This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2797062, IEEE Transactions on Pattern Analysis and Machine Intelligence

Learning Multi-task Correlation Particle Filters for Visual Tracking

Tianzhu Zhang, Changsheng Xu, and Ming-Hsuan Yang

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 interdependencies among different object parts and features into account to learn the correlation filters jointly. Next, the proposed MCPF is introduced 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 MCPF enjoys several merits. First, it exploits the interdependencies among different features to derive the correlation filters jointly, and makes the learned filters complement and enhance each other to obtain consistent responses. Second, it handles partial occlusion via a part-based representation, and exploits the intrinsic relationship among local parts via spatial constraints to preserve object structure and learn the correlation filters jointly. Third, it effectively handles large scale variation via a sampling scheme by drawing particles at different scales for target object state estimation. Fourth, it shepherds the sampled particles toward the modes of the target state distribution via the MCF, and effectively covers object states well using fewer particles than conventional particle filters, thereby resulting in robust tracking performance and low computational cost. Extensive experimental results on four challenging benchmark datasets demonstrate that the proposed MCPF tracking algorithm performs favorably against the state-of-the-art methods.

Index Terms—Visual Tracking, Correlation Filter, Structural Modeling, Particle Filter.

1 Introduction

Visual tracking is one of the most important tasks in computer vision that finds numerous applications such as video surveillance, motion analysis, action recognition, and autonomous driving, to name a few. 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 object state estimation algorithms for tracking scenarios with challenging factors such as illumination changes, fast motion, pose variations, partial occlusions, and background clutters.

Correlation filters have recently been introduced into visual tracking and shown to perform well in terms of speed and accuracy [6], [1], [2], [7], [8], [9], [10], [11], [12], [3], [4]. Existing methods based on correlation filters approximate the dense sampling scheme by generating a circulant matrix, of which each row denotes a circular shifts of a base sample. As such, its regression model can be computed in the Fourier domain with only a base sample, which facilitates achieving significant speed improvement in both training and testing stages. Henriques et al. exploit the circulant structure of shifted image patches in a kernel space and propose the CSK method [7] by using intensity features. Subsequently, a method that extends the input features from one to multiple channels (e.g., HOG descriptors) is presented [2]. In addition, numerous methods based on correlation filters based on conventional features, e.g., KCF [2] and DSST [1], are developed to deal with challenging tracking scenarios.

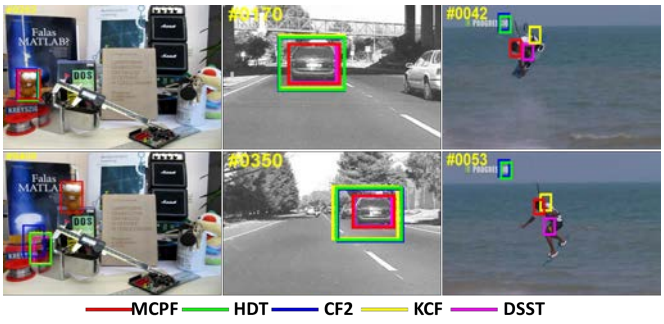


Fig. 1. Comparisons of the proposed MCPF tracking algorithm with the state-of-the-art correlation filter trackers (DSST [1], KCF [2], CF2 [3], and HDT [4]) on the *lemming*, *car4*, and *kitesurf* sequences [5]. These methods perform differently as various features, occlusion handling, and scale adaption strategies are used. The proposed MCPF tracker performs favorably against these trackers.

Recognizing the success of deep convolutional neural networks (CNNs) on a wide range of visual recognition tasks, several CNN based correlation filter trackers, e.g., CF2 [3] and HDT [4], have been developed. Empirical studies on large object tracking benchmark datasets show that these CNN based trackers [3], [4] perform favorably against methods based on hand-crafted features such as intensity [7], HOG descriptors [1], [2], and color attributes [8]. Figure 1 shows some tracking results where the CF2 [3] and HDT [4] algorithms

- *Tianzhu Zhang is with National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, P. R. China. E-mail: tzzhang@nlpr.ia.ac.cn.*
- *Changsheng Xu is with National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, P. R. China. E-mail: csxu@nlpr.ia.ac.cn.*
- *Ming-Hsuan Yang is with School of Engineering, University of California, Merced, CA 95344. E-mail: mhyang@ucmerced.edu.*

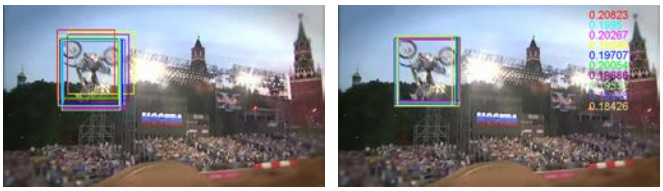
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2797062, IEEE Transactions on Pattern Analysis and Machine Intelligence

perform well against the DSST [1] and KCF [2] methods which achieve the state-of-the-art results [13].

Despite achieving the state-of-the-art performance, existing correlation filter based tracking approaches are limited in several aspects. First, these trackers [3], [4] learn a correlation filter for each feature independently without considering their relationship. In [3], [4], adaptive linear correlation filters are learned over the outputs of each convolutional layer. When inferring the location of targets, the CF2 method [3] uses a coarse-to-fine search strategy on the multi-level correlation response maps, and the HDT scheme [4] combines all correlation filters into a stronger one using an online hedge algorithm. Since features from different layers can enhance and complement each other, CNN based correlation trackers (CF2 [3] and HDT [4]) perform well. Nevertheless, these methods assume that correlation filters of different features are independent. As demonstrated empirically in this work, ignoring the relationship among correlation filters makes the tracker prone to drift away from target objects in cases of significant appearance change.

Second, existing correlation filter based trackers [1], [2], [3], [4] do not deal with partial occlusion well. Figure 1 shows tracking results on the *lemming* sequence of two correlation filter based trackers, DSST [1] and KCF [2], which fail to track the target object when partial occlusion occurs. To deal with the above issues, correlation filters based part-based representations have been developed [12], [11] . In [11], [12], object parts are independently tracked by the KCF tracker [2], and the spatial constraints among parts are not considered. As a result, object parts are tracked independently which eventually leads the tracker to drift away. In most sequences, the appearance change between two consecutive frames is small [10], and most parts should move in similar directions to preserve object structure. Thus, enforcing constraints among object parts in such tracking approaches is likely to achieve more robust performance.

Third, tracking methods based on correlation filters [2], [3], [4] do not handle scale variation well. Danelljan et al. propose the DSST tracker [1] with adaptive multi-scale correlation filters and HOG features to handle scale variation of target objects. However, the adaptive multi-scale strategy does not facilitate the tracking methods based on CNN features and correlation filters [3], [4] well when objects undergoing large scale variation (see Section 4). In this work, we resort to particle filters [14], [15] 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 adequately, but the proposed algorithm performs well by using a 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 environments at the expense of heavy computational loads. On the other hand, particle filter based trackers determine each object state based on each drawn sample. If the sampled particles do not cover object states well as shown in Figure 2(a), the predicted target state may be not correct. Thus, it is critical to shepherd the sampled particles toward the modes of a target state distribution.



(a) Sampled particles (b) Particles after shepherding

Fig. 2. A multi-task correlation filter is used to shepherd the sampled particles toward the modes of the target state distribution. The numbers in (b) are the scores of the 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) to address the above-discussed issues for robust visual tracking. The proposed MCPF learns multiple correlation filters jointly via a Multi-task Correlation Filter (MCF) by exploiting not only interdependencies among different features, but also spatial constraints among parts to preserve object structure. Here, learning the correlation filter of each part with one type of feature is viewed as one task. The MCF learns all correlation filters within a multi-task framework. The proposed MCPF algorithm is designed to exploit and complement the strength of a MCF and a particle filter, which enjoys the merits of multiple features, part-based representations, robustness to scale variation as well as occlusion, and computational efficiency.

Based on the proposed appearance model, we design the MCPF tracking algorithm. Each target object is represented by a set of parts with multiple features, and each part with one type of feature is associated a multi-task correlation filter. We learn the correlation filters for all parts among multiple features jointly. For each sampled particle, the MCPF is used such that particles are shepherded to the local modes of a target state distribution as illustrated in Figure 2(b). The image region corresponding to each particle is divided into several parts, and each part with one type of feature is used as a base sample to construct a block-circulant circulant matrix, of which each block denotes a shifted sample [7]. The MCPF measures the similarity by computing the inner product for each shifted sample related to the learned filter. Finally, we obtain the response map of all parts based on the correlation filters among multiple features, and each particle can be shepherded to the location with the maximal value of the response map. During the tracking process, target object state is estimated as a weighted average of all shepherded particles. Here, the weights are based on the outputs of the proposed MCF. With the use of circulant matrix, each particle can densely cover an image region where the target object may appear, and we do not need to draw particles *densely* to cover object possible states. Consequently, we can cover object possible states well using fewer particles than conventional approaches. We evaluate the proposed tracking algorithm on four standard benchmark

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2797062, IEEE Transactions on Pattern Analysis and Machine Intelligence

datasets [5], [16], [17], [18]. Extensive experimental results show that the proposed MCPF tracking algorithm performs favorably against the state-of-the-art methods in terms of accuracy and robustness.

The contributions of this work are summarized as follows. First, different from existing methods that learn correlation filters for different features independently, the proposed MCPF model considers interdependencies among different features to learn their correlation filters jointly for robust visual tracking. Second, the proposed MCPF appearance model exploits spatial layout structure among object parts, which is not considered by the existing tracking methods based on correlation filters [6], [1], [2], [19], [7], [8], [9], [10], [11], [12]. As such, the proposed model not only exploits the intrinsic relationship among object parts to learn the correlation filters jointly, but also preserves the spatial layout structure among object parts. Third, the MCPF model provides 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 variation can be covered. Therefore, the MCPF algorithm effectively handles scale variation problem. Fourth, the MCPF model shepherds the sampled particles toward the modes of a target state distribution and cover the state space well using fewer particles than conventional approaches, thereby resulting in robust tracking performance at a low computational cost.

2 Related Work

Visual tracking has been studied extensively over the past decades. A comprehensive review of the tracking methods is beyond the scope of the paper, and surveys of this field can be found in [20], [21], [5], [22], [23], [24]. In this section, we discuss the methods closely related to this work in terms of generative and discriminative trackers, and tracking algorithms based on correlation as well as particle filters.

Generative Trackers. Tracking algorithms can be broadly categorized as either generative or discriminative methods. Generative trackers typically formulate the tracking problem as searching for the image regions which are most similar to the target objects [25], [26], [27], [28], [29]. In [25], Black et al. use a subspace model learned off-line to represent target objects holistically for tracking. To account for large appearance change for visual tracking, the IVT method [27] is proposed to learn an incremental subspace model. Instead of using holistic representations, the FragTrack method [30] models object appearance with histograms of local parts. In [26], the mean shift tracking algorithm models a target with nonparametric distributions of color features and locates the object with mode shifts. Kwon et al. [28] decompose the observation model into multiple basic observation models to cover a wide range of pose and illumination variation. On the other hand, Cehovin et al. [31] represent global and local appearance of target objects based on parts for visual tracking.

Discriminative Trackers. Discriminative approaches cast tracking as a classification problem that distinguishes tracked targets from backgrounds [32], [33], [34], [35], [36], [37]. Avidan [32] proposes an ensemble tracking method by combining a set of weak classifiers. In [33], Grabner et al. propose an online boosting tracking method to update discriminative features to account for large appearance change. Babenko et al. [34] introduce multiple instance learning into online tracking where samples are considered in positive and negative bags for dealing with ambiguities of object locations. In [36], Hare et al. use an online structured output support vector machine for adaptive visual tracking such that the dependence of predicted locations is considered. The TLD tracker [35] explicitly decomposes the tracking task into tracking, learning, and detection where both tracker and detector are used to achieve robust performance. Recently, Zhang et al. [37] utilize multiple experts and entropy minimization to address the model drift problem of online visual tracking.

Correlation Filters. Correlation filters have recently attracted considerable attention in visual tracking due to computational efficiency and robustness. Bolme et al. represent target objects by learning adaptive correlation filters [6] for visual tracking. Henriques et al. exploit the circulant structure of shifted image patches in a kernel space and propose the CSK method based on intensity features [7], and extend it to the KCF approach [2] with the HOG descriptors. For robust tracking, Danelljan et al. use adaptive color attributes [8] by mapping multi-channel features into a Gaussian kernel space, and develop adaptive multi-scale correlation filters to handle scale variation [1]. In [19], Zhang et al. incorporate context information into a correlation filter and model the scale change based on consecutive correlation responses. Hong et al. [9] propose a biology-inspired framework (MUSTer) where short-term and long-term tracking processes are used to cooperate with each other. In [10], Ma et al. introduce an online random fern classifier for long-term tracking. Most 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 [38].

Correlation filters based on local representations using patches or parts have also been developed [11], [39]. In [11], object parts are independently tracked by the KCF tracker [2] and the object location is determined based on a confidence map of all the tracking results, in a way similar to the FragTrack [30] approach. Liu et al. [39] develop a part-based structural correlation filter to preserve object structure. In [12], Li et al. present a method to measure how reliably a patch can be tracked and exploit the corresponding trajectories for visual tracking via the Hough voting scheme [40] on a confidence map. Several tracking methods based on correlation filters and multiple deep features have recently been proposed [3], [4], In [41], a multi-kernel correlation filter is designed to combine complementary features for visual tracking. Different from existing tracking methods based on correlation filters, we propose a multi-task formulation to exploit the interdependencies among different features and spatial constraints among parts to preserve object structure to learn correlation filters jointly.

Particle Filters. In visual tracking, particle filters or Sequential Monte Carlo (SMC) methods [15] have been widely used to estimate object states. For robust performance, the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2797062, IEEE Transactions on Pattern Analysis and Machine Intelligence

number of drawn samples must be sufficient to cover the possible states. However, methods using dense sampling of particles generally entail high computational load for visual tracking as each one needs to be evaluated. Consequently, numerous particle filters have been presented to draw samples efficiently and effectively [15], [42], [43], [44], [45]. Importance sampling [15] is introduced to obtain an effective proposal function by combining predictions based on the previous configuration with additional knowledge from auxiliary measurements. In [43], subspace representations are used with the Rao-Blackwell particle filter for visual tracking. On the other hand, the number of particles can be adjusted according to an adaptive noise component [44]. In [42], the observation likelihood is computed in a coarse-to-fine manner, which allows efficient focus on more promising particles. In [46], multiple motion models are combined to improve the dynamics model of a particle filter for tracking of interest points. Different from the above methods, we develop 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 load.

3 Proposed Algorithm

In this section, we present the multi-task correlation particle filter for visual tracking. Different from existing methods [2], [7] 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 the proposed sampling scheme.

3.1 Multi-task Correlation Filter

The key idea of tracking methods based on correlation filters [1], [2], [3], [4] is to use numerous negative samples for enhancing the discriminability of a tracking-by-detection scheme and exploit the circulant matrix of shift samples for computational efficiency. In visual tracking, object appearance is modeled by 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 label $\mathbf{y}_{m,n}$ regressed by a Gaussian function. Given a target object, we sample P parts described by K different features (e.g., intensity, HOG, color, or CNN features) and have $\mathbf{X}_{pk} = [\mathbf{x}^{pk}_{0,0}, \ldots, \mathbf{x}^{pk}_{m,n}, \ldots \mathbf{x}^{pk}_{M-1,N-1}]^\top$ that contains all training samples of the p-th part described by the k-th type of feature ($p = 1,\ldots,P$, $k = 1,\ldots,K$). The goal is to find the optimal weights $\mathbf{w}_{pk}$ for P parts and K different features,

$$\arg\min_{\{\mathbf{w}_{pk}\}} \sum_{p,k} \|\mathbf{X}_{pk}\mathbf{w}_{pk} - \mathbf{y}\|_F^2 + \lambda \|\mathbf{w}_{pk}\|_F^2, \tag{1}$$

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 least-squares minimization problem (1) can equivalently be expressed by its dual form (see [47] for details),

$$\min_{\{\mathbf{z}_{pk}\}} \sum_{p,k} \frac{1}{4\lambda} \mathbf{z}_{pk}{}^\top \mathbf{G}_{pk} \mathbf{z}_{pk} + \frac{1}{4} \mathbf{z}_{pk}{}^\top \mathbf{z}_{pk} - \mathbf{z}_{pk}{}^\top \mathbf{y}. \tag{2}$$

Here, the vector $\mathbf{z}_{pk}$ contains $M \times N$ dual optimization variables $\mathbf{z}^{m,n}_{pk}$, and $\mathbf{G}_{pk} = \mathbf{X}_{pk}\mathbf{X}_{pk}^\top$. These two solutions are related by $\mathbf{w}_{pk} = \frac{\mathbf{X}_{pk}{}^\top \mathbf{z}_{pk}}{2\lambda}$. The learned $\mathbf{z}^{m,n}_{pk}$ selects discriminative training samples $\mathbf{x}^{pk}_{m,n}$ to distinguish the target object from the background. For the p-th part, the corresponding filters of the K different features form a matrix $\mathbf{Z}_p = [\mathbf{z}_{p1}, \cdots, \mathbf{z}_{pk}, \cdots, \mathbf{z}_{pK}] \in \mathbb{R}^{MN \times K}$. Putting the filters of all the K parts together, we obtain $\mathbf{Z} = [\mathbf{Z}_1, \cdots, \mathbf{Z}_p, \cdots, \mathbf{Z}_P] \in \mathbb{R}^{MN \times PK}$. Similarly, for the k-th type of feature, the learned filters of all parts form a matrix $\mathbf{Z}^k = [\mathbf{z}_{1k}, \cdots, \mathbf{z}_{pk}, \cdots, \mathbf{z}_{Pk}] \in \mathbb{R}^{MN \times P}$.

Note that the main idea of (2) is to select discriminative training samples $\mathbf{x}^{pk}_{m,n}$ via $\mathbf{z}^{m,n}_{pk}$ to distinguish the target object from the background. The training samples $\mathbf{x}^{pk}_{m,n}$, $(m,n) \in \{0,1,\ldots,M-1\} \times \{0,1,\ldots,N-1\}$ are the all possible circular shifts, which represent the possible motion of the target object. Therefore, selecting training samples $\mathbf{x}^{pk}_{m,n}$ via $\mathbf{z}^{m,n}_{pk}$ can predict the motion of target object.

Using $\mathbf{Z}_p$, $\mathbf{Z}^k$, and $\mathbf{Z}$ for visual tracking, we have the following observations. First, $\mathbf{Z}_p$ is the learned filter of the p-th part among K different features. Different features should have similar $\mathbf{z}_{pk}$ such that they have consistent localization of the target object, and their correlation filters should be learned jointly to distinguish the target from the background. Second, $\mathbf{Z}^k$ is the learned filters of all parts with the k-th type of feature. As most parts of a target object move similarly between two consecutive frames, they should have similar circular shifts such that all parts have similar motion to preserve target object structure. Third, for the learned filters, only a few possible locations $\mathbf{x}^{pk}_{m,n}$ should be selected to localize the target object in the next frame. Ideally, only one possible location corresponds to the target object.

Based on the above observations, it is clear that most parts will have similar $\mathbf{z}_{pk}$ to make them move similarly to preserve target object structure. In this work, we use the convex $\ell_{\hat{p},\hat{q}}$ 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

$$\min_{\{\mathbf{z}_{pk}\}} \sum_{p,k} \frac{1}{4} \mathbf{z}_{pk}^\top \mathbf{G}_{pk} \mathbf{z}_{pk} + \frac{1}{4} \lambda \mathbf{z}_{pk}^\top \mathbf{z}_{pk} - \lambda \mathbf{z}_{pk}^\top \mathbf{y} + \gamma \|\mathbf{Z}\|_{2,1}, \tag{3}$$

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

To solve (3), we adopt the accelerated proximal gradient (APG) method, which has been extensively used to efficiently solve convex optimization problems with non-smooth terms [48]. While it is time-consuming to compute $\mathbf{G}_k$ directly, the solution can be obtained efficiently in the Fourier domain by considering the circulant structure property of $\mathbf{G}_{pk}$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2797062, IEEE Transactions on Pattern Analysis and Machine Intelligence

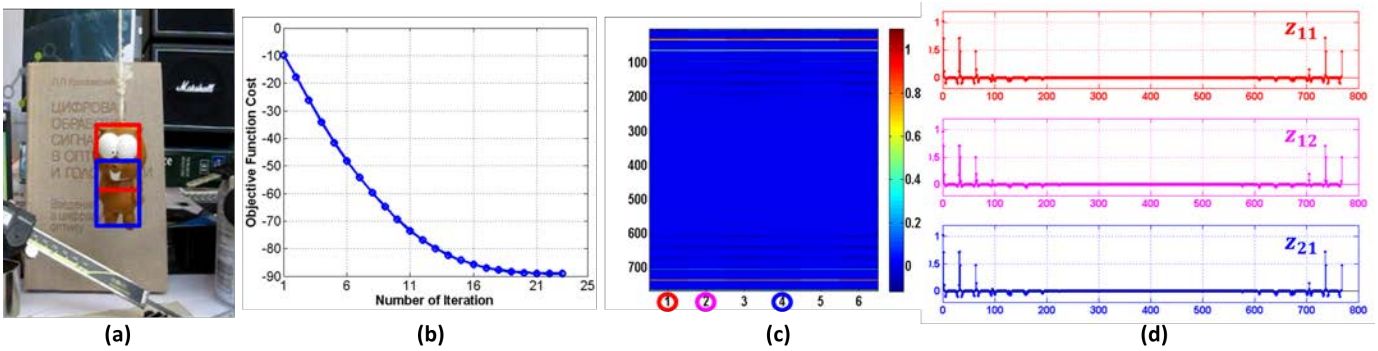


Fig. 3. An illustrative example of the learned multi-task correlation filter via the APG method. Here, $P = 2$ and $K = 3$. (a) A target object is represented by two parts denoted in red and blue bounding boxes. (b) The objective function value vs number of iteration. The problem (3) can be solved efficiently via the APG method and converge quickly in less than 25 iterations. (c) The learned correlation filters $\mathbf{Z} \in \mathbb{R}^{768\times 6}$ where $M = 128$, $N = 96$, and $P \times K = 6$. Notice that the columns of $\mathbf{Z}$ are similar and jointly sparse. It is clear that the learned correlation filters for all parts among multiple features can make similar circular shifts and have the consistent motion to preserve target object structure. (d) Three examples of the learned correlation filters including $\mathbf{z}_{11}$, $\mathbf{z}_{12}$, and $\mathbf{z}_{21}$, which are quite similar to each other.

More details can be found in the supplementary material. After solving this optimization problem, we obtain the multi-task correlation filter $\mathbf{z}_{pk}$ for the p-th part with the k-th feature. To illustrate the proposed formulation clearly, we show an example of the learned correlation filters in Figure 3. In this example, the solution for the optimization problem (3) can be converged in less than 25 iterations by using the APG method as shown in Figure 3(b). In addition, the learned correlation filters for all parts among different features can make similar circular shifts and have the consistent motion to preserve the target object structure as shown in Figure 3(c) and (d).

3.2 Multi-task Correlation Particle Filter

The proposed multi-task correlation particle filter is based on a Bayesian sequential importance sampling algorithm which recursively approximates the posterior distribution using a finite set of weighted samples to estimate 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 function, 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. In this work, we use $p(\mathbf{s}_t^i|\mathbf{s}_{t-1}^i)$ for $q(\cdot)$ and have $w_t^i \propto w_{t-1}^i p(\mathbf{y}_t|\mathbf{s}_t^i)$. Then, a re-sampling algorithm is applied to avoid the degeneracy problem [14]. In this case, the weights are set to $w_{t-1}^i = 1/n \; \forall i$. Therefore, we rewrite the importance weights 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 draws the particles based on the weights of the previous step, and all the new particles are updated by the likelihood function in the next frame.

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

$$\mathbf{r} = \sum_{p,k} \mathcal{F}^{-1}(\mathcal{F}(\mathbf{z}_{pk}) \odot \mathcal{F}(\langle \mathbf{y}_t^i, \bar{\mathbf{x}}_{pk} \rangle)). \quad (9)$$

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 with 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 $\mathbf{s}_t^i \rightarrow$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2797062, IEEE Transactions on Pattern Analysis and Machine Intelligence

$\mathcal{S}_{mcf}(\mathbf{s}_t^i)$. 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

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

In the conventional particle filter, the tracking result is estimated by using the particle with maximum weight or the weighted average of all particles. In this work, we use the weighted average which is more stable as shown in the SCM [49] and ℓ_1 [29] methods. At time t, the object state is estimated by

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

In the proposed multi-task correlation particle filter, we assume an affine motion model between consecutive frames. Therefore, the state variable $\mathbf{s}_t$ consists of the six parameters (2D linear transformation and 2D translation). As the particles are drawn at multiple scales for the 2D linear transformation, the proposed model can handle scale variation. Particles are sampled around the previous object state to predict the state $\mathbf{s}_t$ of the target at time t, from which we crop the corresponding region $\mathbf{y}_t$ in the current image and normalize it to the same size. The state transition function $p(\mathbf{s}_t|\mathbf{s}_{t-1})$ is modeled by an affine motion model with a diagonal Gaussian distribution. The observation model $p(\mathbf{y}_t|\mathbf{s}_t)$ reflects the similarity between an observed image region $\mathbf{y}_t$ corresponding to the state $\mathbf{s}_t$.

3.3 MCPF Tracker

Based on the multi-task correlation particle filter, we propose the MCPF tracker. As summarized in Algorithm 1, the proposed method involves five main steps. First, the proposed method draws particles using the transition model $p(\mathbf{s}_t|\mathbf{s}_{t-1})$ and re-samples them. Second, the proposed multi-task correlation filter is applied to each particle to shepherd it toward the mode of the target state distribution. Third, the weights are updated using the responses of the multi-task correlation filter. Finally, the weighted average of particles is computed using (11). To update the multi-task correlation filter for visual tracking, we adopt an incremental strategy similar to that in [1], [2], [3], [4], which only uses new samples $\mathbf{x}_{pk}$ in the current frame to update a model by

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

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 shown in Figure 4.

First, tracking methods based on conventional particle filters need to draw samples densely to cover the possible states and thus entail high computational load. The MCF can refine particles to cover target states and effectively reduce the number of particles required for accurate tracking. As shown in Figure 4(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 state distribution (where the target object is) as shown in Figure 4(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 ef-

Algorithm 1: Multi-task correlation particle filter tracking algorithm.

Input : Image sequence and initialization.
Output: Tracking results $\mathbf{s}_t$ $\forall t$.

for *each frame* **do**
- Generate particles using the transition model $p(\mathbf{s}_t|\mathbf{s}_{t-1})$ and re-sample them.
- Shift particles with the proposed multi-task correlation filter $\mathbf{s}_t^i \rightarrow \mathcal{S}_{mcf}(\mathbf{s}_t^i)$.
- Update particle importance weights using (10).
- Predict target object state using (11).
- Update tracking models using (12).

end

(a) (b)

Fig. 4. 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 a 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.

fectively help the MCF handle scale variation, as demonstrated in the attribute-based experiments with large scale variation as shown in Figure 6.

4 Experimental Results

In this section, we first present the experimental setups, and then present extensive evaluations of the proposed algorithm against the state-of-the-art trackers on benchmark datasets. In addition, we analyze different components of the proposed algorithm including particle sampling and MCF. The tracking results are available at http://nlpr-web.ia.ac.cn/mmc/homepage/tzzhang/lmcpf.html, and the source code of this work will be made available to the public.

4.1 Experimental Setups

Features and Parts. We use the same experimental protocols as the CF2 method [3] for fair comparisons in which the VGG-Net-19 [50] is adopted for feature extraction. We first remove the fully-connected layers and use the outputs of the *conv3-4*, *conv4-4* and *conv5-4* layers as our features (i.e., K is 3 in this work). Note that, a variety of features can be adopted such as HOG or other layers of CNN features as in the HDT scheme [4]. We use the spatial layout as shown in Figure 3(a) to represent P object parts based on the ratio of height and width. If the ratio of height and width for a target object is greater than 1, we use $2/3$ of the height from the top and bottom as well as in the center to obtain the parts. Similarly, we can represent objects if the ratio of height and width is less than 1 (i.e., $2/3$ of the width from left, right and center). In this work, the P is empirically set to 3. We note that this simple representation performs well in practice, and other part-based methods can also be adopted. More details can be found in the supplementary material.

Implementation Details. We set the regularization parameters λ and γ of (3) to 1 and 10^{-2}, respectively. In addition, we use a kernel width of 0.1 for generating the Gaussian function labels. Their 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 [2]. We implement our tracker in MATLAB on an Intel 3.10 GHz CPU with 256 GB RAM, and use the MatConvNet toolbox [51], 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. All the parameter settings are available in the source code to be released for accessible reproducible research. As in [29], 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. The average run time of the MCPF algorithm is about 0.5 frames per second on the machine with the above-mentioned settings. Note that the deep feature extraction part amounts to more than 80% of the run time.

Datasets. The proposed algorithm is evaluated on four benchmark datasets: OTB2013 [5], OTB2015 [16], Temple Color [17], and VOT2015 [18]. The first two datasets are composed of 50 and 100 sequences, respectively. The sequences

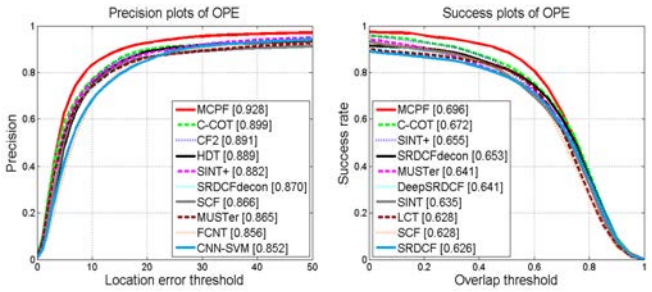


Fig. 5. Precision and success plots over all the 50 sequences using one-pass evaluation on the OTB2013 dataset. The legend contains the area-under-the-curve score and the average distance precision score at 20 pixels for each tracker. The MCPF method performs favorably against the state-of-the-art trackers.

in OTB2013 and OTB2015 datasets are annotated with ground truth bounding boxes and tracking attributes. The Temple Color database [17] contains 128 videos and the VOT2015 dataset [18] consists of 60 challenging image sequences.

Evaluation Metrics. We evaluate the proposed algorithm against the state-of-the-art tracking methods using evaluation metrics and code provided by the respective benchmark dataset. For the OTB2013, OTB2015, and Temple Color datasets, we use the one-pass evaluation (OPE) protocol with 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 of each figure, we report the area under curve (AUC) of success plot and precision score (PS) at 20 pixels threshold corresponding to the one-pass evaluation for each tracking method. For the VOT2015 dataset, the performance is measured both in terms of accuracy (overlap with the ground-truth) and robustness (failure rate).

4.2 OTB2013 Dataset

We evaluate the MCPF algorithm with 29 trackers in [5] and 22 state-of-the-art methods using the source codes including MEEM [37], TGPR [52], KCF [2], RPT [12], MUSTer [9], DSST [1], LCT [10], CF2 [3], SCF [39], HDT [4], Staple [53], SRDCF [54], DeepSRDCF [55], CNN-SVM [56], SRDCFdecon [57], C-COT [38], SINT+ [58], SiamFC [59], DAT [60], FCNT [61], and SCT [62].

Figure 5 shows the one-pass evaluation results using the distance precision and overlap success rate. For presentation clarity, we only show the top 10 trackers. In the figure legend, we report the AUC score and average distance precision score at 20 pixels for each tracker. Among the trackers in the literature, the C-COT method achieves the best results with the distance precision of 89.9% and overlap success rate of 67.2%. The proposed MCPF method performs well with the distance precision of 92.8% and overlap success rate of 69.6%. Compared with other existing trackers, the MCPF tracker achieves significant improvement. The Struck

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2797062, IEEE Transactions on Pattern Analysis and Machine Intelligence

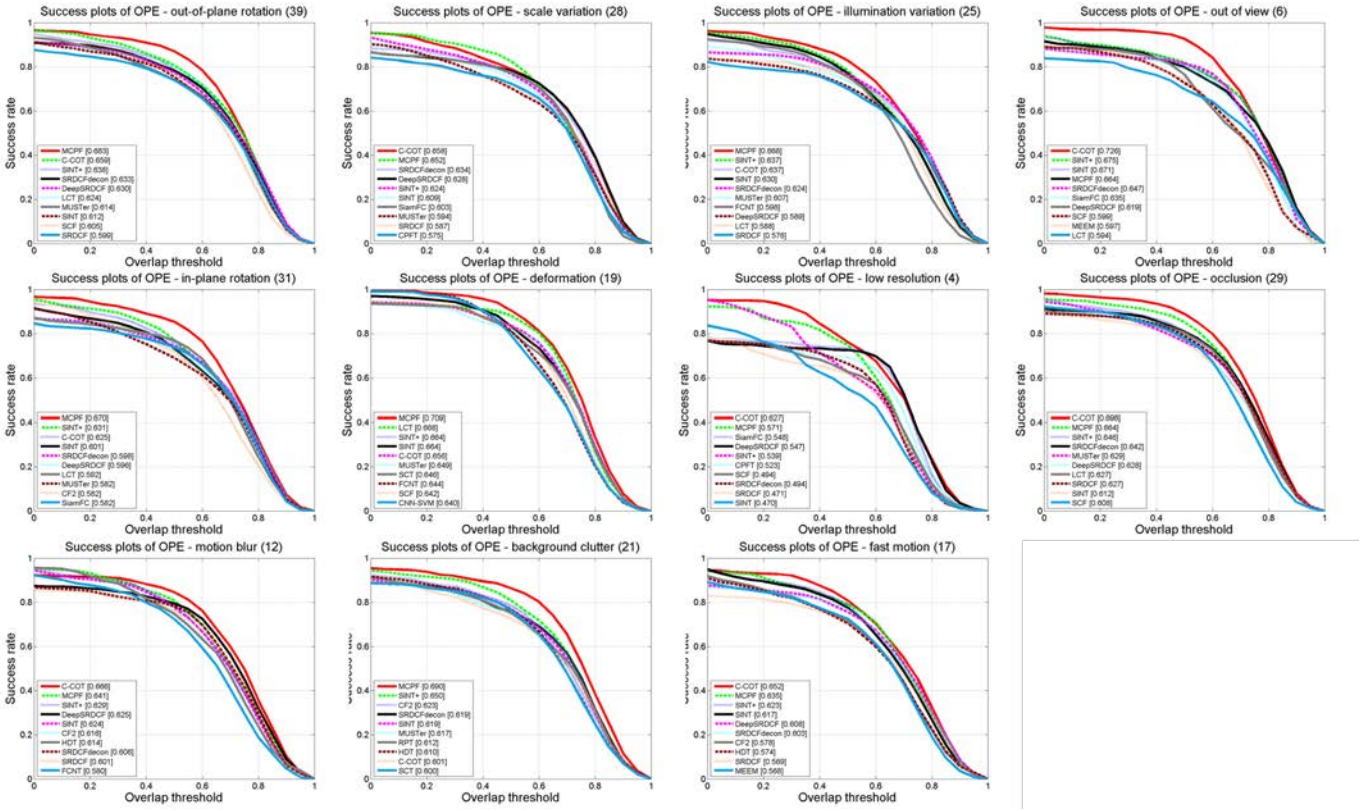


Fig. 6. Tracking performance based on attributes of image sequences on the OTB2013 dataset. Success plots on 11 tracking challenges of scale variation, out of view, out-of-plane rotation, low resolution, in-plane rotation, illumination, motion blur, background clutter, occlusion, deformation, and fast motion. The legend contains the AUC scores for each tracker. Our MCPF method performs favorably against the state-of-the-art trackers.

and SCM methods are top 2 methods in the 29 trackers [5] with $47.4\%/65.6\%$ and $49.9\%/64.9\%$ in terms of the AUC/PS metrics, respectively. The MCPF algorithm performs favorably against these two trackers. Among the other 22 state-of-the-art trackers, the proposed MCPF method performs well against the SINT+ (by 4.1%), SRDCFdecon (by 4.3%), and MUSTer (by 5.5%) methods in terms of the AUC metric. Compared with other correlation filter based trackers, the MCPF algorithm outperforms the CF2 (by $3.7\%/9.1\%$) and HDT (by $3.9\%/9.3\%$) methods in terms of the PS/AUC metrics, respectively. Furthermore, the MCPF algorithm performs well against the KCF (by 18.5%) and DSST (by 18.5%) methods in terms of the PS metric, and achieves performance gain of 17.9% and 13.7% in terms of the AUC metric. In Figure 5, the results by the MDNet [63] and SANet [64] algorithms are not included because these methods use numerous external videos for training. The MDNet and SANet methods achieve $94.8\%/70.8\%$ and $95.0\%/68.6\%$ on the AUC and PS, which are similar to the results achieved by the MCPF algorithm. Furthermore, the proposed tracker achieves comparable results as the ECO [65] tracker ($93.0\%/70.9\%$). Overall, the MCPF algorithm performs well against the state-of-the-art tracking methods based on both metrics.

In Figure 6 we analyze the tracking performance based on sequence attributes [5] including 11 challenging factors in the tracking problem, 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, illumination variation, deformation, background clutter, fast motion, and occlusion. For the sequences with large scale variation, the MCPF algorithm performs well among all the state-of-the-art trackers. Compared with the CF2 and HDT methods, the proposed MCPF algorithm achieves better performance by $12.1\%/3.2\%$ and $12.9\%/4.6\%$ in terms of AUC/PS, respectively. These results show that the proposed MCPF tracker can handle scale variations well. For the sequences with occlusion, the proposed MCPF algorithm achieves better performance than the CF2 and HDT trackers by $1.2\%/5.8\%$ and $1.5\%/6.1\%$ in terms of the PS/AUC, respectively. Overall, these results demonstrate that the proposed MCPF method can improve correlation filter based trackers in handling scale variation via the particle sampling strategy and partial occlusion with the part-based representation scheme.

Figure 7 shows the evaluation results of the MCPF algorithm and 9 state-of-the-art trackers (SCM [49], TGPR [52], KCF [2], CF2 [3], HDT [4], MUSTer [9], DSST [1], SINT+ [58], and C-COT [38]). Here, due to space constraints, we present the main results on 16 challenging sequences

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2797062, IEEE Transactions on Pattern Analysis and Machine Intelligence

Fig. 7. Tracking results of the 10 state-of-the-art trackers (denoted in different colors and lines) on 16 challenging sequences from the OTB2013 dataset (from left to right and top to down are car4, soccer, deer, skating1, shaking, singer1, singer2, couple, jogging-1, walking2, jumping, skiing, lemming, motorRolling, basketball, and tiger1).

from the OTB2013 dataset and more findings on the above-mentioned project website. Although these trackers perform well in general, there are several issues. The SCM tracker is less effective in handling images with occlusion (jogging-1, lemming), fast motion (jumping, skiing, couple), illumination variation (singer2), and rotation (motorRolling) attributes. The TGPR tracker does not perform well in sequences with large illumination variation (skating1), fast motion (couple, skiing, tiger1), and scale variation (car4, singer1) attributes. The correlation filter based trackers, KCF and DSST, drift when target objects undergo occlusion (jogging-1), scale variation (car4, singer1), and fast motion (couple, jumping, and skiing). The CF2 and HDT trackers perform well on most of the sequences because of using deep features. However, the two trackers cannot deal with scale variation well (car4, singer1, and walking2). On the other hand, the SINT+ approach does not perform well in sequences with illumination variation (skating1, singer1), background clutter (basketball), and rotation (motorRolling) attributes. The C-COT performs well on almost all the sequences, but occasionally does not estimate scale well (skating1, shaking, singer2, and motorRolling). We note that the MUSTer tacker drifts off a target object when fast motion occurs (tiger1, motorRolling, couple). Overall, the proposed MCPF tracker is able to track target objects well in most sequences.

4.3 OTB2015 Dataset

We carry out experiments on the OTB2015 dataset with comparisons to 29 trackers in [5] and other 14 state-of-the-art tracking methods including MEEM [37], TGPR [52], KCF [2], MUSTer [9], DSST [1], LCT [10], CF2 [3], HDT [4], Staple [53], SRDCF [54], DeepSRDCF [55], SRDCFdecon [57], CNN-SVM [56], and C-COT [38].

Figure 8 shows the results in one-pass evaluation using the distance precision and overlap success rate. The MCPF tracker achieves the AUC score of 64.3% and PS of 88.7%. Compared with the CF2 and HDT methods based on deep

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2797062, IEEE Transactions on Pattern Analysis and Machine Intelligence

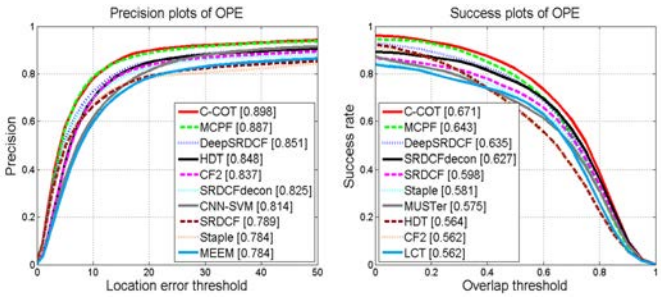


Fig. 8. Precision and success plots over all the 100 sequences using one-pass evaluation on the OTB2015 dataset. The legend contains the area-under-the-curve score and the average distance precision score at 20 pixels for each tracker. Our MCPF method performs favorably against the state-of-the-art trackers.

features as well as correlation filters, the performance gain is $8.1\%/5.0\%$ and $7.9\%/3.9\%$ in terms of AUC and PS, respectively. Furthermore, our tracker performs well against the DeepSRDCF (by 3.6%), SRDCFdecon (by 6.2%), and CNN-SVM (by 7.3%) methods in terms of PS, and achieves favorable performance than the DeepSRDCF (by 0.8%), SRDCFdecon (by 1.6%), and SRDCF (by 4.5%) schemes in terms of AUC. The MDNet [63], SANet [64], and ECO [65] methods perform slightly better with $90.9\%/67.8\%$, $92.8\%/69.2\%$, and $91.0\%/69.1\%$ in terms of PS/AUC, respectively. Overall, the C-COT method performs well but at a lower speed (0.22 FPS), and our MCPF algorithm achieves comparable results.

Figure 9 shows the tracking performance based on attributes of image sequences on the OTB2015 dataset. We show the top 10 methods in each plot for presentation clarity. We note that the proposed tracking method performs well in dealing with most challenging factors, such as out-of-plane rotation, occlusion, illumination variation, and in-plane rotation. The proposed MCPF algorithm performs well against the CF2 (by $10.0\%/5.4\%$) and HDT (by $9.9\%/4.5\%$) methods in terms of AUC/PS for sequences with large scale variation, and outperforms the CF2 (by $7.8\%/6.6\%$) and HDT (by $7.5\%/5.9\%$) schemes in terms of AUC/PS for sequences with occlusion. Furthermore, the MCPF algorithm achieves better performance than the tracker in [66] without considering the part-based representation by $1.5\%/1.4\%$ in terms of AUC/PS. Overall, these results show that the particle sampling strategy helps improve correlation filter based tracking methods to handle target objects undergoing scale variation and partial occlusion.

4.4 Temple Color Dataset

We evaluate the proposed MCPF method on the Temple Color dataset [17] with 16 trackers in [17] and other 9 state-of-the-art tracking methods using the source codes including MUSTer [9], SRDCF [54], CF2 [3], HDT [4], DSST [1], Staple [53], DeepSRDCF [55], SRDCFdecon [57], and C-COT [38]. For fair comparisons, RGB color features are used for all trackers and the same AUC and PS metrics are used.

Figure 10 shows that the proposed MCPF 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. In contrast, the MCPF algorithm performs well in both metrics $(57.1\%, 77.9\%)$. The MCPF method obtains performance gain of 7.6% and 8.7% on the PS and AUC scores against the CF2 method, and outperforms the CNN based correlation filter trackers by a significant margin. The MCPF algorithm performs as well as the C-COT method and significantly outperforms other correlation filter based trackers (DSST and KCF). Compared to the tracker in [66] without considering the part-based representation, the MCPF algorithm achieves performance gain of 2.6% and 0.5% in terms of AUC and PS. Overall, the proposed MCPF tracker performs favorably against the state-of-the-art trackers in terms of both PS and AUC metrics.

4.5 VOT2015 Dataset

We evaluate the proposed MCPF method with 62 trackers in [18] including SAMF [67], SRDCF [54], DeepSRDCF [55], EBT [68], RAJSSC [69], Struck [36], and C-COT [38]. The tracking performance is measured both in terms of accuracy (overlap with the ground-truth) and robustness (failure rate) as in [18]. In the VOT2015 dataset [18], a tracker is restarted when failures occur.

Table 1 shows the evaluation results of the proposed approach with 11 trackers including the C-COT and top 10 methods in the VOT2016. The C-COT method achieves the best tracking results in terms of robustness with a high computational cost, the RAJSSC scheme achieves favorable results in terms of accuracy with a higher failure rate, and the SRDCF tracker performs well in terms of accuracy with some significant degradation in robustness. The proposed MCPF algorithm performs well with robustness (failure rate) of 0.94 and accuracy score of 0.57. For the proposed MCPF method without the part-based representation [66], the robustness and accuracy measures are 0.98 and 0.54. Furthermore, the proposed MCPF algorithm performs well against the the C-COT and EBT methods in both measures.

4.6 Effect of Particle Sampling on Visual Tracking

To evaluate the effect of particle sampling on correlation filter based trackers, the proposed MCPF method is designed with multiple deep features and without the part-based representation as in the CF2 [3] and HDT [4] methods. We evaluate the effects of particle number and scale on visual tracking performance in terms of effectiveness and efficiency.

Table 2 shows the evaluation results on the OTB2013 and OTB2015 datasets using the OPE protocol and run-time performance. The MCPF tracker with 10 particles achieves the AUC and PS scores of $(65.1\%/90.8\%)$ and $(61.0\%/86.7\%)$ on the OTB2013 and OTB2015 datasets, respectively. These results are significantly better than the CF2 $(60.5\%/89.1\%$ on the OTB2013 dataset and $56.2\%/83.7\%$ on the OTB2015 dataset) and HDT $(60.3\%/88.9\%$ on the OTB2013 dataset and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2797062, IEEE Transactions on Pattern Analysis and Machine Intelligence

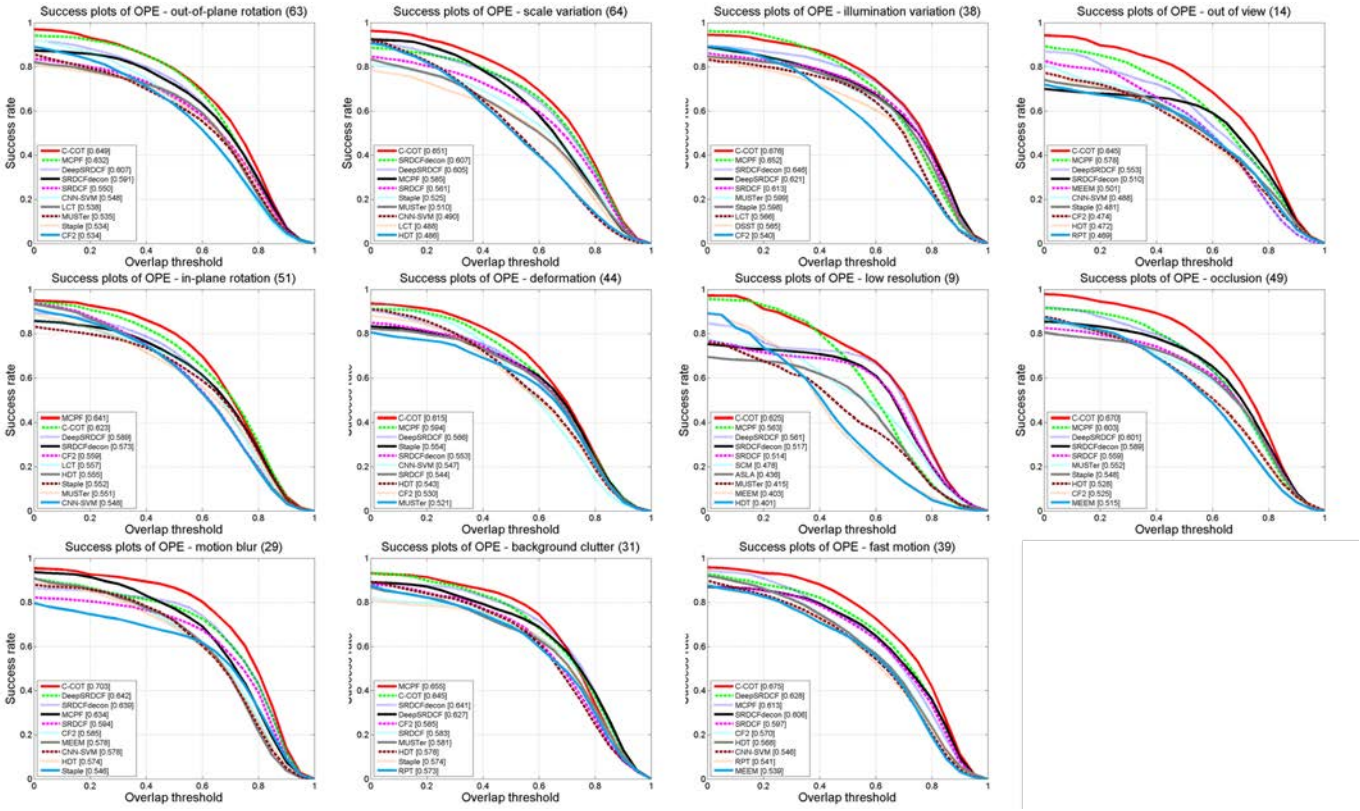


Fig. 9. Tracking performance based on attributes of image sequences on the OTB2015 dataset. Success plots on 11 tracking challenges of scale variation, out of view, out-of-plane rotation, low resolution, in-plane rotation, illumination, motion blur, background clutter, occlusion, deformation, and fast motion. The legend contains the AUC scores for each tracker. Our MCPF method performs favorably against the state-of-the-art trackers.

TABLE 1
Comparison with the state-of-the-art tracking methods on the VOT2015 dataset. The results are presented in terms of robustness and accuracy. The proposed MCPF method performs favorably against the state-of-the-art trackers.

	S3Tracker	RAJSSC	Struck	NSAMF	SC-EBT	sPST	LDP	SRDCF	EBT	DeepSRDCF	C-COT	MCPF
Robustness	1.77	1.63	1.26	1.29	1.86	1.48	1.84	1.24	1.02	1.05	0.82	0.94
Accuracy	0.52	0.57	0.47	0.53	0.55	0.55	0.51	0.56	0.47	0.56	0.54	0.57

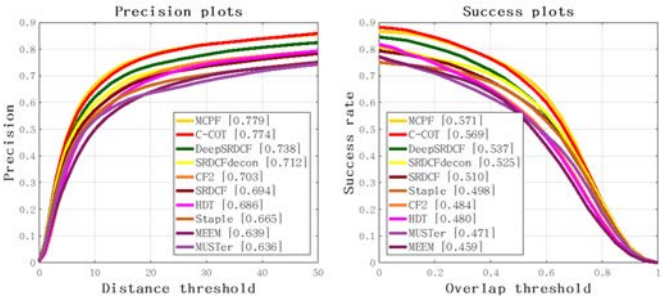


Fig. 10. Precision and success plots over the 128 sequences using OPE on the Temple Color dataset. The legend contains the AUC and PS scores for each tracker. The proposed MCPF method performs favorably against the state-of-the-art trackers.

TABLE 2
Effect of particle numbers on visual tracking performance. For different particle numbers, we report the FPS, AUC, and PS.

# Particles		10	30	50	100
AUC/PS	OTB2013	65.1/90.8	65.9/90.4	66.1/89.4	67.7/91.6
	OTB2015	61.0/86.7	62.7/87.6	62.1/86.7	62.8/87.3
FPS	OTB2013	1.96	1.29	0.85	0.58
	OTB2015	1.80	1.27	0.87	0.54

56.4%/84.8% on the OTB2015 dataset) methods in terms of AUC and PS. The above results demonstrate that the particle sampling strategy can improve correlation filter based trackers. 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 small number of particles. Even with a small number of particles, the proposed MCPF method can achieve comparable performance with much higher efficiency. Compared with the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2797062, IEEE Transactions on Pattern Analysis and Machine Intelligence

SCM [49] method, which is one of the top performing trackers using particle filters on the OTB2013 dataset in [5], the proposed MCPF algorithm achieves $17.8\%/26.7\%$ improvement in terms of AUC and PS. Furthermore, the MCPF algorithm is more efficient than the SCM method (0.4 FPS). These results show that correlation filter can improve existing particle filter based trackers by a large margin. In Table 3, we show the results of our 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 tracker performs robustly within a wide range of scale change.

4.7 Model Analysis

We analyze the effectiveness of the proposed model by using multiple features, part-based representation, particle filter for scale variation handling, and particle shepherding. With different experimental settings, we design 13 different methods including the MCPF, MCF, CPF, and CF2 [3], and their variants as shown in Table 4. The MCF is a multi-task correlation filter tracker (i.e., MCPF without using the particle filtering) and the CPF method is based on a correlation particle filter tracker (i.e., MCPF using a conventional correlation filter instead of the multi-task correlation filter). As discussed in Section 4.1, the MCPF algorithm uses features from the *conv3-4*, *conv4-4* and *conv5-4* layers of the VGG-Net-19. The MCPF-F is the MCPF without the part-based representation, and the MCPF-P is the MCPF with only *conv5-4* features of the VGG-Net-19. Similarly, we design the MCF-F, MCF-P, CPF-F, and CPF-P methods accordingly. The MCF-S is the MCF by applying different scales to the search region, which is the MCPF by sampling particles without considering the translation. The MCPF-S is the MCPF using a conventional particle filter without the particle shepherding scheme. The CF2S is the CF2 [3] method using the adaptive multi-scale strategy as the DSST tracker [1]. The CF2H is the CF2 scheme using HOG features instead of deep features [3].

Table 4 shows that both the multi-task correlation filter and particle filter can improve object tracking performance. We have the following observations from the evaluation results. First, deep features (from different layers) can improve visual tracking performance significantly. The effectiveness of deep features for visual tracking has been demonstrated in the CF2 [3] and HDT [4] methods. Furthermore, the CF2 method (deep features) performs well against the CF2H scheme (HOG features) by $9.8\%/16.3\%$ and $8.3\%/15.3\%$ in terms of AUC/PS on the OTB2013 and OTB2015 datasets.

Second, the tracking method using multiple deep features performs better than the approach with one type of features. Compared with the MCPF-P scheme, the MCPF algorithm achieves $6.3\%/7.2\%$ and $6.8\%/8.4\%$ improvement in terms of AUC/PS on the OTB2013 and OTB2015 datasets. Similarly, the MCF and CPF methods achieve favorable performance than the MCF-P and CPF-P schemes, respectively. These results demonstrate the effectiveness of multiple features for visual tracking.

Third, the part-based representation is effective as the MCPF, MCF, and CPF methods consistently outperform the MCPF-F, MCF-F, and CPF-F schemes, respectively. Furthermore, the MCPF method with the part-based representation performs well for sequences with occlusion attributes as demonstrated in Figure 6 and Figure 9.

Fourth, the multi-task correlation filter improves tracking performance. Compared with the CPF method, the MCPF algorithm achieves $1.0\%/1.3\%$ and $2.0\%/1.2\%$ improvement in terms of AUC/PS on the OTB2013 and OTB2015 datasets. Compared with the CF2 scheme, the MCF-F method achieves 0.4% and 1.0% improvement in terms of AUC and PS on the OTB2015 dataset. Furthermore, the MCPF-F and MCPF-P methods consistently outperform the CPF-F and CPF-P trackers, respectively. These results show the effectiveness of the multi-task correlation filter by exploiting the interdependencies among different parts and features for visual tracking.

Fifth, particle filters can handle scale variation well. Compared with the MCF method, the MCPF algorithm achieves favorable performance with $7.1\%/3.3\%$ and $5.9\%/3.8\%$ improvement on the OTB2013 and OTB2015 datasets. These results show that particle filters complement the MCF and significantly improve tracking performance. Furthermore, both CPF-F and CF2S methods perform better than the CF2 tracker [3], and the CPF-F method achieves better performance than the CF2S scheme in terms of AUC and PS. These results show both particle filters and the adaptive multi-scale strategy [1] improve tracking performance. We note the MCPF tracker with a particle filter handles scale variation well, which is demonstrated in Figure 6 and Figure 9 for sequences with scale variation attribute. Furthermore, compared with the MCF-S, the MCPF achieves better performance, which demonstrates that it is useful to sample particles by considering both different scales and translations. By only applying different scales to the search region, the MCF-S may fail when the search region does not cover the target object well. Different from the MCF-S, the MCPF samples particles to maintain multiple search regions, thereby facilitating the tracker to better handle fast moving objects and recover from failure or temporary distraction.

Finally, the particle shepherding strategy via the correlation filter improves visual tracking performance. Compared with the MCPF-S method, the MCPF algorithm improves significantly by $18.3\%/25.2\%$ and $17.8\%/29.2\%$ in terms of AUC/PS on the OTB2013 and OTB2015 datasets, respectively. In conventional particle filter based trackers, the predicted target state may be not correct if the sampled particles do not cover object states well as shown in Figure 2(a). When we use a correlation filter, each sampled particle can be shepherded to the local modes of a target state distribution as shown in Figure 2(b), which results in robust tracking performance.

TABLE 3
Effect of particle scale (s) on visual tracking in terms of AUC and PS corresponding to the OPE.

Scale	0.005	0.01	0.02	0.05
OTB2013	65.2/90.9	67.7/91.6	66.1/89.4	64.1/89.6
OTB2015	60.2/86.0	62.8/87.3	62.1/86.7	61.0/86.3

TABLE 4
Model analysis by comparing MCPF, MCF, CPF, CF2, and CF2S. The AUC and PS are reported on the OTB2013 and OTB2015 datasets (AUC/PS) corresponding to the OPE.

Dataset	MCPF	MCPF-F	MCPF-P	MCPF-S	MCF	MCF-S	MCF-F	MCF-P	CPF	CPF-F	CPF-P	CF2	CF2S	CF2H
OTB2013	69.6/92.8	67.7/91.6	63.3/85.6	51.3/67.6	62.5/89.5	67.3/92.6	60.7/89.3	58.5/84.1	68.6/91.5	65.7/89.3	60.2/84.5	60.5/89.1	63.4/89.1	50.7/72.8
OTB2015	64.3/88.7	62.8/87.3	57.5/80.3	46.5/59.5	58.4/84.9	61.1/86.7	56.6/84.7	53.6/78.7	62.3/86.5	61.2/86.3	54.5/78.5	56.2/83.7	59.1/84.0	47.9/68.4

5 Conclusion

In this paper, we propose a multi-task correlation particle filter for robust visual tracking. The proposed tracking algorithm effectively handles scale variation via a particle sampling scheme, and deals with partial occlusion via a part-based representation. Furthermore, the proposed MCPF method exploits the interdependencies among different features and the intrinsic relationship among parts to learn their correlation filters jointly by preserving object structure, and shepherds 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.

References

[1] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *British Machine Vision Conference*, 2014. 1, 2, 3, 4, 6, 7, 8, 9, 10, 12

[2] 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*, vol. 37, no. 3, pp. 583–596, 2015. 1, 2, 3, 4, 6, 7, 8, 9

[3] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *IEEE International Conference on Computer Vision*, 2015. 1, 2, 3, 4, 6, 7, 8, 9, 10, 12

[4] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M.-H. Yang, "Hedged deep tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1, 2, 3, 4, 6, 7, 8, 9, 10, 12

[5] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 1, 3, 7, 8, 9, 12

[6] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2544–2550. 1, 3

[7] J. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *European Conference on Computer Vision*, 2012. 1, 2, 3, 4

[8] M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer, "Adaptive color attributes for real-time visual tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1090–1097. 1, 3

[9] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 749–758. 1, 3, 7, 8, 9, 10

[10] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Long-term correlation tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5388–5396. 1, 2, 3, 7, 9

[11] T. Liu, G. Wang, and Q. Yang, "Real-time part-based visual tracking via adaptive correlation filters." in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4902–4912. 1, 2, 3

[12] Y. Li, J. Zhu, and S. C. H. Hoi, "Reliable patch trackers: Robust visual tracking by exploiting reliable patches." in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 353–361. 1, 2, 3, 7

[13] M. K. et al., "The visual object tracking vot2014 challenge results," in *ECCV Workshops*, 2014. 2

[14] M. S. Arulampalam, S. Maskell, and N. Gordon, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, pp. 174–188, 2002. 2, 5

[15] M. Isard and A. Blake, "Condensation - conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, pp. 5–28, 1998. 2, 3, 4

[16] Y. Wu, J. Lim, and M. Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, 2015. 3, 7

[17] P. Liang, E. Blasch, and H. Ling, "Encoding color information for visual tracking: Algorithms and benchmark," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5630–5644, 2015. 3, 7, 10

[18] M. K. et al., "The visual object tracking vot2015 challenge results," in *ICCV Workshop*, 2015. 3, 7, 10

[19] 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*, 2014, pp. 127–141. 3

[20] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, p. 13, 2006. 3

[21] S. Salti, A. Cavallaro, and L. D. Stefano, "Adaptive appearance modeling for video tracking: Survey and evaluation," *IEEE Transactions on Image Processing*, vol. 21, no. 10, pp. 4334–4348, 2012. 3

[22] M. Kristan, L. Cehovin, and et al., "The visual object tracking vot 2013 challenge results," in *ICCV Workshop on Visual Object Tracking Challenge*, 2013. 3

[23] Y. Pang and H. Ling, "Finding the best from the second bests - inhibiting subjective bias in evaluation of visual tracking algorithms," in *IEEE International Conference on Computer Vision*, 2013. 3

[24] A. Smeulders, D. Chu, R. Cucchiara, S. Calderara, A. Deghan, and M. Shah, "Visual tracking: an experimental survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442–1468, 2013. 3

[25] M. J. Black and A. D. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation," *International Journal of Computer Vision*, pp. 63–84, 1998. 3

[26] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-Based Object Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–575, 2003. 3

[27] D. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental Learning for Robust Visual Tracking," *International Journal of Computer Vision*, vol. 77, no. 1, pp. 125–141, 2008. 3

[28] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 3

[29] X. Mei and H. Ling, "Robust Visual Tracking and Vehicle Classification via Sparse Representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2259–2272, 2011. 3, 6, 7

[30] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 798–805. 3

[31] L. Čehovin, M. Kristan, and A. Leonardis, "Robust visual tracking using an adaptive coupled-layer visual model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 4, pp. 941–953, 2013. 3

[32] S. Avidan, "Ensemble tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 494–501. 3

[33] H. Grabner, M. Grabner, and H. Bischof, "Real-Time Tracking via On-line Boosting," in *British Machine Vision Conference*, 2006. 3

[34] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 3

[35] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012. 3

[36] S. Hare, A. Saffari, and P. Torr, "Struck: Structured output tracking with kernels," in *IEEE International Conference on Computer Vision*, 2011. 3, 10

[37] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: Robust tracking via multiple experts using entropy minimization," in *European Conference on Computer Vision*, 2014. 3, 7, 9
[38] M. Danelljan, A. Robinson, F. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *European Conference on Computer Vision*, 2016. 3, 7, 8, 9, 10
[39] S. Liu, T. Zhang, X. Chao, and C. Xu, "Structural correlation filter for robust visual tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5388–5396. 3, 7
[40] M. Godec, P. M. Roth, and H. Bischof, "Hough-based tracking of non-rigid objects," in *IEEE International Conference on Computer Vision*, 2011. 3
[41] M. Tang and J. Feng, "Multi-kernel correlation filter for visual tracking," in *IEEE International Conference on Computer Vision*, 2015, pp. 3038–3046. 3
[42] C.Yang, R. Duraiswami, and L. Davis, "Fast multiple object tracking via a hierarchical particle filter," in *IEEE International Conference on Computer Vision*, 2005. 4
[43] Z. Khan, T. Balch, and F. Dellaert, "A rao-blackwellized particle filter for eigentracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2004. 4
[44] S. K. Zhou, R. Chellappa, and B. Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters," *IEEE Transactions on Image Processing*, vol. 11, no. 1, pp. 1491–1506, 2004. 4
[45] A. Doucet, N. De Freitas, and N. Gordon, "Sequential monte carlo methods in practice," in *Springer-Verlag*. New York, 2001. 4
[46] G. C. Cristina, S. Marc, J. Frédéric, and B. G. J., "Motion models that only work sometimes," in *British Machine Vision Conference*, 2012. 4
[47] R. Rifkin, G. Yeo, and T. Poggio, "Regularized least-squares classification," *NATO Science Series Sub Series III Computer and Systems Sciences*, vol. 190, pp. 131–154, 2003. 4
[48] X. Chen, W. Pan, J. Kwok, and J. Carbonell, "Accelerated gradient method for multi-task sparse learning problem," in *IEEE International Conference on Data Mining*, 2009. 4
[49] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model." in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1838–1845. 6, 8, 12
[50] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015. 7
[51] A. Vedaldi and K. Lenc, "Matconvne: convolutional neural networks for matlab," in *Proceeding of the ACM Int. Conf. on Multimedia*, 2015. 7
[52] J. Gao, H. Ling, W. Hu, and J. Xing, "Transfer learning based visual tracking with gaussian process regression," in *European Conference on Computer Vision*, 2014. 7, 8, 9
[53] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 7, 9, 10
[54] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *IEEE International Conference on Computer Vision*, 2015, pp. 4310–4318. 7, 9, 10
[55] M. Danelljan, G. Hager, F. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *ICCV Workshop*, 2015. 7, 9, 10
[56] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *International Conference on Machine Learning*, 2015. 7, 9
[57] M. Danelljan, G. Hager, F. Khan, and M. Felsberg, "Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 7, 9, 10
[58] R. Tao, E. Gavves, and A. W. M. Smeulders, "Siamese instance search for tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 7, 8
[59] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. Torr, "Fully-convolutional siamese networks for object tracking," *ECCV Workshop*, 2016. 7
[60] H. Possegger, T. Mauthner, and H. Bischof, "In defense of color-based model-free tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 7
[61] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in *IEEE International Conference on Computer Vision*, 2015. 7
[62] J. Choi, H. J. Chang, J. Jeong, Y. Demiris, and J. Y. Choi, "Visual tracking using attention-modulated disintegration and integration," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 7
[63] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 8, 10
[64] H. Fan and H. Ling, "Sanet: Structure-aware network for visual tracking," in *CVPR Workshop on DeepVision: Temporal Deep Learning*, 2017. 8, 10
[65] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 8, 10
[66] T. Zhang, C. Xu, and M.-H. Yang, "Multi-task correlation particle filter for robust object tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1–9. 10
[67] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *ECCV Workshop*, 2014. 10
[68] G. Zhu, F. Porikli, and H. Li, "Beyond local search: Tracking objects everywhere with instance-specific proposals," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 10
[69] M. Zhang, J. Xing, J. Gao, X. Shi, Q. Wang, and W. Hu, "Joint scale-spatial correlation tracking with adaptive rotation estimation," in *ICCV Workshop*, 2015. 10

Tianzhu Zhang is an associate professor at the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences in Beijing, China. He received his B.S. degree in communications and information technology from Beijing Institute of Technology in 2006. He obtained his Ph.D. in pattern recognition and intelligent systems from Institute of Automation, Chinese Academy of Sciences, in 2011. After graduation, he worked at Advanced Digital Sciences Center of Singapore. His research interests are in computer vision and multimedia, including action recognition, object classification and object tracking.

Changsheng Xu is a Professor in National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His research interests include multimedia content analysis, pattern recognition and computer vision. Dr. Xu is an Associate Editor of IEEE Trans. on Multimedia, ACM Trans. on Multimedia Computing, Communications and Applications and ACM/Springer Multimedia Systems Journal. He served as Program Chair of ACM Multimedia 2009. He has served as associate editor, guest editor, general chair, program chair, area/track chair, special session organizer, session chair and TPC member for over 20 IEEE and ACM prestigious multimedia journals, conferences, and workshops. He is IEEE Fellow, IAPR Fellow and ACM Distinguished Scientist.

Ming-Hsuan Yang is a professor in Electrical Engineering and Computer Science at University of California, Merced. He received the PhD degree in computer science from the University of Illinois at Urbana-Champaign in 2000. Yang served as an associate editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence from 2007 to 2011, and is an associate editor of the International Journal of Computer Vision, Image and Vision Computing and Journal of Artificial Intelligence Research. He received the NSF CAREER award in 2012 and the Google Faculty Award in 2009. He is a senior member of the IEEE and the ACM.