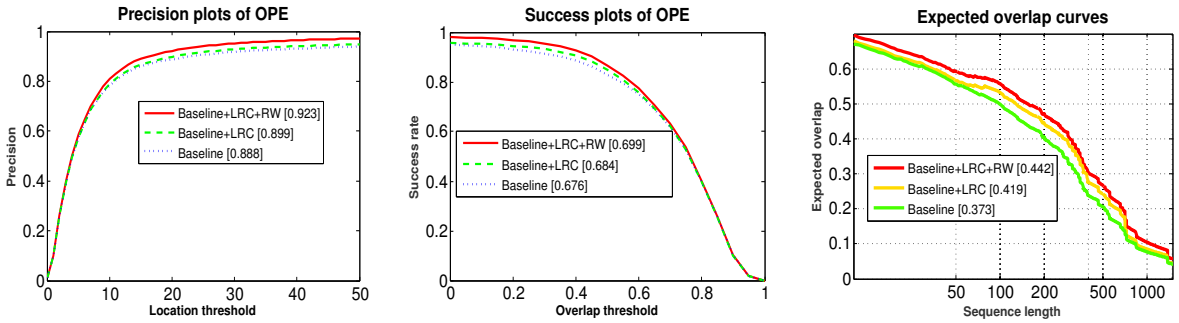


Figure 7. Expected Average Overlap (EAO) curve for 11 state-of-the-art trackers on the VOT-2016 dataset. Our DRT tracker has much better performance than the compared trackers.

VOT-2016 Dataset. The VOT-2016 dataset [15] contains 60 image sequences with 5 challenges including camera motion, illumination change, motion change, occlusion and scale change. Different from the OTB-2013 and OTB-2015 datasets, the VOT-2016 dataset pays much attention to the short-term visual tracking, and thus incorporates the reset-based experiment settings. In this work, we compare the proposed DRT method with 11 state-of-the-art trackers including ECO [5], C-COT [10], TCNN [23], SSAT [15], MLDF [15], Staple [1], DDC [15], EBT [33], SRBT [15] and STAPLE+ [1]. The results of different tracking algorithms are reported in Table 1 (a), using the expected average overlap (EAO), robustness raw value (R) and accuracy raw value (A) criteria.

Before our tracker, the ECO method has the best performance in the VOT-2016 dataset, which achieves an EAO of 0.374. Our DRT method has an EAO of 0.442, which outperforms ECO with a relative performance gain of 18.2%. In addition, our method has the best performance in terms of robustness (*i.e.*, fewer failures) among all the compared methods. Figure 7 shows the EAO curve of the compared trackers, which also demonstrates the effectiveness of our tracker.



OTB-2015 Dataset VOT-2016 Dataset

Figure 8. Performance evaluation for each component of the proposed method.

6.3. Ablation Studies

In this section, we test effectiveness for each component of the proposed joint learning formula on both the OTB-2015 and VOT-2016 datasets. First, we use the notation "Baseline" to denote the baseline method which does not exploit the local consistency constraint and the reliability map (*i.e.* $\beta_m = 1, m \in \{1, ..., M\}$). Like the conventional correlation filter, the baseline method does not separate the discrimination and reliability information. In addition, we also use the notation "Baseline+LRC" to denote the modified baseline tracker with the local response consistency constraint. The "Baseline+LRC" method focuses on learning the discrimination information while ignoring the reliability information of the target. The abbreviation "RW" stands for reliability weight map and "Baseline+LRC+RW' denotes the proposed joint learning method. In Figure 8, we show that the proposed joint learning formula improves the baseline method by 3.5% and 2.3% on the OTB-2015 dataset in terms of the distance precision rate and the AUC score. In addition, the joint learning formula also improves the baseline method by 6.9% in EAO on the VOT-2016 dataset. By comparing our method with "Baseline+LRC", we show the effectiveness of the reliability learning process. Considering the reliability learning, our tracker improves the "Baseline+LRC" method by 1.5% in terms of AUC score on the OTB-2015 dataset, and our tracker also improves it by 2.3% in terms of EAO on the VOT-2016 dataset.

6.4. Failure cases

We show some failure cases of the proposed tracker in Figure 9. In the first and third columns, the cluttered background regions contain numerous distractors, which causes the proposed method to drift off the targets. In the second column, the proposed method does not track the target object well as it undergoes large deformations and rotations in a short span of time. These tracking failures can be partially addressed when the information of the optical flow is considered, which will be the focus of our future work.

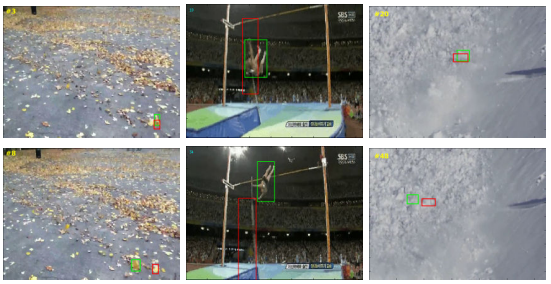


Figure 9. Failure cases of the proposed method, where we use red and green bounding boxes to denote our results and ground-truths.

7. Conclusion

In this paper, we clearly consider the discrimination and reliability information in the correlation filter (CF) formula and rewrite the filter weight as the element-wise product of a base filter and a reliability weight map. First, we introduce a local response consistency constraint for the base filter, which constrains that each sub-region of the target has similar importance. By this means, the reliability information is separated from the base filter. In addition, we consider the reliability information in the filter, which is jointly learned with the base filter. Compared to the existing CF-based methods, our tracker is insusceptible to the non-uniform distributions of the feature map, and can better suppress the background regions. The joint learning of the base filter and reliability term can be preformed by solving the proposed optimization problem and being speeded up in the Fourier domain. Finally, we evaluate our DRT method on the OTB-2013, OTB-2015 and VOT-2016 datasets. Extensive experiments demonstrate that the proposed tracker outperforms the state-of-the-art algorithms over all three benchmarks.

Acknowledgment. This paper is partially supported by the Natural Science Foundation of China #61502070, #61725202, #61472060. Chong Sun and Ming-Hsuan Yang are also supported in part by NSF CAREER (No. 1149783), gifts from Adobe, Toyota, Panasonic, Samsung, NEC, Verisk, and Nvidia.

References

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

[2] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr. Fully-convolutional siamese networks for object tracking. In *ECCV*, 2016.

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

[4] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014.

[5] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg. Eco: Efficient convolution operators for tracking. In *CVPR*, 2017.

[6] M. Danelljan, G. Häger, F. Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *BMVC*, 2014.

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

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

[9] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking. In *CVPR*, 2016.

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

[11] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *ECCV*, 2012.

[12] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596, 2015.

[13] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ICMM*, 2014.

[14] H. Kiani Galoogahi, A. Fagg, and S. Lucey. Learning background-aware correlation filters for visual tracking. In *CVPR*, 2017.

[15] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernandez, T. Vojir, G. Hager, G. Nebehay, and R. Pflugfelder. The visual object tracking vot2016 challenge results. In *ECCV Workshops*, 2016.

[16] P. Li, D. Wang, L. Wang, and H. Lu. Deep visual tracking: Review and experimental comparison. *Pattern Recognition*, 76:323–338, 2018.

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

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

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

[20] A. Lukežič, T. Vojíř, L. Čehovin, J. Matas, and M. Kristan. Discriminative correlation filter with channel and spatial reliability. In *CVPR*, 2017.

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

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

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

[24] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, USA, 2006.

[25] Y. Qi, L. Qin, J. Zhang, S. Zhang, Q. Huang, and M.-H. Yang. Structure-aware local sparse coding for visual tracking. *IEEE Transactions on Image Processing*, PP(99):1–1, 2018.

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

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

[28] L. Wang, W. Ouyang, X. Wang, and H. Lu. Visual tracking with fully convolutional networks. In *ICCV*, 2015.

[29] L. Wang, W. Ouyang, X. Wang, and H. Lu. Stct: Sequentially training convolutional networks for visual tracking. In *CVPR*, 2016.

[30] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *CVPR*, 2013.

[31] Y. Wu, J. Lim, and M.-H. Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1834–1848, 2015.

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

[33] G. Zhu, F. Porikli, and H. Li. Beyond local search: Tracking objects everywhere with instance-specific proposals. In *CVPR*, 2016.