Large Margin Object Tracking with Circulant Feature Maps

Mengmeng Wang[1], Yong Liu[1] *, Zeyi Huang[2]
[1] Institute of Cyber-Systems and Control, Zhejiang University, [2] Exacloud Limited, Zhejiang, China
mengmengwang@zju.edu.cn; yongliu@iipc.zju.edu.cn; haoran@qunhemail.com

Abstract

Structured output support vector machine (SVM) based tracking algorithms have shown favorable performance recently. Nonetheless, the time-consuming candidate sampling and complex optimization limit their real-time applications. In this paper, we propose a novel large margin object tracking method which absorbs the strong discriminative ability from structured output SVM and speeds up by the correlation filter algorithm significantly. Secondly, a multimodal target detection technique is proposed to improve the target localization precision and prevent model drift introduced by similar objects or background noise. Thirdly, we exploit the feedback from high-confidence tracking results to avoid the model corruption problem. We implement two versions of the proposed tracker with the representations from both conventional hand-crafted and deep convolution neural networks (CNNs) based features to validate the strong compatibility of the algorithm. The experimental results demonstrate that the proposed tracker performs superiorly against several state-of-the-art algorithms on the challenging benchmark sequences while runs at speed in excess of 80 frames per second.

1. Introduction

Visual tracking enjoys a wide popularity recently and has been applied in many applications such as robotic services, surveillance, human motion analyses, human-computer interactions and so on. In this paper, we consider the most general scenario of visual tracking, i.e., short-term, single-object tracking with the target given in the first frame. The most difficult point of this problem is to track the target at a high speed for real-time applications while handle all challenging factors simultaneously both from background or the target itself such as occlusions, deformations, fast motions, illumination variations and so on.

Due to the lack of training samples, most existing trackers handle this problem from two aspects. The first one is to explore an effective tracking algorithm which can be designed to be either discriminative [17, 9, 13, 15, 22, 1, 31] or generative [24, 18, 16, 33] models. It seeks to design a robust classifier or filter to detect the target, and establish an optimal mechanism to update the model at each frame. The other one is to exploit the power of the target representation which may come from conventional handcraft features [16, 18, 17, 9, 24] or high-level convolutional features [15, 28, 23, 27, 21] from deep Convolutional Neural Networks (CNNs). These methods improve performance significantly from different aspects. However, to further improve the performance by more complex tracking algorithms or features, it would undoubtedly increase the computational complexity, which would limit the real-time performance of visual tracking.

The most popular and successful framework for visual tracking is tracking-by-detection [14, 17, 34, 13, 22, 13] which treats the tracking problem as a detection task and learns information about the target from each detection online. There are many classification algorithms used in this framework, such as multiple instance learning [1], P-N learning [17], online boosting [11, 12], support vector machines (SVM) [13, 15, 22, 31] and so on. Among them, structured output SVM is demonstrated with an excellent potential in this field [13, 22]. Structured output SVM is a kind of classification algorithm which can deal with complex outputs like trees, sequences, or sets rather than class labels [26]. Hare et al. [13] employ this algorithm in the visual tracking for the first time and improve tracking accuracy considerably in several benchmarks [29, 30]. They propose a tracking algorithm named Struck based on kernelized structured output SVM where the output space is defined as the translations of the target relative to the previous frame. However, Struck suffers from a high computational complexity by its complex optimization while its training samples are still not dense enough. Therefore it operates slowly and limits to extend to higher dimensional features. Ning et al. [22] propose a dual linear structured SVM (DLSSVM) algorithm which approximates nonlinear kernels with explicit feature maps. DLSSVM improves tracking performance significantly, while its tracking speed is not

*Corresponding author

1063-6919/17 $31.00 © 2017 IEEE
DOI 10.1109/CVPR.2017.510

fast enough for realtime applications, especially when scale estimation is considered, as well as feature dimensions and budgets of support vectors are increased. Thus, it is significant to design a novel tracking algorithm based on structured SVM which can not only absorb the strong discrimination from structured SVM, but also process sufficiently fast with higher dimensional features and more dense samples.

Recently, a group of correlation filter (CF) based trackers [9, 14, 5, 2, 32, 4] have attracted extensive attentions due to their significant computational efficiency. CF enables training and detection with densely-sampled examples and high dimensional features in real time by using the fast Fourier transform (FFT). Since Bolme et al. [4] introduce the CF into the visual tracking field, several extensions have been proposed to improve tracking performance. Henriques et al. [14] propose a high speed tracker with kernelized correlation filters (KCF) and multi-channel features which enables further extension for high dimensional features while remaining the real-time capability. Danelljan et al. [5] figure out the fast scale estimation problem by learning discriminative CF based on a scale pyramid representation. One deficiency of CF is the unwanted boundary effects introduced by the periodic assumption for all circular shifts, that would degrade the discriminative ability of tracking models. To resolve this issue, Danelljan et al. [7] introduce a spatially regularized component in the learning to penalize CF coefficients depending on their spatial locations and achieve excellent tracking accuracy. However, this algorithm reduces the computational efficiency of CF and runs at a reported speed of 5 frames per second (FPS). The evolution of these methods motivate us to improve the discriminative ability of CF based tracking algorithm and remain its high operating speed.

With the great power in the feature representations, CNNs have been demonstrated significant success on many computer vision tasks, including visual tracking. Recent studies [27, 21, 28, 23, 15] have shown state-of-the-art results on many object tracking benchmarks. Ma et al. [21] exploit features extracted from a pretrained deep CNN and learn adaptive CFs on several CNN layers to improve tracking accuracy and robustness. Wang et al. [28] present a sequential training method for CNN that is regarded as an ensemble with each channel of the output feature map as an individual base learner. These methods validate the strong capacity of CNNs for the target representation at the cost of time consumption and high requirements of computational resources.

In this paper, we consider the problems mentioned above and propose a large margin object tracking method with circulant feature maps (LMCF). The main contributions of our work can be summarized as follows:

- We propose a novel structured SVM based tracking method which takes dense circular samples into account in both training and detection processes. A bridge is built up to link our problem with CF, which speeds up the optimization process significantly.
- We explore a multimodal target detection technique to prevent the model drift problem introduced by similar objects or background noise.
- We establish a model update strategy to avoid model corruption by the high-confidence selection from tracking results.

2. Large Margin Object Tracking with Circulant Feature Maps

In this section, we first present the problem formulation of the large margin tracking method with circulant feature maps. Next, we deduce a fast optimization algorithm that builds up a bridge between our problem formulation and the well-known correlation filter. Thirdly, a multimodal target detection method is proposed to improve the localization precision and prevent model drift introduced by similar objects or background noise. In the end, we present a model update strategy by exploiting the feedback from tracking results to avoid the model corruption.

2.1. Problem formulation

We consider the tracking-by-detection framework in this paper. When receiving a new frame, our goal is to learn a classifier which can distinguish the target from its surrounding background in real time. The employed classifier is a structured output SVM which is different from conventional binary discriminative classifiers. It can directly estimate the relative movement between adjacent frames rather than discriminate whether it is the target or not. Additionally, the structured output SVM used here is distinct from the methods [13, 22] in both the variable definitions and the objective function.

The object of large margin learning over structured output spaces is to learn a function $f : X \to Y$ based on the input-output pairs, where X is the input spaces and Y is arbitrary discrete output spaces. In our case, all the cyclic shifts of the image patch centered around the target are considered as the training samples, i.e., $Y = \{(w,h)|\, w \in \{0,...,W-1\}, h \in \{0,...,H-1\}\}$, where W and H are the width and the height of the image patch. Hence, the input-output pairs are defined as $(\mathbf{x}, \mathbf{y}_{w,h})$, where $\mathbf{x} \in X$ denotes the image patch which contains and is proportional to the target bounding box at center, $\mathbf{y}_{w,h} \in Y$ represents its corresponding cyclic transform. With different cyclic shifts $\mathbf{y}_{w,h}$, the pairs stand for different image regions which contain diverse translated targets. The joint feature maps of these cyclic image patches are denoted as $\Psi(\mathbf{x},\mathbf{y}_{w,h})$, whose specific form depends on the nature of the problem.

We aim to measure the compatibility between the input-output pairs $(\mathbf{x},\mathbf{y})$ with $F : X \times Y \rightarrow \mathbb{R}$ from which we can acquire a prediction by maximizing F over the response variable for a specific given input $\mathbf{x}$. Then the general form of the function f can be denoted as

$$f(\mathbf{x};\mathbf{w}) = \arg\max_{\mathbf{y}\in Y} F(\mathbf{x},\mathbf{y};\mathbf{w}) \tag{1}$$

where we assume F to be a linear function, $F(\mathbf{x},\mathbf{y};\mathbf{w}) = \langle \mathbf{w}, \Psi(\mathbf{x},\mathbf{y})\rangle$ and $\mathbf{w}$ denotes the parameter vector which can be learned from the soft-margin support vector machine learning over structured outputs. F can also be extended to nonlinear situation which will be discussed in the next section. We penalize the margin violations by a quadratic term, leading to the following optimization problem:

$$\begin{aligned}&\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{w=1}^{W-1}\sum_{h=1}^{H-1}\xi_{w,h}^2\\ &\text{s.t.}\forall w,\forall h,\forall \mathbf{y}_{w,h}\in Y\backslash\mathbf{y}_{0,0}:\\ &F(\mathbf{x},\mathbf{y}_{0,0};\mathbf{w}) - F(\mathbf{x},\mathbf{y}_{w,h};\mathbf{w}) \geqslant \sqrt{\Delta(\mathbf{y}_{0,0},\mathbf{y}_{w,h})} - \xi_{w,h}\end{aligned} \tag{2}$$

where $\mathbf{y}_{0,0}$ denotes the observed output with no cyclic transform and $\xi_{w,h}$ is the slack variable which penalizes the margin violations. The regularization parameter $C > 0$ controls the trade-off between training error minimization and margin maximization. $\Delta(\mathbf{y}_{0,0},\mathbf{y}_{w,h})$ quantifies the loss associated with a prediction $\mathbf{y}_{w,h}$ when the true output value is $\mathbf{y}_{0,0}$. We define the loss function as

$$\Delta(\mathbf{y}_{0,0},\mathbf{y}_{w,h}) = m(\mathbf{y}_{0,0}) - m(\mathbf{y}_{w,h}) \tag{3}$$

where $m(\bullet)$ is designed to follow a Gaussian function that takes a maximum value for the centered target and smoothly reduces to 0 for larger shifts.

The optimization problem in Eq.2 pursues to ensure that the value of $F(\mathbf{x},\mathbf{y}_{0,0};\mathbf{w})$ is greater than $F(\mathbf{x},\mathbf{y}_{w,h};\mathbf{w})$, by a margin which depends on the loss function as Eq.3.

2.2. Fast online optimization

The conventional structured SVM in visual tracking is solved by sequential minimal optimization (SMO) step [13] or the basic dual coordinate descent (DCD) optimization process [22]. Thus the tracking speed is limited due to their high computational complexity. Inspired by [14], we propose a novel algorithm to employ Fourier transform to speed up the optimization.

Following the constraint in Eq.2, we reformulate it by adding Eq.4 into the constraint,

$$F(\mathbf{x},\mathbf{y}_{0,0};\mathbf{w}) - F(\mathbf{x},\mathbf{y}_{0,0};\mathbf{w}) \geqslant \sqrt{\Delta(\mathbf{y}_{0,0},\mathbf{y}_{0,0})} - \xi_{0,0} \tag{4}$$

where $\xi_{0,0}$ denotes the slack variable of the true output which is set to 0. Then the optimization problem can be rewritten as

$$\begin{aligned}&\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{w=0}^{W-1}\sum_{h=0}^{H-1}\xi_{w,h}^2\\ &\text{s.t.}\forall w,\forall h,\forall \mathbf{y}_{w,h}\in Y:\\ &F(\mathbf{x},\mathbf{y}_{0,0};\mathbf{w}) - F(\mathbf{x},\mathbf{y}_{w,h};\mathbf{w}) \geqslant \sqrt{\Delta(\mathbf{y}_{0,0},\mathbf{y}_{w,h})} - \xi_{w,h}\end{aligned} \tag{5}$$

For clarity, we first formulate our optimization method for the joint feature maps defined in the one-dimensional domain, i.e., set W or H to 1. Here we set $H = 1$ and omit h in the subscript temporarily. It can be generalized to two dimensions in the same way. Now Eq.5 is reformulated as

$$\begin{aligned}&\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\|\zeta\|_2^2\\ &\text{s.t.}\forall w,\forall \mathbf{y}_w\in Y:\mathbf{w}^T\Phi_0 - \mathbf{w}^T\Phi \geqslant \Upsilon - \zeta\end{aligned} \tag{6}$$

where $\zeta = [\xi_0,...,\xi_{W-1}]$ represents the vector of slack variables. $\Phi = [\Psi(\mathbf{x},\mathbf{y}_0),...,\Psi(\mathbf{x},\mathbf{y}_{W-1})]$ is a circulant matrix formed by the joint feature maps of all the cyclic training samples and $\Phi_0 = [\Psi(\mathbf{x},\mathbf{y}_0),...,\Psi(\mathbf{x},\mathbf{y}_0)]$ is constructed with W duplicates of $\Psi(\mathbf{x},\mathbf{y}_0)$. $\Upsilon = \left[\sqrt{\Delta(\mathbf{y}_0,\mathbf{y}_0)},...,\sqrt{\Delta(\mathbf{y}_0,\mathbf{y}_{W-1})}\right]$ denotes the loss vector .

To solve the problem online, we define a new variable $\mathbf{z} = \zeta + \mathbf{w}^T\Phi_0 - \mathbf{w}^T\Phi - \Upsilon, \mathbf{z} \geqslant 0$. Plug $\mathbf{z}$ into the Eq.6:

$$\begin{aligned}&\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\left\|\mathbf{w}^T\Phi - \left(\mathbf{w}^T\Phi_0 - \Upsilon - \mathbf{z}\right)\right\|_2^2\\ &\text{s.t. } \mathbf{z} \geqslant 0\end{aligned} \tag{7}$$

with the circulant nature of Φ, we have

$$\mathbf{w}^T\Phi = \left(\mathcal{F}^{-1}\left(\hat{\Psi}^*(\mathbf{x},\mathbf{y}_0)\circ\hat{\mathbf{w}}\right)\right)^T \tag{8}$$

where $\hat{\bullet}$ and $\mathcal{F}^{-1}$ denotes the discrete Fourier transform (DFT) and its inverse, $\circ$ represents the element-wise multiplication, $\hat{\Psi}^*$ means the complex conjugate of $\hat{\Psi}$.

There are two variables $\mathbf{w}$ and $\mathbf{z}$ to be solved in Eq.7. Whenever one of them is known, the subproblem on the other has a closed form solution. Thus similar to [34], we introduce the alternating optimization algorithm to solve the model efficiently by iterating between the following two steps.

Update $\mathbf{z}$. Given $\mathbf{w}$, the subproblem on $\mathbf{z}$ becomes:

$$\min_{\mathbf{z}} \left\|\mathbf{z} - \left(\mathbf{w}^T\Phi_0 - \mathbf{w}^T\Phi - \Upsilon\right)\right\|_2^2, \text{s.t. } \mathbf{z} \geqslant 0 \tag{9}$$

Then the closed form solution of $\mathbf{z}$ is:

$$\mathbf{z} = \max\left\{\mathbf{w}^T\Phi_0 - \mathbf{w}^T\Phi - \Upsilon, 0\right\} \tag{10}$$

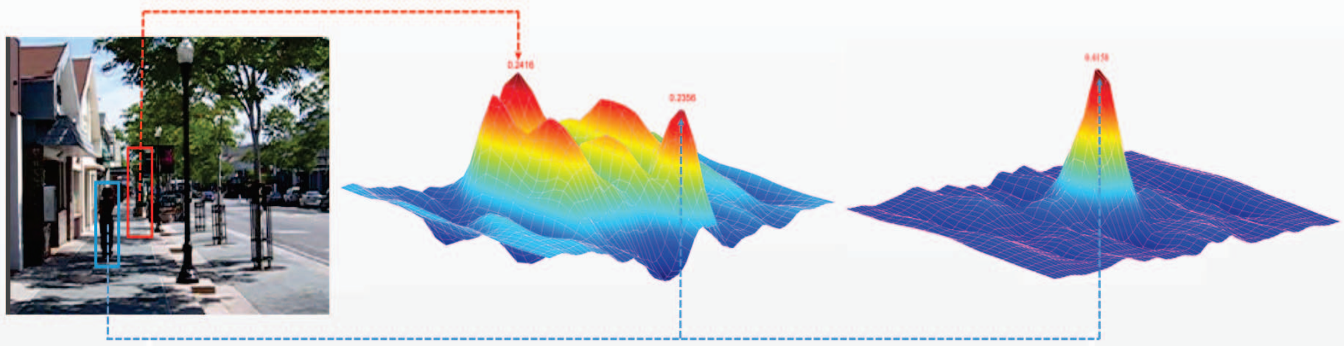


Figure 1. Illustration of multimodal target detection in sequence *human9* from OTB-15 [30]. The blue bounding box indicates the correct location of target, the red one is an incorrect detection. The response of the target is weaker than the background area within the red bounding box as shown in the middle. The unimodal detection will regard the highest peak as the target leading to false detection. The proposed multimodal target detection will redetect the areas centered at other peaks to find the maximum peak among these response maps as the right subfigure and locate the correct position of the target.

Update $\mathbf{w}$. Given $\mathbf{z}$, the subproblem on $\mathbf{w}$ becomes:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\left\|\mathbf{w}^T\Phi - \left(\mathbf{w}^T\Phi_0 - \Upsilon - \mathbf{z}\right)\right\|_2^2 \quad (11)$$

In order to employ the correlation filter theory, we define $\mathbf{u}_0 = \mathbf{w}^T\Phi_0$ which stands for a plane whose height is the highest peak of $\mathbf{w}^T\Phi$ in the last iteration. Then the closed form solution of $\mathbf{w}$ is:

$$\hat{\mathbf{w}} = \frac{\hat{\Psi}^*\left(\mathbf{x}, \mathbf{y}_0\right) \circ \hat{\mathbf{u}}^T}{\hat{\Psi}^*\left(\mathbf{x}, \mathbf{y}_0\right) \circ \hat{\Psi}\left(\mathbf{x}, \mathbf{y}_0\right) + \frac{1}{2C}} \quad (12)$$

where $\mathbf{u} = \mathbf{u}_0 - \Upsilon - \mathbf{z}$ and $\frac{\bullet}{\bullet}$ denotes the element-wise division.

Nonlinear extension. The proposed linear model can be extended to a nonlinear model by the kernel trick $K_{ij} = \langle\varphi\left(\Psi\left(\mathbf{x}, \mathbf{y}_i\right)\right), \varphi\left(\Psi\left(\mathbf{x}, \mathbf{y}_j\right)\right)\rangle$ where $\varphi(\bullet)$ indicates the implicit use of a high-dimensional feature space. The solution w can be represented as $\mathbf{w} = \sum_{w=0}^{W-1} \alpha_w \varphi\left(\Psi\left(\mathbf{x}, \mathbf{y}_w\right)\right)$. The optimization now is rewritten as

$$\begin{aligned} &\min_{\alpha} \alpha^T \mathcal{F}^{-1}\left(\hat{\mathbf{k}}^{\Psi_0\Psi_0} \circ \hat{\alpha}\right) \\ &+ C\left\|\mathcal{F}^{-1}\left(\hat{\mathbf{k}}^{\Psi_0\Psi_0} \circ \hat{\alpha}\right) - \left(\mathbf{u}_0 - \Upsilon - \mathbf{z}\right)^T\right\|_2^2 \\ &\qquad \text{s.t. } \mathbf{z} \geqslant 0 \end{aligned} \quad (13)$$

where $\Psi_0 = \Psi\left(\mathbf{x}, \mathbf{y}_0\right)$ and $\hat{\mathbf{k}}^{\Psi_0\Psi_0}$ denotes the DFT of the first row of the circulant kernel matrix $\mathbf{K}$ whose elements are K_{ij}. The closed form of the subproblem on α is:

$$\hat{\alpha} = \frac{\hat{\mathbf{u}}^T}{\hat{\mathbf{k}}^{\Psi_0\Psi_0} + \frac{1}{2C}} \quad (14)$$

where $\frac{\bullet}{\bullet}$ denotes the element-wise division.

2.3. Multimodal target detection

Intuitively, when a new frame comes out, the transformation of the target $\mathbf{y} = f\left(\mathbf{s}; \mathbf{w}\right)$ is estimated by the Eq.1, where $\mathbf{s}$ is the region in the new frame centered at the target position of the last frame. This can be sped up with the learned model by FFT algorithm. The full detection response map on all cyclic transform is obtained by

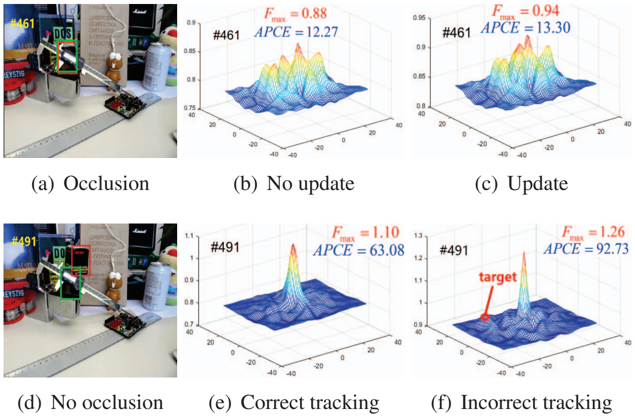
$$F\left(\mathbf{s}, \mathbf{y}; \mathbf{w}\right) = \mathcal{F}^{-1}\left(\hat{\Psi}^*_{\mathbf{s}0} \circ \hat{\mathbf{w}}\right) = \mathcal{F}^{-1}\left(\hat{\mathbf{k}}^{\Psi_{\mathbf{x}0}\Psi_{\mathbf{s}0}} \circ \hat{\alpha}\right) \quad (15)$$

where $\Psi_{\bullet 0}$ is short for $\Psi\left(\bullet, \mathbf{y}_{0,0}\right)$. The localization of the target is estimated on the highest peak of the response map which is defined as the unimodal detection in this paper. However, the unimodal detection may be disturbed by similar objects or certain noise leading to inaccurate detection. The inaccurate detection would further contaminate the learned model due to incorrect training samples. Shown as Figure 1, the peaks located at similar objects or background noise in the response map may approach, or even surpass the peak at the target. As above analysis, the target may locate at one of multiple peaks, all of them should be taken into consideration.

Consequently, a multimodal target detection method is proposed to improve localization precision further. For the unimodal detection response map $F\left(\mathbf{s}, \mathbf{y}; \mathbf{w}\right)$, the multiple peaks are computed by

$$P\left(\mathbf{s}\right) = F\left(\mathbf{s}, \mathbf{y}; \mathbf{w}\right) \circ \mathbf{B} \quad (16)$$

where $\mathbf{B}$ is a binary matrix with the same size as $F\left(\mathbf{s}, \mathbf{y}; \mathbf{w}\right)$, which identifies the locations of local maxima in $F\left(\mathbf{s}, \mathbf{y}; \mathbf{w}\right)$. The elements at the locations of local maxima in $\mathbf{B}$ are set to 1, while others are set to 0. All non-zero elements in $P\left(\mathbf{s}\right)$ indicate multiple peaks in the response map of $\mathbf{s}$.



(a) Occlusion (b) No update (c) Update

(d) No occlusion (e) Correct tracking (f) Incorrect tracking

Figure 2. The first column are the shots of sequence *box* from OTB-15, where the red bounding boxes indicate the tracking results of LMCF with high-confidence update strategy and the green ones belong to the LMCF-NU which updates the tracking model at each frame. The response maps in the second column are corresponding to LMCF and the third column corresponding to LMCF-NU. The red annotation in the last subfigure points out the right position of the target in this response map.

When the ratios between multiple peaks to the highest peak are greater than a predefined threshold θ, the corresponding image regions centered at those peaks are redetected through Eq.15. The target is finally identified to locate at the maximum peak among these response maps as shown in Figure 1.

Furthermore, to handle scale variation, we adopt a scale searching strategy proposed by [5] at the detected location. The difference between ours and [5] lies in that the scale model is only executed when the detected results have high-confidence as discussed in the next section.

2.4. High-confidence update

Most existed trackers update tracking models [5, 22, 14, 2] at each frame without considering whether the detection is accurate or not. Actually, this may cause a deterministic failure once the target is detected inaccurately, severely occluded or totally missing in the current frame. In the proposed method, we utilize the feedback from tracking results during the target detection to decide the necessity of model update.

The peak value and the fluctuation of the response map can reveal the confidence degree about the tracking results to some extent. The ideal response map should have only one sharp peak and be smooth in all other areas when the detected target is extremely matched to the correct target. The sharper the correlation peaks are, the better the location accuracy is. Otherwise, the whole response map will fluctuate intensely, whose pattern is significantly different from normal response maps as shown in the first row of Figure 2. If we continue to use uncertain samples to update the tracking model, it would be corrupted mostly as shown in the second row of the Figure 2. So we explore a high-confidence feedback mechanism with two criteria. The first one is the maximum response score $F_{\max}$ of the response map $F(\mathbf{s}, \mathbf{y}; \mathbf{w})$ defined as

$$F_{\max} = \max F(\mathbf{s}, \mathbf{y}; \mathbf{w}) \tag{17}$$

The second one is a novel criterion called average peak-to-correlation energy (APCE) measure which is defined as

$$APCE = \frac{\left|F_{\max} - F_{\min}\right|^2}{mean\left(\sum\limits_{w,h}\left(F_{w,h} - F_{\min}\right)^2\right)} \tag{18}$$

where $F_{\max}$, $F_{\min}$ and $F_{w,h}$ denote the maximum, minimum and the w-th row h-th column elements of $F(\mathbf{s}, \mathbf{y}; \mathbf{w})$. APCE indicates the fluctuated degree of response maps and the confidence level of the detected targets. For sharper peaks and fewer noise, i.e., the target apparently appearing in the detection scope, APCE will become larger and the response map will become smooth except for only one sharp peak. Otherwise, APCE will significantly decrease if the object is occluded or missing.

When these two criteria $F_{\max}$ and APCE of the current frame are greater than their respective historical average values with certain ratios β_1, β_2, the tracking result in the current frame is considered to be high-confidence. Then the proposed tracking model will be updated online with a learning rate parameter η as

$$\begin{aligned}\hat{\alpha}^t &= (1-\eta)\,\hat{\alpha}^{t-1} + \eta\hat{\alpha}\\ \hat{\Psi}^t_{\mathbf{x}0} &= (1-\eta)\,\hat{\Psi}^{t-1}_{\mathbf{x}0} + \eta\hat{\Psi}_{\mathbf{x}0}\end{aligned} \tag{19}$$

Figure 2 illustrates the importance of the proposed update strategy. As shown in Figure 2, when the target is occluded severely, the response map fluctuates fiercely in the first row so that APCE reduces to about 10, while $F_{\max}$ remains strong enough. Under this circumstance, the proposed high-confidence update strategy will choose not to update the model in this frame, then the tracking model won't be corrupted and the target can be tracked successfully in the subsequent frames. Otherwise, the target will be missed and the right peak will finally fade away.

An overview of the proposed method is summarized in Algorithm 1.

3. Experiments

Since the proposed tracking algorithm is compatible with different kinds of features for representing the targets, we implement experiments with both conventional features based version LMCF and deep CNNs based version

Algorithm 1 LMCF tracking algorithm

```
Input: Frames $\{\mathbf{I}_t\}_1^T$, initial target location $\mathbf{p}_1$, $\mathbf{z} = 0$, $\mathbf{u}_0 = ones\,(W, H)$
Output: Target locations of each frame $\{\mathbf{p}_t\}_2^T$.
repeat
    Crop an image region $\mathbf{s}$ from $\mathbf{I}_t$ at the last location $\mathbf{p}_{t-1}$ and extract its joint feature map $\Psi\,(\mathbf{s}, \mathbf{y}_{0,0})$.
    Detect the target location $\mathbf{p}_t$ with the multimodal detection via Eq.15 and Eq.16.
    Estimate the scale of the target as [5].
    Calculate $F_{\max}$ and APCE with Eq.17 and Eq.18.
    if $F_{\max}$ and APCE satisfy the update condition, then
        Train the $\mathbf{u}_0$, $\mathbf{z}$ and $\hat{\mathbf{w}}$ $(\hat{\alpha})$ with Eq.10 and Eq.12 (14).
        Update the tracking model with Eq.19.
        Update the scale estimation model as [5] with $\eta$.
    end if
until end of video sequence.
```

DeepLMCF to validate the performance of the proposed method.

We implement experiment on the OTB-13 [29] and OTB-15 [30] benchmark datasets. All these sequences are annotated with 11 attributes which cover various challenging factors, including scale variation (SV), occlusion (OCC), illumination variation (IV), motion blur (MB), deformation (DEF), fast motion (FM), out-of plane rotation (OPR), background clutters (BC), out-of-view (OV), in-plane rotation (IPR) and low resolution (LR). To fully assess our method, we use one-pass evaluation (OPE), temporal robustness evaluation (TRE), and spatial robustness evaluation (SRE) metrics as suggested in [29]. The *precision* scores indicate the percentage of frames in which the estimated locations are within 20 pixels compared to the ground-truth positions. The *success* scores are defined as the area under curve (AUC) of each success plot, which is the average of the success rates corresponding to the sampled overlap threshold.

We first analyze LMCF with the improvements from multimodal target detection, high-confidence update strategy and representation power of DeepLMCF on OTB-13. Then we compare LMCF with 9 most related and state-of-the-art trackers based on conventional features on OTB-13 and OTB-15. Finally, we present the attractive performance of DeepLMCF compared with 9 up-to-date CNNs based trackers on OTB-13. All the tracking results are using the reported results to ensure a fair comparison.

3.1. Implementation details

The conventional features used for LMCF are composed of HOG features and color names (CN) [9]. For the CNN features of DeepLMCF, we use imagenet-vgg-verydeep-19 which is available at: http://www.vlfeat.org/matconvnet/. The last three convolutional layers of this network are used to extract the features of the target and the weight of each layer is respectively set to 0.02, 0.5 and 1 similar to [21]. Our tracker is implemented in MATLAB for LMCF with a PC with a 3.60 GHz CPU and DeepLMCF with a tesla k40 GPU. LMCF runs faster than 80 FPS while DeepLMCF runs faster than 10 FPS.

Table 1. Parameters of LMCF and DeepLMCF.

parameters	LMCF	DeepLMCF
padding	1.5	1.8
η	0.015	0.01
θ	0.7	0.7
β_1	0.7	0.4
β_2	0.45	0.3
C	10000	20000

The optimization takes 10 iterations in the first frame and 3 iterations for each online update. Similar to [5], 33 number of scales with a scale factor of 1.02 is used in the scale model. The other parameters setting of LMCF and DeepLMCF are shown in Table 1, where padding means the magnification of the image region samples relative to the target bounding box.

3.2. Analyses of LMCF

To demonstrate the effect of the proposed multimodal target detection, high-confidence update strategy and representation power of DeepLMCF, we first test with different versions of LMCF on OTB-13. We denote LMCF without multimodal detection as LMCF-Uni, without high-confidence update strategy as LMCF-NU and with neither of these two as LMCF-N2. The characteristics and tracking results are summarized in Table 2. The mean FPS here is estimated on the longest sequence *doll* in OTB-13 with 3872 frames.

As shown in Table 2, DeepLMCF shows the best tracking accuracy and robustness in all OPE, TRE and SRE evaluation metrics benefited by the hierarchical CNN features and LMCF performs second while with the fastest speed. Without multimodal detection, LMCF-Uni gets poor performance because of false detection from similar objects or background noise. Additionally, incorrect results are likely leading to unwanted updates, resulting in the fact that operating efficiency is lower than LMCF. Without high-confidence update strategy, LMCF-NU updates the tracking model in each frame, thus the tracking speed is dramatically reduced to nearly half to LMCF and the accuracy is also less than LMCF. Without both of these two, LMCF-N2 reaches the last one in all evaluation metrics. Although the proposed multimodal detection increases the detection time, our high-confidence update strategy speeds up the model update process significantly. Both of them improve the tracking performance observably according to the experimental results.

Table 2. Characteristics and tracking results of LMCF, DeepLMCF, LMCF-Uni, LMCF-NU and LMCF-N2. The entries in red denote the best results and the ones in blue indicate the second best.

Trackers	multimodal detection	high-confidence update	feature representations	OPE		TRE		SRE		mean
				precision	success	precision	success	precision	success	FPS
LMCF-N2	No	No	conventional	0.799	0.586	0.813	0.612	0.740	0.540	60.74
LMCF-Uni	No	Yes	conventional	0.809	0.606	0.815	0.616	0.757	0.549	61.38
LMCF-NU	Yes	No	conventional	0.813	0.605	0.820	0.619	0.750	0.545	46.45
LMCF	Yes	Yes	conventional	0.839	0.624	0.829	0.625	0.760	0.552	85.23
DeepLMCF	Yes	Yes	deep CNNs	0.892	0.643	0.877	0.649	0.850	0.596	8.11

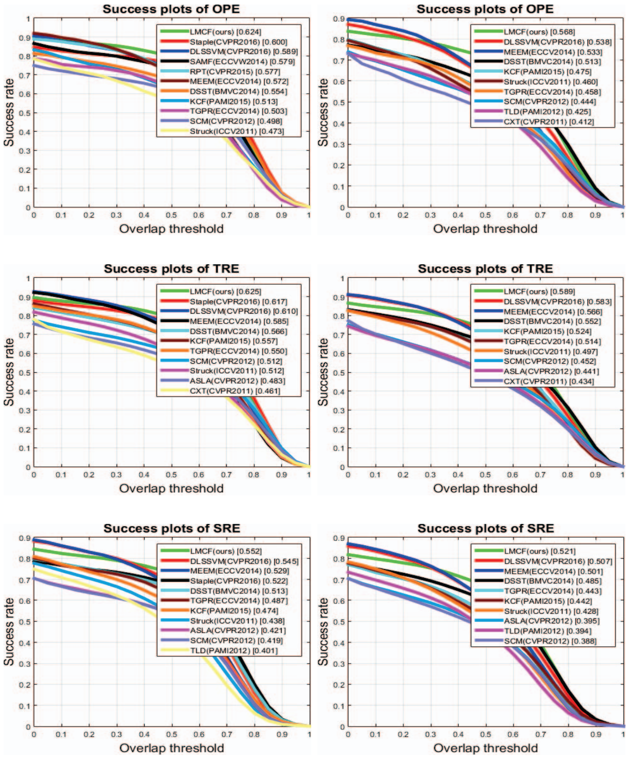


Figure 3. The success plots of OPE, TRE, SRE on OTB-13 (left column) and OTB-15 (right column). The numbers in the legend indicate the average AUC scores for success plots. The years and original sources of these trackers are also shown in the legend. Results are best viewed on high-resolution displays.

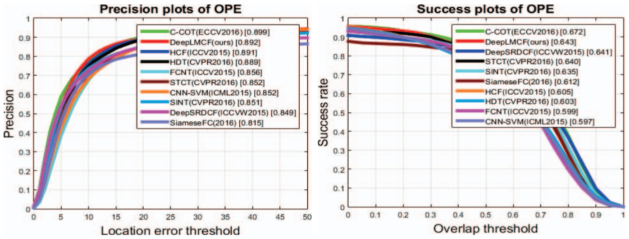


Figure 5. The precision and success plot of OPE on OTB-13. The numbers in the legend indicate the average precision scores for precision plot and the average AUC scores for success plot. Results are best viewed on high-resolution displays.

3.3. Evaluation on LMCF

We evaluate LMCF with 9 state-of-the-art trackers designed with conventional hand-crafted features including Struck [13], MEEM [31], TGPR [10], DLSSVM [22], Staple [2], KCF [14], RPT [20], DSST [5] and SAMF [19]. Among them, Struck and DLSSVM are structured SVM based methods, Staple, KCF, DSST, RPT and SAMF are CF based algorithms, MEEM and TGPR are developed based on regression and multiple trackers.

Figure 3 illustrates the success plots of top ten trackers on both OTB-13 and OTB-15. LMCF performs best with all OPE, TRE and SRE evaluation metrics in the two benchmarks. Struck performed the first when the original benchmark [29] first came out, so that it is a good representation of its previous trackers. LMCF significantly improves Struck by an average improvement of 15% in the average AUC scores. The DSST and SAMF mainly focus on the scale estimation, their speed are 24 FPS and 7 FPS as they reported. Our method employs the scale estimation method from DSST, but the proposed LMCF performs favorably over the DSST as well as SAMF while runs more than 3 times faster than DSST and more than 11 times faster than SAMF. As for tracking efficiency, Staple and KCF are the only two with comparable reported speeds of 80 FPS and 172 FPS, while LMCF outperforms them in all evaluations. Moreover, LMCF is also superior to other up-to-date trackers like MEEM, TGPR, RPT, SAMF and DLSSVM with a significantly higher speed.

For detailed analyses, we also evaluate LMCF with these trackers on various challenging attributes in OTB-13 as shown in Figure 4. The results demonstrate that LMCF performs well on most attributes, especially on occlusion, scale variation, illumination variation, background clutter and out of plane rotation.

3.4. Evaluation on DeepLMCF

To further improve the tracking accuracy and robustness of LMCF, we implement DeepLMCF with deep CNNs based features. It is compared with 9 up-to-date CNNs based trackers including C-COT [8], DeepSRDCF[6],

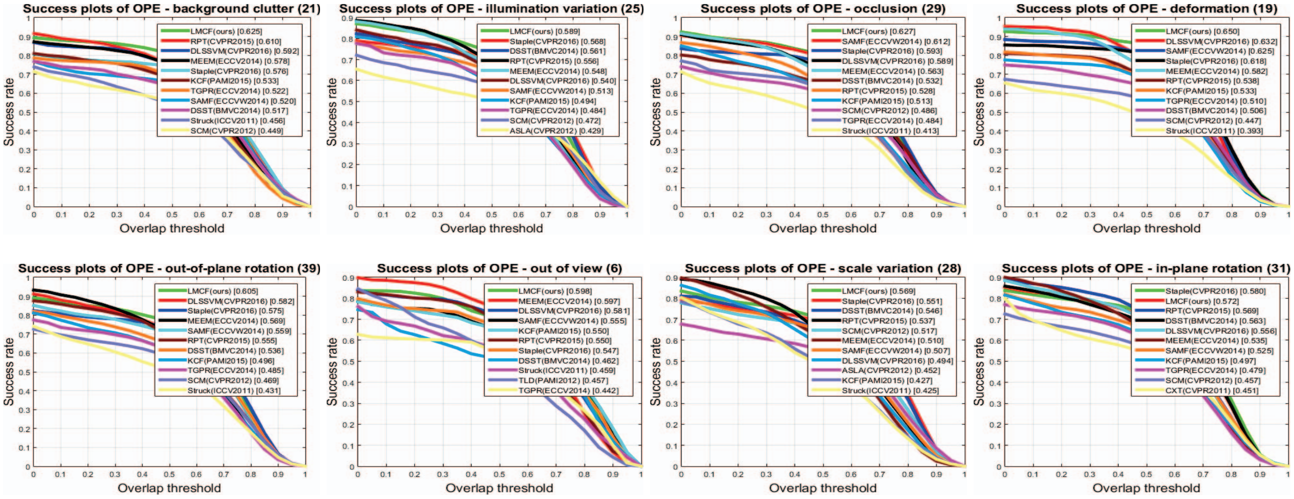


Figure 4. The success plots for 8 challenging attributes including background clutter, illumination variation, occlusion, deformation, out-of-plane rotation, out-of-view, scale variation and in-plane rotation. The proposed LMCF performs best in almost all the attributes. Results are best viewed on high-resolution displays.

HCF[21], HDT[23], STCT[28], CNN-SVM[15], SINT[25], FCNT[27] and SiameseFC[3].

Figure 5 demonstrates the performance of DeepLMCF with the 9 CNNs based trackers on OTB-13. Although the proposed DeepLMCF scores the second following the C-COT tracker on the precision and success scores, the tracking speed of DeepLMCF is 40 times faster than C-COT with the speed from its reported results at about 0.25 FPS, which is a severe limitation of its application. The most related method to DeepLMCF is HCF due to the similar feature hierarchy. But DeepLMCF keeps ahead of it especially on success score mainly because the scale variations of the target are not considered by HCF. Moreover, HCF and SiameseFC are the only two with comparable reported speeds of 10 FPS and 58 FPS, while LMCF performs superiorly against them in both evaluations. In summary, the proposed DeepLMCF outperforms these trackers except for C-COT while remains a comparably fast speed at more than 10 FPS.

4. Conclusion

In this paper, we propose a novel large margin object tracking method with circulant feature maps. A bridge is built up to link the framework with correlation filter. Hence, the proposed LMCF tracker absorbs the strong discriminative ability from structured output SVM and speeds up by the correlation filter algorithm significantly. In order to prevent model drift introduced by similar objects or background noise, a multimodal target detection technique is proposed to ensure the correct detection. Moreover, we establish a high-confidence model update strategy to avoid the model corruption problem. Furthermore, the proposed tracking algorithm is equipped with strong compatibility, thus we also implement a deep CNNs based version DeepLMCF to verify its outstanding performance. Sufficient evaluations on challenging benchmark datasets demonstrate that the proposed LMCF and DeepLMCF tracking algorithms perform well against most state-of-the-art methods including both conventional features and deep CNNs features based trackers. It is worth to emphasize that our proposed algorithm not only performs superiorly, but also runs at a very fast speed which is sufficient for realtime applications.

References

[1] B. Babenko, M.-H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1619–1632, 2011. 1

[2] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr. Staple: Complementary learners for real-time tracking. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2, 5, 7

[3] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr. Fully-convolutional siamese networks for object tracking. *arXiv preprint arXiv:1606.09549*, 2016. 8

[4] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2544–2550. IEEE, 2010. 2

[5] M. Danelljan, G. Häger, F. Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *British Machine Vision Conference, Nottingham, September 1-5, 2014*. BMVA Press, 2014. 2, 5, 6, 7

[6] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Convolutional features for correlation filter based visu-

al tracking. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 58–66, 2015. 7

[7] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Learning spatially regularized correlation filters for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4310–4318, 2015. 2

[8] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg. Beyond correlation filters: learning continuous convolution operators for visual tracking. In *European Conference on Computer Vision*, pages 472–488. Springer, 2016. 7

[9] M. Danelljan, F. Shahbaz Khan, M. Felsberg, and J. Van de Weijer. Adaptive color attributes for real-time visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1090–1097, 2014. 1, 2, 6

[10] J. Gao, H. Ling, W. Hu, and J. Xing. Transfer learning based visual tracking with gaussian processes regression. In *European Conference on Computer Vision*, pages 188–203. Springer, 2014. 7

[11] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via on-line boosting. In *BMVC*, volume 1, page 6, 2006. 1

[12] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *Computer Vision–ECCV 2008*, pages 234–247. Springer, 2008. 1

[13] S. Hare, A. Saffari, and P. H. Torr. Struck: Structured output tracking with kernels. In *2011 International Conference on Computer Vision*, pages 263–270. IEEE, 2011. 1, 2, 3, 7

[14] 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. 1, 2, 3, 5, 7

[15] S. Hong, T. You, S. Kwak, and B. Han. Online tracking by learning discriminative saliency map with convolutional neural network. In D. Blei and F. Bach, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 597–606. JMLR Workshop and Conference Proceedings, 2015. 1, 2, 8

[16] X. Jia, H. Lu, and M.-H. Yang. Visual tracking via adaptive structural local sparse appearance model. In *Computer vision and pattern recognition (CVPR), 2012 IEEE Conference on*, pages 1822–1829. IEEE, 2012. 1

[17] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *IEEE transactions on pattern analysis and machine intelligence*, 34(7):1409–1422, 2012. 1

[18] J. Kwon and K. M. Lee. Visual tracking decomposition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1269–1276. IEEE, 2010. 1

[19] Y. Li and J. Zhu. A scale adaptive kernel correlation filter tracker with feature integration. In *European Conference on Computer Vision*, pages 254–265. Springer, 2014. 7

[20] Y. Li, J. Zhu, and S. C. Hoi. Reliable patch trackers: Robust visual tracking by exploiting reliable patches. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 7

[21] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang. Hierarchical convolutional features for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3074–3082, 2015. 1, 2, 6, 8

[22] J. Ning, J. Yang, S. Jiang, L. Zhang, and M.-H. Yang. Object tracking via dual linear structured svm and explicit feature map. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4266–4274. 1, 2, 3, 5, 7

[23] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, and J. L. M.-H. Yang. Hedged deep tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1, 2, 8

[24] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3):125–141, 2008. 1

[25] R. Tao, E. Gavves, and A. W. M. Smeulders. Siamese instance search for tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 8

[26] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6(Sep):1453–1484, 2005. 1

[27] L. Wang, W. Ouyang, X. Wang, and H. Lu. Visual tracking with fully convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3119–3127, 2015. 1, 2, 8

[28] L. Wang, W. Ouyang, X. Wang, and H. Lu. Stct: Sequentially training convolutional networks for visual tracking. CVPR, 2016. 1, 2, 8

[29] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2411–2418, 2013. 1, 6, 7

[30] Y. Wu, J. Lim, and M.-H. Yang. Object tracking benchmark. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(9):1834–1848, 2015. 1, 4, 6

[31] J. Zhang, S. Ma, and S. Sclaroff. Meem: robust tracking via multiple experts using entropy minimization. In *European Conference on Computer Vision*, pages 188–203. Springer, 2014. 1, 7

[32] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang. Fast visual tracking via dense spatio-temporal context learning. In *European Conference on Computer Vision*, pages 127–141. Springer, 2014. 2

[33] W. Zhong, H. Lu, and M.-H. Yang. Robust object tracking via sparsity-based collaborative model. In *Computer vision and pattern recognition (CVPR), 2012 IEEE Conference on*, pages 1838–1845. IEEE, 2012. 1

[34] W. Zuo, X. Wu, L. Lin, L. Zhang, and M.-H. Yang. Learning support correlation filters for visual tracking. *arXiv preprint arXiv:1601.06032*, 2016. 1, 3