Robust Object Tracking Using Manifold Regularized Convolutional Neural Networks

Hongwei Hu, Bo Ma, *Member, IEEE*, Jianbing Shen, *Senior Member, IEEE*, Hanqiu Sun, *Member, IEEE*, Ling Shao, *Senior Member, IEEE*, and Fatih Porikli, *Fellow, IEEE*

***Abstract*—In visual tracking, usually only a small number of samples are labeled, and most existing deep learning based trackers ignore abundant unlabeled samples that could provide additional information for deep trackers to boost their tracking performance. An intuitive way to explain unlabeled data is to incorporate manifold regularization into the common classification loss functions, but the high computational cost may prohibit those deep trackers from practical applications. To overcome this issue, we propose a two-stage approach to a deep tracker that takes into account both labeled and unlabeled samples. The annotation of unlabeled samples is propagated from its labeled neighbors first by exploring the manifold space that these samples are assumed to lie in. Then, we refine it by training a deep convolutional neural network using both labeled and unlabeled data in a supervised manner. Online visual tracking is further carried out under the framework of particle filters with the presented manifold regularized deep model being updated every few frames. Experimental results on different tracking datasets demonstrate that our tracker outperforms most existing tracking approaches. The source code and results are available at: *https://github.com/shenjianbing/MRCNNTracking*.**

***Index Terms*—Convolutional neural networks, deep learning, deep tracker, manifold regularization, object tracking, online tracking.**

I. Introduction

As a fundamental topic in multimedia processing, visual tracking could be used in different practical applications, especially human-computer interaction, autonomous cars, biomedical image analysis and video surveillance [4], [7], [28], [44]. It is a challenging problem due to target appearance changes caused by illumination variation, occlusion, motion blur, background clutter, etc. Visual tracking has been studied for decades, and researchers have developed and designed various tracking algorithms to handle those challenges [11], [17], [21], [27], [38]. Recently, deep Convolutional Neural Networks (CNNs), being a powerful representation of visual data, have been applied to various multimedia processing topics [31], such as object detection, and image classification. Deep CNNs have also been used in visual tracking and achieve outstanding tracking performance.

Actually, the use of deep networks in visual tracking is never a smooth ride. The deficiency of labeled data in visual tracking limits the application of deep CNNs to the tracking field. The main reason is that obtaining an efficient deep model needs large-scale annotated data. However, in visual tracking, the annotated samples are cropped from the first frame in a video, which are too few to train the complex CNNs with millions of parameters. To handle this problem, two types of CNNs based tracking methods have been proposed from the perspective of sample collection. The first type in [22], [32] adopts the CNNs trained on object detection or image classification datasets to the tracking field and regard them as a feature extraction tool. However, it may not be suitable for the tracking problem by transferring deep models from a non-tracking field. Moreover, a large-scale dataset, which could train a CNN model for visual tracking, is still unavailable. And it needs tremendous efforts to annotate a benchmark for training deep trackers. Hence, the second type of trackers [24], [29] collect training samples from the barely available annotated tracking benchmarks and consider the CNN as a classifier to distinguish the target from the complex background. The above deep methods are trained with labeled samples to achieve impressive tracking performance, but they ignore the unlabeled samples which are the most common data in visual tracking. In this paper, we attempt to make an effective use of those unlabeled and labeled samples together for accurate target location.

Visual data, including labeled and unlabeled samples, on a high dimensional space often lie on a manifold with a smaller

Manuscript received December 25, 2017; revised April 23, 2018; accepted June 22, 2018. Date of publication July 25, 2018; date of current version January 24, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61472036, in part by the Beijing Natural Science Foundation under Grant 4182056, in part by the Australian Research Council's Discovery Projects funding scheme under Grant DP150104645 and in part by Specialized Fund for Joint Building Program of Beijing Municipal Education Commission. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Xavier Giro-i-Nieto. *(Corresponding author: Bo Ma.)*

H. Hu and B. Ma are with the Beijing Laboratory of Intelligent Information Technology, School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China (e-mail: huhongwei@bit.edu.cn; bma000@bit.edu.cn).

J. Shen is with the Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 100081, China, and also with the Inception Institute of Artificial Intelligence, Abu Dhabi, United Arab Emirates (e-mail: shenjianbing@bit.edu.cn).

H. Sun is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: hanqiu@cse.cuhk.edu.hk).

L. Shao is with the Inception Institute of Artificial Intelligence, Abu Dhabi, United Arab Emirates, and also with the School of Computing Sciences, University of East Anglia, Norwich NR5 8HZ, U.K. (e-mail: ling.shao@uea.ac.uk).

F. Porikli is with the Research School of Engineering, Australian National University, Canberra, ACT 0200, Australia (e-mail: fatih.porikli@anu.edu.au).

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

Digital Object Identifier 10.1109/TMM.2018.2859831

dimensionality than the original space [40]. To exploit the geometry of a high dimensional sample space, manifold regularization has been proposed by combining labeled and unlabeled data into a unified semi-supervised learning framework [3]. It has been widely applied in many vision fields, such as data fusion and classification. The deep learning algorithms under a manifold regularization framework [15], [33] are introduced as well. But manifold regularized CNNs have not been generalized to the tracking problem. The major issue is that the training procedure of semi-supervised CNNs is time-consuming, which is not suitable for online visual tracking. To handle this problem, Lee [18] proposed pseudo labels for unlabeled samples to train an efficient semi-supervised deep model, which is equivalent to entropy regularization [19]. Inspired by the aforementioned methods, we generate the deduced labels for unlabeled data points under the manifold assumption and introduce an efficient tracker using CNNs.

We introduce a manifold regularization item in the loss function of CNNs by considering the unlabeled samples arising during tracking, but it will be a time-consuming strategy for online tracking. Therefore, by deducing labels for unlabeled samples according to the manifold structure of the sample space, we could learn a deep model in a supervised manner approximately instead of the complex semi-supervised problem. Motivated by these facts, we introduce a two-stage approach to the deep tracker that takes both labeled and unlabeled samples into account as shown in Fig. 1. The presented tracking approach offers the following three-fold contributions:

- To better exploit both labeled and unlabeled samples, visual data in a manifold space are assumed to have similar labels with their neighbors. This is formulated as a graph Laplacian regularization with a Gaussian random field. The labels of unlabeled samples are calculated by the weighted average of its neighbors using this Gaussian harmonic function.
- Those unlabeled samples with the deduced labels are used to update a pre-trained CNN along with labeled samples. The deduced labels of unlabeled data only relying on the structural space may be inaccurate. Thus, we train a CNN model using the deduced samples as part of training samples and output a more accurate tag for each unlabeled one.
- Our tracker is carried out under the framework of particle filters with online updated CNNs. The proposed tracking approach is then verified on several public tracking benchmarks and achieves better tracking results.

II. Related Works

A large number of visual tracking algorithms [28], [39] have been proposed over decades. They are typically classified into generative and discriminative trackers. It is out of the scope of this work to comprehensively review those tracking methods. We only review the mostly related works, and a more comprehensive overview can be found in [28].

Traditional discriminative tracking methods utilize handcrafted representations such as CIE-Lab and HOG to model the target appearance in general. For example, Henriques *et al.* [10], [12] employed the HOG features to model the target and locate the object center with kernelized correlation filters. Haar-like features [20] were exploited in Struck [9] to perform online tracking using the kernelized structured output SVM. Ning *et al.* [25] used the original features of the color image to formulate the final sample vectors and trained a discriminative classifier using dual linear structured SVM. The SOWP descriptors [14] were designed by combining spatially ordered features to extract multiple local patches for accurate visual tracking based on the Struck tracker. Ma *et al.* [2] also used HOG features to establish the final target representation. Qi *et al.* [48] used the spare codes the represents a target. Those hand-crafted features often rely on professional knowledge and are developed for specialized issues, which means they may not generalize well in the tracking problem.

As the latest feature learning and classifier training techniques, CNNs have been gradually used in visual tracking approaches. Ma *et al.* [22] extracted CNN features for each image patch from VGG-Net trained on ImageNet, and learned multiple linear correlation filters using different features obtained from different convolutional layers. The tracking results were decided by combinations of the response maps calculated by those correlation filters. Wang *et al.* [32] proposed a tracking method using fully convolutional networks, and designed a feature map selection approach to determine discriminative features for tracking. Qi *et al.* [26], [34] presented a tracking method to construct several weak trackers based on correlation filters, where each one was trained using the CNN features extracted from different layers. Wang *et al.* [46] investigated different strategies to tackle the limited training samples in online tracking. The CNN here was pre-trained on VGG-Net, and the adaptive hedging method was applied to ensemble these weak trackers into a stronger one. But the models obtained from other vision tasks may be inappropriate for visual tracking. Nam and Han [24] learned a representation for multi-domain learning based CNNs using a large number of training samples from VOT [16] or OTB [39]. Bertinetto *et al.* [5] trained a fully-convolutional end-to-end Siamese network for visual tracking on the ILSVRC15 dataset [30]. Tao *et al.* [29] constructed a Siamese deep CNN whose inputs are image pairs for visual tracking. They adopted the ALOV dataset [28] to train and test their approach on OTB. These methods all neglect the abundant unlabeled samples available in a visual tracking problem.

Manifold regularization based tracking methods aim to make full use of both labeled and unlabeled data. Bai and Tang [1] proposed a tracking method using online ranking support vector machines, where weakly labeled data in the current frame were used to update their appearance model. Wang *et al.* [45] also proposed a Laplacian regularization method to propagate foreground labels. But the manifold regularized CNNs based tracking method has not been well studied.

III. Label Propagation using Gaussian Fields Harmonic Functions

Now, we focus on the label propagation of unlabeled samples from labeled ones on an embedding manifold space using Gaussian fields harmonic. Suppose that we are given l labeled

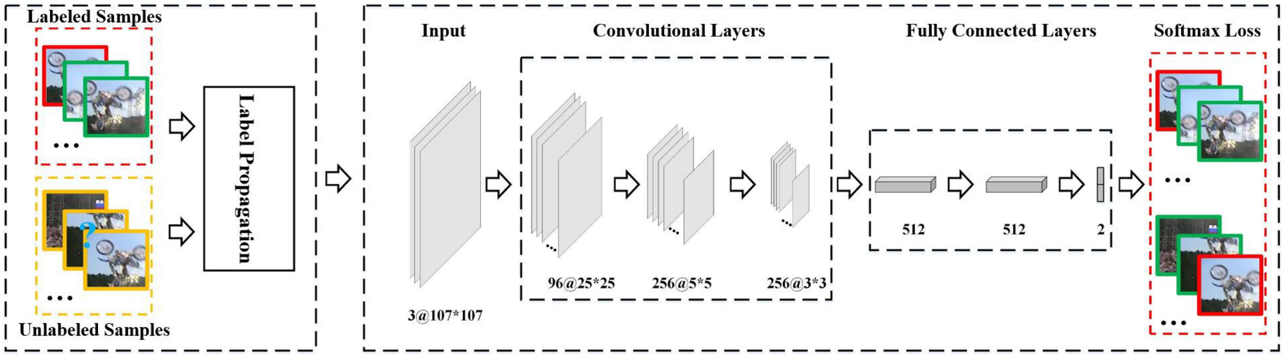

Fig. 1. The proposed architecture for manifold regularized CNNs, where both accurately labeled samples and unlabeled samples with deduced labels are used to train this network. The image patches surrounded with red rectangles indicate positive samples, and the green ones represent negative samples. The unlabeled samples are surrounded with yellow rectangles.

samples $\{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^d, y_i \in \{0,1\}\}_{i=1}^{l}$ where $\mathbf{x}_i$ is the d-dimensional data point and y_i is its corresponding label, and we obtain u unlabeled samples $\{\mathbf{x}_i\}_{i=l+1}^{l+u}$. A graph Laplacian indicates that neighboring samples should share similar labels on the manifold space, which could be formulated as

$$\min_{y_{l+1},\ldots,y_{l+u}} \sum_{i=1}^{l+u}\sum_{j=1}^{l+u} w_{ij}(y_i - y_j)^2, \tag{1}$$

where w_{ij} denotes the similarity weight between samples $\mathbf{x}_i$ and $\mathbf{x}_j$. Generally, the similarity is only relevant to the spatial distance between a sample pair. In practice, we calculate the weights of two samples only if they lie in the k nearest neighbors of each other, i.e.,

$$w_{ij} = \begin{cases} \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\delta), & \mathbf{x}_i \in \mathcal{N}_k(\mathbf{x}_j) \\ \exp(-\|\mathbf{x}_j - \mathbf{x}_i\|^2/\delta), & \mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_i) \\ 0, & \text{otherwise} \end{cases}, \tag{2}$$

where $\mathcal{N}_k(\mathbf{x}_i)$ is a set including the k nearest neighbors of sample $\mathbf{x}_i$. Eq. (1) can be reorganized as

$$\min \operatorname{tr}\left(\mathbf{y}^T \mathbf{L}\mathbf{y}\right), \tag{3}$$

where $\mathbf{y} = [\mathbf{y}_l^T \quad \mathbf{y}_u^T]^T$ with $\mathbf{y}_l = [y_1, y_2, \ldots, y_l]$ represents the labels of labeled samples and $\mathbf{y}_u = [y_{l+1}, \ldots, y_{l+u}]$ denotes the labels of unlabeled samples. And the Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is calculated from similarity matrix $\mathbf{W}$ whose elements are w_{ij}, and the diagonal elements of $\mathbf{D}$ are obtained by $[D]_{ii} = \sum_{j=1}^{l+u} w_{ij}$.

To deduce $\mathbf{y}_u$, Eq. (3) could be reformulated as

$$\min_{\mathbf{y}_u} \operatorname{tr}\left(\begin{bmatrix} \mathbf{y}_l^T & \mathbf{y}_u^T \end{bmatrix} \begin{bmatrix} \mathbf{L}_{ll} & \mathbf{L}_{lu} \\ \mathbf{L}_{ul} & \mathbf{L}_{uu} \end{bmatrix} \begin{bmatrix} \mathbf{y}_l \\ \mathbf{y}_u \end{bmatrix} \right), \tag{4}$$

where $\mathbf{L}_{ll}$ denotes the Laplacian matrix of labeled samples, and $\mathbf{L}_{uu}$ is the Laplacian matrix of all unlabeled samples, and $\mathbf{L}_{lu} = \mathbf{L}_{ul}^T$ represents the Laplacian matrix for labeled and unlabeled samples. The above minimization problem could be solved based on Gaussian fields harmonic functions [43]. Assuming that we have a function f to predict the label y_i of sample $\mathbf{x}_i$, it is formulated as $f(\mathbf{x}_i) = y_i$. The function f is assigned with a Gaussian field probability distribution

$$p(f) = \frac{\exp(-\beta E(f))}{\int_{f|L=f_l} \exp(-\beta E(f))df}, \tag{5}$$

where β is a constant parameter, L indicates the set of labeled samples, and the constraint $L = f_l$ indicates that the predictions of labeled samples calculated by function f should be consistent with their true labels, and $E(f) = \sum_{i=1}^{l+u}\sum_{j=1}^{l+u} w_{ij}(f(\mathbf{x}_i) - f(\mathbf{x}_j))^2$. It can be proved that the optimal prediction function $f = \arg\min_{f|L=f_l} E(f)$ is harmonic, i.e., $\Delta f = 0$ for unlabeled data set where Δ denotes the combinatorial Laplacian operation, and $f = f_l$ for labeled samples [37]. This property means that the label of each unlabeled sample is the weighted average of its labeled neighbors.

Thus, the optimal label vector $\mathbf{y}_u^*$ for unlabeled sample is calculated as

$$\mathbf{y}_u^* = \mathbf{P}\mathbf{y}_l, \tag{6}$$

where $\mathbf{P} = -\mathbf{L}_{uu}^{-1}\mathbf{L}_{ul}$.

From this equation, the matrix $\mathbf{P}$ can be viewed as a projection matrix for label propagation. And $\mathbf{y}_u$ could be regarded as the deduced label vector for an unlabeled sample. But this label vector may not be accurate enough since it only takes the spatial structure of the embedding manifold into consideration and neglects the discriminative information of labeled samples. Thus, we train CNNs using both the spatial structure constituted by all samples and the discriminative information of labeled samples in the next section.

IV. Manifold Regularized CNNs for Online Visual Tracking

A. Model Overview

As provided in the left-side of Fig. 1, we show the proposed manifold regularized CNNs architecture trained with labeled and unlabeled data, where the first three convolutional layers of this network are taken from VGG-net with a resized RGB input. Followed by the convolutional layers, we append three fully connected layers. In our settings, different convolutional layers are usually connected by ReLUs, normalization, and max pooling, and fully connected layers are connected by ReLUs and dropout. The softmax loss is applied as the loss function to train this network for separating targets from the complex background. The design of this network is different from existing deep networks for tracking. Firstly, the present network receives both accurately labeled samples and unlabeled samples with the deduced labels as its inputs, which takes full advantage of the abundant unlabeled samples arising during tracking. Secondly, the network model needs no change for both pre-training and updating, and it makes this deep model more convenient to be applied in tracking. Besides, we treat the samples from different video sequences equally, where the shared information of them is expected to be learned using this deep model. Finally, the single binary classification layer in the last layer is suitable for tracking, since visual tracking is modeled as a binary classification problem in most tracking-by-detection algorithms.

B. Network Update

We aim to train a manifold regularized CNN to distinguish the target and the background. As one of the loss functions, softmax loss has been widely used in many deep neural networks. Its main role is equal to a softmax layer and a multinomial logistic loss layer. We introduce a loss function considering both labeled and unlabeled data. For labeled samples $\{(\mathbf{x}_i, \mathbf{y}_i) \mid \mathbf{x}_i \in \mathbb{R}^d, \mathbf{y}_i \in \{0,1\}^C\}_{i=1}^{l}$ where C is the number of sample categories. In our settings, $C = 2$ indicates the two sample classifications including target and non-target. And $\mathbf{y}_i$ is a label vector with one non-zero element only. The softmax loss in classical deep neural networks is written as

$$\mathcal{L}_l = -\frac{1}{l}\sum_{i=1}^{l}\sum_{j=1}^{C} y_i^j \log\left(p_j(\mathbf{x}_i)\right), \tag{7}$$

where $p_j(\mathbf{x}_i)$ represents the j-th softmax output of sample $\mathbf{x}_i$, and y_i^j the j-th element of label vector $\mathbf{y}_i$.

In order to utilize the unlabeled samples $\{\mathbf{x}_i\}_{i=l+1}^{l+u}$, we take the deduced labels calculated in Section III to participate in the network training. Let $\{\mathbf{y}_i^* \in \{0,1\}^C\}_{i=l+1}^{l+u}$ indicate the label vector of these unlabeled samples using Eq. (6). Thus, a regularization term is defined after Eq. (7), i.e.,

$$\mathcal{L} = (1-\lambda)\mathcal{L}_l + \lambda\mathcal{L}_u, \tag{8}$$

$$\mathcal{L}_u = -\frac{1}{u}\sum_{i=l+1}^{l+u}\sum_{j=1}^{C} y_i^{*j} \log\left(p(\mathbf{x}_i)\right), \tag{9}$$

where y_i^{*j} denotes the j-th element of the deduced label vector for unlabeled sample $\mathbf{x}_i$ calculated by Eq. (6). The function in Eq. (8) is named as a manifold regularized softmax loss function, and its derivative with respect to $\mathbf{x}_i$ is ready to be obtained. If $\mathbf{x}_i$ denotes the labeled samples, only partial derivative of $\mathcal{L}_l$ should be computed, and $\mathcal{L}_u$ otherwise. Let f be the softmax function, and $\mathbf{z}_i \in \mathbb{R}^C$ be the input of softmax loss with respect to sample $\mathbf{x}_i$, i.e., $p_j(\mathbf{x}_i) = f(z_i^j)$ where z_i^j represents the j-th element of $\mathbf{z}_i$.

The final derivative consists of the two parts, and it is balanced by a preset constant factor λ, i.e.,

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial z^j} &= (1-\lambda)\frac{1}{l}\sum_{i=1}^{l}\sum_{j=1}^{C} y_i^j \left(f(z_i^j) - 1\right) \\ &\quad + \lambda\frac{1}{u}\sum_{i=l+1}^{l+u}\sum_{j=1}^{C} y_i^{*j} \left(f(z_i^j) - 1\right).\end{aligned} \tag{10}$$

The CNN model designed in Section IV-A is therefore trained with both labeled and unlabeled samples with the back propagation method in a supervised manner. The presented manifold regularized CNNs are trained by the min-batch stochastic gradient descent method. The labeled training samples are cropped from previous frames. And we take the candidates sampled in the current frame around the previous target location as unlabeled samples. In fact, the number of these samples is quite small in contrast to the complex deep neural networks. It not only may cause overfitting when training the whole network using these samples, but also increases the computational complexity which makes it impractical for online tracking. Thus, we only update the weights of the last three fully connected layers for computational efficiency and keep the weights of other layers unchanged.

C. Offline Pre-training

The three convolutional layers of this network are taken from VGG-net trained with classification data (ImageNet), which may not be appropriate for visual tracking problems. Therefore, we collect training samples from VOT2015 [16], and learn the model parameters with these samples. The positive and negative samples are sampled from every annotated frame in the datasets. The positive samples and negative samples are collected in each frame from the training dataset. The samples are collected around the target region according to a Gaussian distribution, and we choose 50 of those samples whose overlap with ground-truth is larger than a fixed threshold as positive samples and 200 of those ones with their overlap rates that are smaller than a preset constant as negative samples. The network in Fig. 1 is only pre-trained with labeled samples, and all the parameters in it will be updated including the convolutional layers and the fully connected layers. However, this may cause the training ambiguity when sample different targets from the same sequence (like Jogging-1,2). The weights in the convolutional layers are initialized using the corresponding part in VGG-net, and the parameters in the fully connected layers are preset randomly. We employ the softmax loss mentioned in Eq. (7). This model is also trained using the min-batch stochastic gradient descent method.

D. Tracking Approach

The introduced visual tracking method is implemented under the framework of particle filters [23]. The goal is to estimate the maximum of a posterior of an object state $\mathbf{s}_t$ which is modeled by the target central coordinate, height and width, in frame t. It

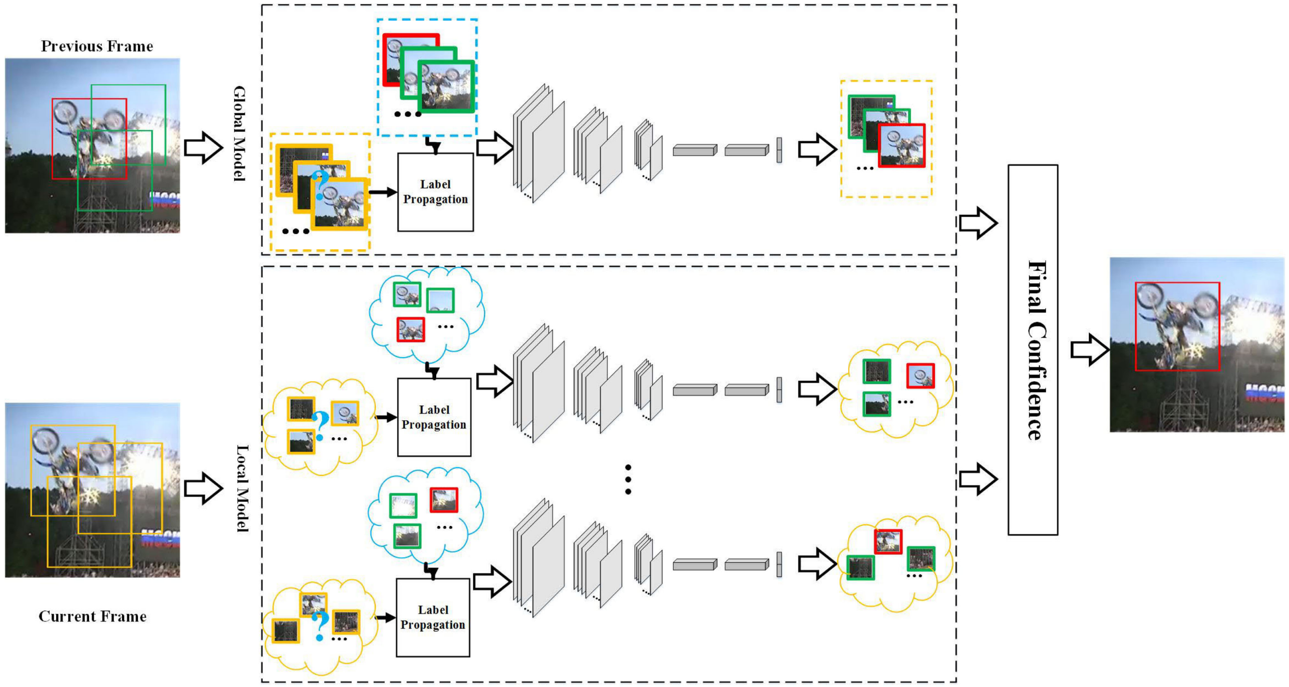

Fig. 2. The tracking flow chart. Global templates are divided into several image patches, and each image part is assigned with a CNN model. The image patches surrounded with red rectangles indicate positive samples, and the green ones represent negative samples. The unlabeled samples are surrounded with yellow rectangles.

can be formulated as

$$\arg\max_{\mathbf{s}_t} p(\mathbf{s}_t \mid \mathbf{o}_{1:t}), \tag{11}$$

where $\mathbf{o}_{1:t} = \{\mathbf{o}_1, \ldots, \mathbf{o}_t\}$ represents the observation sample set up to frame t. According to the Bayesian theorem, the maximum a posterior is proportional to

$$p(\mathbf{o}_t \mid \mathbf{x}_t) \int p(\mathbf{s}_t \mid \mathbf{s}_{t-1}) p(\mathbf{s}_{t-1} \mid \mathbf{o}_{1:t-1}) d\mathbf{s}_{t-1}. \tag{12}$$

where the motion model $p(\mathbf{s}_t \mid \mathbf{s}_{t-1})$ is modeled by a Gaussian distribution, and the posterior $p(\mathbf{s}_t \mid \mathbf{o}_{1:t})$ is approximated by a set of sampled candidates based on an assumed proposal distribution which is the same as the motion model.

We take the positive score in the outputs of manifold regularized CNNs as the likelihood value of a candidate. To be more specific, we crop a set of candidate samples whose positions obey a Gaussian distribution with the previous target state as its mean value. We first update the proposed manifold regularized CNN model with both candidates and labeled samples. And then, the candidates are transferred into the updated deep model, and a confidence value is obtained for each candidate. We could choose the candidate with the highest confidence as the current target. The tracking flow is summarized in Algorithm 1.

Algorithm 1: Manifold Regularized CNNs Based Online Visual Tracking Algorithm

```
Input: The pre-trained CNN model, the target state $\mathbf{s}_1$ of the
       first frame in a video.
Output: The target states $\mathbf{s}_2, \ldots, \mathbf{s}_n$ of subsequent frames.
Crop positive and negative samples $\mathcal{X}_l = \{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$
  in the first frame.
for $t = 2 : n$ do
    Sample target candidates $\mathcal{X}_u = \{\mathbf{x}_i\}_{i=1+1}^{l+u}$ in the $t$-th
      frame around target state $\mathbf{s}_{t-1}$.
    Calculate the deduced the labels for target candidates
      using Eq. (6).
    Update the last three layer of CNNs with manifold
      regularized loss in Eq. (8) with Eq. (10) using both
      the labeled/unlabeled samples $\{\mathcal{X}_l, \mathcal{X}_u\}$ until
      convergence.
    Assign confidence value for each candidate using the
      updated CNNs.
    Determine the current target state $\mathbf{s}_t$ with the highest
      confidence using Eq. (11).
    Collect positive and negative samples
      $\mathcal{X}_l = \{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ based on current target state.
end for
```

Occlusion Handling Strategy: It may not be enough to handle local target appearance changes caused by occlusion during tracking, if we only establish the target model with holistic templates. Hence, we further model global and local target appearances at the same time. As shown in Fig. 2, the holistic templates are divided into several overlapping image parts, and different image parts corresponding to different relative positions on the holistic templates are collected. And then, we train several different CNNs for every local target model with those local samples. Since we only update the parameters in the fully connected layers, the time complexity of tracking is acceptable. Suppose that the global template $\mathbf{z}_g$ is separated into k image patches $\{\mathbf{z}_b^{(1)}, \mathbf{z}_b^{(2)}, \ldots, \mathbf{z}_b^{(k)}\}$, their corresponding confidence

TABLE I
THE DETAILED PARAMETERS FOR EACH LAYER OF THE PROPOSED MANIFOLD REGULARIZED CNNS. (THE 'CONV' INDICATES THE CONVOLUTIONAL LAYER, AND 'MAXPOOL' IS THE MAX POOLING LAYER, AND 'FC' REPRESENTS THE FULLY CONNECTED LAYER. THE SYMBOL '-' MEANS THAT NO PARAMETERS EXIST IN THIS FIELD)

Layer	Kernel Size	Feature Maps	Stride	Padding
conv1	7× 7	96	2	0
maxpool1	3× 3	-	2	0
conv2	5× 5	256	2	0
maxpool2	3× 3	-	2	0
conv3	3× 3	512	1	0
fc4	-	512	-	-
fc5	-	512	-	-
fc6	-	2	-	-

scores s_g and $\{s_b^{(1)}, s_b^{(2)}, \ldots, s_b^{(k)}\}$ are obtained with global and local CNN models. The final confidence s_f of a candidate can be computed as

$$s_f = (1-\gamma)s_g + \gamma\frac{1}{k}\sum_{i=1}^{k} s_b^{(i)}, \tag{13}$$

where γ is a constant to balance the impacts of global and local models.

Once a target is partially occluded, the confidence of occluded parts will decrease, but the confidence of unoccluded parts still maintain high scores, which is helpful for target localization accurately. Finally, the candidate with the highest confidence is selected as the current target. In our implementation, we divided the target into four local patches for efficient computation. We believe that the max pooling or ranked-pooling in [35] will improve the tracking performance especially in case of larger number of local separations. The structure and training procedure of local networks are just similar with the holistic networks, except for the training samples. The training samples of local networks are local patches cropped from a target.

V. EXPERIMENTAL RESULTS

The proposed tracking method is verified on two popular visual tracking datasets, namely, OTB [36] and TB-100 [39]. The manifold regularized CNNs are trained with samples collected from VOT2015 [16], and implemented based on MatConvNet toolbox. Our tracker runs about 1.2 fps using the unoptimized Matlab implementation on an Intel(R) Core i7 CPU with 3.5 GHz and GeForce GTX 1080 GPU.

A. Implementation Details

The detailed parameters of each layer for the proposed manifold regularized CNNs are shown in Table I. Besides, the rectified linear units (ReLU) and local response normalization (LRN) are added after 'conv1' and 'conv2'. And 'conv3' is only followed by the ReLU layer. The 'fc4' and 'fc5' layers are followed by ReLU and the dropout layer.

Our tracker runs under the particle filter framework, and the covariances of the target location and scale are set to $(0.6, 0.6, 1.0)$, which is formulated as a diagonal matrix for Gaussian distribution. For each frame, 50 positive samples and 250 negative samples are collected which means that $l = 300$, and the number of candidates $u = 400$ is the number of particles. Three color channels of CIE Lab are combined row by row of the target as the sample feature. As described in Algorithm 1, the manifold regularized CNNs are updated every frame. But in our implementation, we update this model every three frames, and we could collect more labeled samples to update our deep model for more efficient tracking. Additionally, a bounding box regression technique [8] is utilized to improve the accuracy of the target location. Refer to [24] for more details about the bounding box regression.

B. Results on OTB

OTB [36] contains 50 videos, and each video is annotated by different types of difficulties including out-of-view, occlusion, fast motion, motion blur, and illumination variations. It is one of the most commonly used tracking benchmarks to verify the tracking performance.

We compare the proposed tracking method with several classical trackers such as MDNet [24], Struck [9], WCO [47], CSK [10], SCM [42], KCF [12], and MEEM [41] and some recently proposed deep trackers including HCF [22], HDT [26], SINT [29], CNN-SVM [13] and DeepSRDCF [6]. For performance evaluation, the tracking results from different trackers are estimated by distance precision (DP) and overlap precision (OP) using one-pass evaluation. DP measures the relative number of frames when the center location error is smaller than a threshold (20 in general) in a video, while OP measures the percentage of frames that the overlap rate between ground truth and the tracked bounding box is larger than a threshold. More details about these measurements can be found in [36]. The pre-training videos are collected from VOT2015 [16], which excludes the sequences arising in OTB.

The DP and OP comparisons on OTB are illustrated in Fig. 3, and the performance of the proposed tracker is better than that of existing deep learning based trackers (such as HCF and HDT), as well as traditional trackers. The DP score of our method is 0.931, which is 27.7 percent higher than the Struck tracker. The OP score on the success plots for the proposed tracking method is 0. HCF, HDT, CNN-SVM and SINT perform roughly the same on DP and OP scores, which have a small gap with the proposed tracking method.

To test the performance under different scenes, we compare them according to several annotated attributes. Fig. 4 illustrates the ranking of different tracking algorithms under eight attributes: deformation, occlusion, illumination variation, out-of-plane rotation, in-plane rotation, out-of-view, motion blur, and scale variation. These tracking algorithms are ranked by DP and OP scores on the OTB dataset, and the detailed scores are shown in the legends of each subfigure. Overall, deep learning based tracking methods, such as MDNet, DeepSRDCF, HCF, and HDT, behave better than traditional trackers on these attributes. The proposed method performs the best on the total eight attributes compared with those popular trackers.

C. Results on TB-100

TB-100 [39], as an extended tracking dataset of OTB, adds about 50 annotated videos. We compare the proposed

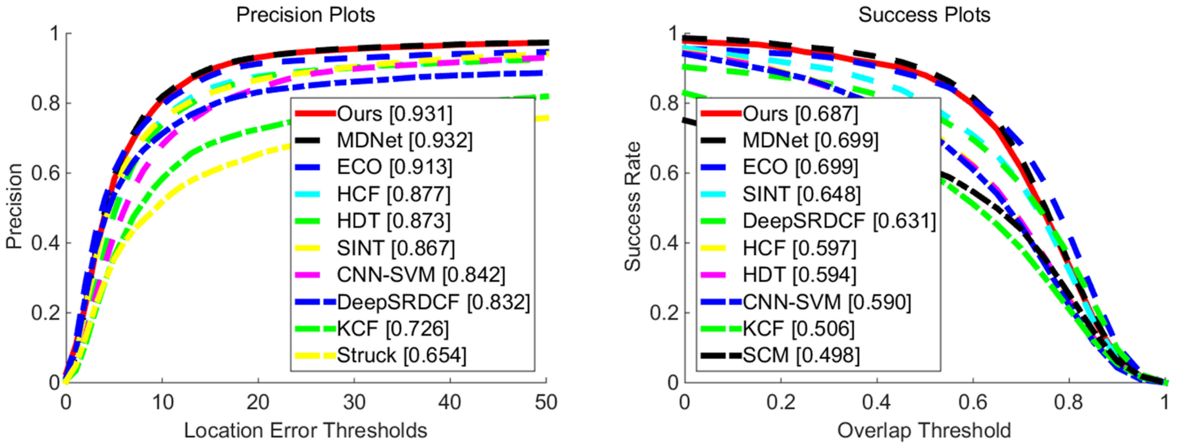

Fig. 3. The comparison with state-of-the-art results on OTB [36]. The left subfigure shows the distance precision evaluation and the right one shows the overlap precision estimation.

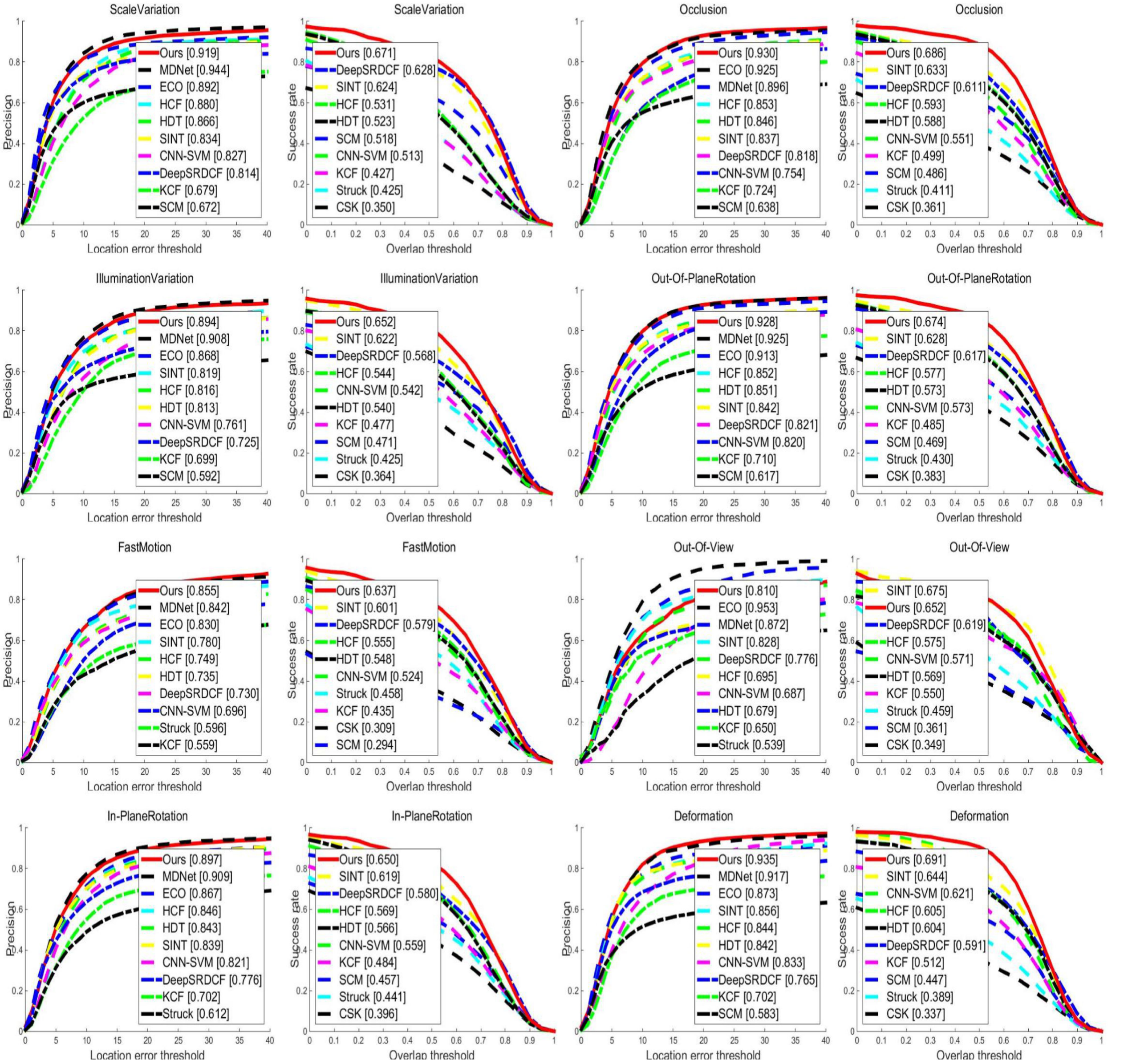

Fig. 4. The comparison with state-of-the-art results on OTB [36] for attribute-based estimation, including deformation, occlusion, illumination variation, out-of-plane rotation, in-plane rotation, out-of-view, motion blur, and scale variation from left to right and up to bottom.

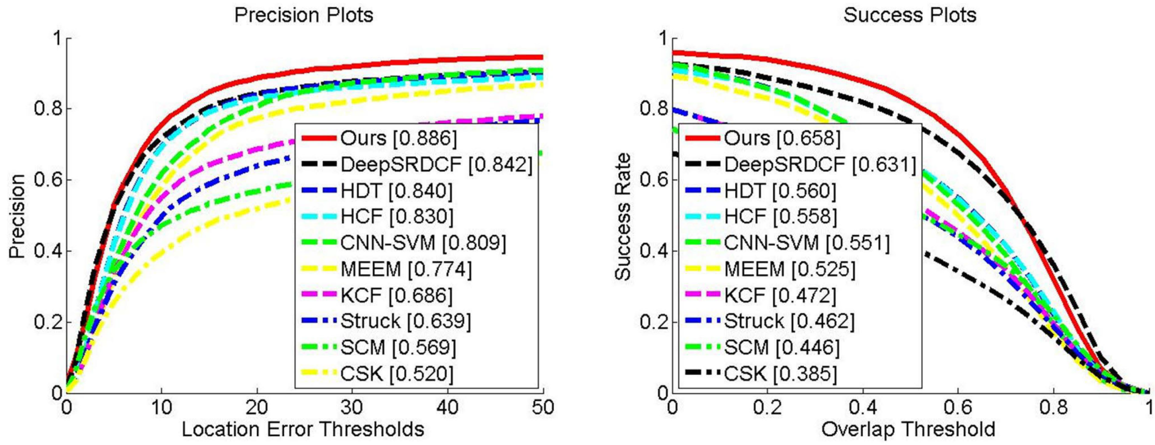

Fig. 5. The state-of-the-art comparison results on TB-100 [39]. The left subfigure shows the distance precision evaluation and the left one shows the overlap precision estimation.

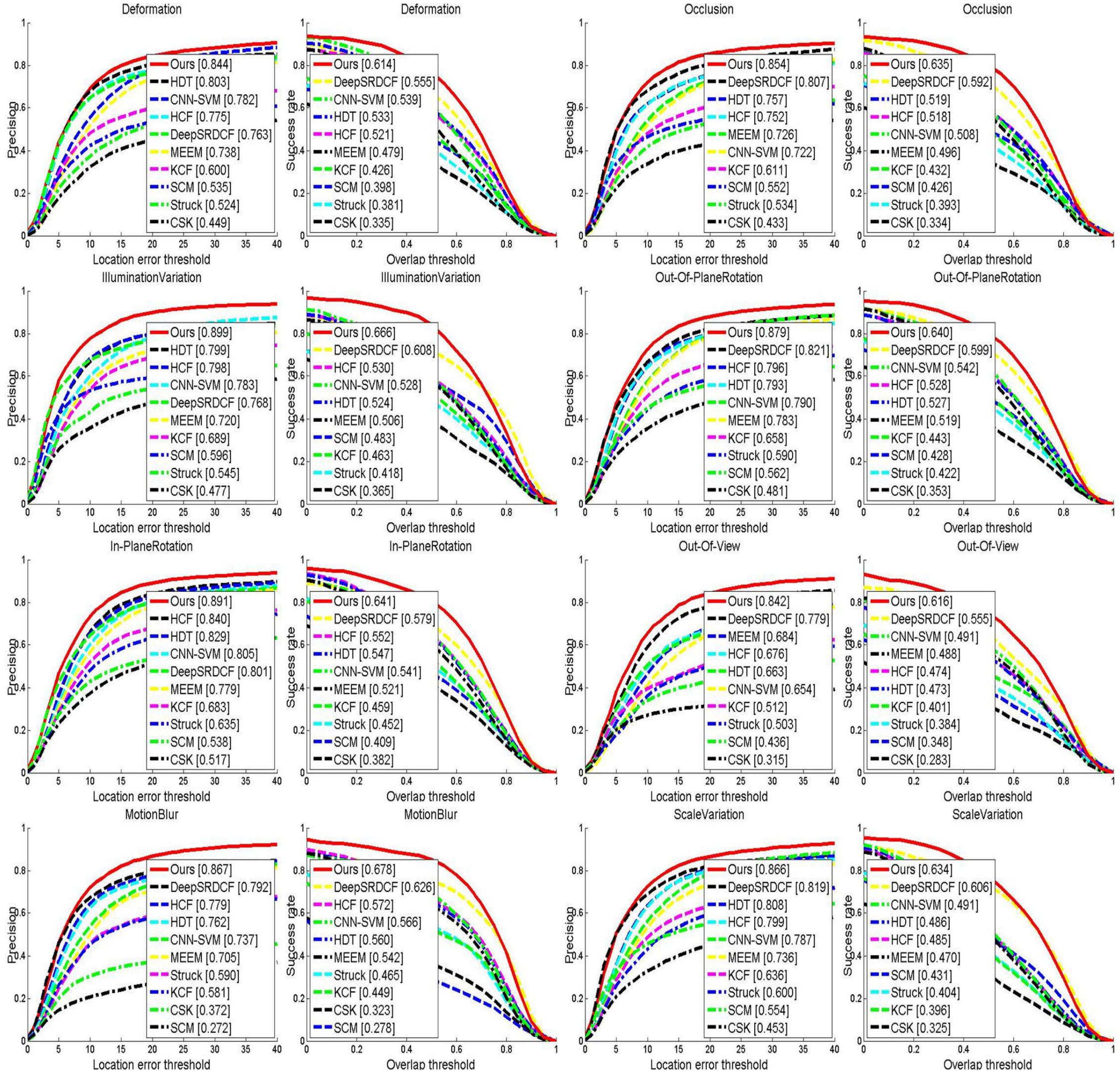

Fig. 6. The comparison with state-of-the-art results on TB-100 [39] for attribute-based estimation, including deformation, occlusion, illumination variation, out-of-plane rotation, in-plane rotation, out-of-view, motion blur, and scale variation from left to right and up to bottom.

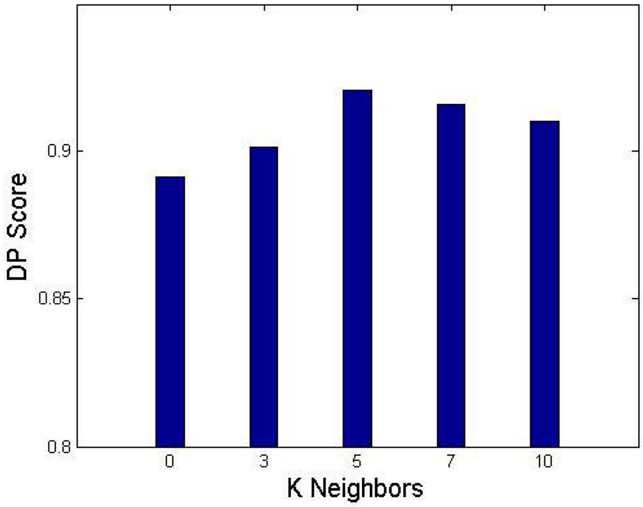

Fig. 7. Comparisons with different numbers of nearest neighbors k.

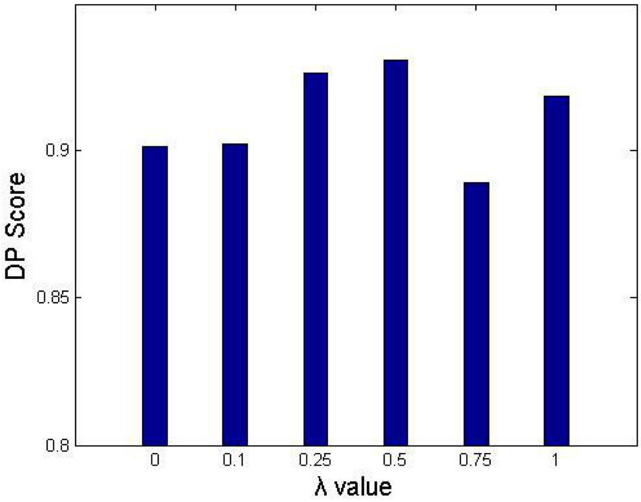

Fig. 8. The analysis of global and local models.

tracking method with the aforementioned popular trackers. The pre-training videos are collected from VOT2015 [16], which excludes the sequences arising in TB-100.

To verify our performance on a larger video dataset, we compare the proposed approach on TB-100 with those aforementioned tracking algorithms using the same evaluation criteria and the same pre-trained deep CNN model. Fig. 5 shows the overall estimations for those trackers on this benchmark. The DP and OP scores of our method are 0.886 and 0.658 respectively, which are higher than those of the compared deep trackers, such as HDT, HCF, CNN-SVM, and those traditional trackers. The DeepSRDCF, HDT, HCF, CNN-SVM tracking methods perform well among them, which demonstrates that deep learning could improve the performance greatly.

In Fig. 6, we also compare our method with those popular trackers under deformation, occlusion, illumination variation, out-of-plane rotation, in-plane rotation, out-of-view, motion blur, and scale variation. These tracking algorithms are ranked by DP and OP scores on the TB-100 dataset, and the detailed scores are shown in the legends of each subfigure. From these figures, we find that our tracker behaves better on these attributes than both deep trackers and conventional trackers.

D. Component Analysis

Effectiveness of Manifold Regularization: To verify the effectiveness of the manifold regularization item in the proposed method, we perform the comparisons of our tracker with different numbers of nearest neighbors in Eq. (2) on OTB. As shown in Fig. 7, the DP scores (vertical axis) of our method with neighbors $k = 0, 3, 5, 7, 10$ (horizontal axis) are illustrated in the histogram. It is noted that the performance will be improved when the manifold regularization item is removed, and the experimental results are shown in Fig. 7. From the figure, the worst tracking performance is produced using our method with $k = 0$, which means that the proposed manifold regularization item works in our setting. The highest DP score is obtained when $k = 5$. Hence, we choose 5 as the number of nearest neighbors in our tracking algorithm. The performance declines with the increase of the number of neighbors, which means that a suitable number of neighbors benefits the tracking performance. In principle, a larger number of neighbors will bring more information in a local region of a sample, and it will violate the assumed manifold structure. Therefore, $k = 5$ is the best choice for the spatial structure of manifold and tracking performance through our experiments.

TABLE II
THE PERFORMANCE COMPARISON RESULTS WITH AND WITHOUT OFFLINE PRE-TRAINING METHOD ON OTB

	Without Pre-training	Without Bounding Box Regression	Ours
DP	0.357	0.902	0.931
OP	0.293	0.668	0.687

Influence of Global and Local Models: To test the influence of global and local models, we implement the experimental comparisons of our tracker with the increase of λ in Eq. (13) from 0 to 1 on OTB. As shown in Fig. 8, the DP scores (vertical axis) of our method with $\lambda = 0, 0.1, 0.25, 0.75, 1$ (horizontal axis) are illustrated in the histogram. From the figure, the best tracking performance is obtained when $\lambda = 0.5$, which is 0.931 for DP score. Hence, we set the value of λ to 0.5 of Eq. (13). Therefore, the local model could bring some performance improvement. But the drawback is that the tracking time increases with the rise of image patches.

Effectiveness of Pre-training and Bounding Box Regression Strategy: We also carry out a comparison experiment with the proposed tracking algorithm without offline pre-training strategy presented in Section IV-C and the bounding box regression Strategy. As shown in Table II, a CNN model without pre-training (only random weights in different layers) based tracker performs pretty bad. For a complex CNN, training samples extracted from the first frame only are far from enough. Thus, the DP and OP scores are pretty low, and most targets could not be located accurately. The bounding box regression strategy is useful for our tracker to improve the tracking performance. The DP and OP scores increase about 2.9 and 1.1 percent respectively.

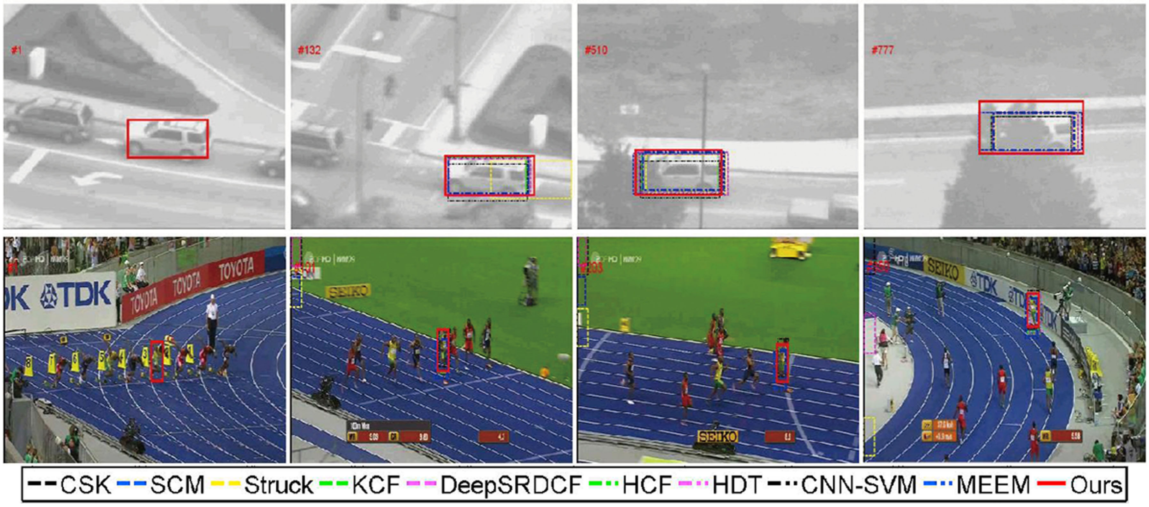

Fig. 9. The visual comparison under partial occlusion (top) and deformation (bottom). The names of clips are “Suv”, and “Bolt” from top to bottom.

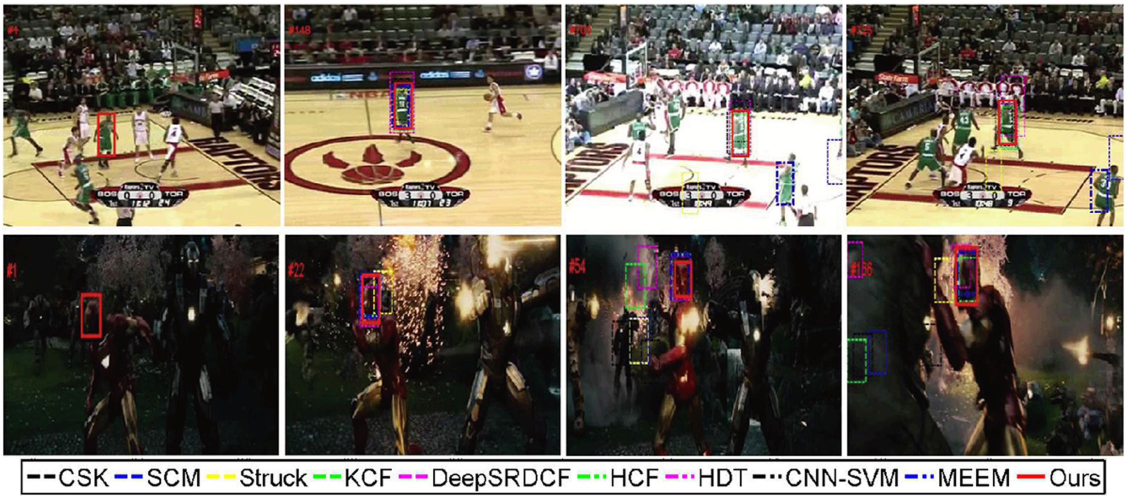

Fig. 10. Visual results under illumination variations (top) and rotation (bottom). The names of clips are “Basketball”, and “Ironman” from top to bottom.

E. Visual Comparisons

Fig. 9 and Fig. 10 show the qualitative comparisons of our tracking algorithm with the state-of-the-art trackers.

Occlusion: Target occlusion brings difficulties for accurate target locating because of the missing of target appearance information. Fig. 9 (top row) shows part of the tracking results under partial occlusion. In clip “Suv”, the interesting targets are occluded by background objects occasionally. We train multiple CNNs for different image parts and decide the final target according to global and local confidences, which alleviates the adverse impacts caused by partial occlusion and verifies the effectiveness of our tracking for handling occlusion. Some trackers (like MEEM) could track interesting targets precisely and even drift under occlusion scenarios.

Deformation: The rigid and non-rigid deformation of the target during tracking often leads to the missing of existing appearance information and the addition of new appearance information. As shown in Fig. 9 (bottom row), the postures of bodies are changing with the swinging of limbs. In addition, motion blur exists in clips such as “BlurBody”, which causes the target appearance changes. Our tracking approach that is trained on CNNs with both labeled and unlabeled samples could improve the accuracy of classification for unlabeled samples containing deformed targets. Thus, the proposed method could handle the deformation problem very well.

Illumination Variation: The target appearance will have shading due to the change of illumination in the surroundings. For example, the target appearance in “Basketball” in Fig. 10 are influenced by the lights, and both the global and local appearances are changed constantly. From the tracking results, our tracker could resist the illumination variations, and most deep learning based trackers also have handle this challenge well. But traditional trackers (such as SCM) could not track the target well from the beginning to the end. It is clear that CNNs have certain advantages for illumination variations.

Rotation: In Fig. 10, targets in “Ironman” have in-plane or out-of-plane rotation. Moreover, the background in most videos are cluttered, which is unfavorable for tracking. Our tracker could update the CNN models only with samples obtained from the current frame online, and make them adapt to appearance changes rapidly caused by rotation. Our tracker could handle rotation well, while other trackers (such as HCF and MEEM) drift more or less during tracking.

VI. Conclusion

We have presented an online visual tracking algorithm based on manifold regularized CNNs using the Gaussian fields harmonic function. The labels of unlabeled samples were first calculated using their local neighbors with graph Laplacian, and then the proposed deep model was trained online with both labeled and unlabeled data. It is a relatively simple network and pre-trained using the samples collected from a public tracking benchmark. The proposed deep tracker was tested on several popular tracking datasets and achieved better tracking performance compared with different tracking approaches.

References

[1] Y. Bai and M. Tang, "Robust tracking via weakly supervised ranking SVM," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1854–1861.

[2] B. Ma *et al.*, "Visual tracking using strong classifier and structural local sparse descriptors," *IEEE Trans. Multimedia*, vol. 17, no. 10, pp. 1818–1828, Oct. 2015.

[3] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, no. 1, pp. 2399–2434, 2006.

[4] J. Shen *et al.*, "Real-time superpixel segmentation by DBSCAN clustering algorithm," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5933–5942, Dec. 2016.

[5] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2016, pp. 850–865.

[6] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2015, pp. 58–66.

[7] W. Wang, J. Shen, and L. Shao, "Video salient object detection via fully convolutional networks," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 38–49, Jan. 2018.

[8] P. F. Felzenszwalb, R. B. Girshick, D. Mcallester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[9] S. Hare *et al.*, "Struck: Structured output tracking with kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2096–2109, Oct. 2016.

[10] J. F. Henriques, C. Rui, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 702–715.

[11] J. Shen, D. Yu, L. Deng, and X. Dong, "Fast online tracking with detection refinement," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 162–173, Jan. 2018

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

[13] 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.*, 2015, pp. 597–606.

[14] H. U. Kim, D. Y. Lee, J. Y. Sim, and C. S. Kim, "SOWP: Spatially ordered and weighted patch descriptor for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3011–3019.

[15] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling, "Semi-supervised learning with deep generative models," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 3581–3589.

[16] M. Kristan *et al.*, "The visual object tracking VOT2015 challenge results," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop*, 2015, pp. 564–586.

[17] J. Shen, Y. Du, W. Wang, and X. Li, "Lazy random walks for superpixel segmentation," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1451–1462, Apr. 2014.

[18] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013.

[19] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2004, pp. 529–536.

[20] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *Proc. Int. Conf. Image Process.*, 2002, pp. I-900–I-903.

[21] W. Wang, J. Shen, and L. Shao, "Consistent video saliency using local gradient flow optimization and global refinement," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4185–4196, Nov. 2015.

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

[23] X. Mei and H. Ling, "Robust visual tracking using L1 minimization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 1436–1443.

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

[25] 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.*, Jun. 2016, pp. 4266–4274.

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

[27] X. Dong *et al.*, "Occlusion-aware real-time object tracking," *IEEE Trans. Multimedia*, vol. 19, no. 4, pp. 763–771, Apr. 2017.

[28] A. W. Smeulders *et al.*, "Visual tracking: An experimental survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1442–1468, Jul. 2014.

[29] R. Tao, E. Gavves, and A. W. M. Smeulders, "Siamese instance search for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1420–1429.

[30] R. Olga *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, pp. 211–252, 2015.

[31] W. Wang and J. Shen, "Deep visual attention prediction," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2368–2378, May 2018.

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

[33] J. Weston, F. Ratle, H. Mobahi, and R. Collobert, "Deep learning via semi-supervised embedding," *Lecture Notes Comput. Sci.*, vol. 7700, pp. 1168–1175, 2012.

[34] Y. Qi *et al.*, "Hedging deep features for visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.

[35] A. Kolesnikov and C. H. Lampert, "Seed, expand and constrain: Three principles for weakly-supervised image segmentation," in *Proc. IEEE Conf. Eur. Conf. Comput. Vis.*, 2016, pp. 695–711.

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

[37] K. Xu, H. Su, J. Zhu, J.-S. Guan, and B. Zhang, "Neuron segmentation based on CNN with semi-supervised regularization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshop*, 2016, pp. 1324–1332.

[38] W. Wang, J. Shen, R. Yang, and F. Porikli, "Saliency-aware video object segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 20–33, Jan. 2018.

[39] W. Yi, L. Jongwoo, and M.-H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.

[40] K. Yu *et al.*, "Nonlinear learning using local coordinate coding," in *Proc. 22nd Int. Conf. Neural Inf. Process. Syst.*, 2009, pp. 2223–2231.

[41] J. Zhang, S. Ma, and S. Sclaroff, "Meem: Robust tracking via multiple experts using entropy minimization," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 188–203.

[42] W. Zhong, H. Lu, and M. H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1838–1845.

[43] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. 20th Int. Conf. Mach. Learning*, 2003, pp. 912–919.

[44] J. Shen *et al.*, "Multiobject tracking by submodular optimization," *IEEE Trans. Cybern.*, doi://10.1109/TCYB.2017.2769097, to be published.

[45] L. Wang, H. Lu, and M. H. Yang, "Constrained superpixel tracking," *IEEE Trans. Cybern.*, vol. 48, no. 3, pp. 1030–1041, Mar. 2018.

[46] L. Wang, W. Ouyang, X. Wang, and H. LU, "STCT: Sequentially training convolutional networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1373–1381.

[47] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6931–6939.

[48] Y. Qi *et al.*, "Structure-aware local sparse coding for visual tracking," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3857–3869, Aug. 2018.

Hongwei Hu is currently working toward the Ph.D. degree with the School of Computer Science, Beijing Institute of Technology, Beijing, China. His current research interests include visual object tracking algorithms.

Bo Ma received the Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 2003. From 2004 to 2006, he was with the Department of Computer Science, City University of Hong Kong, Hong Kong, working on research projects in computer vision and pattern recognition. In June 2006, he joined the Department of Computer Science, Beijing Institute of Technology, Beijing, China, where he is currently an Associate Professor. He has authored or coauthored about 40 journal and conference papers, such as the IEEE Transactions on Image Processing and IEEE International Conference on Computer Vision. His current fields of interest include statistical pattern recognition and computer vision.

Jianbing Shen (M'11–SM'12) received the Ph.D. degree in computer science from Zhejiang University, Hangzhou, China, in 2007. He is currently a Professor with the School of Computer Science, Beijing Institute of Technology, Beijing, China. He has authored or coauthored about 100 journal and conference papers, such as IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Conference on Computer Vision and Pattern Recognition, and IEEE International Conference on Computer Vision. He is also the recipient of many flagship honors including the Fok Ying Tung Education Foundation from Ministry of Education, the Program for Beijing Excellent Youth Talents from Beijing Municipal Education Commission, and the Program for New Century Excellent Talents from Ministry of Education. His research interests include computer vision and machine learning. He is also an Associate Editor for the IEEE Transactions on Neural Networks and Learning Systems, *Neurocomputing*, and the *Visual Computer* journals.

Hanqiu Sun (M'98) received the Ph.D. degree in computer science from University of Alberta, Alberta, ON, Canada. She is currently an Associate Professor with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong. Her current research interests include virtual reality and interactive graphics.

Ling Shao (M'09–SM'10) is currently the Chief Scientist for the Inception Institute of Artificial Intelligence, Abu Dhabi, United Arab Emirates, and is also a Professor with the School of Computing Sciences, University of East Anglia, Norwich, U.K. His research interests include computer vision, image processing, pattern recognition, machine learning, and medical imaging. He is an Associate Editor for the IEEE Transactions on Image Processing, IEEE Transactions on Neural Networks and Learning Systems, and other journals. He is a Fellow of the IAPR, and the IET.

Fatih Porikli (M'96–SM'04–F'14) received the Ph.D. degree from New York University, New York, NY, USA, in 2002. He is currently a Professor with the Research School of Engineering, Australian National University, Canberra, ACT, Australia, and also the Chief Scientist with Huawei, Santa Clara, CA, USA. Previously, he was a Distinguished Research Scientist with Mitsubishi Electric Research Laboratories. His research interests include computer vision, pattern recognition, manifold learning and online learning with commercial applications in video surveillance, car navigation, intelligent transportation and medical systems. He is currently the Associate Editor for five journals including the IEEE Transactions on Multimedia and the IEEE Signal Processing Magnize during the past 10 years.