Robust Distracter-Resistive Tracker via Learning a Multi-Component Discriminative Dictionary

Weichao Shen, Yuwei Wu, Junsong Yuan, *Senior Member, IEEE*, Lingyu Duan, *Member, IEEE*, Jian Zhang, *Senior Member, IEEE*, and Yunde Jia, *Member, IEEE*

***Abstract*— Discriminative dictionary learning (DDL) provides an appealing paradigm for appearance modeling in visual tracking. However, most existing DDL-based trackers cannot handle drastic appearance changes, especially for scenarios with background cluster and/or similar object interference. One reason is that they often suffer from the loss of subtle visual information, which is critical to distinguish an object from distracters. In this paper, we explore the use of activations from the convolutional layer of a convolutional neural network to improve the object representation and then propose a robust distracter-resistive tracker via learning a multi-component discriminative dictionary. The proposed method exploits both the intra-class and inter-class visual information to learn shared atoms and the class-specif c atoms. By imposing several constraints into the objective function, the learned dictionary is reconstructive, compressive, and discriminative, and thus can better distinguish an object from the background. In addition, our convolutional features have structural information for object localization and balance the discriminative power and semantic information of the object. Tracking is carried out within a Bayesian inference framework where a joint decision measure is used to construct the observation model. To alleviate the drift problem, the reliable tracking results obtained online are accumulated to update the dictionary. Both the qualitative and quantitative results on the CVPR2013 benchmark, the VOT2015 data set, and the SPOT data set demonstrate that our tracker achieves substantially better overall performance against the state-of-the-art approaches.**

***Index Terms*— Visual tracking, multi-component discriminative dictionary, appearance changes, multi-object tracking.**

Manuscript received January 22, 2017; revised September 16, 2017; accepted July 20, 2018. Date of publication August 1, 2018; date of current version July 1, 2019. This work was supported in part by the Natural Science Foundation of China under Grant 61702037, in part by the Beijing Municipal Natural Science Foundation under Grant L172027, and in part by the Beijing Institute of Technology Research Fund Program for Young Scholars. This paper was recommended by Associate Editor S. Gong. *(Corresponding author: Yuwei Wu.)*

W. Shen, Y. Wu, and Y. Jia are with the Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 100081, China (e-mail: shenweichao@bit.edu.cn; wuyuwei@bit.edu.cn; jiayunde@bit.edu.cn).

J. Yuan is with the Department of Computer Science and Engineering, The State University of New York, Buffalo, NY 14260-2500 USA (e-mail: jsyuan@buffalo.edu).

L. Duan is with the Institute of Digital Media, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China (e-mail: lingyu@pku.edu.cn).

J. Zhang is with the Global Big Data Technologies Centre, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia (e-mail: jian.zhang@uts.edu.au).

Color versions of one or more of the f gures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifie 10.1109/TCSVT.2018.2862151

I. Introduction

A GOOD appearance model is one of the most critical prerequisites for successful visual tracking. Designing an effective appearance model is still a challenging task due to appearance variations. Efforts dedicated to this issue have leaded to numerous tracking algorithms [1], [2], which can be roughly categorized as either generative [3], [4] or discriminative [5]–[7] approaches. Generative methods build an object representation, and then search for a region most similar to the object. Discriminative methods online train a binary classif er to adaptively separate an object from the background, thereby being more robust against appearance variations of an object.

Recently, the discriminative dictionary learning (DDL) provides an appealing paradigm for appearance modeling due to its superior discrimination power. Considering visual tracking as a binary classificatio problem, however, most DDL based trackers [8]–[10] have diff culties in discriminating the similar visual patterns, especially for objects sharing similar shape and/or visual appearances with *distracters*. Distracters induced by background clutters, illumination changes, partial occlusions, or surrounding crowds, are generally diff cult to handle. In this paper, we focus on handling such scenarios with distracters using a novel discriminative dictionary learning method.

A single object tracking example is shown in the f rst row of Fig.1, where the differences between the object and the background are very subtle. Numerous DDL based trackers often fail to distinguish the differences successfully, and thus the most likely object location is incorrectly determined. The reason can be explained as follows. Since the object and the background are visually similar, the learnt dictionary is likely to be governed by common patterns. Candidates from different classes (*i.e.,* the object and the background) may be encoded by same atoms, as illustrated in the purple rectangle of Fig.1. As a result, representations of candidates could share many similar codes and the proportion of discriminative codes may be very small. Such a property causes the potential loss of subtle image information that is critical to differentiate an object from the similar background. Therefore, for a robust DDL based tracker, what is desired is a new dictionary learning method which can encode subtle visual differences between an object and distracters, especially for cluttered environments or similar object interference scenarios.

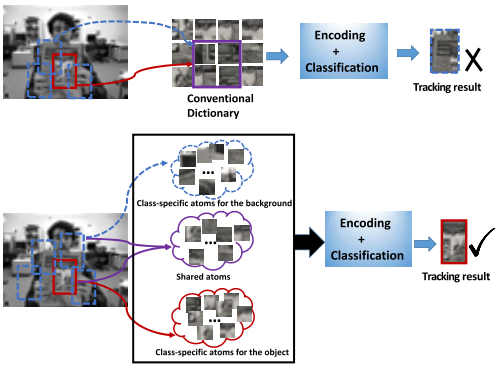

Fig. 1. Comparisons between our tracker and traditional sparse trackers. Row 1: Most DDL based trackers are likely to encode subtly discriminative objects into the same atoms, resulting in the object drift. Row 2: Our method can jointly learn shared atoms and the class-specif c atoms. The learnt dictionary is compact and discriminative, which better distinguish target objects from the background. Red solid rectangles denote the true object and blue dashed rectangles are candidates.

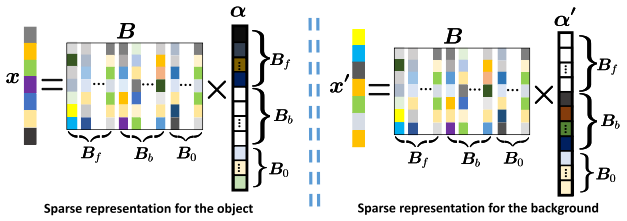

Fig. 2. The learnt dictionary $\boldsymbol{B}$ and its corresponding sparse codes. Taking the single object tracking as an example, $\boldsymbol{B}_f$, $\boldsymbol{B}_b$, and $\boldsymbol{B}_0$ denote the class-specifi atoms for the object, the class-specifi atoms for the background, and the shared atoms, respectively. $\boldsymbol{x}$ and $\boldsymbol{x}'$ represent feature vectors of the object and the background, respectively. For instance, an object candidate should be well represented by its corresponding class-specifi atoms and the shared ones as much as possible. Darker color elements have larger values in sparse codes.

The representation ability of the original hand-crafted features, *e.g.,* HOG [11] and Color Names [12], is another important factor resulting in the struggle performance especially when the target-background similarity is high. Driven by the emergence of large-scale data sets and fast development of computation power, features based on convolutional neural networks (CNNs) have proven to perform remarkably well on a wide range of visual recognition tasks [13]–[15]. Different from hand-crafted features, pre-trained on a large dataset with massive classes, the CNNs contains a great deal of prior knowledge including rich high-level semantic information and powerful inter-class discriminative information, effectively distinguishing the object of interest from the background. Recent study [16], [17] has also shown that local image regions correspond to receptive f elds of the particular features, *i.e.*, convolutional features have structural information for object localization and balance the discriminative power and semantic information of the object [14], [18].

Above observations inspire us to design a new distracter-resistive tracker by learning a multi-component discriminative dictionary appropriately using the convolutional features. The multi-component dictionary consists of class-specifi atoms and shared atoms by concurrently exploiting the intra-class visual information and inter-class visual correlations. In our work, the class-specifi atoms are able to capture the most discriminative feature between an object and distracters. However, the class-specifi atoms usually share some common patterns because of inter-class visual correlations, which may make the learnt dictionary redundant. We should effectively discover the common patterns from class-specifi ones. The shared atoms are mainly used to reconstruct common patterns among all samples, which contribute to the representation of the data rather than discrimination. To enhance the discrimination power of the dictionary, the classificati n error and several discriminative constraints are incorporated into the objective function. In this way, the learnt dictionary is more compact and discriminative, thus can better discriminate an object from distracters. On the other hand, the powerful capabilities of learning representation of the convolutional features is utilized to improve the expressive and discriminative ability of learnt dictionary. This work is, to the best of our knowledge, the f rst to explore the convolutional features to model DDL for visual tracking. The reason may be that, for an input image, usually the limited number of CNNs features extracted from the labeled samples are available, the dimensions of which are inherently much higher, making robust dictionary learning diff cult. We introduce reasonable usage of the convolutional features for our dictionary learning process.

As discussed in [19], a sample should be well represented by its corresponding class-specif c atoms and the shared ones. Considering again the example illustrated in Fig. 1, given an object candidate, ideally, only the coefficient associated with $\boldsymbol{B}_f$ and $\boldsymbol{B}_0$ will be non-zero, as shown in Fig. 2. Tracking is then carried out within a Bayesian inference framework where a joint decision measure is used to construct the observation model. The quality of each candidate is measured by the global coding classif er and the learnt linear classif er instead of relying on only one of them. The candidate with the highest measure score is considered as the f nal tracking result. To alleviate the drift problem, both the precise information of the f rst frame and reliable tracking results obtained online are accumulated to update the dictionary. Compared with existing DDL based visual tracking methods, the proposed multi-component discriminative dictionary method is able to encode subtle visual differences between an object and distracters. The learnt dictionary, therefore, can represent an object well and discriminate an object from distracters simultaneously. Employing a new joint decision measure to construct the observation model, can effectively evaluate how a candidate is resembling the object to improve the tracking performance. Both the qualitative and quantitative results on the CVPR2013 tracking benchmark [20], the VOT2015 dataset [21] and the SPOT dataset [22] demonstrate the superior performance of the proposed approach compared with several state-of-the-art trackers.

II. Related Work

A. Discriminative Dictionary Learning

Dictionary learning (DL) has received considerable attention in signal processing and computer vision community with

a wide range of applications [23]–[25]. Early works learn dictionary in a unsupervised way, which attempt to find a dictionary which linearly reconstruct input signals with a small reconstruction error. However, the unsupervised DL methods often lack of discriminative ability as they are only optimal for reconstruction but not classification. In this paper, we pay more attention to discriminative dictionary learning.

Discriminative DL can be roughly categorized into three categories for visual analysis. In the first category, a *shared dictionary* is learnt for all classes [26], [27]. These methods attempt to enhance the discrimination power of the dictionary by either forcing the sparse coefficients to be more discriminative or promoting the incoherence of dictionary atoms. Nevertheless, the *shared dictionary* does not take the correspondence between the dictionary atoms and the class labels into account. Many works [24], [28] have advocated learning *class-specific dictionaries* whose atoms correspond to the class labels. Since the representation coefficients are not enforced to be discriminative, however, the classification decision solely depend on the reconstruction errors in the class-specific dictionary learning methods. *Hybrid dictionary* has also been proposed to learn a shared dictionary and a set of class-specific dictionaries [19], [29]. In these methods, however, the class-specific dictionaries between different classes usually share some coherent or even the same atoms (*i.e.,* common patterns). These common patterns shared by the visually correlated classes, however, do not contribute to the discrimination of the object, but may even degrade the classification accuracy. Therefore, how to explicitly discover the shared visual atoms from the class-specific dictionaries in the hybrid dictionary is still a challenging task.

Zhou and Fan [19] and Yang *et al.* [28] exploited both the reconstruction errors and classification errors to learn the discriminative dictionary. However, since they took no consideration of the inter-class nor intra-class orthogonality constraints, the redundancy of the dictionary could degrade the performance of the classification. Furthermore, Zhou and Fan [19] ignored the representation ability of the global dictionary, and Yang *et al.* [28] did not explicitly learn a shared dictionary. Though Gao *et al.* [29] and Zheng and Jiang [30] learned the class-specific and shared dictionaries simultaneously, they did not force the sparse coefficient to be discriminative. Gao *et al.* [29] neglected the discriminative fidelity constraint which can ensure that samples of each class can be favorably reconstructed by their corresponding class-specific dictionary. And the intra-class orthogonality constraint is not imposed on each class-specific dictionary in [30], which may result in many atoms being zeros in the class-specific dictionary, but our model can avoid this. Overall, our learnt dictionary is reconstructive, compressive and discriminative, and even better distinguishes the subtle and minute differences among different classes.

B. Tracking With DDL

Sparse representation has been shown to give promising results against object appearance variations for visual tracking. The pioneer work introduced by Mei and Ling [31] models the object appearance as a sparse linear combination of both object templates and trivial templates via ℓ_1 minimization. However, the computational cost of [31] grows linearly with the number of candidates, resulting in high computation burden. To obtain more efficien solutions, many accelerated algorithms have been employed under the framework of ℓ_1 tracker, including multi-task sparse learning [32], Circulant Sparse Tracker [33], bounded particle resampling [34], random projection based dimensionality reduction [10], and accelerated proximal gradient [35].

Although these methods [32], [34], [35] are effective in modeling the object appearance, the observation likelihood measured by the reconstruction error under the generative framework is neither efficie t nor robust.

To tackle this issue, several sparse trackers not only propose new sparse models but also introduce construction schemes of the observation likelihood. For example, Wang *et al.* [36] replaced the target templates with online updated PCA basis vectors, and exploited advantages of both the subspace and sparse representation to deal with the partial occlusion when determining the best candidate. Jia *et al.* [3] introduced an alignment-pooling method across local patches to improve the accuracy of location estimation. The coefficients after pooling are summed to sort the candidates. Mei *et al.* [37] used multiple types of visual features and presented an approximate least absolute deviation (LAD)-based multitask multiview sparse learning method for visual tracking. Zhong *et al.* [38] presented a collaborative appearance model in which candidates are evaluated based on the collaboration of generative and discriminative modules. However, these two modules are independent and combined in a heuristic way. In comparison, we propose a joint decision measure to determine the most likely object location. The quality of each candidate is measured by both the global coding classifier and the learnt linear classifie . More details are discussed in Section IV.

The aforementioned methods achieve promising tracking results. However, since most formulations do not take the background information into account, they are less effective for tracking in cluttered environments due to the lack of discrimination power. Liu *et al.* [39] adopted histograms of sparse coefficients and the mean-shift algorithm to construct a local sparse appearance model for object tracking. Wang *et al.* [9] exploited joint optimization of representation and classification by minimizing the least-squares reconstruction error and discriminative penalties with regularized constraints. Hong *et al.* [40] proposed a distracter-resistant tracking approach by integrating the dualforce metric learning and the ℓ_1 minimization framework in the original image space. In [33], a circulant sparse tracker CST) was proposed by Zhang *et al.* to exploited circulant target templates. This method reduced particles using circular shifts and is solved efficiently in the Fourier domain. Although the work in [9], [39], and [40] considered the background information during dictionary construction, the fixed dictionary makes them lack the ability to adapt to appearance changes.

Most methods updated the dictionary by simply using either newly obtained tracking results [3], [38], [41] or candidates [5] as dictionary atoms, without consideration of the dictionary

learning. Wang *et al.* [42] introduced an online tracking algorithm based on local sparse representation and classifie learning. Wang *et al.* [43] formulated object templates updating as a robust non-negative dictionary learning problem and proposed a novel visual tracking method. Though [42] and [43] used dictionary learning methods for online visual tracking, the dictionary and the classifie are learned separately rather than jointly. Different from methods in [9] and [42], our tracker incorporates the classif cation error into the objective function, thus allowing a linear classif er and a good dictionary being considered simultaneously.

C. Tracking With Deep Features

The strong expressive ability of DNN feature is under explored in visual tracking. Wang and Yeung [44] trained a stacked denoising autoencoder (SDAE) for online tracking process. Wang *et al.* [14] analyzed the properties of different convolutional layers, and proposed a feature map selecting method for visual tracking. Ma *et al.* [18] learned a set of linear correlation f lters on different feature maps come from different convolutional layers. The target location was inferred using correlation response maps. Nam and Han [13] inserted the tracking framework into Multi-Domain Network (MDNet) trained on a large set of videos with tracking groundtruths. Qi *et al.* [45] designed a set of CNN trackers based on different CNN layers. Then all these trackers was hedged into a stronger one by an adaptive Hedge method. In this paper, a new deep feature generation method is proposed to extract the CNN features which are more suitable for our dictionary learning process.

Our tracker also differs from the closely related works [8]. Yang *et al.* [8] presented an online discriminative dictionary learning algorithm for visual tracking, which learns a sparse dictionary and a linear classif er simultaneously. Compared with [8], our dictionary learning method can jointly learn the shared atoms and the class-specifi atoms by imposing the inter-incoherence constraint and the intra-incoherence constraint on the objective function. Such a dictionary can characterize the discriminative information between an object and the background, especially when the object appearance bears some similarity with the background objects. More importantly, our tracker can be naturally extended to track multiple objects by treating each object as an individual class, and achieves the good performance for multi-object tracking [46]–[49]. Thus it can be expected that our algorithm can be extended to other visual applications, such as Person Re-Identif cation [50]–[52].

III. Multi-Component Discriminative Dictionary Learning

Note that we use $\boldsymbol{x}$ and $\mathbf{X}$ to denote a vector and a matrix, respectively. Let $\mathbf{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N\} \in \mathbb{R}^{d\times N}$ be a set of the N samples, $\mathbf{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_K] \in \mathbb{R}^{d\times K}$ be the dictionary where each column represents an atom, and $\mathbf{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_N] \in \mathbb{R}^{K\times N}$ be a coding matrix of $\mathbf{X}$ over $\mathbf{B}$. The goal of dictionary learning is to f nd a dictionary, such that each sample can be linearly reconstructed by a relatively small subset of atoms, while keeping the reconstruction error as small as possible, given by

$$\begin{aligned} &\arg\min_{\mathbf{B},\mathbf{A}} \sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{B}\boldsymbol{a}_i\|_2^2 + \|\boldsymbol{a}_i\|_1 \\ &s.t.\ \|\boldsymbol{b}_k\|_2 \le 1, \quad for\ \forall k = 1, 2, \cdots, K. \end{aligned} \tag{1}$$

However, object representation obtained by Eq. (1) often lacks of discriminative ability as it is only optimal for reconstruction but not classif cation [5], [34], [35], [38]. In this paper, we propose a novel multi-component discriminative dictionary learning method to make the learnt dictionary not only reconstructive and compressive, but also discriminative. In addition, in our model the representation coeff cient $\mathbf{A}$ is more discriminative by coupling classif er parameters.

A. Deep Feature Extractor

Before def ning our dictionary learning problem, we f rstly introduce our deep feature generation method. As analyzed in [14], different convolutional layers encode different property of samples. Higher layers capture semantic class information while lower layers encode more discriminative details to distinguish the intra class variations. When it comes to tracking problem, the target and background samples always share a lot of semantic information, *e.g.,* both the whole face (target) and half of it (background) will be classif ed into face class using VGG net trained for classif cation. If we use the feature of higher layer to describe the samples, the high semantic similarity will reduce the distance between positive and negative samples, which result in the learnt dictionaries lacking discriminative ability. On the other hand, the feature extracted by lower layer lacks the robust ability to the appearance variations (*e.g.,* illumination variation, deformation), which increases the intra-class differences. As a tradeoff between this two consideration, we use the conv4-3 layers of VGG-16 to achieve a satisfactory balance.

The dimension of feature vector is too high if we use the outputs of Conv4-3 layers of VGG-16 directly (512 feature maps whose size are 28*28). The critical issue is to reduce the dimension to improve computational eff ciency without losing too much discrimination. A straightforward method is performing pooling operation on each feature map (max or average) to generate a 512 dimensional feature vector. However, it cannot model the difference between target and background well because this method loses too much partial location information (As illustration in Fig. 3). Another method is subsampling each feature map to a smaller size using a uniform distribution and concatenating them to a new feature vector. The features generated with bigger subsampling size contain more location information. Fig. 3 shows that the features containing more location information are more discriminative. As a tradeoff between computational efficien y and feature expressiveness, we subsampled each feature map to 5×5 using uniform distribution and concatenate all of them as our new feature vector. To make our feature more robust to target variation(*e.g.,* illumination variation), we normalize the feature vector using two-norm normalization.

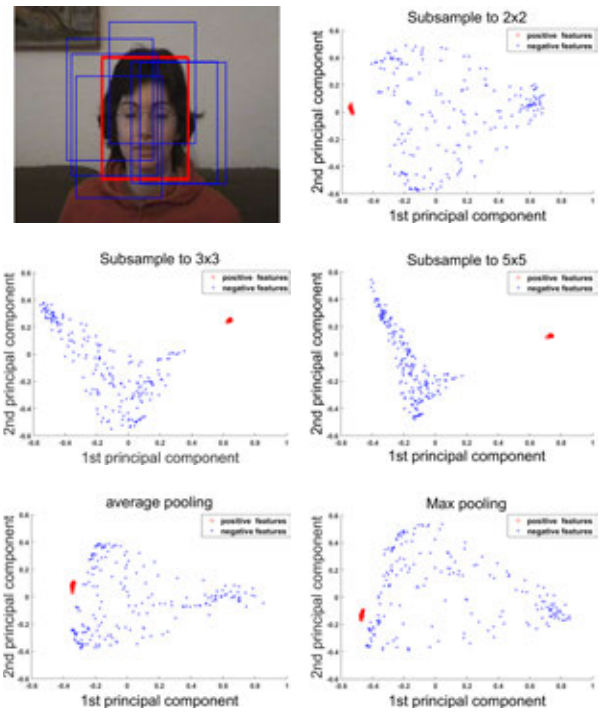

Fig. 3. The first figure shows how we get positive (red rect) and negative (blue rect) samples from image. From top to bottom, left to right, we display the effect of different operations applied on feature maps(VGG16 conv4-3) successively including subsample to 2×2, 3×3, 5×5, average pooling and max pooling. The coordinate axis are the first two principal components of features. We can see that subsample the feature map to 5×5 achieves better discriminative ability.

Finally, the feature dimension is reduced again using PCA method to farther accelerate the dictionary learning process.

B. Problem Description

Given C classes of individual samples, we rewrite the training set as $\mathbf{X} = [\mathbf{X}_1, \cdots, \mathbf{X}_c, \cdots, \mathbf{X}_C] \in \mathbb{R}^{d \times N}$, where $\mathbf{X}_c \in \mathbb{R}^{d \times N_c}$ includes samples of each class and N_c is the number of samples from the c-th class. The class-specific atoms are denoted by $\mathbf{B}_c \in \mathbb{R}^{d \times K_c}$ with $c = 1, 2, \cdots, C$, and the shared atoms are denoted by $\mathbf{B}_0 \in \mathbb{R}^{d \times K_0}$, respectively. K_0 and K_c are the number of atoms from the shared atoms and c-th class-specific dictionary, respectively. $\mathbf{B}_c$ is responsible for describing class-specific visual properties of each class. $\mathbf{B}_0$ is used to describe commonly shared visual patterns for all classes. Thus the complete dictionary is $\mathbf{B} = [\mathbf{B}_0, \mathbf{B}_1, \cdots, \mathbf{B}_c, \cdots, \mathbf{B}_C] \in \mathbb{R}^{d \times K}$, where K is the number of atoms with $K = \sum_{c=0}^{C} K_c$. $\mathbf{A}$ denotes a sparse coefficient matrix of $\mathbf{X}$ over $\mathbf{B}$, *i.e.,* $\mathbf{X} \approx \mathbf{BA}$. Here, $\mathbf{A}$ can be expressed as $\mathbf{A} = [\mathbf{A}_1, \cdots, \mathbf{A}_c, \cdots, \mathbf{A}_C] \in \mathbb{R}^{K \times N}$, where $\mathbf{A}_c \in \mathbb{R}^{K \times N_c}$ is a coefficient matrix of $\mathbf{X}_c$ over $\mathbf{B}$.

In this paper, our goal is to learn a novel multi-component discriminative dictionary which can encode subtle visual differences between an object and distracters. We emphasize that the learnt dictionary should be reconstructive, compressive, and discriminative. Meanwhile, the proposed method allows a linear classifier and a good dictionary being considered simultaneously. Our discriminative dictionary model with ℓ_1 regularization is formulated as

$$\begin{aligned} \langle \mathbf{A}^*, \mathbf{B}^*, \mathbf{W}^* \rangle = \arg\min_{\mathbf{B},\mathbf{A},\mathbf{W}} \Big\{ & \mathcal{C}(\mathbf{X}, \mathbf{A}, \mathbf{B}) \\ & + \mathcal{L}(\mathbf{A}; \mathbf{W}) + \eta ||\mathbf{A}||_1 \Big\} \\ s.t.\ & \mathcal{Q}(\mathbf{A}, \mathbf{B}). \end{aligned} \tag{2}$$

Here, η is a scalar parameter which involves the sparsity of the coefficients. $\mathcal{C}(\cdot)$ is the data fidelity term which aims to get a reconstructive, compressive and discriminative dictionary $\mathbf{B}$. In this paper, our goal is to learn a novel multi-component discriminative dictionary which can encode subtle visual differences between an object and background. As we all known, visual tracking can be treated as a problem of binary classification, and thus C is set to 2, *i.e.*, $\mathbf{B} = [\mathbf{B}_0, \mathbf{B}_1, \cdots, \mathbf{B}_2]$, where $\mathbf{B}_1$, $\mathbf{B}_2$ and $\mathbf{B}_0$ denote the class-specific dictionary for the object, the class-specific dictionary for the background and the shared dictionary, respectively. $\mathcal{L}(\mathbf{A}; \mathbf{W})$ is the discrimination coefficient term which can jointly obtain the discriminative coefficients and classification parameter $\mathbf{W}$. This term can make the coding coefficients more discriminative and propagate indirectly the discrimination power of the coefficients to the dictionary. $\mathcal{Q}(\mathbf{A}, \mathbf{B})$ is the constraint term imposed on the coefficient matrix $\mathbf{A}$ and the dictionary $\mathbf{B}$, which makes $\mathbf{B}$ have not only powerful capability to represent $\mathbf{X}$ (*i.e.,* $\mathbf{X} \approx \mathbf{BA}$), but also discriminative power to distinguish the sample from $\mathbf{X}$. In what follows, we will elaborate each term in Eq.(2).

C. Discrimination Dictionary Term $\mathcal{C}(\mathbf{X}, \mathbf{A}, \mathbf{B})$

The first consideration is that the learned dictionary $\mathbf{B}$ should well represent $\mathbf{X}_c$, *i.e.,* $\mathbf{X}_c \approx \mathbf{BA}_c$. For each class samples, the representation $\mathbf{A}_c$ can be rewritten as $\mathbf{A}_c = \left[\mathbf{A}_c^0; \mathbf{A}_c^1; \cdots; \mathbf{A}_c^c; \cdots; \mathbf{A}_c^C\right] \in \mathbb{R}^{K \times N_c}$, where $\mathbf{A}_c^0 \in \mathbb{R}^{K_0 \times N_c}$ is the coding coefficients of $\mathbf{X}_c$ over the shared dictionary $\mathbf{B}_0$, and $\mathbf{A}_c^c \in \mathbb{R}^{K_c \times N_c}$ is the coding coefficient of $\mathbf{X}_c$ over the sub-dictionary $\mathbf{B}_c$. So there exists $\mathbf{X}_c \approx \mathbf{BA}_c = \sum_{i=0}^{C} \mathbf{B}_i \mathbf{A}_c^i$. In addition, ideally each class sample $\mathbf{X}_c$ should be well represented by $\mathbf{B}_0$ and $\mathbf{B}_c$. That is, only the coefficient associated with $\mathbf{B}_0$ and $\mathbf{B}_c$ can be non-zero, such that $\mathbf{X}_c \approx \mathbf{B}_0\mathbf{A}_c^0 + \mathbf{B}_c\mathbf{A}_c^c$. Mathematically, the data fidelity term is formulated as

$$\begin{aligned} & \mathcal{C}(\mathbf{X}, \mathbf{A}, \mathbf{B}) \\ & = \sum_{c=1}^{C} \mathcal{C}(\mathbf{X}_c, \mathbf{A}_c, \mathbf{B}, \mathbf{B}_0, \mathbf{B}_c) \\ & = \sum_{c=1}^{C} \left(\left\| \mathbf{X}_c - \mathbf{BA}_c \right\|_F^2 + \left\| \mathbf{X}_c - [\mathbf{B}_0, \mathbf{B}_c][\mathbf{A}_c^0; \mathbf{A}_c^c] \right\|_F^2 \right). \end{aligned} \tag{3}$$

The first term is able to guarantee the good representation power of the overall dictionary, and the second term ensures that samples of each class can be favorably reconstructed by $\mathbf{B}_0$ and $\mathbf{B}_c$. Obviously, only using the first term is impractical to learn the discriminative class-specific bases. While only adopting the second term is impossible to obtain an optimal shared dictionary $\mathbf{B}_0$, as there exist some commonly shared visual bases between $\mathbf{B}_0$ and $\mathbf{B}_c$. In what follows,

we will introduce three constraints imposed on the dictionary to enhance its discrimination power.

(1) To make the dictionary more discriminative, it is desired that each class-specific dictionary has poor representation ability for other classes, *i.e.,* $\mathbf{A}_c^j$ should have nearly zero coefficients such that $\sum_{j\neq c, j=1}^{C} \left\|\mathbf{B}_j\mathbf{A}_c^j\right\|_F^2$ is as small as possible. In this case, we consider a strong constraint

$$\min \sum_{j\neq c, j=1}^{C} \|\mathbf{A}_c^j\|_F^2. \tag{4}$$

That is, for samples of the c-th class, it amounts to forcing the coefficients to be zero except the ones corresponding to both the c-th class-specific bases and the shared ones.

(2) However, Eq. (4) does not mean that the learnt individual dictionary bases only contain the visual properties of its corresponding class. The commonly shared visual bases may appear in the different individual dictionaries, which makes the individual dictionary bases redundant and thereby resulting in poor performance [53]. Based on the theoretical analysis in [54], the mutual coherence among all class-specific bases and the shared bases should be as small as possible. Inspired by the incoherence penalty term in [55], the inter-nonredundancy constraint is expressed as

$$\min \sum_{j\neq c, j=0}^{C} \left\|\mathbf{B}_c^\top\mathbf{B}_j\right\|_F^2. \tag{5}$$

Note that j starts with $j = 0$, which means we also consider the nonredundancy of shared bases with all class-specific ones. Clearly, the sub-dictionary among the class-specific bases and the shared ones are nonredundant, and in this case we regard the learnt dictionary $\mathbf{B}$ as most incoherent.

(3) Assume that the dictionary bases are normalized, the mutual coherence is defined as the largest absolute and normalized inner product between different columns in $\mathbf{B}$, *i.e.,*

$$\mu(\mathbf{B}) = \max_{i\neq j} \frac{\left|\boldsymbol{b}_i^\top \boldsymbol{b}_j\right|}{\|\boldsymbol{b}_i\|_2 \cdot \|\boldsymbol{b}_j\|_2}. \tag{6}$$

Furthermore, we can understand the mutual coherence by the Gram matrix $\mathbf{G} = \mathbf{B}^\top\mathbf{B}$. The matrix ℓ_2-norm $\rho(\mathbf{G}) = \|\mathbf{G}\|_2 = \max\{|\delta| : \delta \in \delta(\mathbf{G})\}$, where $\delta(\mathbf{G})$ is the set of eigenvalues of $\mathbf{G}$. Therefore, $\rho(\mathbf{G}) = |\delta_{max}|$, where $|\delta_{max}|$ denote the eigenvalue with largest absolute value. By the Gershgorin Circle Theorem [56], we have

$$\begin{aligned} &\delta(\mathbf{G}) \subseteq \bigcup_{k=1}^{K} \mathcal{G}_k, \\ &s.t. \quad \mathcal{G}_k = \left\{z : |z - g_{kk}| \leq \sum_{r\neq k} |g_{kr}|\right\} \end{aligned} \tag{7}$$

where g_{kr} is an entry of $\mathbf{G}$. The off-diagonal entries in $\mathbf{G}$ are the inner products in Eq. (6). Since the dictionary bases are normalized, $g_{kk} = 1$ and $\sum_{r\neq k} |g_{kr}| \leq \mu(\mathbf{B})$. Substituting these into the Gershgorin Circle Theorem, we have $\rho(\mathbf{G}) = |\delta_{max}| \leq 1 + \mu(\mathbf{B})$. To guarantee the performance of sparse coding, therefore, minimizing the mutual incoherence is equivalent to simultaneously reducing $\delta(\mathbf{G})$ and $\mu(\mathbf{B}$ by imposing small off-diagonal entities in $\mathbf{G}$, which yields an intra-nonredundancy constraint given by

$$\min \left\|\mathbf{B}_c^\top\mathbf{B}_c - \mathbf{I}_{K_c}\right\|_F^2. \tag{8}$$

This constraint is to encourage low mutual coherence and Gram matrix norm of the learned dictionary. Moreover, It makes the learnt class-specifi dictionary more stable, which benefits to improve reconstruction accuracy. Without this constraint, many bases in the class-specific dictionary will be zeros.

We add the terms introduced in Eq. (4), Eq. (5), and Eq. (8) into Eq. (3), leading to a complete data fidelity term $\mathcal{C}(\mathbf{X}, \mathbf{A}, \mathbf{B})$ given by

$$\begin{aligned} &\sum_{c=1}^{C} \mathcal{C}(\mathbf{X}_c, \mathbf{A}_c, \mathbf{B}, \mathbf{B}_0, \mathbf{B}_c) \\ &= \sum_{c=1}^{C} \Big(\left\|\mathbf{X}_c - \mathbf{B}\mathbf{A}_c\right\|_F^2 \\ &\qquad + \left\|\mathbf{X}_c - [\mathbf{B}_0, \mathbf{B}_c][\mathbf{A}_c^0; \mathbf{A}_c^c]\right\|_F^2 + \left\|\mathbf{A}_c^{/0,c}\right\|_F^2 \\ &\qquad + \lambda_1 \left\|\mathbf{B}_c^\top\mathbf{B}_{/c}\right\|_F^2 + \lambda_2 \left\|\mathbf{B}_c^\top\mathbf{B}_c - \mathbf{I}_{K_c}\right\|_F^2 \Big). \end{aligned} \tag{9}$$

Here λ_1 and λ_2 are the trade-off parameters between constraints. $\mathbf{A}_c^{/0,c}$ indicates the submatrix by removing $\mathbf{A}_c^0$ and $\mathbf{A}_c^c$ from $\mathbf{A}_c$, *i.e.,* $\mathbf{A}_c^{/0,c} = \left[\mathbf{A}_c^1; \cdots; \mathbf{A}_c^{c-1}; \mathbf{A}_c^{c+1} \cdots; \mathbf{A}_c^C\right] \in \mathbb{R}^{(K-K_0-K_c)\times N_c}$. $\mathbf{B}_{/c}$ is the submatrix by removing $\mathbf{B}_c$ from $\mathbf{B}$, *i.e.,* $\mathbf{B}_{/c} = [\mathbf{B}_0, \mathbf{B}_1, \cdots, \mathbf{B}_{c-1}, \mathbf{B}_{c+1}, \cdots, \mathbf{B}_C] \in \mathbb{R}^{d\times(K-K_c)}$.

Eq.(9) seems to be incremental, but it is not a trivial combination. The first two terms guarantee the good representation power of the learnt dictionary, and they can discover the hidden patterns shared by the visually correlated classes. The third term makes the sparse coefficient more discriminative, *i.e.,* each class-specifi dictionary has poor representation ability for other classes. More importantly, the inter-nonredundancy and the intra-nonredundancy constraints are considered in the last two terms. These two constraints are meaningful, and make the learnt dictionary non-redundant and more stable.

D. Discrimination Coefficient Term $\mathcal{L}(\mathbf{A}; \mathbf{W})$

Following [19] and [57], to further enhance the discrimination power of the dictionary $\mathbf{B}$, we can force the sparse coefficient $\mathbf{A}$ to be discriminative and indirectly propagate the discrimination power to $\mathbf{B}$. $\mathcal{L}(\mathbf{A}; \mathbf{W})$ aims to incorporate the classification error into the objective function to make coefficients $\mathbf{A}$ more discriminative and reliable for classification. Here, we use a simple linear regression model $f(\mathbf{A}; \mathbf{W}) = \mathbf{W}\mathbf{A}$ to obtain more discriminative coefficients. $\mathcal{L}(\mathbf{A}; \mathbf{W})$ is given by

$$\mathcal{L}(\mathbf{A}; \mathbf{W}) = \left\|\mathbf{H} - \mathbf{W}\mathbf{A}\right\|_F^2 + \beta\left\|\mathbf{W}\right\|_F^2, \tag{10}$$

where β is a trade-off parameter controlling the relative contribution of the corresponding terms. $\left\|\mathbf{H} - \mathbf{W}\mathbf{A}\right\|_F^2$ represents the classification error. $\mathbf{W} \in \mathbb{R}^{C\times K}$ denotes the classifier parameters. $\mathbf{H} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_N\} \in \mathbb{R}^{C\times N}$ is the class labels of $\mathbf{X}$, where class label vector $\boldsymbol{h}_i = [0, 0, \cdots, 1, \cdots, 0, 0]^\top \in$

$\mathbb{R}^C$. For each sample $\boldsymbol{x}_i \in \mathbb{R}^d$, if $\boldsymbol{x}_i$ belongs to the c-th class ($1 \le c \le C$), the c-th entry in $\boldsymbol{h}_i$ is 1 and all the other entries are 0's. $\mathcal{L}(\mathbf{A}; \mathbf{W})$ not only couples the process of learning classifie , but also generates discriminative sparse coefficient . The discriminative property of $\mathbf{A}$ is very important for the performance of the linear classifie [57].

E. The Complete Model

Empirically, if the dictionary $\mathbf{B}$ and the classifie are learnt separately, it might make $\mathbf{B}$ suboptimal for classif cation. An intuitive way is to jointly learn the dictionary $\mathbf{B}$, sparse coefficient $\mathbf{A}$, and classifie parameters $\mathbf{W}$. Plugging Eq.(9) and Eq.(10) into Eq.(2) and choosing ℓ_1-norm as the sparsity constraint, the fi al objective function for the multi-class discriminative dictionary learning model with nonredundancy constraints becomes

$$\begin{aligned}\langle \mathbf{A}_{\mathbf{c}}^*, \mathbf{B}_{\mathbf{0}}^*, \mathbf{B}_{\mathbf{c}}^*, \mathbf{W}^* \rangle &= \arg \min_{\mathbf{A}_c, \mathbf{B}_0, \mathbf{B}_c, \mathbf{W}} \Bigg\{ \sum_{c=1}^{C} \mathcal{C}(\mathbf{X}_c, \mathbf{A}_c, \mathbf{B}, \mathbf{B}_0, \mathbf{B}_c) \\ &\quad + \mathcal{L}(\mathbf{A}; \mathbf{W}) + \eta \sum_{c=1}^{C} ||\mathbf{A}_c||_1 \Bigg\}. \end{aligned} \tag{11}$$

From Eq. (11), we can see that the dictionary $\mathbf{B}$ learnt by the proposed model is reconstructive, compressive and discriminative. Therefore, the samples from c-class will have small reconstruction errors combined with the shared dictionary $\mathbf{B}_0$ and class-specifi dictionary $\mathbf{B}_c$ but have large reconstruction errors with other class-specif c bases. And the representation coefficien $\mathbf{A}$ is more discriminative by coupling the classif er parameters.

F. Optimization

The objective function in Eq. (2) is not jointly convex concerning all variables $\mathbf{A}_c, \mathbf{B}_0, \mathbf{B}_c, \mathbf{W}$, however, it is convex with respect to each variable when others are f xed. Therefore, the optimization procedure of our DDL model can be divided into four sub-procedures by solving $\mathbf{A}_c$, $\mathbf{B}_0$, $\mathbf{B}_c$, and $\mathbf{W}$ alternatively. The alternative procedure is iteratively implemented to fin the local optimum of each variable.

1) Update of $\boldsymbol{A_c}$: Suppose that all other variables f xed except $\mathbf{A}_c$ in our objective function, then Eq. (2) is reduced to a sparse coding problem. We compute sparse coeff cients class by class. Mathematically, $\mathbf{A}_c$ is updated by f xing $\mathbf{A}_j, j \ne c$, and the objective function is given by

$$\begin{aligned}\mathbf{A}_c^* = \arg \min_{\mathbf{A}_c} \Big\{ &\left\| \mathbf{X}_c - [\mathbf{B}_0, \mathbf{B}_c][\mathbf{A}_c^0; \mathbf{A}_c^c] \right\|_F^2 \\ &+ \left\| \mathbf{X}_c - \mathbf{B}\mathbf{A}_c \right\|_F^2 + \left\| \mathbf{A}_c^{/0,c} \right\|_F^2 \\ &+ \left\| \boldsymbol{h}_c - \boldsymbol{w}_c \mathbf{A} \right\|_F^2 + \eta ||\mathbf{A}_c||_1 \Big\}, \end{aligned} \tag{12}$$

where $\boldsymbol{h}_c \in \mathbb{R}^{1 \times N_c}$ and $\boldsymbol{w}_c \in \mathbb{R}^{1 \times K}$ correspond to each row of $\mathbf{H}$ and $\mathbf{W}$, respectively. Many efficien algorithms have been developed to solve Eq. (12). In this work, we adopt the feature-sign search algorithm [58] due to its global convergence.

2) Update of $\boldsymbol{B_c}$: Considering $\mathbf{A}$ is f xed, we f rst update the class-specif c atoms class by class and then update the shared atoms. In this section, we take the c-th dictionary as an example to describe the optimization of $\mathbf{B}_c$. We arrive at the objective function of $\mathbf{B}_c$ by fixin the shared atoms $\mathbf{B}_0$ and all other class-specifi atoms $\mathbf{B}_j$, $j \ne c$, given by

$$\begin{aligned}\mathbf{B}_c^* = \arg \min_{\mathbf{B}_c} \Bigg\{ &\left\| \mathbf{X}_c - \sum_{j=0, j \ne c}^{C} \mathbf{B}_j \mathbf{A}_c^j - \mathbf{B}_c \mathbf{A}_c^c \right\|_F^2 \\ &+ \left\| \mathbf{X}_c - \mathbf{B}_0 \mathbf{A}_c^0 - \mathbf{B}_c \mathbf{A}_c^c \right\|_F^2 + \lambda_1 \left\| \mathbf{B}_c^\top \mathbf{B}_{/c} \right\|_F^2 \\ &+ \lambda_2 \left\| \mathbf{B}_c^\top \mathbf{B}_c - \mathbf{I}_{K_c} \right\|_F^2 \Bigg\}, \end{aligned} \tag{13}$$

Following [55], we use a stochastic gradient descent algorithm to optimize Eq. (13).

3) Update of $\boldsymbol{B_0}$: Different from the class-specifi dictionary $\mathbf{B}_c$, the shared atoms $\mathbf{B}_0$ concentrates on the representation of all samples from all classes. After the class-specifi atoms $\{\mathbf{B}_c\}_{c=1}^C$ are updated, we further update atoms of the shared atoms $\mathbf{B}_0$ by solving the following objective function:

$$\begin{aligned}\mathbf{B}_0^* = \arg \min_{\mathbf{B}_c} \sum_{c=1}^{C} \Bigg\{ &\left\| \mathbf{X}_c - \sum_{j=1}^{C} \mathbf{B}_j \mathbf{A}_c^j - \mathbf{B}_0 \mathbf{A}_c^0 \right\|_F^2 \\ &+ \left\| \mathbf{X}_c - \mathbf{B}_c \mathbf{A}_c^c - \mathbf{B}_0 \mathbf{A}_c^0 \right\|_F^2 \Bigg\} + \lambda_1 \left\| \mathbf{B}_0^\top \mathbf{B}_{/0} \right\|_F^2 \\ &+ \lambda_2 \left\| \mathbf{B}_0^\top \mathbf{B}_0 - \mathbf{I}_{K_0} \right\|_F^2, \end{aligned} \tag{14}$$

where $\mathbf{B}_{/0}$ is a submatrix by removing $\mathbf{B}_0$ from $\mathbf{B}$, *i.e.,* $\mathbf{B}_{/0} = [\mathbf{B}_1, \mathbf{B}_2, \cdots, \mathbf{B}_C] \in \mathbb{R}^{d \times (K - K_0)}$. Following [55], we also use a stochastic gradient descent algorithm to optimize the shared atoms $\mathbf{B}_0$.

4) Update of $\boldsymbol{W}$: Given updated $\mathbf{A}$ and $\mathbf{B}$, the objective function of $\mathbf{W}$ is given by

$$\mathbf{W}^* = \arg \min_{\mathbf{W}} \left\| \mathbf{H} - \mathbf{W}\mathbf{A} \right\|_F^2 + \beta \left\| \mathbf{W} \right\|_F^2. \tag{15}$$

This ridge regression model can be directly solved by setting the partial derivatives *w.r.t.* $\mathbf{W}$ to zero. It yields the global optimal solution $\mathbf{W}^* = \mathbf{H}\mathbf{A}^\top \left(\mathbf{A}\mathbf{A}^\top + \beta \mathbf{I}\right)^{-1}$.

The overall optimization procedure of our model is summarized in Algorithm 1.

IV. Our Tracker

In this section, with the joint discriminative dictionary learning method introduced in Section III, we propose a robust distracter-resistive tracker based on Bayesian inference where a joint decision measure is used to construct the observation model. In our tracker, the candidate with the highest measure score is considered as the tracking result. Both the ground truth information of the fi st frame and the reliable tracking results obtained online are accumulated to update the dictionary, which is effective to alleviate the drift problem. The tracking framework is shown in Fig. 4. The detailed description of the proposed tracking method is summarized in Algorithm 2.

Algorithm 1 Multi-Component Discriminative Dictionaries Learning

Input: Trainning dataset $[\mathbf{X}_1, \cdots, \mathbf{X}_c, \cdots, \mathbf{X}_C] \in \mathbb{R}^{d\times N}$; the size of each class-specifi dictionary K_c; the size of the shared atoms K_0; the trade-off parameters λ_1, λ_2, β, and η.

Output: The class-specifi atoms $\{\mathbf{B}_c\}_{c=1}^{C}$, and the shared atoms $\mathbf{B}_0$.

Initialization: Compute each $\mathbf{B}_c^{(0)}$ with $\mathbf{X}_c$ using K-SVD, and initialize the shared atoms $\mathbf{B}_0^{(0)}$ with $[\mathbf{X}_1, \cdots, \mathbf{X}_c, \cdots, \mathbf{X}_C]$ using K-SVD, such that $\mathbf{B}^{(0)} = \left[\mathbf{B}_0^{(0)}, \mathbf{B}_1^{(0)}, \cdots, \mathbf{B}_C^{(0)}\right]$. Initialize the classifie parameters $\mathbf{W}^{(0)}$.

while *stopping criteria is not reached* **do**
  for $i = 1 \to C$ **do**
    Update the coeff cient matrix $\mathbf{A}_c$ by solving the sparse coding problem Eq.(12);
  end
  for $i = 1 \to C$ **do**
    Update each class-specifi dictionary $\mathbf{B}_c$ by solving Eq.(13);
  end
  Update the shared atoms $\mathbf{B}_0$ by solving Eq.(14);
  Update the classifie parameters $\mathbf{W}$.
end

Algorithm 2 The Proposed Tracking Algorithm

Input: Image frames $F_1, F_2, \cdots, F_n$; Object state $\boldsymbol{s}_1$.

Output: Tracking results $\widehat{\boldsymbol{s}}_t$ at time t.

for $t = 1 \to n$ **do**
  if $t == 1$ **then**
    Obtain labeled samples set $\mathbf{X_1} = \mathbf{X}_{N_p} \bigcup \mathbf{X}_{N_n}$;
    Initialize the sample pool $\mathbf{X}_P = \mathbf{X}_1$;
    Initialize the sample buffer pool $\mathbf{X}' = \emptyset$;
    Initialize $\mathbf{B}^{(0)}$ and $\mathbf{W}^{(0)}$ with $\mathbf{X}_P$.
  end
  ① Sample the object candidates $\widehat{\mathbf{X}}$ according to the motion model $p(\boldsymbol{s}_t|\boldsymbol{s}_{t-1})$;
  ② Compute the classificatio score of each candidate using Eq. (18) and get the best candidate based on Eq. (17);
  ③ Collect training samples set $\widetilde{\mathbf{X}}$ in the current frame and let $\mathbf{X}' = [\mathbf{X}'; \widetilde{\mathbf{X}}]$;
  if $mod(t, T) == 0$ **then**
    Update $\mathbf{X}_P$ with $\mathbf{X}'$;
    if $length(\mathbf{X}_P) > \Theta(\mathbf{X}_P)$ **then**
      randomly remove some samples from $\mathbf{X}_P$.
    end
    Update dictionaries $\mathbf{B}$;
    $\mathbf{X}' = \emptyset$.
  end
  ④ $\widehat{\mathbf{X}} = \emptyset$.
end

A. Bayesian State Inference

Object tracking can be considered as a Bayesian inference task in a Markov model with hidden state variables. Given the observation set of the object $\mathcal{O}_{1:t} = \{\boldsymbol{o_1}, \boldsymbol{o_2}, \cdots, \boldsymbol{o_t}\}$, the optimal state $\boldsymbol{s_t}$ of the tracked object is obtained by the maximum a posteriori estimation $p(\boldsymbol{s}_t^i|\mathcal{O}_{1:t})$, where $\boldsymbol{s}_t^i$ indicates the state of the i-th sample. The posterior probability $p(\boldsymbol{s}_t|\mathcal{O}_{1:t})$ is formulated by Bayes theorem as

$$p(\boldsymbol{s}_t|\mathcal{O}_{1:t}) \propto p(\boldsymbol{o}_t|\boldsymbol{s}_t) \int p(\boldsymbol{s}_t|\boldsymbol{s}_{t-1}) p(\boldsymbol{s}_{t-1}|\mathcal{O}_{1:t-1})\, d\boldsymbol{s}_{t-1}. \tag{16}$$

This inference is governed by the dynamic model $p(\boldsymbol{s}_t|\boldsymbol{s}_{t-1})$ which models the temporal correlation of the tracking results in consecutive frames, and by the observation model $p(\boldsymbol{o}_t|\boldsymbol{s}_t)$ which estimates the likelihood of observing $\boldsymbol{o_t}$ at state $\boldsymbol{s_t}$.

With particle filte ing, the posterior $p(\boldsymbol{s}_t|\mathcal{O}_{1:t})$ is approximated by a f nite set of N_s samples or particles $\{\boldsymbol{s}_t^i\}_{i=1}^{N_s}$ with importance weights $\{\omega_t^i\}_{i=1}^{N_s}$. The particle sample $\boldsymbol{s}_t^i$ is drawn from an importance distribution $q(\boldsymbol{s}_t|\boldsymbol{s}_{1:t-1}, \mathcal{O}_{1:t})$, which for simplicity is set to the dynamic model $p(\boldsymbol{s}_t|\boldsymbol{s}_{t-1})$. The importance weight ω_t^i of particle i is equal to the observation likelihood $p(\boldsymbol{o}_t|\boldsymbol{s}_t^i)$. We apply an aff ne image warp to model the object motion between two consecutive frames. Let $\boldsymbol{s}_t = \{x_t, y_t, \theta_t, s_t, \eta_t, \psi_t\}$, where x_t, y_t, θ_t, s_t, η_t, ψ_t denote x, y translations, rotation angle, scale, aspect ratio and skew at time t, respectively. The dynamic model $p(\boldsymbol{s}_t|\boldsymbol{s}_{t-1})$ is modeled by Gaussian distribution, *i.e.*, $p(\boldsymbol{s}_t|\boldsymbol{s}_{t-1}) = \mathcal{N}(\boldsymbol{s}_t; \boldsymbol{s}_{t-1}, \sum)$, where $\sum$ is a diagonal covariance matrix whose diagonal elements are the corresponding variances of respective parameters. The observation model $p(\boldsymbol{o}_t|\boldsymbol{s}_t)$ is define as

$$p(\boldsymbol{o}_t|\boldsymbol{s}_t) \propto SC_t, \tag{17}$$

where $SC_t = \kappa(\boldsymbol{x}^{(t)})$ is the classificati n decision score at time t which will be explained in the next section.

B. Classifica ion Decision

Given a candidate $\boldsymbol{x} \in \mathbb{R}^{d\times 1}$, we encode it over the learnt dictionary $\mathbf{B} = [\mathbf{B}_0, \mathbf{B}_1, \cdots, \mathbf{B}_c, \cdots, \mathbf{B}_C]$, and obtain the sparse code $\boldsymbol{v} = \arg\min_{\boldsymbol{v}} \|\boldsymbol{x} - \mathbf{B}\boldsymbol{v}\|_2^2 + \eta\|\boldsymbol{v}\|_1$, where $\boldsymbol{v} \in \mathbb{R}^{K\times 1}$. The candidate can be better represented by its corresponding dictionary $\mathbf{B}_c$ and the shared atoms $\mathbf{B}_0$, then its reconstruction error is $\varepsilon_f = \|\boldsymbol{x} - \mathbf{B}_0\boldsymbol{v}_0 - \mathbf{B}_c\boldsymbol{v}_c\|_2^2$, where $\boldsymbol{v}_0 = \left[v_0^1, v_0^2, \cdots, v_0^{K_0}\right]^\top \in \mathbb{R}^{K_0\times 1}$ and $\boldsymbol{v}_c = \left[v_c^1, v_c^2, \cdots, v_c^{K_c}\right]^\top \in \mathbb{R}^{K_c\times 1}$ are the sparse coefficient over $\mathbf{B}_0$ and $\mathbf{B}_c$, respectively. Meanwhile, the candidate should be poor represented by other class-specif c dictionaries and the corresponding reconstruction error is $\varepsilon_b = \|\boldsymbol{x} - \sum_{j=1, j\neq c} \mathbf{B}_j\boldsymbol{v}_j\|_2^2$. For instance, in the case of the single object tracking, the candidate with a smaller foreground error and larger background error is more likely to be the target object, and vice versa. Thus, the global coding classifie is formulated as $f_g = \exp\left(-(\varepsilon_f - \varepsilon_b)/\sigma\right)$, where σ is a constant. To enhance the classificatio accuracy, the linear predictive classifie $f_c = \mathbf{W}_c\boldsymbol{v}$ is jointly used to evaluate how a candidate is resembling the target object. The joint decision

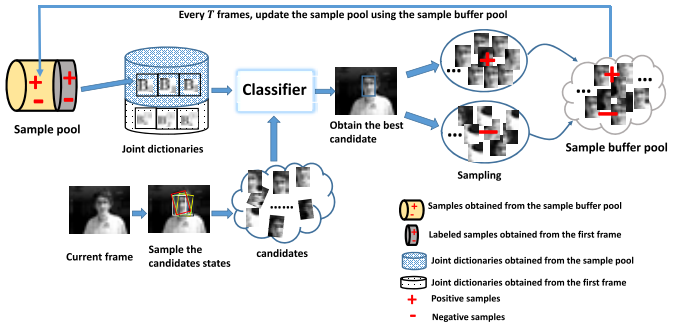

Fig. 4. Tracking framework. We take the single object tracking as an example. For each new frame, the candidate with the highest classif cation score is considered as the tracking result. Every T frames, the sample pool is updated by the sample buffer pool. After updating, we will empty the sample buffer pool and then reconfigur it. Then the updated sample pool is used to update the dictionaries.

measure is define as

$$\kappa(\boldsymbol{x}) = \mathbf{W}_c \boldsymbol{v} + \exp\big(-(\varepsilon_f - \varepsilon_b)/\sigma\big). \tag{18}$$

For each class candidates, the index corresponding to the largest element of $\kappa(\boldsymbol{x})$ is considered as the tracking result. Using the measure Eq.(18), we have a more reliable decision score for the candidate $\boldsymbol{x}$.

C. Initialization

In the f rst frame, we draw positive and negative samples around the object location to initialize the dictionaries. Suppose the object is labeled manually, perturbation (e.g., shifting 1 or 2 pixels) around the object is performed for collecting N_p positive samples $\mathbf{X}_{N_p}$. Similarly, N_n negative samples $\mathbf{X}_{N_n}$ are collected far away from the located object (e.g., within an annular region a few pixels away from the object). $\mathbf{X_1} = \mathbf{X}_{N_p} \bigcup \mathbf{X}_{N_n}$ is the initialized labeled sample set. Using labeled samples, we can obtain the initialized dictionary via the proposed DDL method.

D. Updating the Dictionary

For each new frame, candidates predicted by the particle f lter are denoted by $\widehat{\mathbf{X}}$. According to Eq. (18), we can get the classificatio score of each candidate. A candidate with higher classificati n score indicates that it is more likely to be generated from the target class. The most likely candidate is considered as the tracking result for this frame. Then, perturbation (*i.e.*, the same scheme in the f rst frame) around the tracking result is performed for collecting the sample set $\widetilde{\mathbf{X}}$.

To make our tracker more adaptive to appearance changes, we construct a *sample pool* $\mathbf{X}_P$ and a *sample buffer pool* $\mathbf{X}'$ to update samples and dictionaries, as shown in Fig. 4. We keep a set of T previous $\boldsymbol{X}_C$ to constitute the sample buffer pool $\boldsymbol{X}'$, *i.e.*, $\boldsymbol{X}' = [\boldsymbol{X}_{C-T+1}; \boldsymbol{X}_{C-T+2}; \cdots; \boldsymbol{X}_C]$, where $\boldsymbol{X}_C$ denotes the sample set collected from the current frame. Every T frames, $\mathbf{X}'$ is utilized to update $\mathbf{X}_P$. After updating the sample pool, we will empty $\mathbf{X}'$ and then reconf gure it. In our experiment, we set the sample pool capacity $\Theta(\mathbf{X}_P)$.[1]

[1]The cardinality $\Theta(\mathbf{X}_P)$ denotes the number of samples in the sample pool.

If the total number of samples in the sample pool is larger than $\Theta(\mathbf{X}_P)$, some samples in $\mathbf{X}_P$ will be randomly replaced with samples in $\mathbf{X}'$. To reduce the risk of visual drift, we always retain the samples $\mathbf{X}_1$ obtained from the f rst frame in the sample pool. That is, $\mathbf{X}_P = [\mathbf{X}_1; \mathbf{X}']$, which is able to better characterize the samples distribution. Then the updated sample pool $\Theta(\mathbf{X}_P)$ is utilized to update the dictionaries with our DDL method described in Section III. Meanwhile, we retain the dictionary obtained in the f rst frame for constructing the joint dictionaries to compute the classificati n decision.

V. Experiments

In this section, we concentrate on single object tracking on the benchmark dataset [20] including 51 challenging image sequences. We also show that our tracker can be easily generalized to track multiple objects by treating each object as an individual class. Six challenging sequences[2] are used to illustrate the good performance of the proposed tracker for tracking multiple objects. For the single object tracking, we evaluate the proposed tracker against 9 state-of-the-art tracking algorithms including HDT [45], HCFT [18], FCNT [14], CNN-SVM [59], EBT [60], MEEM [61], DLSSVM [62], KCF [11], AEST [63]. We also test our hand-crafted feature based tracker to evaluate the effectiveness of our dictionary learning model. We evaluate this tracker against 16 state-of-the-art hand-craft based tracking algorithms including ONNDL [43], RET [64], CT [65], MLSAM [6], ODDL [8], CN [12], VTD [66], MIL [67], SCM [38], Struck [68], TLD [69], ASLA [3], LSST [4], MTT [32], LSK [39], and LSPT [7]. In terms of multi-object tracking, we compare the performance of our tracker with OAB [70], TLD [69], and SPOT [22] trackers.

Our approach is implemented in native Matlab. The experiments are performed on an Intel Core2 2.5 GHz processor with 16GB RAM. The computation of forward propagation on VGG-16 is run by MatConvNet toolbox [71] and transferred to a GeForce GTX TITAN Black. The Matlab source code and experimental results are available at http://iitlab.bit.edu.cn/mcislab/~wuyuwei/publication.html (The password of unzipping is *Trans_for_reviewers*).

A. Experimental Setup

The number of particles is 200 and the state transition matrix is $[10, 10, 0.015, 0, 0.005, 0]$ in the particle filte . For each sample we extract deep feature using the same parameters setting as section III-A. In the fir t frame, $N_p = 50$ positive samples and $N_n = 200$ negative samples are used to initialize the dictionary. Once the tracked object is located, 10 positive samples and 80 negative samples are utilized for the dictionary updating. The sample pool capacity $\Theta(\boldsymbol{X}_P)$ is set to 1200, in which the numbers of positive and negative samples are 200 and 1000, respectively. The sizes of each class-specif c dictionary and the shared atoms are 10 and 5, respectively. The constraints parameters are set to $\lambda_1 = 0.1$ and $\lambda_2 = 0.01$. The parameter of the linear regression model is set as $\beta = 1e-10$. The dictionary $\mathbf{B}$ is updated every $T = 10$ frames. Moreover, $\eta = 0.15$, $\sigma = 0.02$.

[2]http://visionlab.tudelft.nl/spot

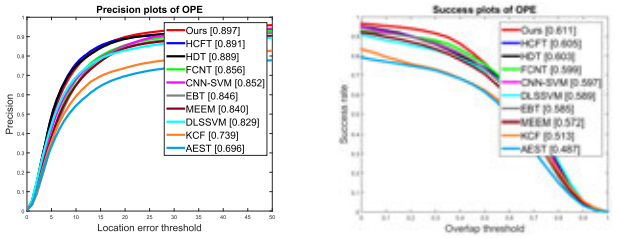

Fig. 5. Overall performance comparisons of OPE in precision plots and success plots. The performance score for each tracker is shown in the legend.

B. Evaluation Criteria

To measure the accuracy of the tracking results, we use the center location error (CLE), the overlapping rate (OR), and the success rate (SR) for quantitative evaluations. The CLE is based on the relative position errors (in pixels) between the central locations of the tracked object and those of the ground truth. Ideally, an optimal tracker is expected to have a small error. The OR is def ned by $score = \frac{area(ROI_T \bigcap ROI_G)}{area(ROI_T \bigcup ROI_G)}$, where ROI_T is the tracking bounding box and ROI_G is the ground truth. If $score$ is larger than 0.5 in one frame, the tracking result is considered as a success. This can be used to evaluate the SR of any tracking approach.

In this paper, the precision plot (PP) and the success plot (SP) are also adopted to measure the overall tracking performance. PP shows the percentage of frames whose estimated location is within the given threshold distance (*e.g.*, 20 pixels) of the ground truth. More accurate trackers have higher precision at lower thresholds. If a tracker loses the object, it is difficul to reach a higher precision. In the SP, we count the number of successful frames as the thresholds vary from 0 to 1 and plot the SP curve for our tracker and the compared trackers. The area under curve (AUC) of each success rate plot is employed to rank the tracking algorithms. More robust trackers have higher success rates at higher thresholds.

C. Experiment 1: Evaluation on the OTB50 Dataset

1) Overall Performance: Followed by [20], one pass evaluation (OPE) is also employed to evaluate the overall performance for 10 trackers on 51 sequence. The OPE curve of both precision plots and success plots are shown in Fig. 5. For precision plots, we use the results at error threshold of 20 pixels for ranking these 10 trackers. The AUC score for each tracker is shown in the legend. Our tracker is 0.6% above the HCFT in the success rate, and outperforms the HCFT by 0.6% in the precision plot. Overall, our tracker outperforms other 9 trackers both in precision plots and success rates.

2) Attribute-Based Performance: Apart from summarizing the performance on the whole sequences, we also construct 11 subsets corresponding to distinctive attributes to test the tracking performance under specifi challenging conditions. Due to the restriction of paper length, We only show the attribute-based performance analysis in precision plots in Fig. 6. Our approach performs favorably on 4 out of 11 attributes: illumination variation (IV), deformation (DEF), in-plane rotation (IPR) and out-of-plane rotation (OPR). In what follows, we analyze four attributes which occur more frequently in the benchmark based on the precision plots.

On the BC subset, our method gets the second best results than others. The results suggest that the learnt dictionaries can characterize the discriminative information between the object and distracters. On the SV subset, our tracker get a satisfactory result as a result of the using of aff ne motion models. Our tracker get a better results than other on the OPR and IPR subsets. The performance can be attributed to the efficien sparse representations of local image patches.

In addition, we see that our tracker obtains worse results in some attributes. For instance, when the object undergoes fast motion and/or motion blur, our method performs worse than HCFT, HDT and FCNT trackers due to the poor dynamic models in the particle filte . Our tracker can be further improved with more effective state transition matrix of the particle filte . In the LR subset, Our tracker does not perform well, because low-resolution objects (resized to 16×16) may not capture suff cient visual information to represent objects for tracking.

3) Qualitative Comparisons: As shown in Fig. 7, we also present a qualitative evaluation of tracking results to illustrate the effectiveness of our tracker. In total 8 representative sequences are chosen from the subsets of four dominant attributes, *i.e.*, occlusion, illumination variations, background clutter and deformation. Other challenges, *e.g.*, out-of-plane rotation, in-plane rotation and scale variations, are also included in the 8 sequences. Due to space limitations, we only analyze 4 sequences in detail.

In the Matrix sequence, the target undergoes illumination variations and fast motion in a complex scenarios. EBT and FCNT lose the target at the begin of the sequence (*e.g.,* ♯6). When the target moves fast, most of the tracker drift except MEEM and our method (*e.g.,* ♯40). At the end of this sequence, only our tracker still lock on the target with a low overlap score (*e.g.,* ♯94). In the Shaking sequence, the object undergoes the illumination change besides pose variations. AEST, HDT, FCNT and KCF deviate from target at the begin of the sequence (*e.g.,* ♯26). EBT, KCF and AEST trackers drift from the object when the spotlight blinks suddenly (*e.g.,* ♯62). All the other trackers are able to successfully track the object throughout the sequence with relatively accurate sizes of the bounding box. In the Soccer sequence, the object undergoes pose variations as well as partial occlusions by red ribbons. The FCNT, AEST and KCF methods lose the target after a drastic pose change (*e.g.,* ♯78). The DLSSVM method drift away when the object is occluded (*e.g.,* ♯128). In comparison, our tracker perform better than the other methods during the whole sequence (*e.g.,* ♯358). In the Tiger2 sequence, there exists lots of object deformation, occlusion, fast motion and illumination variations. In the ♯1204 frame, FCNT and HDT lose the target while part of tiger goes out of view. After the inf uence of illumination variations and scale change, our tracker get a better success rates compare with other trackers.

D. Experiment 2: Evaluation on the VOT2015 Dataset

In this section, we also present the evaluation results on the VOT2015 [21] dataset. We use accuracy and robustness as

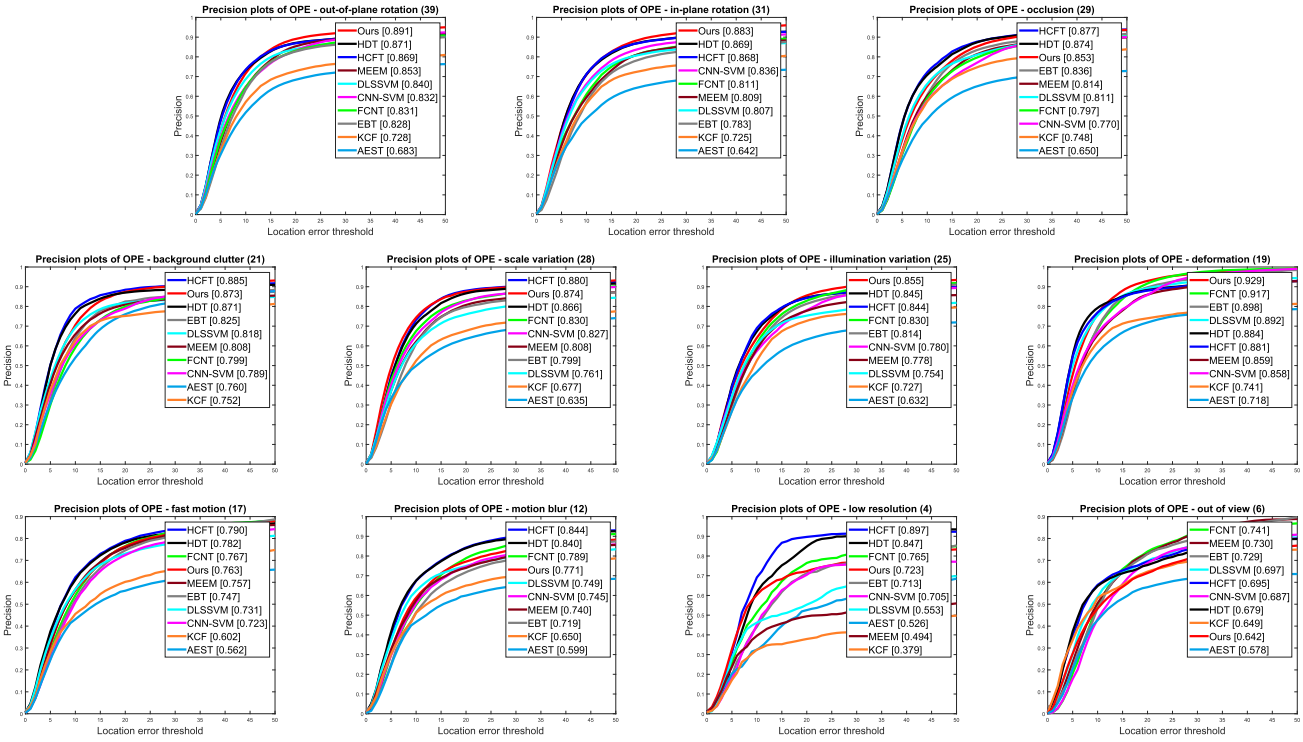

Fig. 6. Attribute-based performance analysis in precision plots. The performance score of each tracker is shown in the legend.

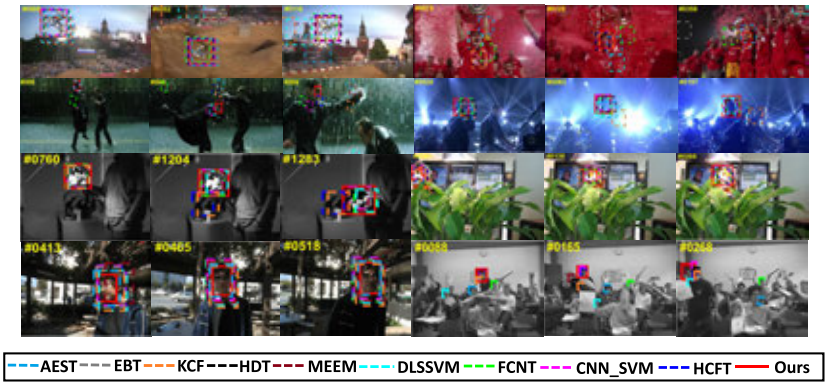

Fig. 7. Qualitative tracking results of 10 trackers over 8 representative sequences (*i.e.,* "MotorRolling", "Soccer", "Matrix", "Shaking", "Sylvester", "Tiger2", "Trellis" and "Freeman4") that are respectively aligned from top to bottom, left to right.

our evaluation criteria. The accuracy measures the bounding box overlap ratio and the robustness counts the number of failures. We compare our tracker with top 9 tracker on this dataset including MDnet [13], DeepSRDCF [72], EBT [60], SRDCF [72], LDP, sPST [73], nsamf and MEEM [61]. Table I shows the expected overlap, AR ranking accuracy and robustness. Fig. 8 shows the baseline results of accuracy and robustness. Our tracker achieve the third best results in overall expected overlap, and the gap between DeepSRDCF and our tracker is only 0.0037. Our tracker cannot achieve the comparable accuracy with MDnet mainly because the VGG-16 utilized in our tracker was trained on ImageNet for the classificati n task while MDnet was trained on tracking benchmark(VOT and OTB) for the tracking task, which leads to the poorer expressive ability of our deep feature compared with MDnet. Overall, the satisfactory performance achieved both in accuracy and robustness show the validity of our tracker.

E. Experiment 3: Evaluation on Hand-Crafted Feature

In this section, we evaluate the overall performance for 16 trackers on 51 sequences. 64 dimensional gray scale feature using subsampling with a step size of 4 and 288 dimensional HOG feature are extracted from each candidate, and they are concatenated into a single feature vector of 352 dimensions as our hand-crafted feature. The OPE curve of both precision plots and success plots are shown in Fig. 9. Only the top 10 trackers are displayed for clarity. The AUC score for each tracker is shown in the legend. Our tracker is 4.1% above

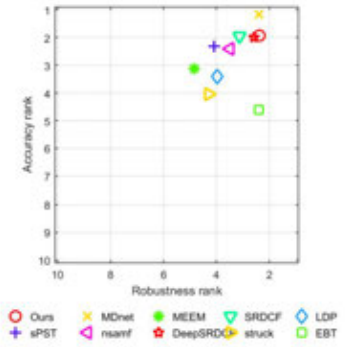

Fig. 8. The robustness-accuracy ranking plots of tested algorithms in VOT2015 dataset. The better trackers are located at the upper-right corner.

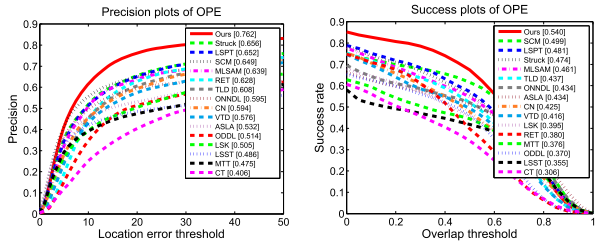

Fig. 9. Overall performance of hand-crafted feature based tracker comparisons of OPE in precision plots and success plots. The performance score for each tracker is shown in the legend.

TABLE I

THE EXPECT OVERLAP AND RANKS OF ACCURACY AND ROBUSTNESS ON THE TWO EXPERIMENTS IN VOT2015. THE FIRST, SECOND AND THIRD BEST SCORES ARE HIGHLIGHTED IN **RED**, **BLUE** AND GREEN COLORS, RESPECTIVELY

	Expected overlap	Accuracy	Robustness
MDnet	**0.3783**	**1.17**	**2.40**
DeepSRDCF	**0.3181**	2.00	2.57
Ours	0.3144	**1.93**	**2.37**
EBT	0.3130	4.60	**2.40**
SRDCF	0.2877	1.95	3.12
LDP	0.2785	3.42	3.97
sPST	0.2767	2.32	4.08
nsamf	0.2536	2.42	3.48
struck	0.2458	4.03	4.28
MEEM	0.2212	3.13	4.83

the SCM in the success rate, and outperforms the Struck by 10.6% in the precision plot. The high accurate achieved both in overlap and location score show the validity of our original dictionary learning model.

Table II shows a comparison of the key dictionary-based methods on the OTB50 benchmark dataset using the average CLE, OR and SR. The key dictionary-based methods include SCM, ODDL, ONNDL, ASLA and ours. We rewritten the original code to make sure that all methods use same features, *i.e.*, HOG and gray value. Our tracker is 4.9% and 7.4% above the SCM in terms of average OR and average SR, respectively.

To better analysis the speed of our original dictionary learning process, we show the speed of our tracker and compare it with other sparse trackers in Table III. Although the run speed of our tracker is not real-time, it is comparable to other sparse trackers. Our tracker cannot achieve a real-time speed due to

TABLE II

QUANTITATIVE COMPARISON OF OUR TRACKERS WITH 6 DICTIONARY-BASED METHODS ON THE CVPR2013 BENCHMARK. THE RESULTS ARE REPORTED IN THE AVERAGE CLE (IN PIXELS), THE AVERAGE OR (%), THE AVERAGE SR (%). RED **BOLD** FONTS INDICATE THE BEST PERFORMANCE AND THE BLUE *Italic* FONTS INDICATE THE SECOND BEST ONES. THE BEST TWO RESULTS ARE SHOWN IN RED BOLD FONTS

	SCM	ODDL	ONNDL	ASLA	MTT	Ours
CLE	58.2	65.4	*52.1*	69.1	78.2	**33.1**
OR	*49.7*	40.1	46.1	45.1	39.1	**54.6**
SR	*59.7*	51.3	55.9	52.4	45.3	**67.1**

TABLE III

QUANTITATIVE COMPARISON OF OUR TRACKERS WITH 8 STATE-OF-THE-ART METHODS ON THE CVPR2013 BENCHMARK [20]. THE RESULTS ARE REPORTED IN THE AVERAGE FPS. RED **BOLD** FONTS INDICATE THE BEST PERFORMANCE AND THE BLUE *Italic* FONTS INDICATE THE SECOND BEST ONES

	SCM	ODDL	ONNDL	ASLA	RET	MLSAM	MTT	LSK	Ours
FPS	0.41	2.6	0.23	**5.4**	*3.7*	0.31	0.67	2.8	2.2

the high computational burden existing in dictionary learning process. We will elaborate the complexity of our model in V-G.3.

F. Experiment 4: Evaluation on Multi-Object Tracking

With some minor modif cations, our tracker can be used to the multi-object tracking by treating each object as an individual class. In this work, tracking multiple objects is fi st to learn multiple class-specifi dictionaries and the shared dictionary. Then each object is determined by Eq. (18). For each multi-object tracking sequence, we pre-defi ed which object should we track and initial their locations. The number of dictionary class also is pre-define according to the number of objects. Fig. 10 shows multi-object tracking results of 6 challenging sequences. In these sequences, the main challenge of the trackers is to distinguish the true object from distracters (*i.e.* objects with a similar appearance). For example, the "Air Show" sequence contains a formation of four similar planes that fl very close to each other, and objects suffer from camera jitter. The "Red Flowers" shows several similar f owers which are moving and changing appearance due to the wind. In the "Skating" sequence, several skaters perform on stage with drastic lighting change as a result of neon and spot lights. Tracking such objects is challenge because objects are almost indistinguishable in the dark environment, even for human eyes. Overall, our tracker achieves good performance.

Following the work of [22], we compare the performance of our tracker with OAB [70], TLD [69], and SPOT [22] trackers. The quantitative results of these four trackers are presented in Table V. In the SPOT tracker, the structural constraints lead to substantial performance improvements when tracking multiple objects. Our tracker achieves comparable results with SPOT without considering the spatial constraints between objects.

To demonstrate the performance of our method, we formulate multiple objects tracking as multiple single-object

Fig. 10. Qualitative multi-object tracking results over 6 representative sequences (*i.e.,* "Air Show", "Parade", "Shaking", "Sky Diving", "Skating", and "Red Flowers") that are respectively aligned from top to bottom. The colors of the rectangles indicate the different objects that are tracked.

TABLE IV
COMPARISONS WITH NAIVE MULTI-OBJECT TRACKING METHOD IN TERMS OF CLE (THE LOWER THE BETTER) AND SR(THE HIGHER THE BETTER)

	Ours_Naive		**Ours**	
	CLE	*SR*	*CLE*	*SR*
Air Show	12.7	0.89	9.2	0.91
Parade	26.4	0.60	17.5	0.69
Red Flowers	10.7	0.92	11.0	0.91
Sky Diving	5.6	0.97	4.7	1.0
Shaking	8.9	0.92	9.6	0.94
Skating	17.8	0.86	15.4	0.88

TABLE V
PERFORMANCE OF FOUR TRACKERS ON MULTI-OBJECT TRACKING IN TERMS OF CLE (THE LOWER THE BETTER) AND SR(THE HIGHER THE BETTER)

	OAB [53]		**TLD** [52]		**SPOT** [22]		**Ours**	
	CLE	*SR*	*CLE*	*SR*	*CLE*	*SR*	*CLE*	*SR*
Air Show	9.3	0.86	31.3	0.53	**5.6**	**1.0**	*9.2*	*0.91*
Parade	*12.7*	**0.82**	**8.8**	*0.71*	9.2	0.63	17.5	0.69
Red Flowers	79.7	0.09	33.3	0.30	**9.5**	**0.99**	*11.0*	*0.91*
Sky Diving	15.5	0.76	35.3	0.13	*5.4*	**1.0**	**4.7**	**1.0**
Shaking	61.9	0.47	14.3	0.47	**7.7**	**0.97**	*9.6*	*0.94*
Skating	100.2	0.05	90.3	0.42	*16.2*	*0.85*	**15.4**	**0.88**

trackers. That is, tracking each object is considered as a binary classif cation problem. The corresponding tracking method is referred to as the *Ours_Naive*. The quantitative results are shown in Table IV. We see that our method is slightly better than *Ours_Naive*. This is because the proposed method can learn the shared atoms and the class-specifi atoms, and discover the shared visual atoms from the class-specifi ones. The learnt discriminative class-specifi atoms are able to encode subtle visual differences between objects and distracters, which prevent the tracker from switching between objects with similar appearance.

G. Diagnostic Analysis

1) Parameter Analysis: There are four parameters in our model, which need to be turned: η, λ_1, λ_2 and β. According to the analysis in Section III, $\mathcal{C}(\mathbf{X}, \mathbf{A}, \mathbf{B})$ is the critical component distinguishing our method from other DDL trackers (*e.g.*, [8], [43]), so we pay more attention on λ_1 and λ_2. To better analyze the inf uence of these two parameters, we set $\eta = 0.15$, $\beta = e^{-10}$ and test our tracker on tracking sequences *CarScale* and *Couple* with different combinations of the values of λ_1 and λ_2. The value range of λ_1 and λ_2 is $[0.01, 0.1, 0.3, 0.5, 2, 10]$ and $[0.05, 0.1, 0.3, 0.5, 1, 2, 5, 10]$ respectively. Fig. 11 shows

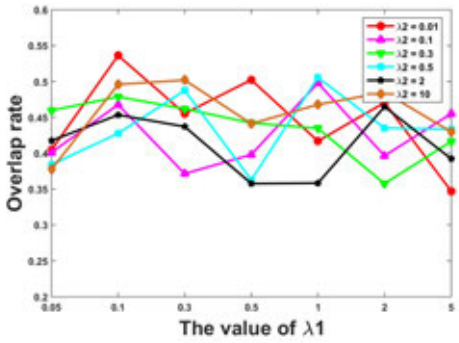

Fig. 11. The overlap rate of different combinations of λ_1 and λ_2. Horizontal axis presents the values of λ_1, vertical axis presents average overlap rate performed on sequence *CarScale* and *Couple*. Lines with different color presents different values of λ_2, the labels are shown in the legend.

the average overlap rate as a function of λ_1 and λ_2 on sequences *CarScale* and *Couple*. We f nd that there is no f xed relationship behind the changes of the overlap rate with different combinations of λ_1 and λ_2. One of the reasons should be that the constraint conditions are variant for different sequences, we thus just choose the best parameters setting for our tracker. As illustrated in Fig. 11, we set $\lambda_1 = 0.1$ and $\lambda_2 = 0.01$ which can achieve the best tracking performance among all combinations.

In addition, η and β are other two scalar parameters in our model which involve the sparsity and discrimination of the dictionary respectively. The analysis of these two items can be found in other DDL methods (*e.g.*, [8], [43]), therefore, we just set them empirically according to the setting of [8] and [43].

2) Effectiveness of Each Component: The main contribution distinguishing our tracker from others is the discrimination dictionary term $\mathcal{C}(\mathbf{X}, \mathbf{A}, \mathbf{B})$, in particular, the inter-nonredundancy constraint $\min \sum_{j \neq c, j=0}^{C} \left\| \mathbf{B}_c^\top \mathbf{B}_j \right\|_F^2$ and the intra-nonredundancy constraint $\min \left\| \mathbf{B}_c^\top \mathbf{B}_c - \mathbf{I}_{K_c} \right\|_F^2$. If we remove these two item, our tracker almost degrades to the same model as ODDL introduced in [8]. The signif cant improvement compared with ODDL tracker shown in Fig. 9 have proved that these two items make a great infl ence to our model. To better analyze the contribution of the inter-nonredundancy constraint and the intra-nonredundancy constraint, we remove one component at a time and report the performance of our tracker. Table VI presents the tracking results of our model with and without different component on OTB50 benchmark. "**Ours** with $\lambda_1 = 0$" means that our

TABLE VI

QUANTITATIVE COMPARISON OF OUR TRACKERS WITH AND WITHOUT DIFFERENT COMPONENTS ON THE BENCHMARK. THE RESULTS ARE REPORTED IN THE AVERAGE CENTER LOCATION ERROR (CLE ↓, IN PIXELS), THE AVERAGE OVERLAP RATE (OR ↑, %), AND THE AVERAGE SUCCESS RATE (SR ↑, %). HERE ↑ MEANS THAT HIGHER SCORES INDICATE BETTER RESULTS, AND ↓ REPRESENTS THAT LOWER IS BETTER

	CLE ↓	OR ↑	SR ↑
Ours with $\lambda_1 = 0$	46.8	45.0	54.1
Ours with $\lambda_2 = 0$	47.9	46.6	56.3
Ours without $\mathbf{B}_0$	35.4	53.8	65.3
Ours	33.1	54.6	67.1

model is trained without the inter-nonredundancy constraint and "**Ours** with $\lambda_2 = 0$" means it is trained without the intra-nonredundancy constraint. The comparisons shown in this table illustrate the effectiveness of each component in the model.

$||\mathbf{A}_c||_1$ is the sparse item which involves the sparsity while $\mathcal{L}(\mathbf{A}; \mathbf{W})$ is the discrimination item which involves the discrimination into dictionaries. Both of them are basic components for DDL-based trackers and have been proved effective for visual tracking [8], [43].

Another important distinction is that besides the class-specifi atoms $\mathbf{B}_c$, we also learned the shared atoms $\mathbf{B}_0$. The shared dictionary $\mathbf{B}_0$ is designed in our dictionary learning method to discover the hidden visual patterns shared by the visually correlated categories. Since the common patterns may make the learnt class-specif c dictionaries redundant, they do not benefi classificatio performance, but may even degrade the classificatio accuracy. Separating them from the class-specifi dictionaries enables our model to learn more discriminative and more compact dictionaries. To evaluate the effectiveness of the shared dictionary $\mathbf{B}_0$, we have trained class-specif c dictionaries without considering the shared dictionary explicitly in the Benchmark2013 dataset. To measure the accuracy of the tracking results, we use the center location error (CLE), the overlapping rate (OR) and the success rate (SR) for quantitative evaluations. The comparisons are reported in Table VI. It shows that separating the hidden visual patterns from the class-specifi ones is effective to ameliorate the discrimination of the dictionaries.

3) Complexity Analysis: As discussed in [74], the time complexity of the sparse coding problem is approximately $\mathcal{O}(d^2\ K^{\varepsilon})$, where d is the feature dimensionality, K is the number of dictionary bases, and $\varepsilon \leq 1.2$ is a constant. In our model, since the coding coeff cients are updated class by class, the time complexity of updating coding coeff cients is $\sum_{c=1}^{C} N_c \mathcal{O}(d^2 K_c^{\varepsilon})$, where N_c is the number of training samples in the c-th class and K_c is the number of class-specif c dictionary bases. The time complexity of updating class-specif c bases is $\sum_{c=1}^{C} K_c \mathcal{O}(d N_c)$. Since the shared bases contribute to the representation of all samples, the time complexity of updating the shared dictionary is $K_0 \mathcal{O}(d N)$. Therefore, the overall time complexity of our model is $n\Big(\sum_{c=1}^{C} N_c \mathcal{O}(d^2 K_c^{\varepsilon}) + \sum_{c=1}^{C} K_c \mathcal{O}(d N_c) + K_0 \mathcal{O}(d N)\Big)$, where n is the total number of iterations. To better elucidate the speed of our original dictionary learning process excluding the inf uence of CNNs feature extractor, we have shown the speed of our tracker based on hand-crafted features and compared it with other hand-crafted based trackers in table III. In particular, we can increase the dictionary updating periods or reduce the number of iteration to accelerate our tracker directly.

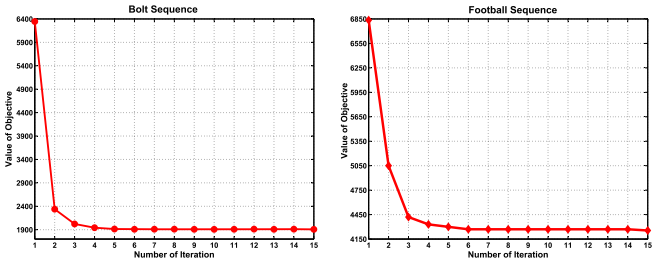

Fig. 12. Examples of the convergence of our model on the "Bolt" sequence and the "Football" sequence.

As discussed in Section III, the objective function of our method is not convex. We alternatively update sparse coeffi ients, class-specifi bases, shared bases, and classifi r parameters. We use a stochastic gradient descent algorithm to obtain the local optimum. The change of the objective value with respect to the number of iteration on the "Bolt" sequence and the "Football" sequence is plotted in Fig. 12. It shows that the objective value converges within about 6 iterations.

VI. CONCLUSION

In this paper, we have presented a joint discriminative dictionary learning method for the robust distracter-resistive tracker. Our method can learn the shared atoms and the class-specifi atoms, and effectively separate commonly shared visual patterns from class-specif c ones. The learnt dictionary is more compact and more discriminative, which makes our tracker have better discriminating power to handle appearance changes. During tracking, the quality of each candidate is measured by the global coding classif er and the learnt linear classif er instead of relying on only one of them. We also successfully apply CNN feature to our dictionary leaning method to improve our tracking results. Comparisons with 9 state-of-the-art tracking methods on the benchmark dataset have demonstrated that our tracker effectively resists distracters and outperforms existing methods. In addition, we show several multi-object tracking results to demonstrate the good performance of the proposed tracker for tracking multiple objects.

REFERENCES

[1] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. Van Den Hengel, "A survey of appearance models in visual object tracking," *ACM Trans. Intell. Syst. Technol.*, vol. 4, no. 4, p. 58, Sep. 2013.

[2] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1442–1468, Jul. 2014.

[3] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1822–1829.

[4] D. Wang, H. Lu, and M.-H. Yang, "Least soft-threshold squares tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 2371–2378.

[5] B. Zhuang, H. Lu, Z. Xiao, and D. Wang, "Visual tracking via discriminative sparse similarity map," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1872–1881, Apr. 2014.

[6] Y. Wu, B. Ma, M. Yang, J. Zhang, and Y. Jia, "Metric learning based structural appearance model for robust visual tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 5, pp. 865–877, May 2014.

[7] R. Yao, Q. Shi, C. Shen, Y. Zhang, and A. van den Hengel, "Part-based visual tracking with online latent structural learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 2363–2370.

[8] F. Yang, Z. Jiang, and L. S. Davis, "Online discriminative dictionary learning for visual tracking," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2014, pp. 854–861.

[9] Q. Wang, F. Chen, W. Xu, and M.-H. Yang, "Object tracking with joint optimization of representation and classif cation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 4, pp. 638–650, Apr. 2015.

[10] S. Zhang, H. Zhou, F. Jiang, and X. Li, "Robust visual tracking using structurally random projection and weighted least squares," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 11, pp. 1749–1760, Nov. 2015.

[11] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation f lters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

[12] M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1090–1097.

[13] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4293–4302.

[14] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3119–3127.

[15] Z. Cui, S. Xiao, J. Feng, and S. Yan, "Recurrently target-attending tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1449–1458.

[16] P. Agrawal, R. Girshick, and J. Malik, "Analyzing the performance of multilayer neural networks for object recognition," in *Computer Vision—ECCV*, vol. 8695. Cham, Switzerland: Springer, 2014, pp. 329–344.

[17] M. Cimpoi, S. Maji, and A. Vedaldi. (2015). "Deep convolutional f lter banks for texture recognition and segmentation." [Online]. Available: https://arxiv.org/abs/1411.6836

[18] C. Ma, J. Huang, X. Yang, and M. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Jun. 2015, pp. 3074–3082.

[19] N. Zhou and J. Fan, "Jointly learning visually correlated dictionaries for large-scale visual recognition applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 715–730, Apr. 2014.

[20] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 2411–2418.

[21] M. Kristan *et al.*, "The visual object tracking VOT2015 challenge results," in *Proc. ICCV*, Dec. 2016, pp. 564–586.

[22] L. Zhang and L. van der Maaten, "Preserving structure in model-free tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 756–769, Apr. 2014.

[23] S. Bahrampour, N. M. Nasrabadi, A. Ray, and W. K. Jenkins, "Multimodal task-driven dictionary learning for image classification" *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 24–38, Jan. 2016.

[24] N. Akhtar, F. Shafait, and A. Mian, "Discriminative Bayesian dictionary learning for classification" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 2374–2388, Dec. 2016.

[25] T. Yao, Z. Wang, Z. Xie, J. Gao, and D. D. Feng, "Learning universal multiview dictionary for human action recognition," *Pattern Recognit.*, vol. 64, pp. 236–244, Apr. 2017.

[26] X.-Y. Jing, F. Wu, Z. Li, R. Hu, and D. Zhang, "Multi-label dictionary learning for image annotation," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2712–2725, Jun. 2016.

[27] C. Bao, J.-F. Cai, and H. Ji, "Fast sparsity-based orthogonal dictionary learning for image restoration," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 3384–3391.

[28] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Sparse representation based Fisher discrimination dictionary learning for image classification" *Int. J. Comput. Vis.*, vol. 109, no. 3, pp. 209–232, 2014.

[29] S. Gao, I. W.-H. Tsang, and Y. Ma, "Learning category-specifi dictionary and shared dictionary for f ne-grained image categorization," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 623–634, Feb. 2014.

[30] J. Zheng and Z. Jiang, "Learning view-invariant sparse representations for cross-view action recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 3176–3183.

[31] X. Mei and H. Ling, "Robust visual tracking using ℓ_1 minimization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Sep./Oct. 2009, pp. 1–8.

[32] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via structured multi-task sparse learning," *Int. J. Comput. Vis.*, vol. 101, no. 2, pp. 367–383, Jan. 2013.

[33] T. Zhang, A. Bibi, and B. Ghanem, "In defense of sparse tracking: Circulant sparse tracker," in *Proc. Comput. Vis. Patter Recognit.*, Jun. 2016, pp. 3880–3888.

[34] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai, "Minimum error bounded eff cient ℓ_1 tracker with occlusion detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 1257–1264.

[35] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust ℓ_1 tracker using accelerated proximal gradient approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1830–1837.

[36] D. Wang, H. Lu, and M.-H. Yang, "Online object tracking with sparse prototypes," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 314–325, Jan. 2013.

[37] X. Mei, Z. Hong, D. Prokhorov, and D. Tao, "Robust multitask multiview tracking in videos," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 11, pp. 2874–2890, Nov. 2015.

[38] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2356–2368, Jun. 2014.

[39] B. Liu, J. Huang, C. Kulikowski, and L. Yang, "Robust visual tracking using local sparse appearance model and K-selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2968–2981, Dec. 2013.

[40] Z. Hong, X. Mei, and D. Tao, "Dual-force metric learning for robust distracter-resistant tracker," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Berlin, Germany: Springer, 2012, pp. 513–527.

[41] T. Zhang, S. Liu, N. Ahuja, M.-H. Yang, and B. Ghanem, "Robust visual tracking via consistent low-rank sparse learning," *Int. J. Comput. Vis.*, vol. 111, no. 2, pp. 171–190, 2015.

[42] Q. Wang, F. Chen, W. Xu, and M.-H. Yang, "Online discriminative object tracking with local sparse representation," in *Proc. IEEE Workshop Appl. Comput. Vis. (WACV)*, Jan. 2012, pp. 425–432.

[43] N. Wang, J. Wang, and D.-Y. Yeung, "Online robust non-negative dictionary learning for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 657–664.

[44] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Proc. Adv. Neural Inf. Process. Syst.*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. New York, NY, USA: Curran Associates, Inc., 2013, pp. 809–817.

[45] Y. Qi *et al.*, "Hedged deep tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4303–4311.

[46] X. Wang, E. Türetken, F. Fleuret, and P. Fua, "Tracking interacting objects using intertwined f ows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2312–2326, Nov. 2016.

[47] X. Wang, E. Türetken, F. Fleuret, and P. Fua, *Tracking Interacting Objects Optimally Using Integer Programming*. Cham, Switzerland: Springer, 2014, pp. 17–32, doi: 10.1007/978-3-319-10590-1_2.

[48] E. Türetken, X. Wang, C. Becker, C. Haubold, and P. Fua. (Jan. 2015). "Globally optimal cell tracking using integer programming." [Online]. Available: https://arxiv.org/abs/1501.05499

[49] A. Maksai, X. Wang, and P. Fua, "What players do with the ball: A physically constrained interaction modeling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 972–981.

[50] T. Wang, S. Gong, X. Zhu, and S. Wang, "Person re-identificatio by discriminative selection in video ranking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 12, pp. 2501–2514, Dec. 2016.

[51] X. Li, W.-S. Zheng, X. Wang, T. Xiang, and S. Gong, "Multi-scale learning for low-resolution person re-identif cation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3765–3773.

[52] W.-S. Zheng, X. Li, T. Xiang, S. Liao, J. Lai, and S. Gong, "Partial person re-identification" in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4678–4686.

[53] S. Arora, R. Ge, and A. Moitra. (2013). "New algorithms for learning incoherent and overcomplete dictionaries." [Online]. Available: https://arxiv.org/abs/1308.6273

[54] E. Candès and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse Problems*, vol. 23, no. 3, p. 969, 2007.

[55] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classif cation and clustering via dictionary learning with structured incoherence and shared features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 3501–3508.
[56] G. H. Golub and C. F. Van Loan, *Matrix Computations*, vol. 3. Baltimore, MD, USA: JHU Press, 2012.
[57] Z. Jiang, Z. Lin, and L. S. Davis, "Label consistent K-SVD: Learning a discriminative dictionary for recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2651–2664, Nov. 2013.
[58] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Eff cient sparse coding algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 801–808.
[59] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 597–606.
[60] G. Zhu, F. Porikli, and H. Li, "Beyond local search: Tracking objects everywhere with instance-specifi proposals," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 943–951.
[61] J. Zhang, S. Ma, and S. Sclaroff, *MEEM: Robust Tracking via Multiple Experts Using Entropy Minimization*. Cham, Switzerland: Springer, 2014, pp. 188–203.
[62] J. Ning, J. Yang, S. Jiang, L. Zhang, and M.-H. Yang, "Object tracking via dual linear structured SVM and explicit feature map," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4266–4274.
[63] M. Yang, Y. Wu, M. Pei, B. Ma, and Y. Jia, "Online discriminative tracking with active example selection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 7, pp. 1279–1292, Jul. 2016.
[64] Q. Bai, Z. Wu, S. Sclaroff, M. Betke, and C. Monnier, "Randomized ensemble tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (CVPR)*, Dec. 2013, pp. 2040–2047.
[65] K. Zhang, L. Zhang, and M. Yang, "Fast compressive tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 10, pp. 2002–2015, Oct. 2014.
[66] J. Kwon and K. Lee, "Visual tracking decomposition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1269–1276.
[67] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, Aug. 2011.
[68] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 263–270.
[69] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, Jul. 2012.
[70] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proc. Brit. Mach. Vis. Conf.*, vol. 1, no. 5, p. 6, 2006.
[71] A. Vedaldi and K. Lenc, "MatConvNet: Convolutional neural networks for MATLAB," in *Proc. ACM Int. Conf. Multimedia*, 2015, pp. 689–692.
[72] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation f lters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4310–4318.
[73] Y. Hua, K. Alahari, and C. Schmid, "Online object tracking with proposal selection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3092–3100.
[74] K. Koh, S.-J. Kim, and S. P. Boyd, "An interior-point method for large-scale l_1-regularized logistic regression," *J. Mach. Learn. Res.*, vol. 8, no. 8, pp. 1519–1555, 2007.

Weichao Shen received the B.S. degree in control engineering from the School of Automation, Beijing Institute of Technology, Beijing, China, in 2014, where he is currently pursuing the Ph.D. degree with the School of Computer Science, under the supervision of Prof. Y. Jia. His current research interests include computer vision, unsupervised representation learning, object tracking, and 3D reconstruction.

Yuwei Wu received the Ph.D. degree in computer science from the Beijing Institute of Technology (BIT), Beijing, China, in 2014. From 2014 to 2016, he was a Post-Doctoral Research Fellow with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. He is currently an Assistant Professor with the School of Computer Science, BIT. He has strong research interests in computer vision and information retrieval. He received the Outstanding Ph.D. Thesis Award from BIT and a Distinguished Dissertation Award Nominee from the China Association for Artif cial Intelligence.

Junsong Yuan (M'08–SM'14) received the M.Eng. degree from National University of Singapore in 2005 and the Ph.D. degree from Northwestern University in 2009. Before that, he graduated from the Special Class for the Gifted Young of Huazhong University of Science and Technology, Wuhan, China, in 2002. He was an Associate Professor with the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore. He is currently an Associate Professor with the Department of Computer Science and Engineering, The State University of New York, Buffalo, USA. He received the 2016 Best Paper Award in IEEE TRANSACTIONS ON MULTIMEDIA, the Doctoral Spotlight Award from the IEEE Conference on Computer Vision and Pattern Recognition (CVPR09), a Nanyang Assistant Professorship from NTU, and the Outstanding EECS Ph.D. Thesis Award from Northwestern University. He is the Program Co-Chair of ICME18 and VCIP15, and the Area Chair of ACM MM18, ACCV1814, ICPR1816, CVPR17, and ICIP1817. He served as the Guest Editor of *International Journal of Computer Vision*. He is currently a Senior Area Editor of *Journal of Visual Communication and Image Representation* and an Associate Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.

Lingyu Duan (M'06) has been the Associate Director of the Rapid-Rich Object Search Laboratory (ROSE), a joint lab between Nanyang Technological University, Singapore, and Peking University (PKU), China, since 2012. He is currently a Full Professor with the National Engineering Laboratory of Video Technology, PKU. His research interests include multimedia indexing, search, and retrieval, mobile visual search, visual feature coding, and video analytics. He received the *EURASIP Journal on Image and Video Processing* Best Paper Award in 2015, the Ministry of Education Technology Invention Award (First Prize) in 2016, the National Technology Invention Award (Second Prize) in 2017, the China Patent Award for Excellence in 2017, and the National Information Technology Standardization Technical Committee "Standardization Work Outstanding Person" Award in 2015. He was a Co-Editor of the MPEG Compact Descriptor for Visual Search Standard (ISO/IEC 15938-13) and is serving as the Co-Chair of MPEG Compact Descriptor for Video Analytics. He is currently an Associate Editor of *ACM Transactions on Intelligent Systems and Technology* and *ACM Transactions on Multimedia Computing, Communications, and Applications*.

Jian Zhang received the B.S. degree in electronics from East China Normal University, China, the M.S. degree in computer science from Flinders University, Australia, and the Ph.D. degree in electrical engineering from University of New South Wales (UNSW), Australia. From 1997 to 2003, he was with the Visual Information Processing Laboratory, Motorola Labs, Sydney, as a Principal Research Engineer and the Research Manager of visual communications. From 2004 to 2011, he was a Principal Researcher and a Project Leader with Data61 (formerly INCTA), Australia, and a Conjoint Associate Professor with the School of Computer Science and Engineering, UNSW. He is currently an Associate Professor with the Global Big Data Technologies Centre, School of Electrical and Data Engineering, Faculty of Engineering and Information Technology, University of Technology Sydney, Australia. He is the author or co-author of more than 140 paper publications and book chapters and holds six issued U.S. and Chinese patents. His current interests include social multimedia signal processing, large-scale image and video content analytics, retrieval and mining, 3D-based computer vision, and intelligent video surveillance systems.

Dr. Zhang was the General Co-Chair of the International Conference on Multimedia and Expo in 2012 and the Technical Program Co-Chair of the IEEE Visual Communications and Image Processing in 2014. He was an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY from 2006 to 2015. He has been an Associate Editor for IEEE TRANSACTIONS ON MULTIMEDIA and *EURASIP Journal on Image and Video Processing* since 2016.

Yunde Jia (M'11) received the B.S., M.S., and Ph.D. degrees from Beijing Institute of Technology (BIT) in 1983, 1986, and 2000, respectively. He was a Visiting Scientist with the Robot Institute, Carnegie Mellon University, from 1995 to 1997. He is currently a Professor with the School of Computer Science, BIT, and the Team Head of BIT innovation on vision and media computing. He serves as the Director of the Beijing Laboratory of Intelligent Information Technology. He has authored over 300 publications in computer vision and media computing. In recent years, his interests have extended to vision-based HCI and HRI, intelligent robotics, and cognitive systems.