# **Robust Structural Sparse Tracking**

Tianzhu Zhang, Changsheng Xu, and Ming-Hsuan Yang

**Abstract**—Sparse representations have been applied to visual tracking by finding the best candidate region with minimal reconstruction error based on a set of target templates. However, most existing sparse trackers only consider holistic or local representations and do not make full use of the intrinsic structure among and inside target candidate regions, thereby making them less effective when similar objects appear at close proximity or under occlusion. In this paper, we propose a novel structural sparse representation, which not only exploits the intrinsic relationships among target candidate regions and local patches to learn their representations jointly, but also preserves the spatial structure among the local patches inside each target candidate region. For robust visual tracking, we take outliers resulting from occlusion and noise into account when searching for the best target region. Constructed within a Bayesian filtering framework, we show that the proposed algorithm accommodates most existing sparse trackers with respective merits. The formulated problem can be efficiently solved using an accelerated proximal gradient method that yields a sequence of closed form updates. Qualitative and quantitative evaluations on challenging benchmark datasets demonstrate that the proposed tracking algorithm performs favorably against several state-of-the-art methods.

Index Terms—Visual Tracking, Sparse Tracking, Structural Modeling, Sparse Representation.

# **1** INTRODUCTION

Visual tracking aims to estimate the states of moving targets in an image sequence. It has long been one of the most important and fundamental topics in computer vision with a wide range of applications such as surveillance, vehicle navigation, human computer interface, and human motion analysis, to name a few. Despite numerous object tracking methods [1], [2], [3], [4], [5], [6], [7] having been proposed in recent years, it remains a challenging task to develop a robust algorithm for complex and dynamic scenes due to the factors such as partial occlusions, illumination, pose variations, scale, camera motion, background clutters, and viewpoints.

Recently, sparse representations have been developed for visual tracking [8], [9], [10], [11], [12], [13], [14], [18], [16], [15], [17], [19]. These trackers can be categorized based on global, local, and joint sparse appearance models as shown in Figure 1. In [8], [10], [11], [12], [13], [18], each target candidate region  $\mathbf{x}_i$  is represented by a sparse linear combination of target templates T that can be dynamically updated to describe appearance changes. While these methods perform well in challenging scenarios, these trackers are less effective when objects are heavily occluded due to the adopted global sparse appearance models (See Figure 1(a)).

Sparse appearance models [9], [15] have been used for visual tracking as illustrated in Figure 1(b) where patches of a target candidate region  $x_i$  are sparsely represented with templates. In [9], Liu et al. propose an algorithm based on

a local sparse appearance model which employs histograms of sparse coefficients and the mean-shift algorithm for visual tracking. However, this method is based on a static local sparse dictionary and likely to fail when similar objects appear at close proximity or with occlusion in the scenes. Jia et al. [15] develop a tracking method based on a local sparse appearance model using a set of overlapped image patches inside the target region with the corresponding spatial layout. These local patches are used to form a dictionary for encoding possible candidate regions. For a target candidate region, its local patches are extracted in the same way. Since each local patch represents one fixed part of the target object, the whole set altogether represents the overall structure of the target. With the sparsity assumption, the local patches within the target candidate region can be represented as the linear combination of a few dictionary bases by solving an  $\ell_1$  minimization problem. Although this model addresses some issues of global sparse appearance models, such tracking algorithms [9], [15] do not consider the spatial structure among the local patches inside each target candidate region or the correlations among local patches from all target candidate regions. For example, as shown in Figure 1(b), local patches inside a target candidate region  $x_i$  may be sparsely represented by those from different dictionary templates. To maintain the spatial layout among the local patches, the purple local patch of  $\mathbf{x}_i$  is best represented by the local patch of the first dictionary basis, and the blue local patch of  $x_i$  should also be represented by the corresponding blue local patch of the first dictionary basis.

1

The joint sparse appearance models [14], [16], [17], shown in Figure 1(c), are developed to exploit structure information based on a few assumptions. In particle filter-based tracking methods, particles at and around a target object are densely sampled based on the previous states. Each particle shares dependency with other particles and their corresponding image regions are likely to be similar. In [14], learning the representation of each particle is viewed as an individual task and

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.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE



Fig. 1. Sparse representation based tracking methods [8], [9], [10], [11], [12], [13], [14], [15], [16], [17]. Given an image with n sampled particles  $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_i, \cdots, \mathbf{x}_n]$  and the dictionary templates T. (a) Global sparse appearance model [8], [10], [11], [12], [13]. These tracking methods model holistic appearance of a target object with sparse representations. As a result, the target candidate region  $x_i$  is represented by a sparse number of elements in T. (b) Local sparse appearance model [9], [15]. These tracking methods represent each local patch inside one target candidate region  $x_i$  by a sparse linear combination of templates in T. Note that, the local patches inside a target candidate region  $x_i$  may be sparsely represented by the corresponding patches inside different dictionary templates. (c) Joint sparse appearance model[14], [16], [17]. These tracking methods exploit the intrinsic relationships among particles X to learn the sparse representations jointly. The joint sparsity constraints encourage all particle representations to be jointly sparse and share the same dictionary templates that reliably represent them. (d) Proposed structural sparse appearance model incorporates the above three models together. The proposed model exploits the intrinsic relationships among particles X and their local patches to learn their sparse representations jointly. In addition, our method also preserves the spatial layout structure among the local patches inside each target candidate region. which is ignored by the above three models [8], [10], [11], [12], [13], [9], [15], [14], [16], [18], [17]. In the proposed algorithm, all particles X and their local patches are represented with joint sparsity, i.e., only a few (but the same) dictionary templates are used to represent all the particles and their local patches at each frame. We note that the local patches inside all particles X are represented with joint sparsity by the corresponding local patches inside the same dictionary templates used to represent X.

a multi-task learning formulation for all particles with joint sparsity is proposed. In addition, a low-rank sparse learning approach is introduced to model all particles jointly [16] for visual tracking. On the other hand, a multi-task multi-view joint sparse representation for visual tracking [17] is proposed. All the above-mentioned tracking methods based on joint sparse appearance models represent holistic object appearance.

In this work, we propose a novel structural sparse appearance model (See Figure 1(d)) for robust visual tracking. First, the proposed structural sparse appearance model incorporates the properties of the above three approaches (local, global, and joint sparse representations), which is more robust to partial occlusion [9], [15], as well as computationally efficient [14], [16], [17] by considering the correlations among the target candidate regions. Second, the proposed model exploits the intrinsic relationships among the particles **X**, and corresponding local image patches to learn sparse representations jointly. Third, the proposed model preserves spatial layout structure among local patches inside each target candidate region, which is not exploited in the previous sparse trackers [8], [10], [11], [12], [13], [9], [15], [14], [16], [18], [17]. As shown in Figure 1(d), since all particles  $\mathbf{X}$  and their local patches are represented with joint sparsity, only a few (but the same) dictionary templates are used to represent all the particles and their local patches inside all particles  $\mathbf{X}$  are represented with joint sparsity by the corresponding local patches inside the same dictionary templates for modeling  $\mathbf{X}$ .

2

Based on the structural sparse appearance model, we propose a computationally efficient structural sparse tracking (SST) algorithm within the particle filter framework. All particles and their local patches are represented via the proposed structural sparse appearance model, and the next target state is the particle that it and its local patches have the smallest

reconstruction error with target dictionary templates and their corresponding patches. Unlike previous methods, the SST algorithm not only exploits the intrinsic relationships among particles and their local patches to learn their sparse representations jointly, but also preserves the spatial layout structure among the local patches inside each target candidate region. In the SST formulation, we use the  $\ell_{p,q}$  mixed-norm regularizer, which is optimized with an accelerated proximal gradient method for fast convergence. In addition, we show that existing  $\ell_1$  tracker [12], LST [15], and MTT [14] methods are special cases of our SST formulation. The SST algorithm assumes that the same local patches of all particles are expected to be similar, and the local patches of a particle should be represented by the local patches of the same target templates. This assumption generally does not usually hold in visual tracking applications, since outlier patches often exist. For example, a small number of particles sampled far away from the majority of particles are likely to have little overlap with other particles and thus considered as outliers. Furthermore, due to occlusions or noises, some local patches of a particle may select different target templates for representation. Based on the fact that most of the particles are relevant and outliers often exist, we improve the SST and introduce a robust structural sparse tracking (RSST) algorithm to capture the underlying relationships shared by all local patches and outliers due to occlusion and noise. Experiments on challenging benchmark datasets demonstrate that our algorithm performs favorably

against several state-of-the-art methods. Preliminary results of this work based on the SST method are presented in [20]. Compared with [20], a number of improvements are made in the proposed RSST algorithm: (i) We propose a robust structural sparse tracker, which not only captures the underlying relationships among all local patches as the SST method, but also effectively models the outliers due to occlusion and noise. (ii) Detailed analysis of most recent tracking methods is performed and experiments on larger benchmark datasets [5], [21] with state-of-the-art

# 2 RELATED WORK

approaches are conducted.

Numerous tracking algorithms have been proposed in recent years [1], [5]. In this section, we discuss the most relevant methods to our work in terms of generative tracking, discriminative tracking, and tracking algorithms based on sparse representation, correlation filter, and deep learning.

**Generative Methods.** A generative tracking method typically learns a representation model of a target object and uses it to search for the image region with the minimal reconstruction error [22], [23], [24], [25], [26], [27]. Black et al. [22] learn a subspace model off-line to represent the object of interest for tracking. The mean shift tracking algorithm [23] models a target object with a nonparametric distribution of features (e.g., color pixels) and locates the object based on mode shifts. In [24], an adaptive appearance model based on the mixture of Gaussians is used to represent a target object with three components. The Frag tracker [25] addresses the partial occlusion problem by modeling object appearance with histograms of local patches. In [26], the incremental visual tracking method utilizes a subspace model with online update to account for appearance changes. In contrast, Kwon et al. [27] use multiple observation models to describe a wide range of appearance changes caused by pose and illumination variation for tracking.

Discriminative Schemes. A discriminative approach formulates the tracking task as a detection problem based on a binary classifier that separates the target object from the background [28], [29], [30], [31], [32], [33], [34]. Collins et al. [28] demonstrate that the most discriminative features can be learned online to separate the target object from the background for visual tracking. In [29], Avidan combines a set of weak classifiers in an ensemble for visual tracking. Grabner et al. propose an online boosting method to update discriminative features [30] and a semi-online boosting algorithm [31] to handle the drifting problem in object tracking. The multiple instance learning approach is incorporated in an online object tracking method [32] where samples are considered within positive and negative bags or sets. Kalal et al. [33] propose the P-N learning algorithm to exploit the underlying structure of positive and negative samples to learn classifiers for object tracking. An online tracking-by-detection algorithm that integrates template matching, optical flow, and an online random forest is proposed for object tracking [34].

**Sparse Representation.** Sparse representation methods have been introduced to object tracking with particle filters [9], [10], [11], [14], [18], [15], [17]. The basic idea is to represent each target candidate as a sparse linear combination of dictionary templates that can be updated to account for appearance change. This representation has been shown to be robust against partial occlusions, thereby facilitating the tracking task. In [12], the sparse tracking algorithm solves a  $\ell_1$  minimization problem for each particle which requires high computational complexity. To reduce the computational cost, numerous methods have been developed [11], [10]. Most recently, an algorithm that learns the sparse representations of all particles jointly [14], [16], [17] is proposed for object tracking.

**Correlation Filters.** Tracking methods based on correlation filters have been shown to achieve high speed and robust performance in recent years [35], [36], [37], [38], [39]. For tracking, a correlation filter evaluates the similarity between the translated image region and a learned template via the inner product. The computation of correlation filters can be significantly reduced based on the convolution theorem. Henriques et al. exploit the circulant structure of shifted image patches in a kernel space and propose the KCF approach [35]. In [36], the DSST tracker is proposed with adaptive multi-scale correlation filters to handle scale variations. Hong et al. [37] develop the MUSTer tracker under the biology-inspired framework. Most recently, several tracking methods based on correlation filters and deep features have also been introduced [38], [39].

**Deep Learning.** In recent years, deep learning methods, e.g., Convolutional Neural Networks (CNNs), have been applied to visual tracking with demonstrated success [38], [39], [40], [41], [42], [43], [44], [45]. Due to limited training samples,

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE



Fig. 2. Spatial layout for sampling local patches to describe a target. Note that other sampling methods can also be used to extract local patches for representation.

most deep learning based trackers adopt a pre-trained model from other vision tasks [38], [39], [45], [46], [44]. In [46], an offline pre-trained stacked denoising autoencoder is used for visual tracking. Hong et al. [44] develop a tracking method based on a discriminative saliency model and a pre-trained CNN. Although CNN features from all layers provide rich descriptions of objects, Wang et al. [43] show that only a subset of them are useful for tracking. On the other hand, tracking methods that exploit combinations of hierarchical deep features have been developed [38], [39]. Different from the above methods using pre-trained deep learning models, Li et al. [47] present a target-specific CNN with a truncated structural loss to construct an online tracker, which learns two-layer CNN models from binary samples without pretraining. Most recently, deep tracking methods using external videos [42], [41] have been shown to achieve the state-of-theart performance in terms of accuracy and efficiency.

# **3** STRUCTURAL SPARSE TRACKING

In this section, we present the tracking algorithm based on the proposed structural sparse appearance models and the particle filtering framework to represent particles and the corresponding local patches jointly. First, particles are sampled at and around the previous object location to predict the state  $s_t$  of the target at time t, from which we crop the region of interest  $y_t$  in the current image and normalize it to the template size. For computational efficiency, the state transition function  $p(\mathbf{s}_t | \mathbf{s}_{t-1})$  is modeled by an affine motion model with a diagonal Gaussian distribution although other transformations can be used. The observation model  $p(\mathbf{y}_t|\mathbf{s}_t)$  reflects the similarity between an observed image region  $\mathbf{y}_t$  corresponding to a particle  $s_t$  and the templates of the current dictionary. In this work, the likelihood function  $p(\mathbf{y}_t|\mathbf{s}_t)$  is computed by the reconstruction error by linearly representing an image region  $y_t$  and its local patches using the template dictionary. The particle that maximizes this function is selected to be the tracked target at each time instance. Next, we show how to use the structural sparse appearance models to represent particles and their local patches in details, respectively.

#### 3.1 Structural Sparse Appearance Model

Given the image set of the target templates  $\mathbf{T} = [\mathbf{T}_1, \mathbf{T}_2, \cdots, \mathbf{T}_m]$ , we sample K local image patches inside each target region with a spatial layout. For simplicity, the spatial layout as shown in Figure 2 is used although any other

local patch sampling methods can also be adopted. These samples are used to form a dictionary for encoding local patches inside any candidate region. For the k-th local image patch among these m target templates, we obtain the corresponding dictionary templates  $\mathbf{D}^k = [\mathbf{d}_1^k, \mathbf{d}_2^k, \cdots, \mathbf{d}_m^k] \in \mathbb{R}^{d_k \times m}$ , where  $k = 1, \dots, K$ ; K is the number of local patches sampled within the target region;  $d_k$  is the dimension of the k-th image patch vector; and m is the number of target templates. Each column in  $\mathbf{D}^k$  is obtained by  $\ell_2$  normalization on the vectorized gray-scale image observations extracted from T. Each local patch represents one fixed part of the target, and hence the local patches altogether can represent the whole structure of the target. Since the image patches are collected from numerous templates, this dictionary captures the commonality of different templates and is able to represent various forms of these parts.

4

To account for appearance variations of a target object for robust visual tracking, the dictionary template set T is progressively updated. The dictionary update scheme in this work is similar to [12]. Each target template in T is assigned a weight that indicates how representative the template is. When a template is frequently used to represent tracking results, it has a higher weight. When the set T does not represent particles well, the target template with the lowest weight is replaced by the current tracking result. To initialize the mtarget templates, we sample equal-sized patches at and around the initial position of the target.

At time t, n particles are drawn and the corresponding vectorized gray-scale image observations form a matrix  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]$  where the observation with respect to the *i*-th particle is denoted as  $x_i$ . For a target candidate region  $\mathbf{x}_i$ , we extract K local patches within it to construct a dictionary of templates  $\mathbf{D}^k$ . For the k-th local image patches of these n particle samples, their corresponding vectorized gray-scale image observations form a matrix  $\mathbf{X}^k =$  $\left[\mathbf{x}_{1}^{k}, \mathbf{x}_{2}^{k}, \cdots, \mathbf{x}_{n}^{k}\right] \in \mathbb{R}^{d_{k} imes n}$ . We represent each observation from  $\mathbf{X}^k$  by a linear combination of templates from the dictionary  $\mathbf{D}^k$  such that  $\mathbf{X}^k = \mathbf{D}^k \mathbf{Z}^k$ . The columns of  $\mathbf{Z}^k = \begin{bmatrix} \mathbf{z}_1^k, \mathbf{z}_2^k, \cdots, \mathbf{z}_n^k \end{bmatrix} \in \mathbb{R}^{m imes n}$  denote the representations of the k-th local patch observations with respect to  $\mathbf{D}^k$ . Putting the representations of all the K local patches together, we obtain  $\mathbf{Z} = [\mathbf{Z}^1, \mathbf{Z}^2, \cdots, \mathbf{Z}^K] \in \mathbb{R}^{m \times nK}$ . For the *i*-th particle, the corresponding representations of its local patches form a matrix  $\mathbf{Z}_i = \begin{bmatrix} \mathbf{z}_i^1, \mathbf{z}_i^2, \cdots, \mathbf{z}_i^K \end{bmatrix} \in \mathbb{R}^{m \times K}$ .

As shown in Figure 3, we have the following assumptions of  $\mathbf{Z}^k$  and  $\mathbf{Z}_i$  for visual tracking. First, for  $\mathbf{Z}^k$ , it consists of the representations of all k-th image patches from n sampled particles. As these particles are densely sampled at and around the target, these particles are likely to be similar, and the corresponding k-th image patches are also expected to be similar. Therefore, the underlying relationships among local patches should be exploited. In contrast, existing methods based on local sparse representations [9], [15] do not take these properties into account. Second, for  $\mathbf{Z}_i$ , it contains the corresponding representations of all local patches of the *i*th particle. Since these local patches are sampled inside the target candidate region, their spatial layout structure should be preserved. Namely, after sampling these local patches via

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE



Fig. 3. Illustration for the structure of the learned coefficient matrix **Z**, where entries of different color represent different values, and the white entries indicate zero values in rows and columns.

the spatial layout as shown in Figure 2, their representations should meet the following constraints. If the *k*-th image patch inside the *i*-th particle  $\mathbf{z}_i^k$  is represented by the *k*-th element of the target template  $\mathbf{T}_j = \{\mathbf{d}_j^1, \mathbf{d}_j^2, \cdots, \mathbf{d}_j^K\}$ , the other image patches should also be represented by the corresponding elements in the same target template  $\mathbf{T}_j$ .

Motivated by the above assumptions, we use the convex  $\ell_{p,q}$  mixed norm, e.g.,  $\ell_{2,1}$ , to model the structure information of  $\mathbf{Z}^k$  and  $\mathbf{Z}_i$ , and obtain the structural sparse appearance model for visual tracking as

$$\min_{\mathbf{Z}} \frac{1}{2} \sum_{k=1}^{K} \left\| \mathbf{X}^{k} - \mathbf{D}^{k} \mathbf{Z}^{k} \right\|_{F}^{2} + \lambda \|\mathbf{Z}\|_{2,1},$$
(1)

where  $\mathbf{Z} = [\mathbf{Z}^1, \mathbf{Z}^2, \dots, \mathbf{Z}^K] \in \mathbb{R}^{m \times nK}, \|\cdot\|_F$  denotes the Frobenius norm, and  $\lambda$  is a tradeoff parameter between reconstruction error and joint sparsity regularization. The  $\ell_{p,q}$ mixed norm is defined by  $\|\mathbf{Z}\|_{p,q} = \left(\sum_i \left(\sum_j |[\mathbf{Z}]_{ij}|^p\right)^{\frac{q}{p}}\right)^{\frac{q}{q}}$ and  $[\mathbf{Z}]_{ij}$  denotes the entry at the *i*-th row and *j*-th column of  $\mathbf{Z}$ . Figure 3 illustrates the structure of the learned matrix  $\mathbf{Z}$ . After learning the representation  $\mathbf{Z}$ , the observation likelihood of a target candidate region *i* is defined by

$$p(\mathbf{y}_t|\mathbf{s}_t) = \frac{1}{\beta} \exp(-\alpha \sum_{k=1}^{K} \left\| \mathbf{x}_i^k - \mathbf{D}^k \mathbf{z}_i^k \right\|_F^2), \qquad (2)$$

where  $\mathbf{z}_i^k$  is the coefficient of the *i*-th candidate corresponding to the target templates of the *k*-th image patch, and  $\alpha$  and  $\beta$  are normalization parameters. The tracking result is the particle that has the maximum observation likelihood.

We illustrate the proposed SST algorithm using one example in Figure 4. Given all particles sampled around the tracked object, the local patches  $(\mathbf{X}^k, k = 1, \dots, K)$  of these observations can be obtained based on the spatial layout as shown in Figure 2. Based on the corresponding dictionary templates



5

Fig. 4. An illustrative example of the proposed tracking algorithm. (a) Objective value vs the number of iteration. The proposed algorithm can converge in several iterations. (b) The learned matrix  $\mathbf{X} \in \mathbb{R}^{20 \times 5600}$  where m = 20, K = 14, and n = 400. Notice that the columns of  $\mathbf{Z}$  are jointly sparse, i.e., a few (but the same) dictionary templates are used to represent all image patches together. (c) The particle  $\mathbf{x}_i$  is selected as the tracking result since it has the smallest reconstruction error.

 $(\mathbf{D}^k, k = 1, \dots, K)$ , we learn the representation matrix  $\mathbf{Z} = [\mathbf{Z}^1, \mathbf{Z}^2, \dots, \mathbf{Z}^K]$  by solving (1). Note that a brighter color square in  $\mathbf{Z}$  indicates a larger value in the corresponding entry. In addition, the white entries denote the elements with zero values. Clearly, the columns of  $\mathbf{Z}$  are jointly sparse, i.e., a few (but the same) dictionary templates are used to represent all the image patches together. The particle  $\mathbf{x}_i$  is determined as the current tracking result  $\mathbf{y}_t$  because the reconstruction error of its image patches with respect to the target templates is the smallest among all particles. Since particle  $\mathbf{x}_j$  corresponds to a misaligned image of the target, it has larger reconstruction error.

#### 3.2 Robust Structural Sparse Appearance Model

The structural sparse appearance model (1) is motivated by two assumptions of  $\mathbf{Z}^k$  and  $\mathbf{Z}_i$  as discussed in Section 3.1. The first assumption is concerned with  $\mathbf{Z}^k$ , which describes the representations of all the k-th image patches of the n sampled particles. With this assumption, the k-th image patches are expected to be similar. However, it generally does not hold in visual tracking since image outliers and noise often exist. For example, a small number of particles sampled far away from the majority of particles may have little overlap with other particles and are considered as outliers.

The second assumption is on  $\mathbf{Z}_i$ , i.e., the corresponding representations of all local patches of the *i*-th particle. To preserve the spatial layout structure, if the *k*-th image patch inside the *i*-th particle is represented by the *k*-th element of the target template  $\mathbf{T}_j = \{\mathbf{d}_j^1, \mathbf{d}_j^2, \cdots, \mathbf{d}_j^K\}$ , the corresponding elements in the same target template  $\mathbf{T}_j$  are assumed to be selected for the other image patches for representation. Due to image noise or occlusion, the *k*-th element of the other target template  $\mathbf{T}_{j^*}$  ( $j^* \neq j$ ) may be selected to represent some local patches of the *i*-th particle for visual tracking. These local patches are outliers and cannot be modeled well via (1).

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.2018.2797082, IEEE Transactions on Pattern Analysis and Machine Intelligence

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE



Fig. 5. Illustration of the coefficient matrix  $\mathbf{Z} = \mathbf{P} + \mathbf{Q}$ , where squares with white background denote zero entries. There are 6 local patches, where the fourth local patch is an outlier patch and has different representations from other patches.

To deal with these issues, we propose a robust structural sparse appearance model for visual tracking as follows,

$$\min_{\mathbf{Z},\mathbf{P},\mathbf{Q}} \frac{1}{2} \sum_{k=1}^{K} \left\| \mathbf{X}^{k} - \mathbf{D}^{k} \mathbf{Z}^{k} \right\|_{F}^{2} + \lambda_{1} \left\| \mathbf{P} \right\|_{2,1} + \lambda_{2} \left\| \mathbf{Q}^{\top} \right\|_{2,1}$$
  
s.t.  $\mathbf{Z} = \mathbf{P} + \mathbf{Q}, \quad \mathbf{Z} = \begin{bmatrix} \mathbf{Z}^{1}, \mathbf{Z}^{2}, \cdots, \mathbf{Z}^{K} \end{bmatrix},$  (3)

where  $\lambda_1$  and  $\lambda_2$  are nonnegative parameters to control these two components. In the above equation, we decompose the representation matrix Z (which consists of the coefficients of all local patches) into the sum of two components P and Q. Similar to the formulation in (1), we use the  $\ell_{2,1}$  mixed norm on row groups of P such that relevant local patches have similar representations. In addition, we use the group Lasso penalty on column groups of Q to simultaneously identify the outlier patches. Intuitively, if the *i*-th column of Q is nonzero, then the *i*-th column of  $\mathbf{Z}$  is also nonzero. Thus, the *i*-th local patch does not share a common representation with other local patches, and is identified as an outlier. Meanwhile, for the remaining local patches corresponding to the zero columns of Q, they share similar representations indicated by the nonzero rows of P (See an example as shown in Figure 5. As a result, the proposed robust structural sparse appearance model can simultaneously capture the shared representations among relevant local patches and detect outliers, which can effectively deal with the issues with the formulation in (1). To illustrate the the proposed formulation clearly, we show an example of the learned sparse coefficients in Figure 6.

For presentation clarity, we show how the tracking problem with the proposed structural appearance models (1) and (3) can be efficiently solved in the supplementary material.

#### 3.3 Connection to Other Sparse Trackers

As discussed in Section 1 and Figure 1, existing sparse tracking methods [8], [9], [10], [11], [12], [13], [14], [16], [15], [17] are developed based on global, local, and joint sparse appearance models. In this work, we propose two novel structural sparse appearance models as shown in (1) and (3) for visual tracking. Our formulations in (1) and (3) are generic and encompass the above three models with the



Fig. 6. An example of the learned coefficients by the proposed RSST algorithm. In the top figure, we show the learned coefficient matrices  $\mathbf{P}$  and  $\mathbf{Q}$  for all particles as shown in the image. Each matrix consists of 14 column components corresponding to 14 different parts, where brighter color entries represent larger values in the corresponding entry. Three elements (3, 4, 5) in the dictionary  $\mathbf{D}$  are the most representative with larger values in the third, fourth, fifth rows of  $\mathbf{P}$  across all parts. On the other hand, some columns in  $\mathbf{Q}$  have large values which indicate the presence of outliers. The bottom figures illustrate the coefficients of two particles  $\mathbf{x}_i$  and  $\mathbf{x}_j$ .

corresponding properties. It is worthwhile emphasizing the differences between the proposed algorithms (SST and RSST) and related sparse tracking methods [8], [9], [10], [11], [12], [13], [14], [16], [15], [17] as follows:

- Global sparse appearance models for tracking [8], [10], [11], [12], [13]: When K = 1 (only  $1 \times 1$  as shown in Figure 2) and  $\ell_{1,1}$  mixed norm is adopted, the proposed formulation (1) is reduced to a global sparse appearance model which describes an object as one single entity and learns the sparse representations of target candidate regions independently without considering their intrinsic relationships.
- Local sparse appearance models for tracking [9], [15]: In (1), with the image patch sampling methods as [9], [15] and  $\ell_{1,1}$  mixed norm, the proposed formulation (1) is reduced to a local sparse representation model, which does not consider the correlations of image patches among multiple target candidate regions or the spatial layout structure of image patches inside each target candidate region.
- Joint sparse appearance model for tracking [14], [16], [17]: In (1), when K = 1 (only  $1 \times 1$  as shown in Figure 2) with  $\ell_{2,1}$  mixed norm, the proposed formulation (1) is reduced to a joint sparse representation model, which considers the intrinsic relationships among target candidate regions. However, this model uses a holistic object representation.

• Proposed structural sparse appearance models for tracking: The proposed SST tracker has the following properties. (1). It considers both the global and local sparsity constraints. (2). It considers the intrinsic relationships among not only the target candidate regions but also their local image patches. (3). It considers the spatial layout structure of image patches inside each target candidate region. Furthermore, the RSST tracker not only has the above properties but also takes outliers (due to occlusion and noise) into account.

# 4 EXPERIMENTAL RESULTS

We first present experimental results of the proposed SST and RSST algorithms with comparisons to 15 state-of-theart tracking methods on a set of 40 challenging image sequences that are not included in the recent benchmark dataset [5]. We then evaluate the proposed methods on the two recent visual tracking benchmark datasets OTB50 [5] and OTB100 [21] with comparisons to state-of-the-art trackers on 50 image sequences and 100 image sequences, respectively. The tracking results are available on the project websites: http://faculty.ucmerced.edu/mhyang/project/rsst.html and http: //nlpr-web.ia.ac.cn/mmc/homepage/tzzhang/rsst.html. All the source codes and datasets are made available to the public on the same web sites.

#### 4.1 Datasets

For thorough evaluations, we use a set of 40 challenging videos (denoted as OTB40) with ground truth object locations including the tunnel, tud, trellis70, surfer, surfing, sphere, shaking, sunshade, singer, jumping, fernando, football, girl, david indoor (david), faceocc, faceocc2, carchase, car4, car11, biker, bicycle, human, One Leave Shop Reenter1cor (olsr), One Leave Shop Reenter2cor (olsr2), KITTI 0000 (KIT00), KITTI 0004 (KIT04), KITTI 0005 (KIT05), KITTI 0008 (KIT08), KITTI 0010 (KIT10), KITTI 0011 (KIT11), KITTI 0012 (KIT12), KITTI 0016 (KIT16), KITTI 0017 (KIT17), KITTI 0018 (KIT18), MOT ETH-Sunnyday (MOTE), MOT Venice-2 (MOTV), MOT ADL-Rundle-6 (MOT6), MOT ADL-Rundle-8 (MOT8), MOT PETS09-S2L1 (MOTP) sequences. These videos are publicly available online<sup>1</sup> and contain complex scenes with challenging factors for visual tracking, e.g., cluttered backgrounds, moving cameras, fast movements, large variations in pose and scale, occlusions, shape deformations and distortions (See Figure 8). For the second set of experiments, our method is evaluated on the OTB50 [5], OTB100 [21], and VOT2014 [50] datasets. The first two datasets are composed of 50 and 100 sequences, respectively. The images are annotated with ground truth bounding boxes and 11 visual attributes (illumination variation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-view, background clutters, and low resolution) for performance analysis. More detailed descriptions of the OTB50 and OTB100 datasets can be found in [5], [21], [51]. The VOT2014 dataset [50] consists of 25 challenging videos from a set of more than 300 sequences.

# 4.2 Comparison Methods

We compare the proposed tracking RSST algorithm with 15 state-of-the-art methods including the online multiple instance learning (MIL) [32], online Adaboost boosting (OAB) [30], tracking by detection (TLD) [33], Struck [52], circulant structure tracking (CST) [53], part-based visual tracking (PT) [54], real time compressive tracking (RTCT) [13],  $\ell_1$  tracking ( $\ell_1$ T) [12], local sparse tracking (LST) [15], multi-task tracking (MTT) [14], incremental visual tracking (IVT) [26], distribution field tracking (DFT) [55], fragments-based (Frag) [25], multiple experts using entropy minimization (MEEM) [56], and local-global tracking (LGT) [57] schemes. The MIL, OAB, TLD, Struck, CST, MEEM, and PT are discriminative trackers, and the others (IVT, DFT, Frag, LGT, RTCT,  $\ell_1$ T, MTT, and LST) are generative methods. In addition, the RTCT and  $\ell_1$ T, LST, and MTT methods are based on global, local, and joint sparse models, respectively. For fair comparisons, we use the publicly available source or binary codes provided by the authors. In addition, we use the same initialization and parameter settings in all experiments. The details of the 29 trackers in the benchmark evaluation can be found in [5]. For through evaluations, we also compare the proposed tracker with other state-of-the-art methods including MEEM [56], TGPR [58], DSST [36], KCF [35], MUSTer [37], SRDCF [59], and SAMF [60].

#### 4.3 Evaluation Metrics

We use two metrics including the center location error and the overlapping rate for quantitative evaluation of tracking methods. The center location error is the Euclidean distance between the center of the tracking result and the ground truth for each frame. The overlapping rate is based on the PASCAL challenge object detection score [61]. Given the tracked bounding box  $ROI_T$  and the ground truth bounding box  $ROI_{GT}$ , the overlap score is computed by  $score = \frac{area(ROI_T \cap ROI_{GT})}{area(ROI_T \cup ROI_{GT})}$ . To rank the tracking performance, we compute the average center location error and average overlap score across all frames of each image sequence as existing methods[11], [12], [13], [14], [15], [17]. In addition, we plot the precision-recall curve and compute the area under curve (AUC) for each evaluated method on every image sequence, where the success rate is defined with a threshold of 20 pixels center location error and 0.5 for overlap ratio [5]. For the VOT2014 dataset, the performance is measured both in terms of accuracy (average bounding box overlap) and robustness (failure rate).

## 4.4 Implementation Details

For all experiments, we set  $\eta = 10$ ,  $\lambda = 5$ ,  $\lambda_1 = 5$ ,  $\lambda_2 = 1$ , the number of image patches K = 14 as shown in Figure 2, the number of templates m = 20, the number of particles n = 400 (same for  $\ell_1$ T and MTT). The variances of affine parameters for particle sampling are set to

<sup>1.</sup> http://vision.ucsd.edu/~bbabenko/project\_miltrack.html [32];

http://www.cs.toronto.edu/~dross/ivt/ [26];

http://cv.snu.ac.kr/research/~vtd/ [27];

http://www.cvlibs.net/datasets/kitti/eval\_tracking.php [48]; https://motchallenge.net/ [49].

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE

## TABLE 1

Image feature evaluation using area under curve of success plot and precision score (20 pixels threshold) reported on the OTB50 and OTB100 datasets (AUC/PS) corresponding to the one-pass evaluation.

Trackers	RSST-	Color	RSST-	HOG	RSST-Deep			
Metrics	AUC	PS	AUC	PS	AUC	PS		
OTB50 [5]	52.0	69.1	54.3	72.6	59.0	78.9		
OTB100 [21]	49.4	66.5	51.4	69.3	58.3	78.9		

(0.005, 0.0005, 0.0005, 0.005, 4, 4), and updated based on the tracking results. The template size d is set to half the size of the target object manually initialized in the first frame. The likelihood  $p(\mathbf{y}_t|\mathbf{s}_t)$  is computed as the linear combination of the confidence scores of all parts as in (2). All the parameter settings are available in the source code to be released for accessible reproducible research. We implement the proposed RSST algorithm in MATLAB with the MatConvNet toolbox [62] on an Intel 3.10 GHz CPU with 256 GB RAM where the computation of forward propagation on CNNs is carried out on a GeForce GTX Titan X GPU. The RSST algorithm runs at 4 and 1.8 frames per second by using color and deep feature, respectively.

#### 4.5 Image Features

We implement the proposed RSST method with three different features: gray color (RSST-Color) [12], HOG (RSST-HOG) [35], and deep features (RSST-Deep) [38]. For the deep features, we use the same experimental protocols in the CF2 [38] method in which the VGG-Net-19 [63] is used for feature extraction. We use the outputs of the conv5-4 convolutional layer as our features. To reduce the feature dimensionality, we use the principal component analysis and retain the top 1120 dimensions (90% spectrum energy). In the experiments, we use the one-pass evaluation (OPE) criterion in terms of the area under curve of success plot and precision score (20 pixels threshold) on the OTB50 and OTB100 datasets (AUC/PS). Table 1 shows that proposed method with deep features outperform the one with intensity values by 7.0%/9.8%and 8.9%/12.4% in terms of AUC/PS on the OTB50 and OTB100 datasets, respectively. The use of HOG features helps improve tracking performance of the proposed algorithm on these datasets, which is similar to what is observed in the performance gain from the KCF [35] to CST [53] methods.

#### 4.6 Model Analysis

We evaluate the sparse representation schemes of the SST (1), RSST (3), and MTT [14] tracking methods. Here, these methods adopt the gray color feature as in [12], [14]. In (1), when K = 1 (only  $1 \times 1$  as shown in Figure 2) with  $\ell_{2,1}$  mixed norm, the proposed formulation (1) is reduced to a joint sparse representation scheme as the MTT [14] method, which considers the intrinsic relationships among target candidate regions. However, the MTT method uses a holistic object representation. Different from the MTT method, the proposed SST approach considers the spatial layout structure of image



8

Fig. 7. Precision and success plots on the OTB40 Dataset. The legend contains the area-under-the-curve score and the average distance precision score at 20 pixels for each tracker. Our trackers perform favorably against the state-of-the-art trackers.

patches inside each target candidate region, and the intrinsic relationships among not only the target candidate regions but also their local image patches. On the other hand, the RSST model (3) not only has the properties of the SST method but also takes outliers (due to occlusion or noise) into account.

Table 4 shows the experimental evaluations of these methods on the OTB40 and OTB50 datasets. The SST method achieves better tracking performances than the MTT method by 3.3%/3.9% and 10.8%/17.3% in terms of success rate and precision on the OTB40 and OTB50 datasets, respectively. Compared with the SST method, the proposed RSST algorithm obtains 8.0%/8.3% and 3.6%/4.3% improvements in terms of success rate and precision on the OTB40 and OTB50 datasets, respectively. These results demonstrate the effectiveness of each component in the proposed model for visual tracking.

## 4.7 Results on the OTB40 Dataset

We evaluate the proposed algorithms against 15 state-of-theart single object tracking methods on the OTB40 benchmark dataset which contains 40 fully annotated sequences from MOT [49], KITTI [48], IVT [26], VTD [27], and MIL [32]. Table 2 and 3 show the average one pass evaluation results based on center location error and overlap score of 18 trackers. Figure 7 shows the average success and precision plots. Overall, the proposed RSST-Deep and RSST-Color algorithms perform favorably against the state-of-the-art methods on this dataset. For example, the proposed RSST-Color algorithm performs well against the MTT (by 11.3%), MEEM (by (7.6%), SST (by (8.0%)), Struck (by (8.3%)) and LST (by (12.9%)) methods in terms of success rate. In addition, the proposed RSST-Color tracking algorithm performs favorably against the MEEM (by 0.6%), Struck (by 2.1%), PT (by 4.6%), SST (by 8.3%) and MTT (by 12.2%) methods in terms of precision. The proposed RSST-Deep method achieves better tracking performance than the RSST-Color scheme by 4.8% and 8.9%in terms of success rate and precision, respectively. These experimental results show that the proposed methods can track objects undergoing large appearance variations caused by occlusions, pose variations, illumination change and abrupt motions in cluttered backgrounds.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE

9

# TABLE 2

Average center location error of 18 different trackers on 40 different videos. Our trackers perform favorably against the other trackers. For each video, the smallest and second smallest distances are denoted in red and blue, respectively.

Video	RSST-Deep	RSST-Color	SST	RTCT	IVT	MIL	OAB	Frag	Struck	MTT	$\ell_1 T$	TLD	CST	DFT	LST	PT	LGT	MEEM
tunnel	4.5	4.0	4.7	15.6	27.9	53.6	54.1	114.2	9.9	24.9	57.8	20.8	11.0	18.5	6.4	12.1	38.0	8.2
tud	5.8	5.7	6.2	55.1	25.9	51.2	26.2	10.8	17.8	14.3	6.8	16.7	58.0	10.6	42.9	27.4	59.6	16.4
trellis70	6.7	10.0	12.0	42.4	54.0	37.3	41.5	55.7	28.3	10.3	31.1	50.9	6.6	60.1	8.9	43.9	8.5	6.9
surfing	1.8	1.5	1.3	4.4	1.7	4.1	2.1	27.1	1.5	1.4	1.8	4.6	2.3	32.1	1.5	2.0	7.8	2.0
surfer	9.3	11.3	18.9	29.8	75.1	8.4	8.1	186.1	9.2	22.3	28.0	12.5	78.4	139.4	150.0	14.2	53.0	7.3
sphere	4.1	11.0	14.2	35.6	23.5	41.5	18.5	138.2	12.0	23.7	89.3	25.4	6.6	189.4	162.4	14.8	14.2	11.0
shaking	3.1	4.7	11.8	86.6	52.2	7.9	100.2	15.2	54.8	8.4	37.7	21.0	13.6	95.4	3.7	34.7	58.8	7.4
sunshade	4.0	3.6	3.9	19.5	74.3	62.5	7.2	28.1	3.8	53.4	45.7	37.9	4.7	52.7	43.4	8.1	10.8	4.9
singer	1.1	1.4	2.7	5.9	9.8	11.1	63.0	26.9	4.5	1.8	5.3	44.1	6.9	6.6	2.8	5.4	20.4	11.3
jumping	3.7	4.3	4.5	47.4	81.7	7.6	86.7	58.8	5.8	31.9	42.5	4.7	95.1	71.8	68.2	11.5	110.1	4.2
girl	3.0	3.0	3.5	17.4	4.2	12.4	11.0	7.4	18.6	4.5	5.0	8.3	38.2	19.1	3.2	4.1	18.2	5.0
football	4.1	12.8	4.7	123.3	5.2	8.0	53.3	6.3	6.9	4.7	15.4	6.0	7.2	5.2	10.5	7.5	11.9	16.7
fernando	49.9	36.4	37.1	64.0	50.7	61.2	72.9	56.3	68.7	48.9	50.4	65.2	58.2	56.6	69.9	58.0	33.0	56.0
faceocc	6.7	8.2	9.5	19.0	9.1	34.3	17.2	7.9	8.4	7.7	7.0	14.8	4.5	4.7	98.1	7.5	28.8	7.6
faceocc2	9.5	5.0	6.1	24.0	6.5	10.2	20.8	48.2	6.5	8.1	15.2	13.3	6.1	7.2	7.3	6.9	26.8	7.8
david	5.4	7.4	12.0	32.4	13.1	30.3	26.4	73.0	46.7	16.0	16.2	11.3	19.7	15.4	53.5	11.7	15.5	11.3
carchase	2.4	2.6	2.3	19.1	18.5	20.4	3.2	11.1	2.5	10.9	21.7	2.9	2.7	46.3	3.1	7.0	58.8	10.9
car4	2.9	1.6	2.2	86.3	6.4	53.8	88.1	127.3	2.3	2.2	8.5	6.9	18.3	89.6	2.0	2.6	91.3	16.9
car11	1.9	1.9	2.1	117.8	5.4	53.8	5.7	72.7	1.8	1.9	19.2	29.0	2.1	8.1	1.8	1.9	23.9	2.2
biker	89.1	15.8	16.0	16.0	76.8	29.6	22.0	104.4	48.0	17.3	29.4	86.9	18.4	122.6	92.6	27.7	34.3	25.4
bicycle	3.4	4.5	4.8	50.6	61.5	6.7	50.1	120.9	4.8	5.5	73.6	56.7	62.5	72.5	4.9	51.2	50.5	3.8
human	1.3	1.3	1.5	23.3	2.2	2.7	5.3	5.2	4.3	2.0	1.5	72.1	3.0	22.6	1.3	4.6	29.3	1.8
osow	2.4	1.3	1.4	15.2	3.0	11.6	4.6	5.6	4.7	2.5	1.8	11.1	4.6	3.8	1.3	4.9	12.9	3.4
olsr2	4.0	2.9	3.1	56.8	24.0	23.8	12.5	57.6	14.3	4.9	4.7	49.5	17.9	29.6	38.1	13.9	34.5	29.2
olsr1	1.5	1.2	1.2	15.2	3.0	11.6	4.6	5.6	4.7	2.5	1.8	11.1	4.6	3.8	1.3	4.9	12.9	3.4
KIT00	24.8	78.6	80.6	62.0	124.1	52.2	163.9	156.7	47.5	443.1	375.7	192.6	46.7	176.1	124.7	44.6	280.8	268.2
KIT04	3.5	40.6	105.9	284.7	528.3	516.5	330.5	406.0	374.5	519.2	395.2	4.9	42.1	392.5	309.2	315.0	309.1	145.0
KIT05	1.7	2.0	2.4	9.3	2.5	8.8	3.0	3.2	2.0	2.0	2.2	3.1	3.7	3.5	1.3	2.1	5.7	4.2
KIT08	6.2	6.5	197.6	467.5	13.7	34.0	6.1	106.5	16.4	6.4	120.8	43.9	4.4	6.5	1.9	122.9	4.3	6.5
KIT10	6.1	3.9	4.6	14.1	10.4	8.8	7.7	16.1	4.1	2.4	4.0	4.5	6.1	8.1	5.9	4.4	6.3	6.5
KIT11	37.8	72.2	67.4	157.3	30.6	37.4	60.7	159.6	16.6	27.0	154.2	7.3	5299.0	21.6	15.3	15.2	311.4	13.3
KIT12	2.3	5.1	299.1	256.4	304.6	10.7	11.6	302.4	2.7	303.9	304.0	289.6	5.1	304.4	304.9	17.0	23.7	8.7
KIT16	7.5	11.3	27.2	16.0	229.6	18.0	74.4	268.4	12.0	156.4	165.6	260.6	104.6	275.4	243.3	11.3	19.2	10.4
KIT17	7.6	15.4	16.3	15.4	31.5	14.6	29.8	58.0	16.3	44.9	163.9	15.5	48.4	28.7	39.7	17.4	22.0	11.4
KIT18	3.2	3.0	5.8	20.4	4.3	50.5	11.1	13.5	4.2	10.3	11.3	6.5	3.1	9.5	3.4	3.5	13.2	3.6
MOT6	22.5	43.0	43.0	51.3	85.6	62.2	62.2	199.1	47.6	52.8	69.0	219.3	38.1	64.9	60.5	54.0	71.1	48.0
MOT8	9.1	357.1	307.1	352.9	373.3	410.3	410.3	378.3	14.6	542.6	387.2	116.7	403.8	368.5	353.8	14.4	532.5	31.0
MOTE	3.3	61.0	85.9	209.5	8.5	136.4	14.9	193.5	5.4	261.8	219.1	315.4	255.8	11.9	4.5	6.1	31.7	8.0
MOTP	261.8	264.7	278.5	493.0	10.6	355.5	354.4	454.2	6.9	356.0	118.2	178.4	4.0	461.8	482.1	415.5	19.1	261.3
MOTV	35.4	84.8	413.3	84.8	399.2	528.1	528.1	271.2	71.5	496.1	512.9	470.2	402.2	382.0	56.8	505.3	311.2	84.8

Figure 8 shows qualitative comparisons with the 15 trackers on the 40 challenging sequences (Here, due to the space limitation, we only show the results of 18 sequences). The MEEM tracker performs well in the most sequences. However, it drifts away from the target object when partial occlusion occurs at frame 300 in the olsr sequence. In addition, the MEEM tracker does not handle scale variation well in the car4, tunnel, and singer sequences. The Struck method drifts when target objects undergo heavy occlusion bicycle and fast motion (shaking and KIT04). The TLD method does not follow targets well when significant deformation and fast motion occur (trellis70, MOTE, KIT00, and fernando). The CST tracker does not perform well due to varying lighting conditions and background clutter encountered in the Shaking sequence. In addition, it drifts when the target objects undergo heavy occlusions (bicycle), scale variation (MOTE), and outof-view (surfer). The PT scheme does not handle partial occlusion well (bicycle). Furthermore, it is not effective in tracking objects when fast motion (fernando and shaking) and scale variation (tunnel) occur. The MTT approach drifts when object motion is large (jumping). The LST tracker fails in the presence of occlusion (faceocc), fast motion (surfer and *jumping*), heavy occlusion or out-of-view (*bicycle* and *biker*). It is also less effective in tracking objects with deformation (sphere) and scale variation (KIT04 and KIT16). The SST

approach does not perform well in significant deformation (*KIT16* and *MOT8*). The IVT method does not track targets undergoing significant occlusions or out-of-view (*girl*) well, and fails when fast motion (*surfer* and *jumping*), and scale variation (*tunnel* and *MOTV*) occur. Overall, the proposed RSST algorithm performs well in tracking objects on these challenging sequences.

## 4.8 Results on the OTB50 Dataset

We evaluate the proposed tracker on the OTB50 dataset with comparisons to 36 trackers including 29 methods in [5] and 7 recent algorithms (MEEM [56], TGPR [58], DSST [36], KCF [35], MUSTer [37], SRDCF [59], and SAMF [60]) using the source codes. Figure 9 shows the OPE success and precision plots of the top 10 performing tracking methods. Overall, the proposed algorithms perform well against the state-of-the-art methods. It is worth noticing that the proposed RSST methods perform significantly better than other sparse trackers [11], [12], [13], [14], [18], [15].

Among the 29 methods in [5], the SCM and Struck methods perform better than the other trackers. The RSST-Deep algorithm performs better than the SCM and Struck methods by 9.1% and 11.6% in terms of success rate, respectively. Compared with the SCM and Struck methods in terms of

10

## TABLE 3

Average overlap score of 18 different trackers on 40 different videos. On average, our trackers outperform the other 15 trackers. For each video, the biggest and second biggest scores are denoted in red and blue, respectively.

Video	RSST-Deen	RSST-Color	SST	RTCT	IVT	MIL	OAB	Frag	Struck	MTT	l <sub>1</sub> T	TLD	CST	DFT	LST	PT	LGT	MEEM
tunnel	0.69	0.65	0.64	0.29	0.21	0.08	0.09	0.04	0.32	0.23	0.15	0.34	0.32	0.23	0.63	0.31	0.15	0.32
tud	0.84	0.89	0.87	0.32	0.56	0.38	0.56	0.68	0.61	0.67	0.84	0.71	0.36	0.67	0.44	0.57	0.24	0.62
trellis70	0.68	0.64	0.61	0.22	0.39	0.35	0.46	0.29	0.50	0.60	0.38	0.21	0.72	0.32	0.62	0.39	0.63	0.70
surfing	0.70	0.88	0.88	0.78	0.84	0.79	0.82	0.50	0.87	0.84	0.85	0.60	0.79	0.40	0.73	0.82	0.48	0.80
surfer	0.50	0.50	0.34	0.15	0.16	0.57	0.59	0.03	0.56	0.27	0.16	0.41	0.21	0.03	0.04	0.41	0.07	0.62
sphere	0.87	0.72	0.70	0.42	0.54	0.36	0.60	0.08	0.68	0.56	0.18	0.49	0.68	0.06	0.11	0.64	0.66	0.69
singer	0.71	0.82	0.78	0.45	0.48	0.41	0.18	0.26	0.46	0.86	0.70	0.40	0.47	0.47	0.73	0.46	0.29	0.42
shaking	0.71	0.70	0.43	0.02	0.02	0.58	0.01	0.40	0.15	0.55	0.18	0.34	0.44	0.15	0.66	0.29	0.09	0.58
sunshade	0.78	0.75	0.73	0.35	0.26	0.14	0.69	0.38	0.78	0.38	0.39	0.39	0.75	0.35	0.40	0.62	0.54	0.73
jumping	0.72	0.69	0.67	0.05	0.12	0.58	0.04	0.14	0.65	0.12	0.13	0.64	0.05	0.11	0.06	0.48	0.06	0.71
girl	0.75	0.73	0.73	0.32	0.68	0.45	0.53	0.60	0.41	0.71	0.68	0.59	0.35	0.38	0.73	0.71	0.25	0.68
football	0.71	0.56	0.65	0.02	0.64	0.52	0.23	0.59	0.60	0.66	0.45	0.60	0.57	0.68	0.58	0.56	0.35	0.55
fernando	0.29	0.33	0.30	0.31	0.30	0.27	0.28	0.32	0.30	0.30	0.25	0.26	0.31	0.27	0.30	0.30	0.43	0.33
faceocc	0.84	0.85	0.76	0.73	0.84	0.58	0.77	0.87	0.85	0.84	0.86	0.57	0.92	0.91	0.30	0.87	0.57	0.86
faceocc2	0.71	0.78	0.73	0.54	0.79	0.72	0.59	0.38	0.77	0.74	0.67	0.57	0.77	0.78	0.77	0.77	0.46	0.75
david	0.69	0.73	0.60	0.41	0.36	0.42	0.43	0.23	0.38	0.53	0.50	0.60	0.50	0.57	0.45	0.64	0.58	0.65
carchase	0.85	0.86	0.87	0.29	0.44	0.53	0.82	0.60	0.85	0.58	0.59	0.80	0.84	0.40	0.79	0.72	0.31	0.77
car4	0.87	0.89	0.89	0.24	0.74	0.27	0.22	0.23	0.49	0.80	0.62	0.57	0.47	0.23	0.87	0.49	0.15	0.47
car11	0.72	0.78	0.77	0.00	0.51	0.22	0.55	0.10	0.83	0.80	0.52	0.28	0.80	0.52	0.79	0.82	0.43	0.77
biker	0.43	0.67	0.68	0.45	0.31	0.43	0.44	0.27	0.38	0.44	0.39	0.30	0.45	0.27	0.39	0.37	0.42	0.47
bicycle	0.64	0.55	0.59	0.33	0.32	0.54	0.31	0.11	0.40	0.64	0.29	0.39	0.25	0.25	0.54	0.28	0.35	0.51
human	0.71	0.73	0.78	0.33	0.66	0.48	0.54	0.47	0.53	0.65	0.73	0.08	0.53	0.31	0.74	0.49	0.23	0.54
osow	0.88	0.93	0.92	0.56	0.83	0.56	0.71	0.77	0.81	0.89	0.91	0.65	0.81	0.82	0.90	0.80	0.54	0.80
olsr2	0.77	0.82	0.81	0.29	0.44	0.35	0.47	0.27	0.50	0.76	0.78	0.28	0.46	0.40	0.34	0.50	0.29	0.42
olsr1	0.81	0.92	0.86	0.71	0.86	0.67	0.17	0.78	0.77	0.88	0.87	0.68	0.84	0.86	0.87	0.77	0.56	0.80
KIT00	0.46	0.35	0.27	0.32	0.26	0.34	0.24	0.14	0.39	0.20	0.14	0.11	0.39	0.15	0.26	0.39	0.17	0.25
KIT04	0.31	0.13	0.10	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.58	0.12	0.01	0.01	0.01	0.04	0.09
KIT05	0.86	0.77	0.77	0.49	0.62	0.53	0.60	0.63	0.61	0.74	0.73	0.75	0.59	0.59	0.83	0.60	0.59	0.60
KIT08	0.50	0.48	0.01	0.00	0.32	0.05	0.48	0.07	0.34	0.49	0.02	0.17	0.50	0.33	0.76	0.25	0.45	0.46
KIT10	0.76	0.74	0.71	0.54	0.58	0.61	0.61	0.54	0.64	0.71	0.76	0.70	0.61	0.64	0.72	0.64	0.67	0.63
KIT11	0.56	0.29	0.31	0.14	0.44	0.31	0.23	0.16	0.47	0.39	0.16	0.68	0.16	0.41	0.46	0.47	0.16	0.48
KIT12	0.86	0.86	0.49	0.13	0.08	0.77	0.76	0.08	0.90	0.08	0.08	0.10	0.86	0.08	0.08	0.69	0.54	0.81
KIT16	0.61	0.43	0.04	0.47	0.10	0.45	0.41	0.04	0.47	0.30	0.18	0.05	0.34	0.04	0.06	0.43	0.48	0.45
KIT17	0.54	0.43	0.43	0.42	0.41	0.42	0.40	0.38	0.42	0.40	0.18	0.56	0.40	0.41	0.46	0.41	0.39	0.42
KIT18	0.70	0.63	0.42	0.25	0.36	0.04	0.33	0.31	0.33	0.38	0.40	0.61	0.33	0.33	0.61	0.33	0.42	0.33
MOT6	0.72	0.53	0.50	0.42	0.42	0.42	0.42	0.35	0.44	0.49	0.42	0.16	0.43	0.44	0.47	0.44	0.45	0.44
MOT8	0.70	0.04	0.04	0.03	0.03	0.03	0.03	0.02	0.56	0.03	0.03	0.07	0.07	0.03	0.03	0.55	0.06	0.24
MOTE	0.78	0.31	0.21	0.04	0.72	0.29	0.62	0.13	0.69	0.16	0.11	0.06	0.05	0.68	0.83	0.68	0.41	0.69
MOTP	0.19	0.29	0.19	0.03	0.53	0.15	0.16	0.02	0.60	0.15	0.23	0.33	0.64	0.08	0.03	0.03	0.39	0.19
MOTV	0.61	0.32	0.13	0.30	0.11	0.09	0.09	0.20	0.31	0.12	0.11	0.12	0.13	0.11	0.41	0.13	0.32	0.34

# TABLE 4

Model analysis by comparing MTT, SST, and RSST. The area under curve of success plot and prevision score (20 pixels threshold) of these three methods reported on the OTB50 and OTB40 datasets (AUC/PS) corresponding to the one-pass evaluation.

Dataset	MTT [14]	SST [20]	RSST
OTB40	47.9/64.2	51.2/68.1	59.2/76.4
OTB50 [5]	37.6/47.5	48.4/64.8	52.0/69.1

precision, the proposed RSST-Deep algorithm achieves 14% and 13.3% improvements, respectively. Compared with other recent trackers, the proposed RSST-Deep method achieves better performance than the TGPR, DSST, and KCF trackers, and shows comparable results as the SAMF and MEEM methods. The MUSTer and SRDCF methods show slightly better performance than other trackers. When compared with the SST method as shown in Table 4, the proposed RSST algorithm achieves the performance gain of 3.6% and 4.3% in terms of success rate and precision. These experimental results can be attributed to that the proposed method is designed to deal with outliers by considering signals and noise in the proposed RSST-Deep method shows better or comparable

results than some deep learning based methods including GOTURN [41] (44.4%/62.0%), SiamFC [40] (61.2%/81.5%), CNN-SVM [44] (59.7%/85.2%), FCNT [43] (59.9%/85.6%), DLSSVM [64] (58.9%/82.9%) in terms of success rate and precision.

#### 4.9 Results on the OTB100 Dataset

We use the OTB100 dataset to evaluate the proposed RSST algorithms against 36 trackers including 29 methods in [5] and other 7 recent approaches (MEEM [56], TGPR [58], DSST [36], KCF [35], MUSTer [37], SRDCF [59], and SAMF [60]). Figure 10 shows the OPE success and precision plots of the top 10 performing tracking methods.

Overall, the proposed RSST-Deep algorithm performs well against the state-of-the-art tracking methods. Compared with the top-performing SCM and Struck methods in the benchmark study [5], the proposed RSST-Deep tracker achieves performance gains of 13.8% and 11.9% in terms of success rate, and 21.9% and 14.8% in terms of precision, respectively. Among the other 7 state-of-the-art trackers, the proposed MCPF method performs well against the MEEM (by 1.1%/5.0%), TGPR (by 7.4%/8.0%), MUSTer (by 1.5%/0.8%), DSST (by 9.4%/6.1%), KCF (by 9.0%/10.3%), MUSTer (by 1.5%/0.8%), and SAMF (by 3.5%/3.0%)

#### IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE



Fig. 8. Tracking results of 16 trackers (denoted in different colors and lines) on 18 image sequences. Frame indexes are shown in the top left of each figure in yellow color. Results are best viewed on high-resolution displays.



Fig. 9. Precision and success plots of overall performance comparison for the 50 videos on the OTB50 Dataset. The legend contains the area-under-the-curve score and the average distance precision score at 20 pixels for each tracker. Our trackers perform favorably against the state-of-the-art trackers.

schemes in terms of the precision as well as success rate metrics, and shows comparable results as the SRDCF method. Furthermore, the proposed RSST-Deep method achieves better or comparable results than recent tracking methods based on deep features including CF2 [38] (56.2%/83.7%), GO-TURN [41] (42.7%/57.2%), CNN-SVM [44] (55.4%/81.4%), DLSSVM [64] (54.1%/76.7%) in terms of success rate and precision.

We further analyze the tracking performance based on attributes of image sequences [21]. Due to space constraints, we present the success and precision plots of OPE for 8



11

Fig. 10. Precision and success plots of overall performance comparison for the 100 videos on the OTB100 Dataset. The legend contains the area-under-the-curve score and the average distance precision score at 20 pixels for each tracker. Our trackers perform favorably against the state-of-the-art trackers.

attributes in Figure 11 and provide more results in the supplementary material. For presentation clarity, we show the top 10 performing methods in each plot. Overall, the proposed RSST-Deep algorithm performs well in all attribute-based evaluation against the 29 state-of-the-art methods [5]. Compared with the other 7 trackers (MEEM [56], TGPR [58], DSST [36], KCF [35], MUSTer [37], SRDCF [59], and SAMF [60]), the proposed RSST-Deep method achieves comparable results. We note that the proposed RSST-Deep method performs well in dealing with challenging factors including low resolution, scale variation, deformation, in-plane rotation, occlusion,

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE



Fig. 11. Attribute based success plots and precision plots of OPE for the 100 videos in the benchmark [21]. Experimental results on 8 different challenging factors are presented.

background clutters, and out of view.

#### 4.10 Results on the VOT2014 Dataset

We follow the protocol of the VOT2014 [50] where trackers are initialized using the ground truth annotation in the first frame of a video and re-initialized once they drift away from the target. We evaluate our method with 38 trackers in [50] including DSST, SAMF, KCF, DGT, PLT\_14, PLT\_13, eASMS, HMM-TxD, MCT, and ACT with the published results from the VOT2014 website.

Table 5 shows the results of the proposed algorithm with the top 10 methods in the VOT2014 challenge according to the evaluation metrics. The proposed RSST-Deep performs well with robustness of 1.30 and accuracy of 0.62. Among the evaluated methods, the DSST, RSST-Deep, KCF, and SAMF schemes achieve comparable results. The PLT\_13 method achieves the best tracking results in terms of robustness with significant degradation in accuracy. Overall, our tracker performs favorably against the state-of-the-art methods in terms of accuracy and robustness.

#### 4.11 Discussions

Although the proposed RSST algorithm performs well against the state-of-the-art methods on the evaluation datasets, it is

less effective in tracking target objects undergoing drastic deformation (MotorRoling and Ironman) and fast motion (Matrix and Skiing). Most sparse trackers use particle filters [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], and need to use dense sampling schemes to cover target object states. Thus, these sparse trackers are limited in several aspects. First, the computational load is high. As most trackers need to solve one optimization problem for each particle, it is computationally expensive for these methods to locate target objects. Second, it is difficult to sample all possible target object states with particle filters. Different from the sparse trackers with particle filters, the MEEM [56], KCF [35], and DSST [36] use dense sampling schemes for efficient and effective tracking. Third, these sparse methods do not use hierarchical and discriminative features. These sparse trackers use intensity or color features as it is computationally expensive to extract hierarchical and discriminative features. However, feature representations play an important role on tracking performance as discussed in Section 4.5. Fourth, these methods use simple template updates by using the most recent tracking results. A better strategy is to use different target templates by considering shot-term and long-term information, or using multiple templates with entropy minimization [56]. Our future work will focus on developing adaptive representation schemes with high dimensional features to account for large object

12

#### TABLE 5

Comparison with the state-of-the-art methods on the VOT2014 dataset. The results are presented in terms of robustness and accuracy. The proposed RSST-Deep method performs favorably against the state-of-the-art trackers.

	DSST	SAMF	KCF	DGT	PLT_14	PLT_13	eASMS	HMM-TxD	MCT	ACAT	MatFlow	RSST-Deep
Robustness	1.16	1.28	1.32	1.00	0.16	0.08	1.12	1.52	0.99	1.56	0.76	1.30
Accuracy	0.62	0.61	0.62	0.58	0.56	0.55	0.54	0.58	0.54	0.55	0.49	0.61

deformation and motion.

# 5 CONCLUSIONS

In this paper, we propose a novel structural sparse appearance model for object tracking within the particle filter framework, where the representations of target candidate regions and their image patches, regularized by a sparsity-induced  $\ell_{2,1}$  mixed norm, are learned jointly. The proposed appearance model is general and accommodates most existing methods based on sparse representations. Based on the fact that most of the particles are relevant and outliers often exist, we propose the RSST algorithm to not only capture the underlying relationships shared by all local patches as the SST, but also model the outliers due to occlusion or noise. Experimental results with evaluations against the state-of-the-art methods on challenging image sequences demonstrate the effectiveness and robustness of the proposed RSST tracking algorithm.

# REFERENCES

- A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," ACM Computing Surveys, vol. 38, no. 4, p. 13, 2006. 1, 3
- [2] K. Cannons, "A Review of Visual Tracking," York University, Canada, Tech. Rep. CSE-2008-07, 2008. 1
- [3] 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. 1
  [4] T. Zhang, B. Ghanem, and N. Ahuja, "Robust multi-object tracking
- [4] T. Zhang, B. Ghanem, and N. Ahuja, "Robust multi-object tracking via cross-domain contextual information for sports video analysis," in *International Conference on Acoustics, Speech and Signal Processing*, 2012. 1
- [5] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 1, 3, 7, 9, 11
- [6] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual Tracking: An Experimental Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442–1468, 2013. 1
- [7] Y. Pang and H. Ling, "Finding the best from the second bests inhibiting subjective bias in evaluation of visual tracking algorithms," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013. 1
- [8] B. Liu, L. Yang, J. Huang, P. Meer, L. Gong, and C. Kulikowski, "Robust and fast collaborative tracking with two stage sparse optimization," in *Proceedings of European Conference on Computer Vision*, 2010, pp. 1–14. 1, 2, 6, 12
- [9] B. Liu, J. Huang, L. Yang, and C. Kulikowski, "Robust visual tracking with local sparse appearance model and k-selection," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 1, 2, 3, 4, 6, 12
- [10] H. Li, C. Shen, and Q. Shi, "Real-time visual tracking with compressed sensing," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1305–1312. 1, 2, 3, 6, 12
- [11] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai, "Minimum error bounded efficient l<sub>1</sub> tracker with occlusion detection," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1257–1264. 1, 2, 3, 6, 7, 11, 12
- [12] 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. 1, 2, 3, 4, 6, 7, 8, 11, 12

- [13] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Proceedings of European Conference on Computer Vision*, 2012. 1, 2, 6, 7, 11, 12
- [14] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 1, 2, 3, 6, 7, 8, 9, 11, 12
- [15] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 1, 2, 3, 4, 6, 7, 11, 12
- [16] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Low-rank sparse learning for robust visual tracking," in *Proceedings of European Conference on Computer Vision*, 2012. 1, 2, 3, 6, 12
- [17] Z. Hong, X. Mei, D. Prokhorov, and D. Tao, "Tracking via robust multitask multi-view joint sparse representation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013. 1, 2, 3, 6, 7, 12
- [18] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust l<sub>1</sub> tracker using accelerated proximal gradient approach," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 1, 2, 3, 11
- [19] T. Zhang, C. Jia, C. Xu, Y. Ma, and N. Ahuja, "Partial occlusion handling for visual tracking via robust part matching," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 1
- [20] T. Zhang, S. Liu, C. Xu, S. Yan, B. Ghanem, N. Ahuja, and M.-H. Yang, "Structural sparse tracking," in *Proceedings of IEEE Conference* on Computer Vision and Pattern Recognition, 2015. 3, 9
- [21] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, 2015. 3, 7, 11, 12
- [22] 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*, vol. 26, no. 1, pp. 63–84, 1998.
- [23] 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
- [24] A. Jepson, D. Fleet, and T. El-Maraghi, "Robust on-line appearance models for visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1296–1311, 2003. 3
- [25] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 798–805. 3, 7
- [26] 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, 7, 10
- [27] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proceedings* of *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1269–1276. 3, 7, 10
- [28] R. T. Collins and Y. Liu, "On-line selection of discriminative tracking features," in *Proceedings of the IEEE International Conference on Computer Vision*, 2003, pp. 346–352. 3
- [29] S. Avidan, "Ensemble tracking," in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2005, pp. 494–501. 3
- [30] H. Grabner, M. Grabner, and H. Bischof, "Real-Time Tracking via Online Boosting," in *Proceedings of British Machine Vision Conference*, 2006, pp. 1–10. 3, 7
- [31] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Proceedings of European Conference* on Computer Vision, 2008, pp. 234–247. 3
- [32] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 983–990. 3, 7, 10
- [33] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N Learning: Bootstrapping Binary Classifiers by Structural Constraints," in *Proceedings of IEEE* Conference on Computer Vision and Pattern Recognition, 2010. 3, 7

#### IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE

- [34] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof, "PROST: Parallel robust online simple tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, June 2010. 3
- [35] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, 2015. 3, 7, 8, 11, 12
- [36] M. Danelljan, G. Hager, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proceedings of the British Machine Vision Conference*, 2014. 3, 7, 11, 12
- [37] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "Multistore tracker (muster): A cognitive psychology inspired approach to object tracking," in *Proceedings of IEEE Conference on Computer Vision* and Pattern Recognition, 2015, pp. 749–758. 3, 7, 11
- [38] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015. 3, 4, 8, 11
- [39] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M.-H. Yang, "Hedged deep tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 3, 4
- [40] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. Torr, "Fully-convolutional siamese networks for object tracking," in ECCV Workshop, 2016. 3, 11
- [41] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 fps with deep regression networks," in *Proceedings of European Conference on Computer Vision*, 2016. 3, 4, 11
- [42] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 3, 4
- [43] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015. 3, 4, 11
- [44] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *Proceedings of the International Conference on Machine Learning*, 2015. 3, 11
- [45] J. Fan, W. Xu, Y. Wu, and Y. Gong, "Human tracking using convolutional neural networks," *IEEE Transactions on Neural Networks*, vol. 21, no. 10, pp. 1610–1623, 2010. 3
- [46] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in Advances in Neural Information Processing Systems, 2013, pp. 809–817. 3
- [47] H. Li, Y. Li, and F. Porikli, "Deeptrack: Learning discriminative feature representations by convolutional neural networks for visual tracking," in *Proceedings of British Machine Vision Conference*, 2014. 4
- [48] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 7, 10
- [49] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "Motchallenge 2015: Towards a benchmark for multi-target tracking," arXiv:1504.01942 [cs], Apr. 2015, arXiv: 1504.01942. 7, 10
- [50] M. K. et al., "The visual object tracking vot2014 challenge results," in ECCV workshop, 2014. 7, 12
- [51] N. Wang, J. Shi, D.-Y. Yeung, and J. Jia, "Understanding and diagnosing visual tracking systems," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015. 7
- [52] S. Hare, A. Saffari, and P. Torr, "Struck: Structured output tracking with kernels," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011. 7
- [53] J. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proceedings* of European Conference on Computer Vision, 2012. 7, 8
- [54] R. Yao, Q. Shi, C. Shen, Y. Zhang, and A. van den Hengel, "Part-based visual tracking with online latent structural learning," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2013.
- [55] L. Sevilla-Lara and E. Learned-Miller, "Distribution fields for tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1910–1917. 7
- [56] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: robust tracking via multiple experts using entropy minimization," in *Proceedings of European Conference on Computer Vision*, 2014. 7, 11, 12, 13
- [57] 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.

[58] J. Gao, H. Ling, W. Hu, and J. Xing, "Transfer learning based visual tracking with gaussian process regression," in *Proceedings of European Conference on Computer Vision*, 2014. 7, 11

14

- [59] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4310–4318. 7, 11
- [60] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in European Conference on Computer Vision Workshop VOT2014, 2014. 7, 11
- [61] M. Everingham, L. Gool, C. Williams, J. Winn, and A. Zisserman, "The pascal visual object class (voc) challenge," *International Journal* of Computer Vision, vol. 88, no. 2, pp. 303–338, 2010. 7
- [62] A. Vedaldi and K. Lenc, "Matconvne: convolutional neural networks for matlab," in CoRR, 2014, p. abs/1412.4564.
- [63] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015. 8
- [64] J. Ning, J. Yang, S. Jiang, L. Zhang, and M.-H. Yang, "Object tracking via dual linear structured svm and explicit feature map," in *Proceedings* of *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 11



**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 re-

search 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 and Executive Director of China-Singapore Institute of Digital Media. His research interests include multimedia content analysis, pattern recognition, and computer vision. Dr. Xu is an Associate Editor of ACM Transactions on Multimedia Computing, Communications and Applications and IEEE Transactions on Multimedia. He served as Program Chair of ACM Multimedia 2009. He is an

ACM Distinguished Scientist, IEEE Fellow, and IAPR Fellow.



0162-8828 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications\_standards/publications/rights/index.html for more information.

**Ming-Hsuan Yang** is an associate 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.