Robust Object Tracking by Nonlinear Learning

Bo Ma, Hongwei Hu, Jianbing Shen, *Senior Member, IEEE*, Yuping Zhang,

Ling Shao, *Senior Member, IEEE*, and Fatih Porikli, *Fellow, IEEE*

***Abstract*—We propose a method that obtains a discriminative visual dictionary and a nonlinear classifier for visual tracking tasks in a sparse coding manner based on the globally linear approximation for nonlinear learning theory. Traditional discriminative tracking methods based on sparse representation learn a dictionary in an unsupervised way and then train a classifier, which may not generate both descriptive and discriminative models for targets by treating dictionary learning and classifier learning separately. In contrast, the proposed tracking approach can construct a dictionary that fully reflects the intrinsic manifold structure of visual data and introduces more discriminative ability in a unified learning framework. Finally, an iterative optimization approach, which computes the optimal dictionary, the associated sparse coding and a classifier, is introduced. Experiments on two benchmarks show that our tracker achieves better performance compared with some popular tracking algorithms.**

***Index Terms*—Object tracking, Nonlinear learning, Local coordinate coding, Global linear approximation**

I. Introduction

Visual tracking is a popular topic in computer vision, and numerous tracking methods [1], [2], [3], [43], [54], [26] have been proposed to deal with challenges, such as illumination variation [12], [16], global or partial occlusion, shape deformation, in-plane rotation and background clutters. To evaluate the performance of different tracking algorithms quantitatively, several tracking datasets, such as visual tracking benchmark [4] and VOT2014 [5], have been established.

Among current tracking approaches, much attention has been attracted by sparse representation based approaches because of their robust performances in vision tasks [6], [36], [41]. Many sparse coding based tracking methods [7] have been proposed by researchers. Broadly speaking, a given candidate sample can be encoded by linear combination of a few atoms spanning an over-complete dictionary that is learned from a training set of samples in sparse coding based methods. Different dictionary learning approaches [8], [9], [10], [52], [47], [49], [55] based on sparse coding have been proposed for signal reconstruction and classification in the audio and image processing domains. In visual tracking, the learning and updating of a dictionary are crucial steps to handle and adapt to appearance variation during tracking procedure as well. Therefore, a suitable dictionary selection carries significant importance. Mei and Ling [11] constructed a dictionary to encode the candidate targets by employing the global target templates. Bao *et al.* [13] introduced the accelerated proximal gradient approach to promote the real time l_1 tracker with the same dictionary learning strategy. Zhong *et al.* [14] introduced a sparsity-based collaborative model with a generative and a discriminative model. The discriminative classifier was trained by a dictionary with holistic templates, while the generative model was based on a local dictionary from local patches by k-means. This method performs well under drastic appearance changing. Most sparse coding based discriminative object tracking methods [15] learn dictionary and train classifier using a separated mechanisms. To acquire the dictionary for coding, some unsupervised clustering methods, such as k-means are usually adopt, but the produced dictionary may not suitable for tracking task. Yang *et al.* [17] proposed an online discriminative dictionary learning approach for visual tracking. But this method leaves the locality of sparse codes out, and has no consideration on the underlying manifold geometry structure of neither labeled samples nor unlabeled samples during dictionary learning.

A classification function learning using visual data is restricted to be nonlinear as a necessity, because of the target appearance variation during tracking process. In principle, the so called "curse of dimensionality" may arise due to high dimensional feature for modeling target appearance changing. This phenomenon is seldom observed during practical tracking procedure. Moreover, a satisfying tracking performance could be obtained using only a handful of visual training samples. One possible reason is that typically visual data represented by high dimensional vectors reside in a low dimensionality embedding manifold of the high dimensional space that they lie in. Based on this inference, a nonlinear learning theory using Local Coordinate Coding (LCC) is proposed in [18], [19]. LCC is a general coding framework that approximates any nonlinear Lipschitz smooth function using linear functions. It consists of a sparse coding scheme that defines the local coordinates and a dictionary that contains the local coordinates. It shows that under some Lipschitz continuity assumption the computation complexity for learning a nonlinear classification function relys on the dimensionality of inherent manifold sample space. Considering the manifold geometry structure of sample space, we think that those dictionary items close

This work was supported in part by the National Natural Science Foundation of China (61472036, 61272359, 61672099, 81627803), the National Key $R\&D$ Program of China (2017YFC0112000), the Australian Research Council's Discovery Projects funding scheme (DP150104645), and the Fok Ying-Tong Education Foundation for Young Teachers. Specialized Fund for Joint Building Program of Beijing Municipal Education Commission. (Corresponding author: Jianbing Shen)

B. Ma, H. Hu, J. Shen, and Y. Zhang are with Beijing Laboratory of Intelligent Information Technology, School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China (email: bma000@bit.edu.cn, shenjianbing@bit.edu.cn)

L. Shao is with JD Artificial Intelligence Research (JDAIR), Beijing 100176, P. R. China, and also with the School of Computing Sciences, University of East Anglia, Norwich NR4 7TJ, UK. (email: ling.shao@uea.ac.uk)

F. Porikli is with the Research School of Engineering, Australian National University. (email: fatih.porikli@anu.edu.au)

to samples to be encoded should be activated. LCC could keep locality of codes while reconstructing original samples using the learned dictionary items. Impressive performance is exhibited on nonlinear learning using LCC [20]. It also shows high classification accuracies on large-scale image classification [21] and object recognition [22], [24]. However, those methods formulate their visual dictionary with a simple unsupervised way. And they all treat dictionary learning and classifier learning as a separated way, and take a two-stage learning strategy. It may not generate an optimal dictionary that owns discriminative power and reflects the spatial geometry structure of sample space. Even so, the nonlinear theory using LCC endows a firm theoretical foundation to promote sparse coding based discriminative visual tracking algorithm.

In this work, we present a well-designed tracking algorithm that aims to learn visual dictionary and nonlinear classification function jointly enlighten by above mentioned nonlinear learning theory under a semi-supervised framework. The dictionary is learned to describe the embedded manifold structure constructed by samples with and without labels, and it is also expected to maintain approving discriminative power. Therefore, the proposed method could overcome several limitations arisen in most of existing visual tracking approaches efficiently. Furthermore, it employs the localized sparse representation to provide guidance for discriminative visual tracking, which has a solid theoretical basis. The final discriminative dictionary, classification function and sparse codes are calculated by an iterative optimization algorithm. One preliminary version of this work was presented in [23], and this work is significantly different. First, more theoretical foundations about the theory of globally linear approximation to nonlinear learning are introduced (Section II-A). Second, theoretical analysis of each item is added and explained concretely, such as the semi-supervised learning item (3) and the discriminative item (4). Third, a new analysis about the relationships between the classifier and the learned dictionary in our optimization algorithm is explained in Section II-B. Fourth, drifting often occurs during long term tracking due to occlusion and deformation in many tracking tasks. In order to alleviate this issue, a target re-detection method is introduced to relocate target once tracking fails (Section III-D). Fifth, a center refining scheme is introduced in the experiments (Section IV-A) to further improve the tracking performance. Finally, we compare the present work with more recent tracking methods on OTB2013 (Fig. 3 and Table I). And more experimental analysis (Section IV-C) and experiments on VOT2015 (Fig. 5) are added as well. Our source code will be available online [1].

II. Globally Linear Approximation to Nonlinear Learning

A. Problem Description

Given a set of labeled samples $X_l = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ with their labels $Y = \{y_1, \cdots, y_n\}$ where $\mathbf{x}_i \in \mathbb{R}^d$ and a group of unlabeled samples $X_u = \{\mathbf{x}_{n+1}, \cdots, \mathbf{x}_{n+u}\}$, our goal is to learn a nonlinear classification function, a discriminative dictionary, and sparse coefficients for samples represented under dictionary. We aim to learn a nonlinear classifier on a very high dimensional sample space originated from visual tracking problem. In view of traditional statistical theory, the performance decreases when dimensionality of sample feature exceeds an optimal number. Thus, learning a nonlinear function from this sample space is inaccurate. Fortunately, the theory of globally linear approximation to nonlinear learning show that a nonlinear function $f(\mathbf{x})$ could be approximated by a linear function with regard to local coordinate coefficients of samples under manifold assumption [18].

$$\left| f(\mathbf{x}) - \sum_{i=1}^{|\mathbf{D}|} \alpha_i f(\mathbf{d}_i) \right| \leq \beta \left\| \mathbf{x} - \gamma(\mathbf{x}) \right\| + \delta \sum_{i=1}^{|\mathbf{D}|} |\alpha_i| \left\| \mathbf{d}_i - \gamma(\mathbf{x}) \right\|^{1+p}, \quad (1)$$

where $\gamma(\mathbf{x}) = \mathbf{D}\boldsymbol{\alpha}$, and $\mathbf{D} = [\mathbf{d}_1, \cdots, \mathbf{d}_m] \in \mathbb{R}^{d \times m}$ is the dictionary, and $\boldsymbol{\alpha}$ is the code of sample $\mathbf{x}$. This equation means that a very high dimensional nonlinear learning problem can be translated into a much simpler linear learning problem. By minimizing the upper bound, we could obtain a much simpler approximated linear function with codes of original samples as its input instead of original complicated nonlinear function. The upper bound of the approximation error is bounded by the reconstruction error of a sample and the affinity between the sample and dictionary items. For a sample $\mathbf{x}_i$, LCC, which is just the upper bound, is approximated as

$$\min_{\mathbf{D}, \boldsymbol{\alpha}_i} \|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|^2 + \mu \sum_{j=1}^{m} |\alpha_i^j| ||\mathbf{d}_j - \mathbf{x}_i||^2, \quad (2)$$
$$s.t. \quad \mathbf{1}^T \boldsymbol{\alpha}_i = 1,$$

where μ is a constant factor to balance reconstruction error and locality, α_i^j denotes the j-th element of $\boldsymbol{\alpha}_i$ which is the local coordinate code of sample $\mathbf{x}_i$ under dictionary $\mathbf{D}$, and each element in vector $\mathbf{1}$ is a set to one.

Considering both labeled and unlabeled samples, we extend the above equation as

$$\min_{\mathbf{D}, \mathbf{A}} \sum_{i=1}^{u+n} \left(\|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|^2 + \mu \sum_{j=1}^{m} |\alpha_i^j| ||\mathbf{d}_j - \mathbf{x}_i||^2 \right), \quad (3)$$
$$s.t. \quad \mathbf{1}^T \boldsymbol{\alpha}_i = 1, \quad i = 1, \cdots, n+u.$$

We denote $\mathbf{A} = [\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_{n+u}] \in \mathcal{R}^{m \times (n+u)}$ as the code matrix of all training samples. The locality of LCC brings sparsity, but it is not true contrarily. The globally linear approximation of $f(\mathbf{x}_i)$ is formulated as $f(\mathbf{x}_i) \approx \boldsymbol{\alpha}_i^T \mathbf{w}$ under the nonlinear learning theory using LCC. And the labeled samples should be considered for discriminative dictionary learning. Therefore, we introduce the discriminative item as

$$\min_{\mathbf{W}, \mathbf{A}_l} \|\mathbf{A}_l^T \mathbf{w} - \mathbf{y}\|^2, \quad s.t. \quad \mathbf{1}^T \boldsymbol{\alpha}_i = 1, \quad i = 1, \cdots, n \quad (4)$$

where code matrix $\mathbf{A}_l = [\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_n]$ corresponds to labeled samples.

We intend to assign similar labels to those neighbor samples considering the geometry spatial structure constructed by

[1] *http://github.com/shenjianbing/LLCtracking*

samples. In LCC, a sample could be sparsely constructed by the bases in advance with a linear combination manner, like other sparse coding approaches. For simplicity, a novel sparse coding method is introduced as an approximation of LCC [21]. It encodes a sample with k dictionary items which are the nearest neighbors of the sample. And the corresponding sparse code is obtained by solving a least squares problem with some equality constrains. It will reach our objective obviously, since neighbor samples will be encoded by several same dictionary items. Therefore, we introduce a Laplacian regularization item to handle it. More theoretical analysis about Laplacian regularization refers to [25]. Finally, the proposed semi-supervised learning method for dictionary, sparse codes and classification function is formulated as

$$\begin{aligned}\min_{\mathbf{D},\mathbf{A},\mathbf{w}} \sum_{i=1}^{u+n} & \left(\|\mathbf{x}_i-\mathbf{D}\boldsymbol{\alpha}_i\|^2+\mu\sum_{j=1}^{m}|\alpha_i^j|\|\mathbf{d}_j-\mathbf{x}_i\|^2\right)\\ & +\lambda_1||\mathbf{A}_l^T\mathbf{w}-\mathbf{y}||^2+\lambda_2\sum_{i=1}^{n+u}\sum_{j=1}^{n+u}\|\boldsymbol{\alpha}_i^T-\boldsymbol{\alpha}_j^T\mathbf{w}\|^2B_{ij}, \end{aligned} \tag{5}$$

$$s.t. \quad \mathbf{1}^T\boldsymbol{\alpha}_i=1, \quad i=1,\cdots,n+u,$$

where the last item is the Laplacian constraint with $B_{ij}=\boldsymbol{\alpha}_i^T\boldsymbol{\alpha}_j$, and λ_1,λ_2 are two preset constants that balance the discriminative ability and manifold spatial constrain.

Manifold regularization has been used in several earlier works [51]. But there are significant differences between our method and [51], [53]. First, our method aims to track a single general object in one video clip, while [51] performs tracking for multiple persons in multiple videos using person detection and face recognition techniques. Thus, these two methods belong to different topics of visual tracking tasks. Second, our tracking method is solved as a regression problem, and the label of a sample is represented by a regression value. While the label of sample in [51] is represented by a label vector, and the tracking problem is treated as a multi-class classification problem. Third, the proposed optimization algorithm utilizes a linear regression model with respect to LCC codes of samples to predict the labels of samples. The linear regression model is a globally linear approximation of original nonlinear function under nonlinear learning theory using LCC. But [51] and [53] predict the labels vectors of data points using a linear regression model with respect to original samples, which does not satisfy the basic theory assumption of our approach. All the Laplacian matrices in our method and theirs consider the manifold structure of samples, and the main idea behind them assumes that neighbor samples should have similar labels. However, the Laplacian matrix in our method includes a variable (sparse code matrix A) to be solved iteratively, while the Laplacian matrices in [51] and [53] are constant matrices. Additionly, the distance between different samples (B_{ij} in (5)) in our method has been continuously updated during the iteration. The learned sparse code for each sample reflects its real spatial position in the manifold space according to both discriminative information and local geometry structures of all data points. The optimization algorithm of the proposed model is introduced in the next section, and the proposed algorithm is summarized in Algorithm 1.

Algorithm 1 The Proposed Learning Algorithm

```
Input: ${(\mathbf{x}_i, y_i)}_{i=1}^{n}$, ${\mathbf{x}_i}_{i=n+1}^{n+u}$, $\mu, \lambda_1$ and $\lambda_2$.
Output: $\mathbf{D}$, $\mathbf{A}$ and $\mathbf{w}$.
Initialization: $\mathbf{D}$ is achieved using $k$-means algorithm, $\mathbf{A}_l=(\mathbf{D}^T\mathbf{D})^{-1}(\mathbf{D}^T\mathbf{X})$, and $\mathbf{X}=[\mathbf{x}_1,\cdots,\mathbf{x}_{n+u}]$.
$t=1$;
while $t<T$ do
    Classifier learning: Solve $\mathbf{w}$ using (7) with fixed $\mathbf{D}$, $\mathbf{A}$;
    Coding: Solve $\mathbf{A}$ by Algorithm 2 with fixed $\mathbf{D}$, $\mathbf{w}$;
    Dictionary learning: Learn $\mathbf{D}$ with fixed $\mathbf{A}$, $\mathbf{w}$ by (17);
    $t=t+1$.
end while
```

B. Optimization Algorithm

(5) could be solved directly, since it is not convex jointly over variables $\mathbf{D}$, $\mathbf{w}$, and $\mathbf{A}$. In this work, we seek to optimize one variable while fixing the other two ones. To this end, the objective function is decomposed into three sub-problems, and the optimal results will be acquired using an iterative way.

Sub-problem A: Classifier Learning. By fixing dictionary $\mathbf{D}$ and sparse code matrix $\mathbf{A}$, the following optimization problem is presented to learn classification function:

$$\min_{\mathbf{w}} \lambda_1||\mathbf{A}_l^T\mathbf{w}-\mathbf{y}||^2+\lambda_2\sum_{i=1}^{n+u}\sum_{j=1}^{n+u}\|\boldsymbol{\alpha}_i^T\mathbf{w}-\boldsymbol{\alpha}_j^T\mathbf{w}\|^2B_{ij}, \tag{6}$$

$$s.t. \quad \mathbf{1}^T\boldsymbol{\alpha}_i=1, \quad i=1,\cdots,n.$$

The optimal solution of $\mathbf{w}$ could be achieved by setting the derivative of (6) to zero. And the final closed-form solution is calculated as

$$\mathbf{w}=\left(\lambda_1\mathbf{A}_l\mathbf{A}_l^T+\lambda_2\mathbf{A}(\boldsymbol{\Delta}-\mathbf{A}^T\mathbf{A})\mathbf{A}^T\right)^{-1}(\lambda_1\mathbf{A}_l\mathbf{y}), \tag{7}$$

where $\boldsymbol{\Delta}=\text{diag}(\Delta_1,\Delta_2,\cdots,\Delta_{u+n})$ with $\Delta_i=\sum_{j=1}^{u+n}B_{ij}$.

Sub-problem B: Coding. To obtain the sparse codes matrix $\mathbf{A}$, we solve the objective function with $\mathbf{D}$ and $\mathbf{w}$ fixed. In fact, the resulting minimization problem is just (5). But it couldn't be solved directly by derivation, since this function is non-differentiable with regard to sparse codes matrix. Thus, locality-constrained linear coding [20] is introduced as an approximate formulation of (5), which could be solved analytically. In this work, we neglect the Laplacian regularization term here for simplicity. Let $\mathbf{c}_i=[c_i^1,\cdots,c_i^m]^T$ where $c_i^j=\|\mathbf{x}_i-\mathbf{d}_j\|$ denotes the Euclidean distance between dictionary entry $\mathbf{d}_j$ and sample $\mathbf{x}_i$. The approximated minimization problem could be formulated as

$$\min_{\mathbf{A}} \sum_{i=1}^{u+n}\left(\|\mathbf{x}_i-\mathbf{D}\boldsymbol{\alpha}_i\|^2+\mu\|\mathbf{c}_i\odot\boldsymbol{\alpha}_i\|^2\right)+\lambda_1||\mathbf{A}_l^T\mathbf{w}-\mathbf{y}||^2$$

$$s.t. \quad \mathbf{1}^T\boldsymbol{\alpha}_i=1, \quad i=1,\cdots,n+u, \tag{8}$$

where $\odot$ represents the Hadamard product. This minimization problem could be solved by calculating one column of $\mathbf{A}$

Algorithm 2 The Coding Algorithm

```
Input: ${(\mathbf{x}_i, y_i)}_{i=1}^{n}$, ${\mathbf{x}_i}_{i=n+1}^{n+u}$, $\mu, \lambda_1$, $\mathbf{D}$ and $\mathbf{w}$.
Output: $\mathbf{A}$
$t = 1$;
while $t < T$ do
    for $i = 1 : n + u$ do
        $\mathbf{P} = \mathbf{D}^T\mathbf{D} + \mu \text{diag}(\mathbf{c}_i^2)$;
        $\mathbf{Q} = \mathbf{D}^T\mathbf{x}_i$;
        if $i \leq n$ then
            $\mathbf{P} = \mathbf{P} + \lambda_1 \mathbf{w}\mathbf{w}^T$;
            $\mathbf{Q} = \mathbf{Q} + \lambda_1 \mathbf{w} y_i$;
        end if
        $\boldsymbol{\alpha}_i = \mathbf{P}^{-1}\left(\mathbf{Q} - \frac{\mathbf{1}^T\mathbf{P}^{-1}\mathbf{Q}\mathbf{1}-1}{\mathbf{1}^T\mathbf{P}^{-1}\mathbf{1}}\mathbf{1}\right)$;
    end for
    $t = t + 1$.
end while
```

with fixed others. The solving procedure is iterated until convergence. Then, each $\boldsymbol{\alpha}_i$ could be calculated analytically with a closed-form solution:

$$\boldsymbol{\alpha}_i = \mathbf{P}^{-1}\left(\mathbf{Q} - \frac{\mathbf{1}^T\mathbf{P}^{-1}\mathbf{Q}\mathbf{1}-1}{\mathbf{1}^T\mathbf{P}^{-1}\mathbf{1}}\mathbf{1}\right). \tag{9}$$

For the $\boldsymbol{\alpha}_i$s corresponding to sample $\mathbf{x}_i$s with labels,

$$\mathbf{P} = \mathbf{D}^T\mathbf{D} + \mu \text{diag}(\mathbf{c}_i^2) + \lambda_1 \mathbf{w}\mathbf{w}^T, \tag{10}$$

$$\mathbf{Q} = \mathbf{D}^T\mathbf{x}_i + \lambda_1 \mathbf{w} y_i, \tag{11}$$

where $(c_i^j)^2$ is the j-th element of diagonal matrix diag$(\mathbf{c}_i^2)$. For those $\boldsymbol{\alpha}_i$s corresponding to samples without labels,

$$\mathbf{P} = \mathbf{D}^T\mathbf{D} + \mu \text{diag}(\mathbf{c}_i^2), \tag{12}$$

$$\mathbf{Q} = \mathbf{D}^T\mathbf{x}_i. \tag{13}$$

The detailed derivation of sub-problem A is given in **Appendix A**. Algorithm 2 summarizes the proposed coding algorithm.

Sub-problem C: Dictionary Learning. To learn the dictionary $\mathbf{D}$, we aim to minimize the following problem with fixed $\mathbf{A}$ and $\mathbf{w}$

$$\min_{\mathbf{D}} \sum_{i=1}^{u+n} \left(\|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|^2 + \mu \sum_{j=1}^{m} |\alpha_i^j| ||\mathbf{d}_j - \mathbf{x}_i||^2 \right).$$

After derivation (refer to [22] for more details), the above minimization problem is equivalent to

$$\min_{\mathbf{D}} tr(\mathbf{D}^T\mathbf{D}\mathbf{G}) - 2tr(\mathbf{D}^T\mathbf{S}), \tag{14}$$

$$\mathbf{G} = \sum_{i=1}^{n+u} \left(\boldsymbol{\alpha}_i \boldsymbol{\alpha}_i^T + \mu \text{diag}(|\boldsymbol{\alpha}_i|) \right), \tag{15}$$

$$\mathbf{S} = \sum_{i=1}^{n+u} \left(\mathbf{x}_i \boldsymbol{\alpha}_i^T + \mu \mathbf{x}_i |\boldsymbol{\alpha}_i|^T \right), \tag{16}$$

where the trace operator $tr(\cdot)$ acts on a square matrix. The block-coordinate descent algorithm in [22] can be a viable approach to obtain the optimal dictionary. But in this work, it exists a closed-form formulation, which could be written as

$$\mathbf{D} = \mathbf{S}\mathbf{G}^{-1}. \tag{17}$$

The detailed derivation of sub-problem B is given in **Appendix B**. Actually, the dictionary entry could be regarded as labeled item, and the sign of its corresponding element in classifier $\mathbf{w}$ is just its label.

III. Tracking Approach

A. Samples Collection

Generally, the initial state of an interesting object in tracking methods is annotated in the first frame manually. In this work, we crop a group of holistic templates $\{\mathbf{x}_i\}_{i=1}^{n}$ within a certain scope of the object region randomly according to a Gaussian distribution as the labeled samples. Most discriminative visual tracking methods are usually seen as a binary classification problem, and the label of each sample is annotated as a discrete value, such as 0 or 1. For more accurate annotation, we set the labels of samples using continuous values which lie in $[0, 1]$ in our method. The label of a sample $\mathbf{x}_i$ is computed as

$$y_i = \frac{\text{Ar}_s \cap \text{Ar}_t}{\text{Ar}_s \cup \text{Ar}_t}, \tag{18}$$

where Ar$_t$ is the area of object region, and Ar$_s$ is the area of a template. It is observed that the similarity between the target and a sample increases with rise of the sample label value. The value of the label is 1 for a sample overlapped with the target region completely, and 0 if no overlap exists between them. It is reasonable because the samples drifting from the target are between target and non-target, and we could not assign them to 1 or 0 crudely. Thus, continuous labels are a good choice.

The optimization of (5) needs n labeled samples and u unlabeled samples, and we assign similar labels for neighbor samples in the manifold space. Thus, unlabeled samples are also needed to the proposed method. The target candidates $\{\mathbf{x}_i\}_{i=n+1}^{n+u}$ selected in the current frame within a certain scope of the previous target state are treated as unlabeled ones. Then we train a classifier to assign labels for target candidates using all these samples in an online manner.

B. Confidence Calculation

The training samples containing labeled and unlabeled ones collected in previous sub-section will be utilized to train the proposed model in (5), and then the optimal dictionary $\mathbf{D}$, LCC matrix $\mathbf{A}$ of samples and the linear classifier $\mathbf{w}$ will be obtained. The regression value of a target candidate $\mathbf{x}_i$ is calculated as $f(\mathbf{x}_i) = \boldsymbol{\alpha}_i^T\mathbf{w}$ where $\boldsymbol{\alpha}_i$ is the LCC of $\mathbf{x}_i$. It measures the affinity similarity of a candidate target to real object. Thus, we could obtain all the confidences of candidates.

Nevertheless, only global template considered in tracking is inadequate to cope with partial occlusion problem arisen in tracking. To handle that, the target region is separated to several small blocks, and several groups of samples of the blocks are obtained. We assign the label of different blocks as the way mentioned in previous sub-section. Meanwhile, we divide the target candidates in current frame into several

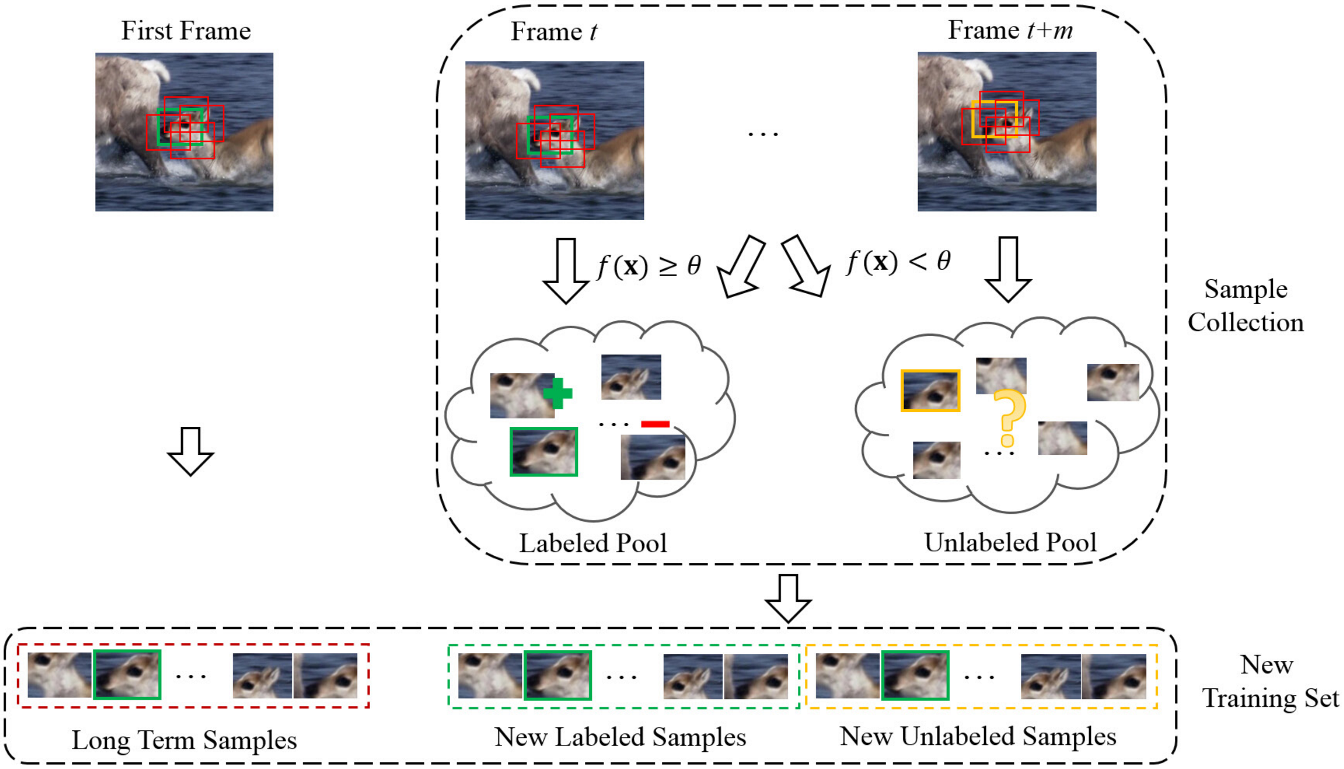

Fig. 1. Updating Scheme. The new training sample set is obtained from labeled and unlabeled pools. The samples collected from the first frame are involved in calculation all the time.

blocks as well, and they are used as unlabeled samples of different blocks. Denote $f_H(\mathbf{x}_i)$ and $\{f_B^j(\mathbf{x}_i^j)\}_{j=1}^b$ indicate the regression values of holistic candidate template $\mathbf{x}_i$ and its corresponding blocks respectively. Each block classifier f_B^j is trained using the block samples extracted from all the j-th blocks of holistic templates, thus different block dictionaries will be learned for different classifiers. Finally, the confidence value of a sample $\mathbf{x}_i$ is calculated as

$$f(\mathbf{x}_i) = \nu f_H(\mathbf{x}_i) + (1-\nu)\frac{1}{b}\sum_{j=1}^{b} f_B^j(\mathbf{x}_i^j), \tag{19}$$

where $\mathbf{x}_i^j$ is the j-th block of sample $\mathbf{x}_i$, and ν is the balance weight between holistic candidates and partial candidate templates. Those classifiers are retained every a few frames for computational efficiency. And the sparse codes of samples are obtained by executing (2) using the learned dictionary.

C. Particle Filter Framework

The proposed tracking algorithm is implemented under the particle filter framework. Given the observation $\mathbf{o}_{1:t} = \{\mathbf{o}_i\}_{i=1}^t$ up to time t, the maximum a posterior (MAP) estimation of object state $\mathbf{s}_t$ can be estimated by

$$\arg\max_{\mathbf{s}_t} p(\mathbf{s}_t \mid \mathbf{o}_{1:t}), \tag{20}$$

which is inferred based on the Bayesian theorem

$$p(\mathbf{s}_t|\mathbf{o}_{1:t}) \propto p(\mathbf{o}_t|\mathbf{s}_t)\int p(\mathbf{s}_t|\mathbf{s}_{t-1})p(\mathbf{s}_{t-1}|\mathbf{o}_{1:t-1})d\mathbf{s}_{t-1}, \tag{21}$$

where $p(\mathbf{s}_t|\mathbf{s}_{t-1})$ is the motion model and $p(\mathbf{o}_t|\mathbf{s}_t)$ the likelihood function. The posterior $p(\mathbf{s}_t|\mathbf{o}_{1:t})$ is approximated by a set of samples $\{\mathbf{s}_t^1, \mathbf{s}_t^2, \cdots, \mathbf{s}_t^N\}$ with their corresponding weights $\{w_t^1, w_t^2, \cdots, w_t^N\}$. The candidates are sampled from a proposal distribution $q(\mathbf{s}_t|\mathbf{s}_{1:t-1}, \mathbf{o}_{1:t}) = p(\mathbf{s}_t|\mathbf{s}_{t-1})$. In our tracking algorithm, the target motion between two consecutive frames is modeled by an affine image warp. And the state $\mathbf{s}_t$ is modeled by $(\xi_x, \xi_y, \theta, s, \eta, \phi)$ where (ξ_x, ξ_y) is the target center coordinate in the image and θ, s, η, ϕ the parameters of rotation angle, scale, aspect ratio and skew, respectively. Without loss of generality, a Gaussian distribution is used to model the motion model with $p(\mathbf{s}_t|\mathbf{s}_{t-1}) = \mathcal{N}(\mathbf{s}_t; \mathbf{s}_{t-1}, \Sigma)$ where Σ is a diagonal covariance matrix.

The likelihood function $p(y_i|\mathbf{s}_i)$ of candidate $\mathbf{x}_i$ is constructed by

$$p(o_i|\mathbf{s}_i) \propto f(\mathbf{x}_i). \tag{22}$$

The target candidate with highest probability is determined as current estimated target state.

D. Target Re-detection

It is common that drifting occurs during long term tracking and causes tracking failure. Especially, when a target is fully occluded by background, trackers are difficult to locate where a target is, and the estimated targets in these frames are almost random. Therefore, a target re-detection algorithm is necessary once the target reappears in new frames. Thus, we introduce the target re-detection strategy [27] to handle this problem. Different from [27], we train a SVM classifier [28] based on

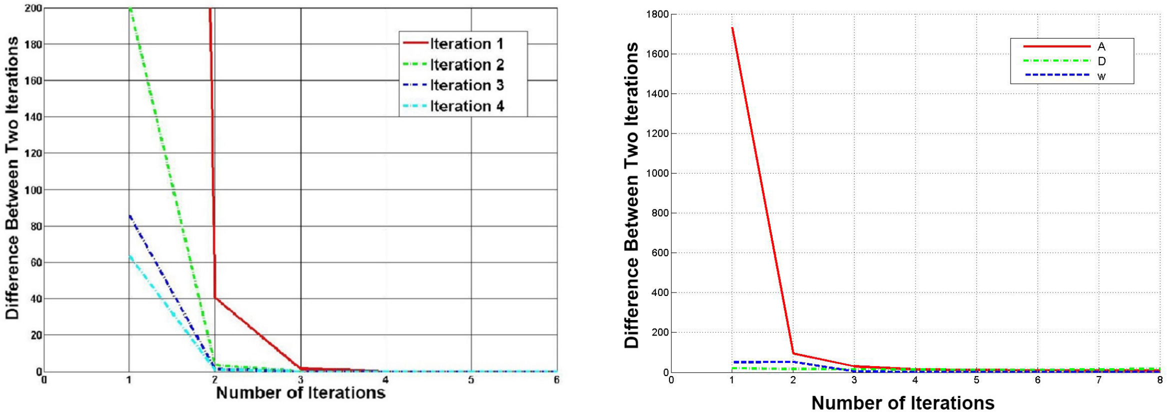

Fig. 2. Convergence of our coding algorithm (left) and the proposed joint discriminative dictionary, sparse codes and classifier learning algorithm (right).

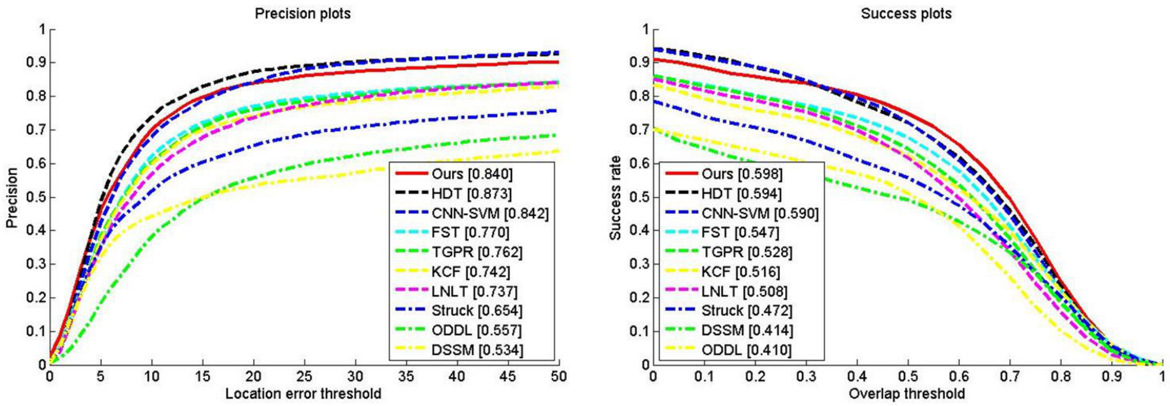

Fig. 3. Overall performance comparisons of precision plots (left side) and success rates (right side) for these trackers. The overall performance scores at 20 pixels are presented in the legend.

the initial target appearance instead of an online random fern classifier [29]. The re-detector is activated in case that the maximum response of the target center location is below a predefined threshold τ, which means that tracking fails in the current frame. With this re-detection strategy, the performance of our tracker is promoted as we can see in Section. IV.

E. Update Strategy

Target appearance changes continuously caused by illumination variations, occlusions, deformation, *etc*. during tracking. The manifold geometry structure constructed with samples will be different with the changing of appearance. Thus, we should re-update the dictionary, linear classifiers and sparse codes to adapt to appearance variation. The target re-detector should also be updated.

Dictionary Updating: We retain two sample pools as shown in Fig. 1 during tracking. The labeled samples are stored in labeled pool, and the unlabeled samples are contained in unlabeled ones. The labeled samples will be cropped based on current target location and then added into labeled pool when the confidence value of current target is greater than a constant θ. Otherwise, we will consider the candidates in current frame to be unlabeled samples and place them into unlabeled pool. Then, a certain amount of samples will be selected from these pools randomly every a few frames, and they are regarded as a new training sample set. The discriminative dictionary and classifier will be recalculated by Algorithm 1. To alleviate the pollution of current training set, we remain the samples collected from the first frame in our new training sample set, which is efficient for long term tracking. The updating scheme is applied on both holistic and block templates.

Re-detector Updating: In order to get more accurate re-detecting results, the re-detector should be updated during tracking as well. An online passive-aggressive algorithm [30] is applied to update the SVM re-detector using features sampled in the current frame. More details about the online SVM algorithm can be found in [30].

IV. Experiments

The proposed object tracking approach is verified on two challenging tracking benchmarks including OTB2013 [4] and VOT2015 [31]. We set the number of particles (600) to the same number of these tracking algorithms under framework of particle filters. The affine parameters of particle filters are set to $[10, 10, 0.04, 0, 0.001, 0]$. The number of positive, negative and unlabeled samples are set to 20, 200 and 200

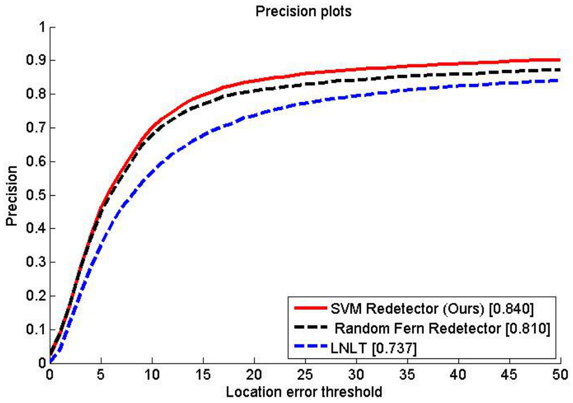

Fig. 4. Comparison results between our tracker using the proposed SVM re-detector, our tracker with the random fern re-detector [27], and the LNLT tracker [23].

respectively for the proposed learning algorithm. We then intended to separate the holistic template into four parts which were located in the upper left, upper right, bottom left, and bottom right of the template. Thus, the block size was set to half of template size. We normalize the size of holistic templates to 24×24, the block size and step size are set to 12×12 and $[12\ 12]$ respectively. The number of dictionary items for global templates is 20. The parameters μ, λ_1, and λ_2 in (5) are set to 0.03, 0.09, and 0.1 respectively. The balance factor is set to 0.8 to determine the impact of holistic and block classifiers. We set the threshold $\theta = 0.65$ for pool updating.

A. Implementation Details

We combine the three channels of CIE Lab color features [32] and the histograms of oriented gradients (HOG) feature [33] of samples as the final template feature vector. The feature in our experiments for each template is a vector combined by the three channels of CIE Lab color features [32] stretched row by row of the target and the histograms of oriented gradients (HOG) feature [33] of samples.

The final confidence in (19) of a target candidate is influenced by the weight ν. Thus, some inaccuracy may exist in the center location of the estimated target. To refine the estimated target state, we train a correlation filter [3], [34] using the holistic target template cropped in the initial frame. Different from their methods, the correlation filter is trained using only the target template without any background information in our implementation. Then, once a candidate state is chosen as the state of the current estimated target by (22), we resize the current target as the same size with target template and calculate a refined center location of confidence response map using the correlation filter learned ahead. The refined center location is determined as the position corresponding to the maximum response in this map. Other parameters such as rotation angle, scale, aspect ratio and skew, in the estimated target state are then transferred to the refined target. The correlation filter updating method is the same as [3].

B. Convergence Analysis

To verify the convergence of the proposed globally linear approximation to nonlinear learning algorithm intuitively, the iterations of the proposed algorithm are calculated. To verify the convergence of the coding algorithm with the increase of iteration number, we show the difference between two iterations on the experimental data in the left of Fig. 2. It can be found that the proposed algorithm converges rapidly. In fact, 4 rounds of iteration are enough for the experiments. The right sub-figure in Fig. 2 shows the convergence curve of the whole learning algorithm which is used to calculate the discriminative dictionary, sparse codes and linear classifier. All of these variables converge rapidly, and 8 rounds of iterations are needed at most. The values of these parameters are set to μ=0.03, λ_1=0.09, and λ_2=0.1 in our experiments.

C. Experimental Results on OTB2013

OTB2013 is a tracking benchmark [4] with 51 videos, where different difficulties encountered in visual tracking are contained. Our tracker is compared with 10 popular tracking methods, including Struck [35], KCF [3], FST [38], TGPR [39], two related methods (DSSM [37] and ODDL [17]), two deep learning trackers (CNN-SVM [40], HDT [42]), and our original version (LNLT) [23] is also included in the comparisons. We apply two measurements including distance precision (DP) and overlap precision (OP) by one-pass evaluation to evaluate the tracking performance. More details about measurements refer to [4].

Overall Estimation: The tracking results of our tracker are compared with those of different popular tracking approaches, and Fig. 3 shows the precision plots and success plots. We collect the center location errors on all sequences of all those tracking methods, and regard the performance scores at 20 pixels as the ranking criterion. The legend of precision plots shows the performance scores. We compute the areas under curves in the success plots as the performance scores of overlap rate. Visual tracking methods based on sparse representation are DSSM and ODDL among these trackers. In precision plots, Struck obtains the best score except for a few other popular tracking methods, which does good work on these image sequences. However, these trackers have no consideration on the embedding manifold spatial structure. Besides, dictionary learning is crucial for tracking methods, for ODDL obtains comparable tracking performance on this benchmark. The proposed tracking approach performs well on the benchmark and even better than the newly proposed works such as KCF and TGPR. The location error performance score is 0.840, and the overlap performance score is 0.598. In addition, all performance scores of the mentioned trackers are listed in the first two rows of Table I. It verifies the effectiveness of the proposed joint learning algorithm. Those deep learning [57] based trackers (such as CNN-SVM and HDT) generally learn their classifiers with a large number of training samples from different image datasets. These methods are easy to obtain appearance models with more discriminative power than that in other method, while our method only trains the proposed model with samples from several frames. In

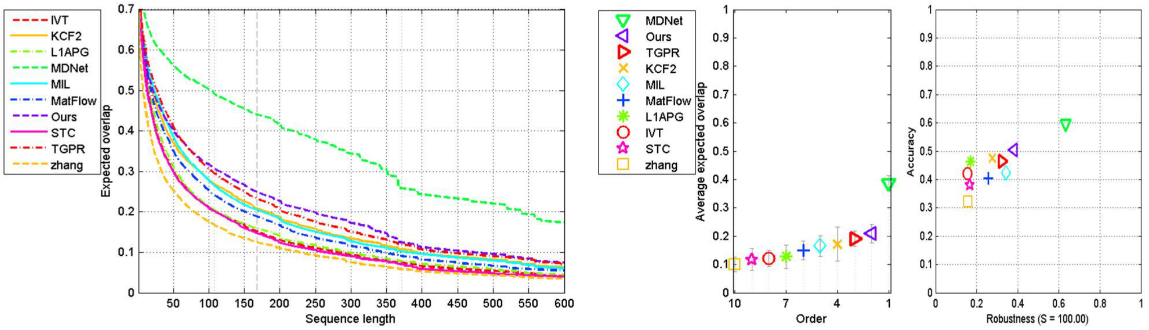

Fig. 5. Results on VOT2015. Left: the expected overlap curves. Middle: the expected overlap graph with the ranked trackers. The right-most tracker is the top-performing according to the VOT2015 expected average overlap values. The horizontal axis denotes the orders of different trackers. Right: the Accuracy Robustness (AR) plot for experiment baseline where the sensitive S= 100. For more details, please refer to [31].

TABLE I
THE PERFORMANCE SCORES FOR THE POPULAR TRACKERS ON OTB2013.

	HDT	DSSM	Struck	FST	CNN-SVM	ODDL	KCF	TGPR	WDL	LNLT	Ours
DP	0.873	0.533	0.654	0.770	0.842	0.561	0.742	0.762	0.689	0.737	0.840
OP	0.594	0.414	0.472	0.547	0.590	0.410	0.516	0.528	0.493	0.508	0.598

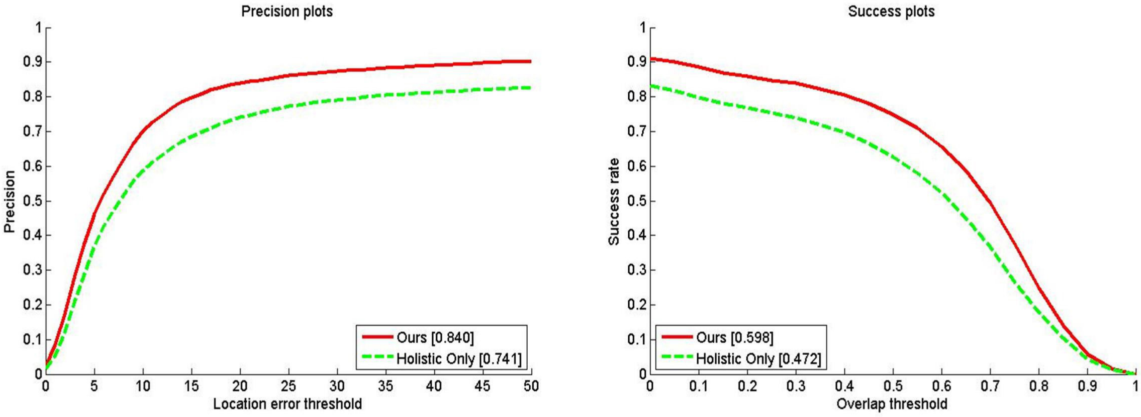

Fig. 6. Experiments comparison between our tracker and the one using holistic template only.

spite of this, our tracker still performs comparable results with CNN-SVM and HDT, and even better than them according to the overlap evaluation in Fig. 3.

Attributes-based Estimation: Different attributes encountered during tracking are annotated for these videos in the benchmark. Eleven attributes, such as background clutters, deformation, fast motion, illumination variation, in-plane rotation, motion blur, occlusion, out-of-plane rotation, low resolution, out-of-view and scale variation, are designed based on different challenging situations. The performance scores of different tracking methods estimated on this attributes to prove the effectiveness of them. As shown in Fig. 7, the success plots of all tracking algorithms under these challenges are presented the precision plots. The proposed tracking method performs well on most of these factors.

Analysis of Dictionary Learning: Our tracker is also compared with ourself without dictionary learning (denoted as WDL) to further prove the validation of discriminative dictionary learning. The parameter setting is consistent with the proposed tracking method. The performance scores of WDL are shown in Table I. The distance precision and overlap precision of this method are 0.689 and 0.493 respectively, which are inferior to the scores of the proposed tracking approach without target re-detection, which are 0.737 and 0.508 respectively. It is observed that the performance of the tracking method with discriminative dictionary learning performs better than the version without discriminative dictionary learning.

Re-detector Comparison: We compare our tracking performances using the introduced re-detection method and the random fern re-detector used in [27], respectively. As shown in Fig. 4, the DP value of our tracking algorithm on OTB2013 is 84.0% which is also better than the one using random fern classifier (81.0%). Compared with our original tracker (LNLT) [23], the introduced re-detection method improves the DP performance about 10 percent, while the random fern re-detector promotes about 8 percent. Thus, these two re-

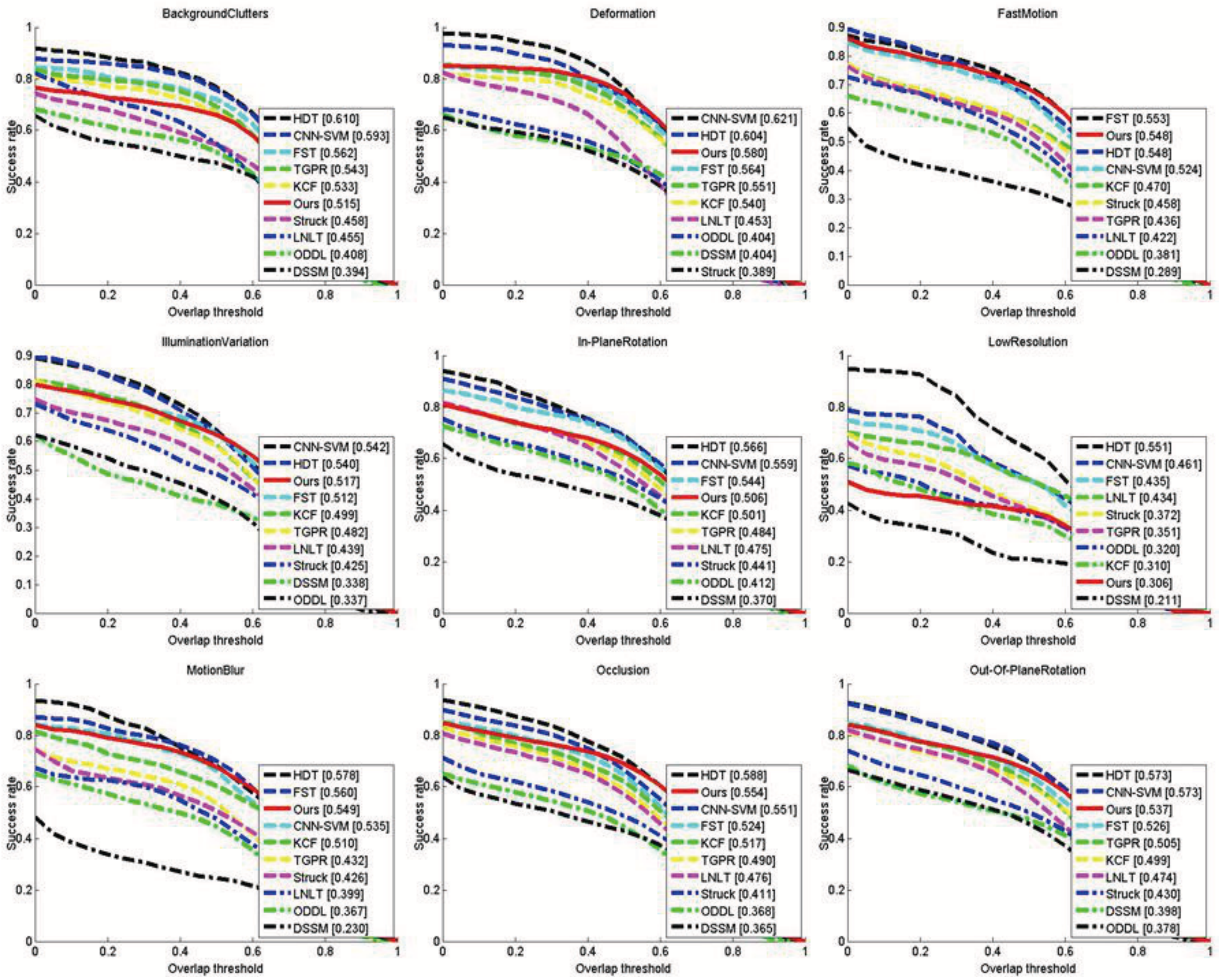

Fig. 7. Attribute-based estimation comparisons of success plots for these trackers.

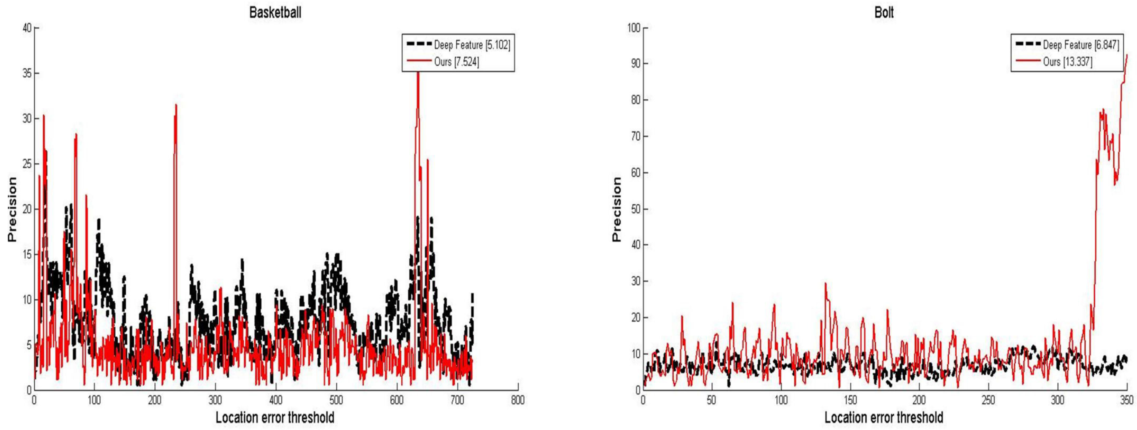

Fig. 8. Experiments with deep features. The average center location errors are shown in the legend.

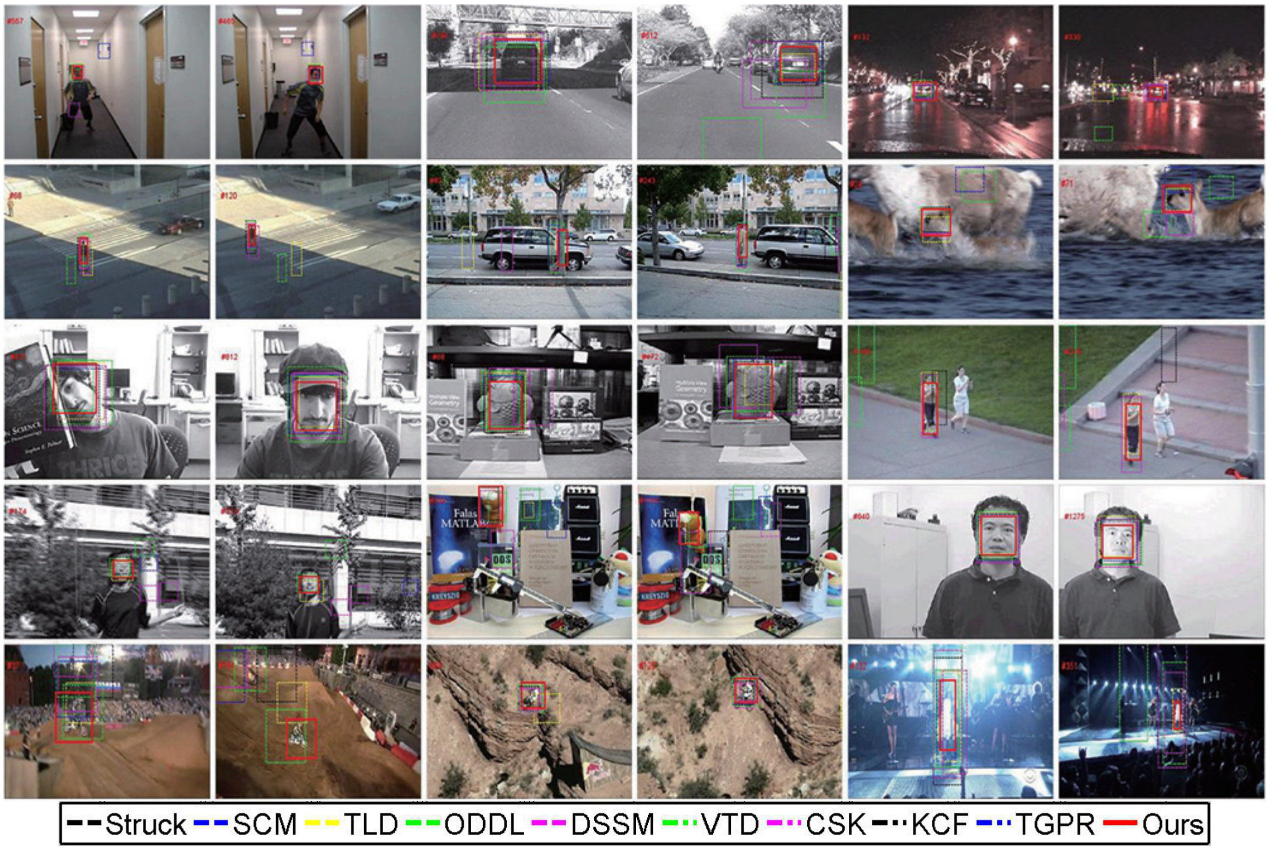

Fig. 9. Visual comparison. These sequences are ‘Boy’, ‘Car4’, ‘CarDark’, ‘Crossing’, ‘David3’, ‘Deer’, ‘Faceocc2’, ‘Fish’, ‘Jogging-1’, ‘Jumping’, ‘Lemming’, ‘Mhyang’, ‘MotorRolling’, ‘MountainBike’, and ‘Singer1’ from left to right and top to bottom.

detection methods both improve the final performance.

Effectiveness of Block Classifiers: We have also tested our tracker with only holistic templates. As shown in Fig. 6, our method with both holistic and block templates achieves better performance than the one with only holistic target templates. The DP score of our method is higher than that with only holistic templates about 10 percent and the OP score is higher than that about 12 percent. Thus, the block classifiers improve the final tracking performance.

We further test our tracking approach on a more challenging benchmark, the visual object tracking challenge 2015 (VOT2015) [31]. The performance of our tracker is compared with IVT [44], KCF2 [3], L1APG [13], MDNet [45], MIL [46], MatFlow [48], STC [50], TGPR [39], zhang (see A.4 in [31]) reported in this benchmark. In these experiments, each tracker is initialized with the ground truth bounding box, and it is also initialized by a perturbed bounding box centered around the ground truth bounding box randomly. The estimation toolkit [2] reports the final tracking results including accuracy and robustness. The accuracy measures the bounding box overlap ratio with ground truth, and the robustness assesses the number of failures which indicate when the overlap measure equals zero. As is shown in Fig. 5, we compare our tracking method with 16 popular tracking approaches in VOT2015 by the expected overlap curves, scores and AR plots of all these trackers. Our method performs better than these popular trackers on this more challenging dataset.

D. More Experiments and Visual Comparisons

Experiments with Deep Features. We have also applied the deep feature to our method. We extract convolutional feature from ResNet [57] for each particle including holistic and block templates. The pertained ResNet model “imagenet-resnet-50-dag” is used as deep features. Under the particle filter tracking framework, the computational complexity is related to the particle numbers. Two video clips including ‘Basketball’ and ‘Bolt’ are randomly tested to verify the effectiveness of deep feature. As shown in Fig. 8, deep feature improves the tracking performance. However, the average tracking speed of the deep feature version is 1200s per frame, while our original tracking speed is about 3s per frame. Our method is much faster than the one with deep feature.

We further show a part of the tracking results obtained by the proposed tracking approach and other trackers in Fig. 9. In the ‘Car4’, ‘CarDark’, and ‘Singer1’ sequences from OTB2013, the illumination of these targets change drastically. Benefiting from the discriminative appearance model, the proposed tracker is robust to illumination changes and can track these targets all the time. In ‘David3’ and ‘Faceocc2’, we show the tracking results on the videos where targets are confronted with heavy global and partial occlusions. For

[2] http://www.votchallenge.net/vot2015/

example, in ‘David3’ , the pedestrian suffers from global occlusion when he walks behind a tree (#83). TLD and CXT fail to track the target even when he is occluded by a lamp pole (#24). Struck and SCM also fail after the pedestrian walks away from the tree (#141). Only our tracker and TGPR could track the pedestrian in the whole sequence successfully. The appearances of some targets change caused by scale variation such as ‘Singer1’.

V. Conclusions

Based on the theory of globally linear approximations to nonlinear learning, a principled method has been presented to learn sparse codes, discriminative dictionary and nonlinear classifier jointly for visual tracking. We then introduce an optimization algorithm to calculate the discriminative dictionary, sparse codes and classifier iteratively. We develop a visual tracking method under the particle filter framework, and adopt an online updating scheme to adapt to target appearance changes. To further improve the tracking performance, the target re-detection strategy is introduced. Experiments on challenging video clips demonstrate the superior performance of the proposed method in comparison to popular trackers. In the future, we will attempt to extend our method to multiple target tracking with multi-task spare learning [56].

Appendix A
Derivations of Sub-Problem B

The approximated objective function can be written as

$$\min_{\mathbf{A}} \sum_{i=1}^{u+n} \left(\|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|^2 + \mu\|\mathbf{c}_i \odot \boldsymbol{\alpha}_i\|^2\right)$$

$$+ \lambda_1 ||\mathbf{A}_l^T \mathbf{w} - \mathbf{y}||^2 + \lambda_3 \sum_{i=1}^{n+u}\sum_{j=1}^{n+u} \|\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_j\|^2 B_{ij},$$

$$s.t. \quad \sum_{j=1}^{m} \boldsymbol{\alpha}_i^j = 1, \quad i = 1, \cdots, n, n+1, \cdots, n+u$$

where $\mathbf{c_i}$ represents the distance vector composed by i-th sample and all the dictionary items with its j-th elements $\mathbf{c}_i^j = \|\mathbf{x}_i - \mathbf{d}_j\|$.

Let

$$\begin{aligned} f &= \sum_{i=1}^{u+n} \left(\|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|^2 + \mu\|diag(\mathbf{c}_i)\boldsymbol{\alpha}_i\|^2\right) \\ &+ \lambda_1 \sum_{i}^{n} ||\boldsymbol{\alpha}_i^T \mathbf{w} - y_i||^2 + \nu(\sum_{i=1}^{n+u}(\boldsymbol{\alpha}_i^T \mathbf{1} - 1)) \\ &+ \lambda_3 \sum_{i=1}^{n+u}\sum_{j=1}^{n+u} \|\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_j\|^2 B_{ij} \end{aligned}$$

When $i \leq n$, we take the derivative of f with respect to $\boldsymbol{\alpha}_i$. Namely,

$$\begin{aligned} \frac{\partial f}{\partial \boldsymbol{\alpha}_i} &= 2\mathbf{D}^T(\mathbf{D}\boldsymbol{\alpha}_i - \mathbf{x}_i) \\ &+ 2\mu diag(\mathbf{c}_i^2)\boldsymbol{\alpha}_i + 2\lambda_1 \mathbf{w}(\mathbf{w}^T\boldsymbol{\alpha}_i - y_i) \\ &+ 4\lambda_3 \sum_{j=1}^{n+u}(\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_j)B_{ij} + \nu\mathbf{1}. \end{aligned}$$

Let the derivation be zero, and we denote

$$\mathbf{P} = \mathbf{D}^T\mathbf{D} + \mu diag(\mathbf{c}_i^2) + \lambda_1 \mathbf{w}\mathbf{w}^T + 2\lambda_3 \sum_{j=1}^{n+u} B_{ij},$$

$$\mathbf{Q} = \mathbf{D}^T\mathbf{x}_i + \lambda_1 \mathbf{w} y_i + 2\lambda_3 \sum_{j=1}^{n+u} \boldsymbol{\alpha}_j B_{ij}.$$

We can get the result of $\boldsymbol{\alpha}_i$ as

$$\boldsymbol{\alpha}_i = \mathbf{P}^{-1}\left(\mathbf{Q} - \frac{\mathbf{1}^T\mathbf{P}^{-1}\mathbf{Q}\mathbf{1} - 1}{\mathbf{1}^T\mathbf{P}^{-1}\mathbf{1}}\mathbf{1}\right).$$

$$When \quad i > n, \quad \mathbf{P} = \mathbf{D}^T\mathbf{D} + \mu diag(\mathbf{c}_i^2) + 2\lambda_3 \sum_{j=1}^{n+u} B_{ij},$$

$$\mathbf{Q} = \mathbf{D}^T\mathbf{x}_i + 2\lambda_3 \sum_{j=1}^{n+u} \boldsymbol{\alpha}_j B_{ij}.$$

Appendix B
Derivations of Sub-Problem C

The original problem is equivalent to

$$\min_{\mathbf{D}} \sum_{i=1}^{n+u} \left(\|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|^2 + \mu \sum_{j=1}^{m} |\alpha_i^j| \|\mathbf{d}_j - \mathbf{x}_i\|^2\right)$$

$$\Rightarrow$$

$$\min_{\mathbf{D}} \sum_{i=1}^{n+u} \left(\boldsymbol{\alpha}_i^T\mathbf{D}^T\mathbf{D}\boldsymbol{\alpha}_i - 2\mathbf{x}_i^T\mathbf{D}\boldsymbol{\alpha}_i + \mu\sum_{j=1}^{m} |\alpha_i^j| (\mathbf{d}_j^T\mathbf{d}_j - 2\mathbf{x}_i^T\mathbf{d}_j)\right)$$

$$\Rightarrow$$

$$\min_{\mathbf{D}} \sum_{i=1}^{n+u} \big(tr(\mathbf{D}^T\mathbf{D}\boldsymbol{\alpha}_i\boldsymbol{\alpha}_i^T) - 2tr(\mathbf{D}^T\mathbf{x}_i\boldsymbol{\alpha}_i^T)$$
$$+\mu tr(\mathbf{D}^T\mathbf{D}diag(|\boldsymbol{\alpha}_i|)) - 2\mu tr(\mathbf{D}^T\mathbf{x}_i|\boldsymbol{\alpha}_i|^T)\big)$$

$$\Rightarrow$$

$$\min_{\mathbf{D}} tr\left(\mathbf{D}^T\mathbf{D}\sum_{i=1}^{n+u}\left(\boldsymbol{\alpha}_i\boldsymbol{\alpha}_i^T + \mu diag(|\boldsymbol{\alpha}_i|)\right)\right)$$
$$- 2tr\left(\mathbf{D}^T\sum_{i=1}^{n+u}\left(\mathbf{x}_i\boldsymbol{\alpha}_i^T + \mu\mathbf{x}_i|\boldsymbol{\alpha}_i|^T\right)\right).$$

Let us denote

$$\mathbf{G} = \sum_{i=1}^{n+u}\left(\boldsymbol{\alpha}_i\boldsymbol{\alpha}_i^T + \mu diag(|\boldsymbol{\alpha}_i|)\right),$$

$$\mathbf{S} = \sum_{i=1}^{n+u}\left(\mathbf{x}_i\boldsymbol{\alpha}_i^T + \mu\mathbf{x}_i|\boldsymbol{\alpha}_i|^T\right).$$

By substituting $\mathbf{G}$ and $\mathbf{S}$ in above equation, the original objective is equivalent to minimizing

$$\min_{\mathbf{D}} tr(\mathbf{D}^T\mathbf{D}\mathbf{G}) - 2tr(\mathbf{D}^T\mathbf{S}).$$

The minimization problem has a closed-form solution by setting the derivative of above equation to zero. Finally, we get the final solution

$$\mathbf{D} = \mathbf{S}\mathbf{G}^{-1}.$$

References

[1] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, “Recent advances and trends in visual tracking: A review,” *Neurocomputing*, vol. 74, no. 18, pp. 3823–3831, 2011.

[2] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, “Visual tracking: an experimental survey,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442–1468, 2014.

[3] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, “High-speed tracking with kernelized correlation filters,” *IEEE Transa. on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[4] Y. Wu, J. Lim, and M.-H. Yang, “Online object tracking: A benchmark,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2411–2418, 2013.

[5] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, L. Čehovin, G. Nebehay, T. Vojíř, G. Fernandez, A. Lukežič, A. Dimitriev *et al.*, “The visual object tracking vot2014 challenge results,” in *European Conference on Computer Vision Workshops*, pp. 191–217, 2014.

[6] Z. Jiang, Z. Lin, and L. S. Davis, “Learning a discriminative dictionary for sparse coding via label consistent k-svd,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1697–1704, 2011.

[7] S. Zhang, H.,ao, X. Sun, and X. Lu, “Sparse coding based visual tracking: review and experimental comparison,” *Pattern Recognition*, vol. 46, no. 46, pp. 1772-1788, 2013.

[8] M. Yang, L. Zhang, X. Feng, and D. Zhang, “Fisher discrimination dictionary learning for sparse representation,” in *IEEE International Conference on Computer Vision*, pp. 543–550, 2011.

[9] L. Shao, L. Liu, and X. Li, “Feature learning for image classification via multiobjective genetic programming,” *IEEE Trans. on Neural Networks and Learning Systems*, vol. 25, no. 7, pp. 1359-1371, 2014.

[10] L. Liu, and L. Shao, “Sequential compact code learning for unsupervised image hashing,” *IEEE Trans. on Neural Networks and Learning Systems*, vol. 27, no. 12, pp. 2526-2536, 2016.

[11] X. Mei and H. Ling, “Robust visual tracking using l1 minimization,” in *IEEE International Conference on Computer Vision*, pp. 1436–1443, 2009.

[12] J. Shen, X. Yang, Y. Jia, X. Li, “Intrinsic images using optimization,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3481-3487, 2011.

[13] C. Bao, Y. Wu, H. Ling, and H. Ji, “Real time robust l1 tracker using accelerated proximal gradient approach,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1830–1837, 2012.

[14] W. Zhong, H. Lu, and M.-H. Yang, “Robust object tracking via sparsity-based collaborative model,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1838–1845, 2012.

[15] X. Lu, Y. Yuan, and P. Yan, “Robust visual tracking with discriminative sparse learning,” *Pattern Recognition*, vol. 46, no. 7, pp. 1762–1771, 2013.

[16] J. Shen, X. Yang, X. Li, Y. Jia, “Intrinsic images decomposition using optimization and user scribbles,” *IEEE Trans. on Cybernetics*, vol. 43, no. 2, pp. 425-436, 2013.

[17] F. Yang, Z. Jiang, and L. S. Davis, “Online discriminative dictionary learning for visual tracking,” in *IEEE Winter Conference on Applications of Computer Vision*, pp. 854–861, 2014.

[18] K. Yu, T. Zhang, and Y. Gong, “Nonlinear learning using local coordinate coding,” in *Advances in Neural Information Processing Systems*, pp. 2223–2231, 2009.

[19] K. Yu and T. Zhang, “Improved local coordinate coding using local tangents,” in *International Conference on Machine Learning*, pp. 1215–1222, 2010.

[20] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, “Locality-constrained linear coding for image classification,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3360–3367, 2010.

[21] Y. Lin, F. Lv, S. Zhu, M. Yang, T. Cour, K. Yu, L. Cao, and T. Huang, “Large-scale image classification: fast feature extraction and svm training,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1689–1696, 2011.

[22] B. Xie, M. Song, and D. Tao, “Large-scale dictionary learning for local coordinate coding,” in *British Machine Vision Conference*, pp. 1–9, 2010.

[23] B. Ma, H. Hu, J. Shen, Y. Zhang, and F. Porikli, “Linearization to nonlinear learning for visual tracking,” in *IEEE International Conference on Computer Vision*, pp. 4400–4407, 2015.

[24] W. Wang, J. Shen, X. Li, and F. Porikli, “Robust video object cosegmentation,” *IEEE Trans. on Image Processing*, vol. 24, no. 10, pp. 3137–3148, 2015.

[25] M. Belkin, P. Niyogi and V. Sindhwani, “Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples,” *Journal of Machine Learning Research*, vol. 7, pp. 2399–3434, 2006.

[26] X. Dong, J. Shen, D. Yu, W. Wang, J. Liu, and H. Huang, “Occlusion-aware real-time object tracking,” *IEEE Trans. on Multimedia*, vol. 19, no. 4, pp. 763-771, 2017.

[27] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, “Long-term correlation tracking,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5388–5396, 2015.

[28] J. A. Suykens and J. Vandewalle, “Least squares support vector machine classifiers,” *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[29] Z. Kalal, K. Mikolajczyk, and J. Matas, “Tracking-learning-detection,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.

[30] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, “Online passive-aggressive algorithms,” *The Journal of Machine Learning Research*, vol. 7, pp. 551–585, 2006.

[31] M. Kristan et al., “The Visual Object Tracking VOT2015 Challenge Results,” in *IEEE International Conference on Computer Vision Workshop (ICCVW)*, pp. 564-586, 2015.

[32] M. D. Fairchild, *Color appearance models*, John Wiley & Sons, 2013.

[33] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” *IEEE Trans.on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[34] J. Henriques, R. Caseiro, P. Martins, and J. Batista, “Exploiting the circulant structure of tracking-by-detection with kernels,” in *European Conference on Computer Vision*, pp. 702–715, 2012.

[35] S. Hare, A. Saffari, and P. H. Torr., “Struck: Structured output tracking with kernels,” in *IEEE International Conference on Computer Vision*, pp. 263–270, 2011.

[36] J. Shen, Y. Du, W. Wang, and X. Li, “Lazy random walks for superpixel segmentation,” *IEEE Trans. on Image Processing*, vol. 23, no. 4, pp. 1451-1462, 2014.

[37] B. Zhuang, H. Lu, Z. Xiao, and D. Wang, “Visual tracking via discriminative sparse similarity map,” *IEEE Trans. on Image Processing*, vol. 23, no. 4, pp. 1872–1881, 2014.

[38] S. Zhang, S. Zhao, Y. Sui and L. Zhang, “Single Object Tracking With Fuzzy Least Squares Support Vector Machine,” *IEEE Trans. on Image Processing*, vol. 24, no. 12, pp. 5723-5738, 2015.

[39] J. Gao, H. Ling, W. Hu, and J. Xing, “Transfer learning based visual tracking with gaussian processes regression,” in *European Conference on Computer Vision*, pp. 188–203, 2014.

[40] S. Hong, T. You, S. Kwak, and B. Han, “Online tracking by learning discriminative saliency map with convolutional neural network,” In *International Conference on Machine Learning*, 2015.

[41] W. Wang, J. Shen, and F. Porikli, “Saliency-aware geodesic video object segmentation,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3395-3402, 2015.

[42] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, and J. L. M.-H. Yang, “Hedged deep tracking,” In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[43] B. Ma, L. Huang, J. Shen, L. Shao, M-H. Yang, and F. Porikli, “Visual tracking under motion blur,” *IEEE Trans. on Image Processing*, vol. 25, no. 12, pp. 5867-5876, 2016.

[44] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, “Incremental learning for robust visual tracking,” *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125–141, 2008.

[45] H. Nam and B. Han, “Learning multi-domain convolutional neural networks for visual tracking,” In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[46] B. Babenko, M. H. Yang, and S. Belongie, “Robust object tracking with online multiple instance learning,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1619-1632, 2011.

[47] B. Ma, H. Hu, J. Shen, Y. Liu, and L. Shao, “Generalized pooling for robust object tracking,” *IEEE Trans. on Image Processing*, vol. 25, no. 9, pp. 4199-4208, 2016.

[48] M. E. Maresca and A. Petrosino, “Matrioska: A multi-level approach to fast tracking by learning,” in *International Conference On Image Analysis and Processing*, pp. 419C428, 2013.

[49] B. Ma, L. Huang, J. Shen, and L. Shao, “Discriminative tracking using tensor pooling,” *IEEE Trans. on Cybernetics*, vol. 46, no. 11, pp. 2411-2422, 2016.

[50] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang, “Fast visual tracking via dense spatio-temporal context learning,” In *European Conference on Computer Vision*, pp. 127C141, 2014.

[51] S. I. Yu, Y. Yang, and A. Hauptmann, "Harry potter's marauder's map: Localizing and tracking multiple persons-of-interest by nonnegative discretization," *IEEE Computer Vision and Pattern Recognition*, pp. 3714-3720, 2013.

[52] B. Ma, J. Shen, Y. Liu, H. Hu, L. Shao, and X. Li, "Visual tracking using strong classifier and structural local sparse descriptors," *IEEE Trans. on Multimedia*, vol. 17, no. 10, pp. 1818-1828, 2015.

[53] Y. Yang, F. Nie, D. Xu, J. Luo, Y. Zhuang, Y. Pan, "A Multimedia Retrieval Framework Based on Semi-Supervised Ranking and Relevance Feedback," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 723-742, 2012.

[54] J. Shen, D. Yu, L. Deng, and X. Dong, "Fast Online Tracking with Detection Refinement," *IEEE Trans. on Intelligent Transportation Systems*, doi://10.1109/TITS.2017.2750082, in press, 2017.

[55] L. Huang, B. Ma, J. Shen, L. Shao, and F. Porikli, "Visual tracking by sampling in part space," *IEEE Trans. on Image Processing*, vol. 26, no. 12, pp. 5800-5810, 2017.

[56] Y. Yang, Z. Ma, A. G. Hauptmann and N. Sebe, "Feature selection for multimedia analysis by sharing information among multiple tasks," *IEEE Trans. on Multimedia*, vol. 15, no. 3, pp. 661-669, 2013.

[57] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," In *IEEE Computer Vision and Pattern Recognition*, pp. 770-778, 2015.

Bo Ma Received the Ph.D. degree in computer science in 2003 from Harbin Institute of Technology, Harbin, China. In June 2006, he joined the Department of Computer Science, Beijing Institute of Technology, Beijing, China, where he is now an associate professor. He has published about 40 journal and conference papers such as *IEEE Transactions on Image Processing* and *IEEE ICCV*. His current fields of interest include statistical pattern recognition and computer vision.

Hongwei Hu is currently pursuing the PhD degree in the School of Computer Science, Beijing Institute of Technology, Beijing, China. His current research interests include visual object tracking algorithms.

Jianbing Shen (M'11-SM'12) is a Professor with the School of Computer Science, Beijing Institute of Technology, Beijing, China. He has published about 70 journal and conference papers such as *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Image Processing*, *IEEE CVPR*, and *IEEE ICCV*. His research interests include computer vision and machine learning. He serves as an Associate Editor of journal *Neurocomputing*.

Yuping Zhang is currently working toward the Ph.D. degree in the School of Computer Science, Beijing Institute of Technology, Beijing, China. His current research interests include efficient visual object tracking in videos.

Ling Shao (M'09-SM'10) is Ling Shao is the Chief Scientist of JD Artificial Intelligence Research (JDAIR), Beijing, China, and is also a professor with the School of Computing Sciences, University of East Anglia, Norwich, UK. His research interests include Computer Vision, Image Processing, Pattern Recognition and Machine Learning. He is an Associate Editor of *IEEE Transactions on Image Processing*, *IEEE Transactions on Neural Networks and Learning Systems*. He is a Fellow of the IET.

Fatih Porikli is an IEEE Fellow and a Professor with the Research School of Engineering, Australian National University, Canberra, ACT, Australia. He has contributed broadly to object and motion tracking, and video analytics. Prof Porikli was the recipient of the R&D 100 Scientist of the Year Award in 2006. He has won 4 best paper awards at premier IEEE conferences including the Best Paper Runner-Up at IEEE CVPR. He serves as the Associate Editor of 5 premier journals including *IEEE Signal Processing Magazine* and *IEEE Trans. on Multimedia*.