Multi-task Correlation Particle Filter for Robust Object Tracking

Tianzhu Zhang[1,2] Changsheng Xu[1,2] Ming-Hsuan Yang[3]

[1] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences

[2] University of Chinese Academy of Sciences [3] University of California at Merced

Abstract

In this paper, we propose a multi-task correlation particle filter (MCPF) for robust visual tracking. We first present the multi-task correlation filter (MCF) that takes the inter-dependencies among different features into account to learn correlation filters jointly. The proposed MCPF is designed to exploit and complement the strength of a MCF and a particle filter. Compared with existing tracking methods based on correlation filters and particle filters, the proposed tracker has several advantages. First, it can shepherd the sampled particles toward the modes of the target state distribution via the MCF, thereby resulting in robust tracking performance. Second, it can effectively handle large-scale variation via a particle sampling strategy. Third, it can effectively maintain multiple modes in the posterior density using fewer particles than conventional particle filters, thereby lowering the computational cost. Extensive experimental results on three benchmark datasets demonstrate that the proposed MCPF performs favorably against the state-of-the-art methods.

1. Introduction

Visual tracking is one of the most important tasks in computer vision that finds numerous applications such as video surveillance, motion analysis, and autonomous driving, to name a few [38, 13, 36, 46, 35, 31, 14]. The main challenge for robust visual tracking is to account for large appearance changes of target objects over time. Despite significant progress in recent years, it remains a difficult task to develop robust algorithms to estimate object states in tracking scenarios with challenging factors such as illumination changes, fast motions, pose variations, partial occlusions and background clutters.

Correlation filters have recently been introduced into visual tracking and shown to achieve high speed as well as robust performance [4, 9, 16, 15, 18, 26, 24, 21, 25, 29]. Recognizing the success of deep convolutional neural networks (CNNs) on a wide range of visual recognition tasks, several tracking methods based on deep features and correlation filters have been developed [25, 29]. Empirical studies using large object tracking benchmark datasets show that these CNN based trackers [25, 29] perform favorably against methods based on hand-crafted features. Figure 1 shows some tracking results where the CF2 [25] and HDT [29] perform well against the DSST [9] and KCF [16] schemes which achieve the state-of-the-art results in the VOT challenge.

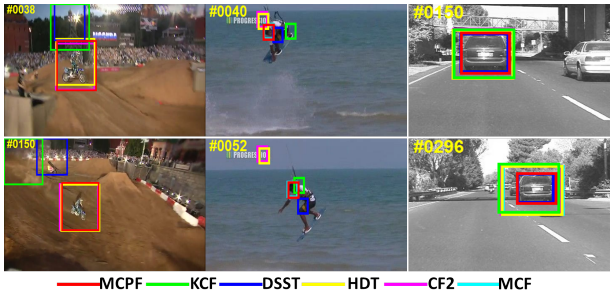
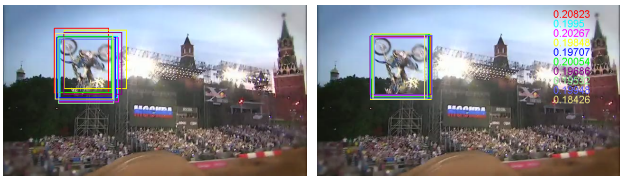


Figure 1. Comparisons of the proposed MCPF tracker with the state-of-the-art correlation filter trackers (DSST [9], KCF [16], CF2 [25], and HDT [29]) on the *motorRolling*, *KiteSurf*, and *car4* sequences [36]. These trackers perform differently as various features and scale handling strategies are used. The proposed algorithm performs favorably against these trackers.

Despite achieving the state-of-the-art performance, existing CNN based correlation filter trackers [25, 29] have several limitations. (1) These trackers learn correlation filter for each layer independently without considering their relationship. In [25, 29], adaptive linear correlation filters rather than the outputs of each convolutional layer are used. Since features from different layers can enhance and complement each other, existing CNN based correlation trackers (CF2 [25] and HDT [29]) perform well. Nevertheless, these methods assume that correlation filters of different features are independent. Ignoring the relationships between correlation filters tends to make the tracker more prone to drift away from target objects in cases of significant changes in appearance. To deal with this issue, we propose a multi-task correlation filter (MCF) to exploit interdependencies among different features to obtain their correlation filters jointly. Here, learning the correlation filter of each type of feature is viewed as an individual task. As shown in Figure 1, the

MCF achieves better performance than the CF2 and HDT in the *KiteSurf* sequence. (2) These trackers [25, 29] do not handle scale variation well. Recently Danelljan et al. propose the DSST method [9] with adaptive multi-scale correlation filters using HOG features to handle the scale variation of target objects. However, the adaptive multi-scale strategy does not facilitate the tracking methods based on CNN features and correlation filters [25, 29] well (see Section 4). To overcome this issue, we resort to particle filters [1, 19] to handle large-scale variation. In a particle-based tracking method, the state space for target objects undergoing large-scale variation can be covered with dense sampling. As shown in Figure 1, the HDT and CF2 methods do not track the target object with scale variation in the *car4* sequence well, but the proposed algorithm performs well by using particle filter.

In general, when more particles are sampled and a robust object appearance model is constructed, particle filter based tracking algorithms are likely to perform reliably in cluttered and noisy scenes. However, the computational cost of particle filter based trackers usually increases significantly with the number of particles. Furthermore, particle filter based trackers determine each target object state based on the sampled particle separately. If the sampled particles do not cover target object states well as shown in Figure 2(a), the predicted target state may be not correct. To overcome this problem, it is better to shepherd the sampled particles toward the modes of the target state distribution. In this work, we exploit the strength of the MCF and particle filter, and complement each other: (1) Particle filters provide a probabilistic framework for tracking objects by propagating the posterior density over time based on a factored sampling technique. With dense sampling, the states for target objects undergoing large-scale variations can be covered. Therefore, particle filters can effectively help the MCF handle scale variation problem. (2) For each sampled particle, the MCF can be applied such that particles are shepherded toward the modes of the target state distribution as shown in Figure 2(b). Here, each particle is used as a base sample to construct a block-circulant circulant matrix, of which each block denotes a shifted sample [15]. Then, the MCF evaluates the similarity by computing the inner product for each shifted sample relative to the learned filter. Finally, the response map is obtained, and the maximum response is used to shepherd this particle. It is clear that each particle can densely cover a state subspace with the MCF, and we do not need to draw particles *densely* to maintain multiple possible states. As a result, we can maintain multiple modes using fewer particles in comparison to the conventional particle filter. Since the computational load of a particle-based tracking method depends heavily on the number of drawn particles, the multi-task correlation filter can be used in these methods for efficient and effective visual tracking.



(a) Sampled particles (b) Particles after shepherding

Figure 2. The multi-task correlation filter can be used to shepherd the sampled particles toward the modes of the target state distribution. The numbers in (b) are the scores of correlation filter for the particles. Different colored boxes indicate the respective locations and scores.

In this work, we propose a Multi-task Correlation Particle Filter (MCPF) for robust visual tracking, which enjoys the merits of both particle filters and correlation filters, e.g., robustness to scale variation, and computational efficiency. The contributions of the proposed MCPF tracking method are as follows. (1) Different from existing methods that learn correlation filters for different features independently, the proposed MCPF model can exploit interdependencies among different features to learn their correlation filters jointly to improve tracking performance. (2) The proposed MCPF tracker can effectively overcome the scale variation problem via a particle sampling strategy as in traditional particle filter. In particular, our MCPF tracker can cover multiple modes in the posterior density using fewer particles than conventional particle filters do, resulting in low computational cost. (3) The proposed MCPF tracker can shepherd the sampled particles toward the modes of the target state distribution using the proposed MCF, resulting in robust tracking performance. During tracking, a target object state is estimated as a weighted average of all particles. Here, the weights are based on the outputs of the proposed MCF. We evaluate the proposed tracking algorithm on three tracking benchmark datasets [36, 37, 22]. Extensive experimental results on three benchmark datasets show that the proposed MCPF tracking algorithm performs favorably against the state-of-the-art methods regarding accuracy, efficiency, and robustness.

2. Related Work

A comprehensive review of the tracking methods is beyond the scope of the paper, and surveys of this field can be found in [38, 36, 31]. In this section, we discuss the methods closely related to this work, mainly regarding correlation and particle filters.

Correlation Filters. Correlation filters have recently attracted considerable attention in visual tracking due to computational efficiency and robustness. Bolme et al. model target appearance by learning an adaptive correlation filter which is optimized by minimizing the output sum of

squared error (MOSSE) [4]. Henriques et al. exploit the circulant structure of shifted image patches in a kernel space and propose the CSK method based on intensity features [15], and extend it to the KCF approach [16] with the HOG descriptors. Danelljan et al. propose the DSST method [9] with adaptive multi-scale correlation filters using HOG features to handle the scale change of target object. In [40], Zhang et al. incorporate circulant property of target template to improve sparse based trackers. Hong et al. [18] propose a biology-inspired framework (MUSTer) where short-term processing and long-term processing are cooperated with each other. In [26], Ma et al. introduce an online random fern classifier as a re-detection component for long-term tracking. Recently, Danelljan et al. propose a continuous convolution filters for tracking with multi-scale deep features to account for appearance variation caused by large scale change [11].

Correlation filters based on local patches or parts have also been developed [24, 23]. In [24], a part-based method is proposed where object parts are independently tracked by the KCF tracker [16]. Liu et al. [23] propose a part based structural correlation filter to preserve target object structure for visual tracking. In [21], Li et al. introduce reliable local patches to exploit the use of local contexts and treat the KCF as the base tracker. Recently, in [25, 29], correlation filters are learned independently for each type of feature. Different from existing tracking methods based on correlation filters, we propose a multi-task correlation filter to exploit interdependencies among different features to learn their correlation filters jointly.

Particle Filters. In visual tracking, particle filters or Sequential Monte Carlo (SMC) methods [19, 45, 47] have been widely adopted. For robust performance, the number of drawn samples must be sufficient to cover the possible states. However, the dense sampling of particles generally results in high computation load for visual tracking as each one needs to be evaluated. Consequently, numerous techniques have been presented to improve the sampling efficiency of particle filtering [19, 6, 20, 48]. Importance sampling [19] is introduced to obtain better proposal by combining prediction based on the previous configuration with additional knowledge from auxiliary measurements. In [20], subspace representations are used with the Rao-Blackwell particle filtering for visual tracking. On the other hand, the number of particle samples can be adjusted according to an adaptive noise component [48]. In [6], the observation likelihood is computed in a coarse-to-fine manner, which allows efficient focus on more promising particles. Different from the above methods, we adopt a multi-task correlation filter to shepherd particles toward the modes of a target state distribution and thereby reduce the number of particles and computational cost.

3. Proposed Algorithm

In this section, we present the multi-task correlation particle filter for visual tracking. Different from existing methods [16, 15] that learn correlation filter independently, the proposed MCF considers the interdependencies among different features and parts, and learns the correlation filters jointly. Furthermore, our tracker can effectively handle scale variation via particle sampling strategy.

3.1. Multi-task Correlation Filter

The key idea of tracking methods based on correlation filters [9, 16, 25, 29] is that numerous negative samples are used to enhance the discriminability of the tracking-by-detection scheme while exploring the circulant matrix for computational efficiency. In visual tracking, object appearance is modeled via a correlation filter $\mathbf{w}$ trained on an image patch $\mathbf{x}$ of $M \times N$ pixels, where all the circular shifts of $\mathbf{x}_{m,n}$, $(m,n) \in \{0,1,\ldots,M-1\} \times \{0,1,\ldots,N-1\}$, are generated as training samples with Gaussian function label $\mathbf{y}_{m,n}$. Given K different features (HOG, color, or CNN features), we use $\mathbf{X}_k = [\mathbf{x}_{0,0},\ldots,\mathbf{x}_{m,n},\ldots\mathbf{x}_{M-1,N-1}]^\top$ to denote all training samples of the k-th type of feature ($k = 1,\ldots,K$). The goal is to find the optimal weights $\mathbf{w}_k$ for K different features,

$$\underset{\{\mathbf{w_k}\}_{k=1}^{K}}{\arg\min} \sum_k \|\mathbf{X}_k\mathbf{w}_k - \mathbf{y}\|_F^2 + \lambda\|\mathbf{w_k}\|_F^2, \tag{1}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, $\mathbf{y} = [\mathbf{y}_{0,0},\ldots,\mathbf{y}_{m,n},\ldots,\mathbf{y}_{M-1,N-1}]^\top$, and λ is a regularization parameter. The objective function (1) can equivalently be expressed in its dual form,

$$\min_{\{\mathbf{z}_k\}_{k=1}^{K}} \sum_k \frac{1}{4\lambda}\mathbf{z}_k^\top\mathbf{G}_k\mathbf{z}_k + \frac{1}{4}{\mathbf{z}_k}^\top\mathbf{z}_k - {\mathbf{z}_k}^\top\mathbf{y}. \tag{2}$$

Here, the vector $\mathbf{z}_k$ contains $M \times N$ dual optimization variables $\mathbf{z}_k^{m,n}$, and $\mathbf{G}_k = \mathbf{X}_k{\mathbf{X}_\mathbf{k}}^\top$. These two solutions are related by $\mathbf{w}_k = \frac{{\mathbf{X}_k}^\top\mathbf{z}_k}{2\lambda}$. The learned $\mathbf{z}_k^{m,n}$ selects discriminative training samples $\mathbf{x}_k^{m,n}$ to distinguish the target object from the background. Here, the training samples $\mathbf{x}_k^{m,n}$, $(m,n) \in \{0,1,\ldots,M-1\} \times \{0,1,\ldots,N-1\}$ are the all possible circular shifts, which represent the possible locations of the target object. Putting the learned $\mathbf{z}_k$ of the K different features together, we obtain $\mathbf{Z} = [\mathbf{z}_1,\mathbf{z}_2,\cdots,\mathbf{z}_K] \in \mathbb{R}^{MN\times K}$.

For $\mathbf{Z}$, we have the following observations: (1) For each feature k, only a few possible locations $\mathbf{x}_k^{m,n}$ need to be selected to localize where the target object is in the next frame. Ideally, only one possible location corresponds to the target object. (2) Among K different features, the learned $\mathbf{z}_k$ should select similar circular shifts such that they have similar motion. As a result, the learned $\mathbf{z}_k$ should be similar.

Based on the above observation, it is clear that different features should have similar $\mathbf{z}_k$ to make them have consistent localization of the target object, and their correlation filters should be learned jointly to distinguish the target from the background. In this work, we use the convex $\ell_{p,q}$ mixed norm, especially, $\ell_{2,1}$ to model the underlying structure information of $\mathbf{Z}$ and obtain the multi-task correlation filter for object tracking as

$$\min_{\{\mathbf{z}_k\}_{k=1}^K} \sum_k \frac{1}{4\lambda}\mathbf{z}_k^\top \mathbf{G}_k \mathbf{z}_k + \frac{1}{4}\mathbf{z}_k{}^\top \mathbf{z}_k - \mathbf{z}_k{}^\top \mathbf{y} + \gamma \|\mathbf{Z}\|_{2,1}, \quad (3)$$

where γ is a tradeoff parameter between reliable reconstruction and joint sparsity regularization. The definition of the $\ell_{p,q}$ mixed norm is $\|\mathbf{Z}\|_{p,q} = \left(\sum_i \left(\sum_j |[\mathbf{Z}]_{ij}|^p\right)^{\frac{q}{p}}\right)^{\frac{1}{q}}$ and $[\mathbf{Z}]_{ij}$ denotes the entry at the i-th row and j-th column of $\mathbf{Z}$.

To solve (3), we use the Accelerated Proximal Gradient method, which has been widely used to efficiently solve convex optimization problems with non-smooth terms [42, 43]. Although it is time-consuming to compute $\mathbf{G}_k$ directly, it can be computed efficiently in the Fourier domain by considering the circulant structure property of $\mathbf{G}_k$. More details can be found in the supplementary material. After solving this optimization problem, we obtain the multi-task correlation filter $\mathbf{z}_k$ for each type of feature.

3.2. Multi-task Correlation Particle Filter

The proposed multi-task correlation particle filter is based on Bayesian sequential importance sampling. which recursively approximates the posterior distribution using a finite set of weighted samples for estimating the posterior distribution of state variables. Let $\mathbf{s}_t$ and $\mathbf{y}_t$ denote the state variable (e.g., location and scale) of an object at time t and its observation respectively. The posterior density function $p(\mathbf{s}_t|\mathbf{y}_{1:t-1})$ at each time instant t can be obtained recursively in two steps, namely prediction and update. The prediction stage uses the probabilistic system transition model $p(\mathbf{s}_t|\mathbf{s}_{t-1})$ to predict the posterior distribution of $\mathbf{s}_t$ given all available observations $\mathbf{y}_{1:t-1} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_{t-1}\}$ up to time $t-1$, and is recursively computed by

$$p(\mathbf{s}_t|\mathbf{y}_{1:t-1}) = \int p(\mathbf{s}_t|\mathbf{s}_{t-1}) p(\mathbf{s}_{t-1}|\mathbf{y}_{1:t-1}) d\mathbf{s}_{t-1}, \quad (4)$$

where $p(\mathbf{s}_{t-1}|\mathbf{y}_{1:t-1})$ is known at time $t-1$, and $p(\mathbf{s}_t|\mathbf{s}_{t-1})$ is the state prediction. When the observation $\mathbf{y}_t$ is available, the state is predicted by

$$p(\mathbf{s}_t|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{s}_t) p(\mathbf{s}_t|\mathbf{y}_{1:t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})}, \quad (5)$$

where $p(\mathbf{y}_t|\mathbf{s}_t)$ denotes the likelihood function. The posterior $p(\mathbf{s}_t|\mathbf{y}_{1:t})$ is approximated by n particles $\{\mathbf{s}_t^i\}_{i=1}^n$,

$$p(\mathbf{s}_t|\mathbf{y}_{1:t}) \approx \sum_{i=1}^n w_t^i \delta(\mathbf{s}_t - \mathbf{s}_t^i), \quad (6)$$

where $\delta(\cdot)$ is the Dirac delta measure, and w_t^i is the weight associated to the particle i. Each particle weight is computed by

$$w_t^i \propto w_{t-1}^i \frac{p(\mathbf{y}_t|\mathbf{s}_t^i) p(\mathbf{s}_t^i|\mathbf{s}_{t-1}^i)}{q(\mathbf{s}_t^i|\mathbf{s}_{t-1}^i, \mathbf{y}_t)}, \quad (7)$$

where $q(\cdot)$ is the importance density function which is chosen to be $p(\mathbf{s}_t^i|\mathbf{s}_{t-1}^i)$ and this leads to $w_t^i \propto w_{t-1}^i p(\mathbf{y}_t|\mathbf{s}_t^i)$. Then, a re-sampling algorithm is applied to avoid the degeneracy problem [1]. In this case, the weights are set to $w_{t-1}^i = 1/n \; \forall i$. Therefore, we can rewrite the importance weights in (8), which are proportional to the likelihood function $p(\mathbf{y}_t|\mathbf{s}_t^i)$,

$$w_t^i \propto p(\mathbf{y}_t|\mathbf{s}_t^i). \quad (8)$$

The above re-sampling step derives the particles based on the weights of the previous step, and all the new particles are updated by the next frame likelihood function.

Given the learned MCF $\mathbf{z}_k$ and target appearance model $\bar{\mathbf{x}}$, each particle can be shepherded toward the modes of the target state distribution by using its circular shifts. For particle i with the search window size $M \times N$, we can compute its response map by

$$\mathbf{r} = \sum_k \mathcal{F}^{-1}(\mathcal{F}(\mathbf{z}_k) \odot \mathcal{F}(\langle \mathbf{y}_t^i, \bar{\mathbf{x}} \rangle)). \quad (9)$$

Here, $\mathbf{y}_t^i$ is the observation of particle i, $\odot$ is the Hadamard product, and $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the Fourier transform and its inverse, respectively. Then, the particle i is shepherded by searching for the location of the maximal value of $\mathbf{r}$. For simplicity, we define the above process as a MCF operator for state calculation $\mathcal{S}_{mcf} : \mathcal{R}^d \rightarrow \mathcal{R}^d$, where d is the state space dimensionality, and the state of each particle is shifted $\mathbf{s}_t^i \rightarrow \mathcal{S}_{mcf}(\mathbf{s}_t^i)$. We define the response of the MCF for particle $\mathbf{s}_t^i$ as the maximal value of $\mathbf{r}$, which is denoted as $\mathcal{R}_{mcf}(\mathbf{s}_t^i)$. Then we set $p(\mathbf{y}_t|\mathbf{s}_t^i) = \mathcal{R}_{mcf}(\mathbf{s}_t^i)$. As a result, the particle weights are proportional to the response of the MCF and defined by

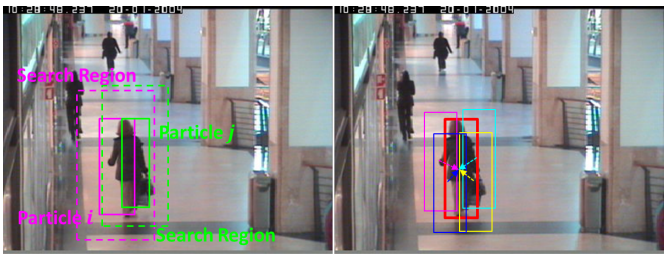
$$w_t^i \propto \mathcal{R}_{mcf}(\mathbf{s}_t^i). \quad (10)$$

Finally, the state of target object is estimated as

$$\mathbf{E}[\mathbf{s}_t|\mathbf{y}_{1:t}] \approx \sum_{i=1}^n w_t^i \mathcal{S}_{mcf}(\mathbf{s}_t^i). \quad (11)$$

3.3. MCPF Tracker

Based on the multi-task correlation particle filter, we propose a MCPF tracker. The first step generates particles using the transition model $p(\mathbf{s}_t|\mathbf{s}_{t-1})$ and re-samples them. The second step applies the proposed MCF to each particle such that it is shifted to a stable location. The third step updates the weights using the responses of the MCF. Finally,



(a) (b)

Figure 3. The MCPF can cover object state space well with a few particles. Each particle corresponds to an image region enclosed by a bounding box. (a) The MCPF can cover object state space well by using few particles with the search region where each particle covers the state subspace corresponding to all shifted region of the target object. (b) The MCPF can shepherd the sampled particles toward the modes of the target state distribution, which correspond to the target locations in the image.

the optimal state is obtained using (11). To update the MCF for visual tracking, we adopt an incremental strategy similar to that in [9, 16, 25, 29], which only uses new samples $\mathbf{x}_k$ in the current frame to update models by

$$\begin{aligned} \mathcal{F}(\bar{\mathbf{x}}_k)^t &= (1-\eta)\mathcal{F}(\bar{\mathbf{x}}_k)^{t-1} + \eta\mathcal{F}(\mathbf{x}_k)^t, \\ \mathcal{F}(\mathbf{z}_k)^t &= (1-\eta)\mathcal{F}(\mathbf{z}_k)^{t-1} + \eta\mathcal{F}(\mathbf{z}_k)^t, \end{aligned} \tag{12}$$

where η is the learning rate parameter.

3.4. Discussion

We discuss how the MCPF tracker performs with particles, correlation filters and circular shifts of target objects for visual tracking using an example.

First, tracking methods based on conventional particle filters need to draw samples densely to cover the possible states and thus entail a high computational cost. The MCF can refine particles to cover target states and effectively reduce the number of particles required for accurate tracking. As shown in Figure 3(a), for a particle j (denoted in a green bounding box), its search region (denoted in a green bounding box with dashed line) is twice the size of the possible object translations, which determines the total number of possible circulant shifts of a correlation filter. Although this particle is not drawn at the location where the target object is, its search region (with possible circulant shifts) covers the state of the target object. For each particle with a search region of $M \times N$ pixels, it contains $M \times N$ circular shifts, which are all shifts of this particle. Here, each particle can be viewed as a base particle, and its circular shifts are all virtual particles with the same scale. With the proposed MCF, each particle can be shepherded toward the modes of the target object distribution (where the target object is) as shown in Figure 3(b). Therefore, we do not need to draw particles densely as each particle can cover a local search region including many possible states of a target object, and reduce computational load.

Second, the proposed MCPF can handle scale variation well via a particle sampling strategy. Particle filters can use dense sampling techniques to cover the state space of target object undergoing large-scale variation. Thus, particle filters can effectively help the MCF handle scale variation, as demonstrated in the attribute-based experiments with large-scale variation as shown in Figure 5.

4. Experimental Results

We evaluate the proposed MCPF algorithm with the state-of-the-art trackers on benchmark datasets. The source code is available at `http://nlpr-web.ia.ac.cn/mmc/homepage/tzzhang/mcpf.html` and more results can be found in the supplementary material.

4.1. Experimental Setups

Implementation Details. We use the same experimental protocols in the CF2 method [25] for fair comparisons in which the VGG-Net-19 [30] is used for feature extraction. We first remove the fully-connected layers and use the outputs of the *conv3-4*, *conv4-4* and *conv5-4* convolutional layers as our features. Note that, a variety of features can be adopted, such as HOG, other layers of CNN features as in the HDT [29]. We set the regularization parameters of (3) to $\lambda = 10^{-4}$ and $\gamma = 10^{-2}$, and use a kernel width of 0.1 for generating the Gaussian function labels. The learning rate η in (12) is set to 0.01. To remove the boundary discontinuities, the extracted feature channels of each convolutional layer are weighted by a cosine window [16]. We implement our tracker in MATLAB on an Intel 3.10 GHz CPU with 256 GB RAM and use the MatConvNet toolbox [33] where the computation of forward propagation on CNNs is carried out on a GeForce GTX Titan X GPU. We use the same parameter values for all the experiments. Furthermore, all the parameter settings are available in the source code. As in [41, 44], the variances of affine parameters for particle sampling are set to $(0.01, 0.0001, 0.0001, 0.01, 2, 2)$, and the particle number is set to 100.

Datasets. Our method is evaluated on three benchmark datasets: OTB-2013 [36], OTB-2015 [37], and Temple Color [22]. The first two datasets are composed of 50 and 100 sequences, respectively. The images are annotated with ground truth bounding boxes and various visual attributes. The Temple Color dataset [22] contains 128 videos.

Evaluation Metrics. We compare the proposed algorithm with the state-of-the-art tracking methods using evaluation metrics and code provided by the respective benchmark dataset. For the OTB-2013, OTB-2015, and Temple Color datasets, we employ the one-pass evaluation (OPE) and

Table 1. Model analysis by comparing MCPF, MCF, CPF, CF2, and CF2S. The AUC and PS are reported on the OTB-2013 and OTB-2015 datasets (AUC/PS) corresponding to the OPE.

Dataset	MCPF	MCF	CPF	CF2	CF2S
OTB-2013	**67.7/91.6**	60.7/89.3	65.7/89.3	60.5/89.1	63.4/89.1
OTB-2015	**62.8/87.3**	56.6/84.7	61.2/86.3	56.2/83.7	59.1/84.0

use two metrics: precision and success plots. The precision metric computes the rate of frames whose center location is within some certain distance with the ground truth location. The success metric computes the overlap ratio between the tracked and ground truth bounding boxes. In the legend, we report the area under curve (AUC) of success plot and precision score at 20 pixels threshold (PS) corresponding to the one-pass evaluation for each tracking method.

4.2. Model Analysis

In the proposed MCPF tracker, we adopt the MCF to exploit interdependencies among different features and particle filters to handle scale variation. With different experimental settings, we have six different trackers including MCPF, MCF, CPF, CF2 [25], and CF2S. Here, MCF is our MCPF without using particle filters, CPF is the MCPF using traditional correlation filter instead of the multi-task correlation filter, and CF2S is the CF2 [25] using the adaptive multi-scale strategy as the DSST [9].

Table 1 shows that both the multi-task correlation filter and particle filters can improve object tracking performance. We have the following observations from the experimental results. First, multi-task correlation filter can improve tracking performance. Compared with CPF, MCPF achieves about $2.0\%/2.3\%$ and $1.6\%/1.0\%$ improvement with AUC and PS metrics on the OTB-2013 and OTB-2015 datasets. Furthermore, compared with CF2, MCF achieves about 0.4% and 1.0% improvement with AUC and PS on the OTB-2015 dataset.

Second, particle filters can handle scale variation well. Compared with MCF, MCPF achieves much better performance with about $7.0\%/2.3\%$ and $6.2\%/2.6\%$ improvement on the OTB-2013 and OTB-2015 datasets. These results show that particle filters can complement multi-task correlation filter and significantly improve tracking performance. Furthermore, both CPF and CF2S perform much better than CF2 [25], and CPF achieves better performance than CF2S. These results show both particle filter and the adaptive multi-scale strategy [9] can improve tracking performance. However, our tracker with a particle filter can deal with scale variation better, which is also demonstrated in Figure 5 for scale variation attribute evaluation.

4.3. Effect of Particle Sampling on Visual Tracking

In this section, we evaluate the effects of particle number and scale on visual tracking performance in terms of effectiveness and efficiency. As shown in Table 2, the proposed MCPF tracker is evaluated with different particle numbers on the OTB-2013 and OTB-2015 datasets, and the AUC and PS corresponding to the OPE are reported for each experiment. Furthermore, the run-timer performance in terms of frame-per-second (FPS) is also provided for analyzing the trade-off between accuracy and efficiency in Table 2. Based on the results, it is clear that increasing the number of particles can improve tracking performance. However, the tracker becomes slower. Note that, the MCPF tracker with 10 particles achieves comparable results to the one with 50 particles. These results show that the multi-task correlation filter can enhance and complement particle filters, and help cover the target state space well with a few number of particles. Even with a fewer number of particles, the proposed MCPF method can achieve comparable performance with much higher efficiency.

Table 2. Effect of particle numbers on visual tracking performance. For different particle numbers, we report frame per second, AUC, and PS. Increasing particle numbers can improve visual tracking performance. However, the tracker becomes slower.

# Particles		10	30	50	100
AUC/PS	OTB-2013	65.1/90.8	65.9/90.4	66.1/89.4	67.7/91.6
	OTB-2015	61.0/86.7	62.7/87.6	62.1/86.7	62.8/87.3
FPS	OTB-2013	1.96	1.29	0.85	0.58
	OTB-2015	1.80	1.27	0.87	0.54

Table 3. Effect of particle scales (s) on visual tracking performance on the AUC and PS metrics corresponding to the OPE.

Scale	0.005	0.01	0.02	0.05
OTB-2013	65.2/90.9	67.7/91.6	66.1/89.4	64.1/89.6
OTB-2015	60.2/86.0	62.8/87.3	62.1/86.7	61.0/86.3

Compared with the SCM method, which is one of the top performing trackers based on particle filters [36], the proposed MCPF method has about $17.8\%/26.7\%$ improvement with AUC and PS metrics. Moreover, the proposed tracker is faster than the SCM (about 0.4 FPS). In Table 3, we show the results of the proposed MCPF with different particle scales s. Here, the variances of affine parameters for particle sampling are set to $(s, 0.0001, 0.0001, s, 2, 2)$. Overall, the proposed MCPF performs robustly within a wide range of scale change.

4.4. OTB-2013 Dataset

We evaluate our MCPF algorithm with 29 trackers in [36] and other 22 state-of-the-art trackers using the source codes including MEEM [39], TGPR [12], KCF [16], RPT [21], MUSTer [18], DSST [9], LCT [26], CF2 [25], SCF [23], HDT [29], Staple [2], SRDCF [10], DeepSRDCF [7], SRDCFdecon [8], CNN-SVM [17], C-COT [11], SINT [32], SiamFC [3], DAT [28], FCNT [34], and SCT [5]. We show the results in OPE using the distance precision and overlap success rate in Figure 4. For presentation clarity, we only show the top 10 trackers. In the figure leg-

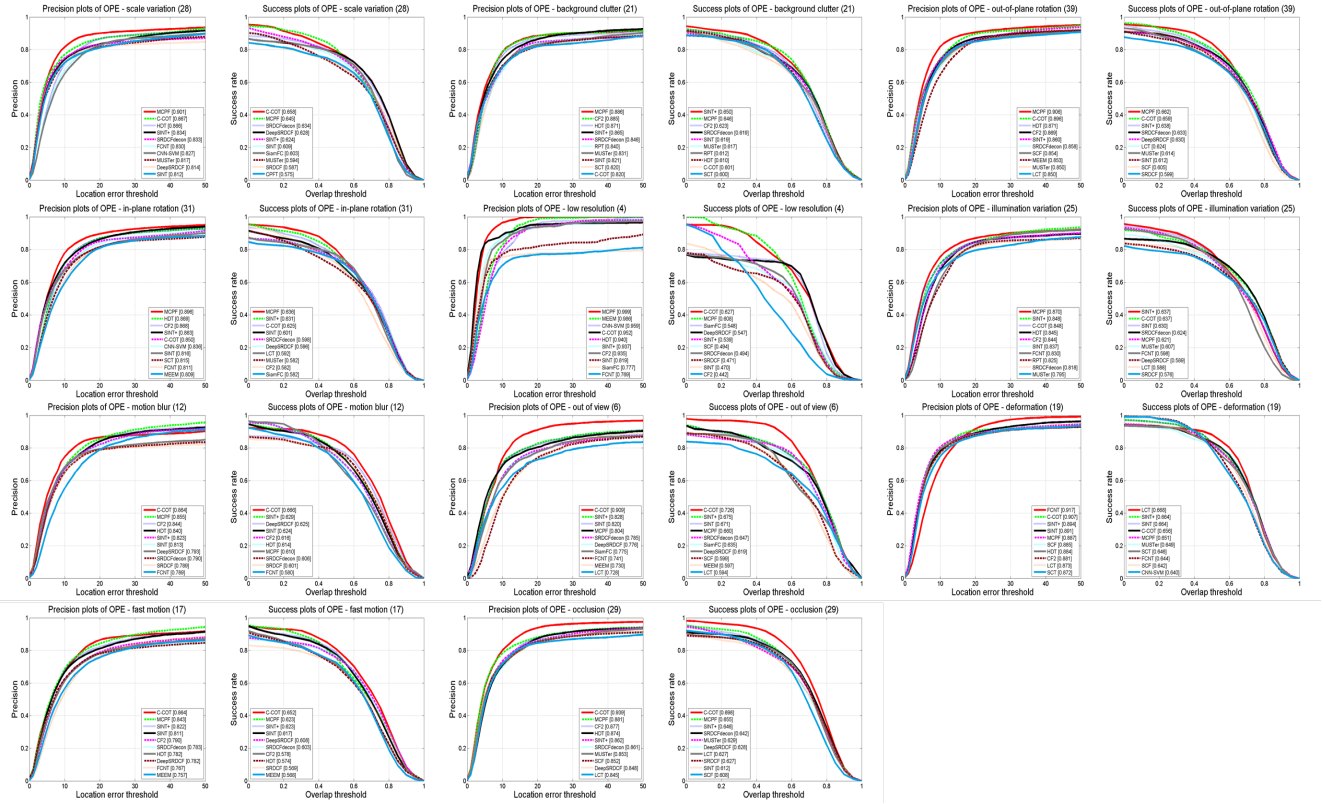


Figure 5. Success and precision plots on 11 tracking challenges of scale variation, out of view, out-of-plane rotation, low resolution, in-plane rotation, illumination, motion blur, background clutter, occlusion, deformation, and fast motion. The legend contains the AUC and PS scores for each tracker. Our MCPF method performs favorably against the state-of-the-art trackers.

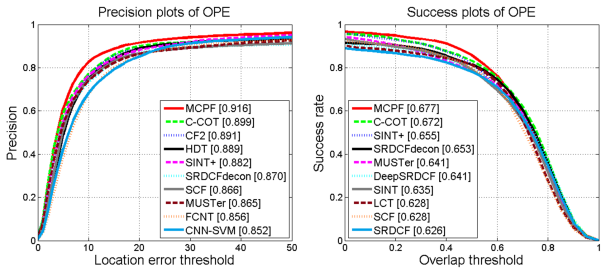


Figure 4. Precision and success plots over all the 50 sequences using one-pass evaluation on the OTB-2013 Dataset. The legend contains the area-under-the-curve score and the average distance precision score at 20 pixels for each tracker. Our MCPF method performs favorably against the state-of-the-art trackers.

end, we report the AUC score and average distance precision score at 20 pixels for each tracker.

Among all the trackers, the proposed MCPF method performs well on the distance precision and overlap success rate. Compared with other correlation filter based trackers, the proposed MCPF algorithm performs favorably against the C-COT method. In Figure 4, we do not show the results by the MDNet [27] method, because it uses many external videos for training. The MDNet method achieves 94.8% and 70.8% on the area-under-the-curve score and the precision at a threshold of 20 pixels, which are comparable to the proposed tracker. Overall, the precision and success plots demonstrate that our approach performs well against the state-of-the-art methods.

In Figure 5, we analyze the tracking performance based on attributes of image sequences [36] in terms of 11 challenging factors, e.g., scale variation, out of view, occlusion, and deformation. These attributes are useful for analyzing the performance of trackers in different aspects. For presentation clarity, we present the top 10 methods in each plot. We note that the proposed tracking method performs well in dealing with challenging factors including scale variation, in-plane rotation, out-of-plane rotation, low resolution, and background clutter. For the sequences with large-scale variations, our MCPF algorithm performs well among all the state-of-the-art trackers (e.g., CF2 and HDT), which demonstrates that the proposed MCPF can handle scale variation by integrating the MCF and a particle filter.

4.5. OTB-2015 Dataset

We carry out experiments on the OTB-2015 dataset with comparisons to 29 trackers in [36] and other 14 state-of-the-

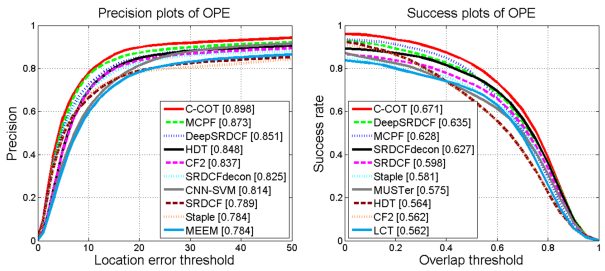


Figure 6. Precision and success plots over all 100 sequences using one-pass evaluation on the OTB-2015 dataset. The legend contains the area-under-the-curve score and the average distance precision score at 20 pixels for each tracker. Our MCPF method performs favorably against the state-of-the-art trackers.

art tracking methods including MEEM [39], TGPR [12], KCF [16], MUSTer [18], DSST [9], LCT [26], CF2 [25], HDT [29], Staple [2], SRDCF [10], DeepSRDCF [7], SRDCFdecon [8], CNN-SVM [17], and C-COT [11]. We show the results in one-pass evaluation using the distance precision and overlap success rate in Figure 6. The proposed MCPF algorithm achieves the AUC score of 62.8% and PS of 87.3%. Compared with the CF2 and HDT methods based on deep features as well as correlation filters, the performance gain is $6.6\%/3.6\%$ and $6.4\%/2.5\%$ in terms of AUC and PS, respectively. Overall, the C-COT method performs well but at a lower speed (0.22 FPS), and the proposed MCPF as well as DeepSRDCF algorithms achieve comparable results.

4.6. Temple Color Dataset

We evaluate the proposed MCPF algorithm on the Temple Color dataset [22] with 16 trackers in [22] and other 9 state-of-the-art tracking methods using their shared source codes, including MUSTer [18], SRDCF [10], CF2 [25], HDT [29], DSST [9], Staple [2], DeepSRDCF [7], SRDCFdecon [8], and C-COT [11]. For fair comparisons, RGB color features are used for all trackers and the same evaluation metrics with the OTB-2013 and OTB-2015 datasets, i.e. AUC and PS, are adopted.

Figure 7 shows that our algorithm performs favorably against the state-of-the-art methods. Among the evaluated trackers, the CF2, HDT, Staple, and SRDCF methods achieve the AUC and PS scores of $(48.4\%, 70.3\%)$, $(48.0\%, 68.6\%)$, $(49.8\%, 66.5\%)$, and $(51.0\%, 69.4\%)$, respectively. Our MCPF algorithm achieves the AUC and PS scores of $(54.5\%, 77.4\%)$. In both precision and success plots, our method obtains performance gain of 6.1% and 7.1% on the AUC and PS scores against the CF2 method. Overall, the proposed MCPF method shows comparable results compared to the C-COT and significantly outperforms other correlation filter based trackers (DSST and KCF).

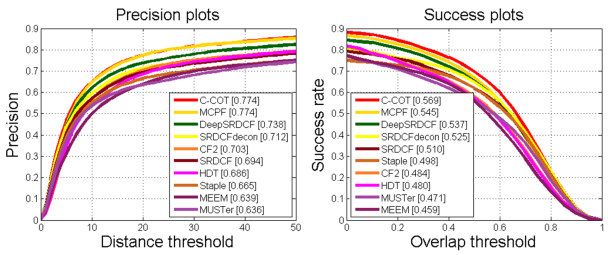


Figure 7. Precision and success plots over the 128 sequences using one-pass evaluation on the Temple Color dataset. The legend contains the area-under-the-curve score and the average distance precision score at 20 pixels for each tracker. Our MCPF method performs favorably against the state-of-the-art trackers.

5. Conclusion

In this paper, we propose a multi-task correlation particle filter for robust visual tracking. The proposed tracking algorithm can effectively handle scale variation via a particle sampling strategy, and exploit interdependencies among different features to learn their correlation filters jointly. Furthermore, it can shepherd the sampled particles toward the modes of the target state distribution to obtain robust tracking performance. Extensive experimental results on benchmark datasets demonstrate the effectiveness and robustness of the proposed algorithm against the state-of-the-art tracking methods.

Acknowledgments

This work is supported by National Natural Science Foundation of China (No.61432019, 61532009, 61572498, 61572296), Beijing Natural Science Foundation (4172062), and US National Science Foundation CAREER grant 1149783.

References

[1] M. S. Arulampalam, S. Maskell, and N. Gordon. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *TSP*, 50:174–188, 2002. 2, 4

[2] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr. Staple: Complementary learners for real-time tracking. In *CVPR*, 2016. 6, 8

[3] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. Torr. Fully-convolutional siamese networks for object tracking. *ECCV Workshop*, 2016. 6

[4] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In *CVPR*, pages 2544–2550, 2010. 1, 3

[5] J. Choi, H. J. Chang, J. Jeong, Y. Demiris, and J. Y. Choi. Visual tracking using attention-modulated disintegration and integration. In *CVPR*, June 2016. 6

[6] C.Yang, R. Duraiswami, and L. Davis. Fast multiple object tracking via a hierarchical particle filter. In *ICCV*, 2005. 3

[7] M. Danelljan, G. Hager, F. Khan, and M. Felsberg. Convolutional features for correlation filter based visual tracking. In *ICCV workshop*, 2015. 6, 8

[8] M. Danelljan, G. Hager, F. Khan, and M. Felsberg. Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking. In *CVPR*, 2016. 6, 8

[9] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *BMVC*, 2014. 1, 2, 3, 5, 6, 8

[10] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Learning spatially regularized correlation filters for visual tracking. In *ICCV*, pages 4310–4318, 2015. 6, 8

[11] M. Danelljan, A. Robinson, F. Khan, and M. Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *ECCV*, 2016. 3, 6, 8

[12] J. Gao, H. Ling, W. Hu, and J. Xing. Transfer learning based visual tracking with gaussian process regression. In *ECCV*, 2014. 6, 8

[13] J. Gao, T. Zhang, X. Yang, and C. Xu. Deep relative tracking. *TIP*, 26(4):1845–1858, 2017. 1

[14] W. Guo, L. Cao, T. X. Han, S. Yan, and C. Xu. Max-Confidence Boosting With Uncertainty for Visual Tracking. *IEEE Trans. Image Processing*, 24(5):1650–1659, 2015. 1

[15] J. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *ECCV*, 2012. 1, 2, 3

[16] J. F. Henriques, R. Caseiro, P. M. 0004, and J. Batista. High-speed tracking with kernelized correlation filters. *TPAMI*, 37(3):583–596, 2015. 1, 3, 5, 6, 8

[17] S. Hong, T. You, S. Kwak, and B. Han. Online tracking by learning discriminative saliency map with convolutional neural network. In *ICML*, 2015. 6, 8

[18] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao. Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking. In *CVPR*, pages 749–758, 2015. 1, 3, 6, 8

[19] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *IJCV*, 29:5–28, 1998. 2, 3

[20] Z. Khan, T. Balch, and F. Dellaert. A rao-blackwellized particle filter for eigentracking. In *CVPR*, 2004. 3

[21] Y. Li, J. Zhu, and S. C. H. Hoi. Reliable patch trackers: Robust visual tracking by exploiting reliable patches. In *CVPR*, pages 353–361, 2015. 1, 3, 6

[22] P. Liang, E. Blasch, and H. Ling. Encoding color information for visual tracking: Algorithms and benchmark. *TIP*, 24(12):5630–5644, 2015. 2, 5, 8

[23] S. Liu, T. Zhang, X. Chao, and C. Xu. Structural correlation filter for robust visual tracking. In *CVPR*, pages 5388–5396, 2016. 3, 6

[24] T. Liu, G. Wang, and Q. Yang. Real-time part-based visual tracking via adaptive correlation filters. In *CVPR*, pages 4902–4912, 2015. 1, 3

[25] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang. Hierarchical convolutional features for visual tracking. In *ICCV*, 2015. 1, 2, 3, 5, 6, 8

[26] C. Ma, X. Yang, C. Zhang, and M.-H. Yang. Long-term correlation tracking. In *CVPR*, pages 5388–5396, 2015. 1, 3, 6, 8

[27] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, June 2016. 7

[28] H. Possegger, T. Mauthner, and H. Bischof. In defense of color-based model-free tracking. In *CVPR*, 2015. 6

[29] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M.-H. Yang. Hedged deep tracking. In *CVPR*, 2016. 1, 2, 3, 5, 6, 8

[30] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 5

[31] A. Smeulders, D. Chu, R. Cucchiara, S. Calderara, A. Deghan, and M. Shah. Visual tracking: an experimental survey. *TPAMI*, 36(7):1442–1468, 2013. 1, 2

[32] R. Tao, E. Gavves, and A. W. M. Smeulders. Siamese instance search for tracking. In *CVPR*, 2016. 6

[33] A. Vedaldi and K. Lenc. Matconvne: convolutional neural networks for matlab. In *CoRR*, page abs/1412.4564, 2014. 5

[34] L. Wang, W. Ouyang, X. Wang, and H. Lu. Visual tracking with fully convolutional networks. In *ICCV*, 2015. 6

[35] B. Wu, S. Lyu, B.-G. Hu, and Q. Ji. Simultaneous clustering and tracklet linking for multi-face tracking in videos. In *ICCV*, pages 2856–2863, 2013. 1

[36] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *CVPR*, 2013. 1, 2, 5, 6, 7

[37] Y. Wu, J. Lim, and M. Yang. Object tracking benchmark. *TPAMI*, 37(9):1834–1848, 2015. 2, 5

[38] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4):13, 2006. 1, 2

[39] J. Zhang, S. Ma, and S. Sclaroff. MEEM: Robust tracking via multiple experts using entropy minimization. In *ECCV*, 2014. 6, 8

[40] T. Zhang, A. Bibi, and B. Ghanem. In defense of sparse tracking: Circulant sparse tracker. In *CVPR*, 2016. 3

[41] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Low-rank sparse learning for robust visual tracking. In *ECCV*, 2012. 5

[42] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Robust visual tracking via multi-task sparse learning. In *CVPR*, 2012. 4

[43] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Robust visual tracking via structured multi-task sparse learning. *International Journal of Computer Vision*, 101(2):367–383, 2013. 4

[44] T. Zhang, B. Ghanem, S. Liu, C. Xu, and N. Ahuja. Robust Visual Tracking via Exclusive Context Modeling. *IEEE transactions on cybernetics*, 46(1):51–63, 2016. 5

[45] T. Zhang, C. Jia, C. Xu, Y. Ma, and N. Ahuja. Partial occlusion handling for visual tracking via robust part matching. In *CVPR*, 2014. 3

[46] T. Zhang, S. Liu, N. Ahuja, M.-H. Yang, and B. Ghanem. Robust Visual Tracking via Consistent Low-Rank Sparse Learning. *International Journal of Computer Vision*, 111(2):171–190, 2015. 1

[47] T. Zhang, S. Liu, C. Xu, S. Yan, B. Ghanem, N. Ahuja, and M.-H. Yang. Structural sparse tracking. In *CVPR*, 2015. 3

[48] S. K. Zhou, R. Chellappa, and B. Moghaddam. Visual tracking and recognition using appearance-adaptive models in particle filters. *TIP*, 11(1):1491–1506, 2004. 3