Robust Visual Tracking via Convolutional Networks Without Training

Kaihua Zhang, Qingshan Liu, Yi Wu, and Ming-Hsuan Yang, *Senior Member, IEEE*

***Abstract*— Deep networks have been successfully applied to visual tracking by learning a generic representation offline from numerous training images. However, the offline training is time-consuming and the learned generic representation may be less discriminative for tracking specific objects. In this paper, we present that, even without offline training with a large amount of auxiliary data, simple two-layer convolutional networks can be powerful enough to learn robust representations for visual tracking. In the first frame, we extract a set of normalized patches from the target region as fixed filters, which integrate a series of adaptive contextual filters surrounding the target to define a set of feature maps in the subsequent frames. These maps measure similarities between each filter and useful local intensity patterns across the target, thereby encoding its local structural information. Furthermore, all the maps together form a global representation, via which the inner geometric layout of the target is also preserved. A simple soft shrinkage method that suppresses noisy values below an adaptive threshold is employed to de-noise the global representation. Our convolutional networks have a lightweight structure and perform favorably against several state-of-the-art methods on the recent tracking benchmark data set with 50 challenging videos.**

***Index Terms*— Visual tracking, convolutional networks, deep learning.**

I. Introduction

VISUAL tracking is a fundamental problem in computer vision with a wide range of applications. Although much progress has been made in recent years [1]–[6], it remains a challenging task due to many factors such as illumination changes, partial occlusion, deformation, as well as viewpoint variation [7]. To address these challenges for robust tracking, recent state-of-the-art approaches [2]–[4], [8]–[12] focus on exploiting robust representations with hand-crafted features (e.g., local binary patterns [3], Haar-like features [4], [13], [14], histograms [8], [10], HOG descriptors [11], and covariance descriptors [12]). However, these hand-crafted features are not tailored for all generic objects, and hence require sophisticated learning techniques to improve their representative capabilities.

Deep networks can directly learn features from raw data without resorting to manual tweaking, and have gained much attention with state-of-the-art results in complicated tasks such as image classification [15], object recognition [16], detection and segmentation [17]. However, considerably less attention has been made to apply deep networks for visual tracking. The main reason may be that there exists scarce amount of data to train deep networks in visual tracking because only the target state (i.e., position and size) in the first frame is available. Li *et al*. [18] incorporate a convolutional neural network (CNN) to visual tracking with multiple image cues as inputs. In [19] an ensemble of deep networks have been combined by online boosting method for visual tracking. Due to the lack of sufficient training data, both methods have not demonstrated competitive results compared to the state-of-the-art methods. Another line of research resorts to numerous auxiliary data for offline training the deep networks, and then transfer the pre-trained model to online visual tracking. Fan *et al*. [20] present a human tracking algorithm that learns a specific feature extractor with CNNs from an offline training set (about 20000 image pairs). In [6] Wang and Yeung develop a deep learning tracking method that uses stacked de-noising auto-encoder to learn the generic features from a large number of auxiliary images (1 million images). Recently, Wang *et al*. [21] use a two-layer CNN to learn hierarchical features from auxiliary video sequences, which takes into account complicated motion transformations and appearance variations in visual tracking. All these methods pay focus on learning an effective feature extractor offline with a large amount of auxiliary data, and do not fully take into account the similar local structural and inner geometric layout information among the targets over consequent frames, which is handy and effective to discriminate the target from background for visual tracking. For instance, when tracking a face, the appearance and background in consecutive frames change gradually, thereby providing strong similar local structure and geometric layout in each tracked face (rather any arbitrary pattern from a large dataset that covers numerous types of objects).

In this paper, we present a convolutional network based tracker (CNT) which exploits the local structure and inner

Manuscript received September 7, 2015; revised December 8, 2015 and February 9, 2016; accepted February 10, 2016. Date of publication February 18, 2016; date of current version March 8, 2016. The work of K. Zhang, Q. Liu, and Y. Wu was supported in part by the Natural Science Foundation of China under Grant 61402233, Grant 41501377, Grant 61532009, Grant 61272223, and Grant 61370036, in part by the National Science Foundation of Jiangsu Province under Grant BK20151529, and in part by the Startup Foundation for Introducing Talent of Nanjing University of Information Science and Technology under Grant S8113049001. The work of M.-H. Yang was supported in part by the National Science Foundation CAREER under Grant 1149783 and in part by the Information and Intelligent Systems Program under Grant 1152576. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Dimitrios Tzovaras.

K. Zhang, Q. Liu, and Y. Wu are with the Jiangsu Key Laboratory of Big Data Analysis Technology, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: cskhzhang@nuist.edu.cn; qsliu@nuist.edu.cn; ywu@nuist.edu.cn).

M.-H. Yang is with the School of Engineering, University of California at Merced, Merced, CA 95344 USA (e-mail: mhyang@ucmerced.edu).

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/TIP.2016.2531283

1057-7149 © 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

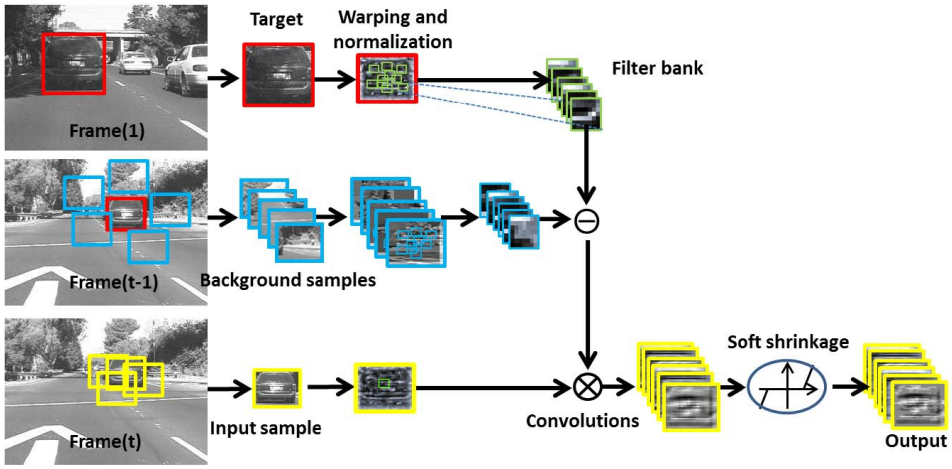


Fig. 1. Overview of the proposed representation. Input samples are warped into canonical 32×32 images. We first use the k-means algorithm to extract a set of normalized local patches from the warped target region in the first frame, and extract a set of normalized local patches from the contextual region surrounding the target. We then use them as filters to convolve each normalized sample extracted from subsequent frames, resulting in a set of feature maps. Finally, the feature maps are de-noised by a soft shrinkage method, which results in a robust sparse representation.

geometric layout information of the target. The proposed CNT has a simple architecture, and yet effectively constructs a robust representation. Figure 1 shows an overview of our algorithm. Different from the traditional CNNs [22], [23] that use pooling with local averaging and subsampling to address distortion variance, our algorithm employs a different pooling process with an effective soft shrinkage strategy. The final image representation in our method is global and sparse, which is a combination of local feature maps. Such global image representations are constructed based on the mid-level features which extract low-level properties but remain close to image-level information [24].

The main contributions of this work are summarized as follows:

1) We present a convolutional network with a lightweight structure for visual tracking. It is fully feed-forward and achieves high speed performance for online tracking even on a CPU.
2) Our method directly exploits local structural and inner geometric layout information from data without manual tweaking, which provides additional useful information in addition to appearance for visual tracking.
3) Our method achieves competitive results based on the recent tracking benchmark dataset with 50 challenging videos [7] among 32 tracking algorithms including the state-of-the-art kernel correlation filter (KCF) based method [11] and transfer learning with transformation Gaussian progress regression (TGPR) approach [12]. In particular, it outperforms the recently proposed deep learning tracker (DLT) [6] (which requires offline training with 1 million auxiliary images) by a large margin (more than 10 percents in terms of area under curve (AUC) of success rate).

II. Related Work and Problem Context

Most tracking methods emphasize on designing effective object representations [25]. The holistic templates (i.e., raw image intensity) have been widely used in visual tracking [26], [27]. Subsequently, the online subspace-based method has been introduced to visual tracking that handles appearance variations well [28]. Mei and Ling [29] utilize a sparse representation of templates to account for occlusion and appearance variation of target objects, which has been further improved [30], [31].

Meanwhile, the local templates have attracted much attention in visual tracking due to their robustness to partial occlusion and deformation. Adam *et al.* [32] use a set of local image patch histograms in a predefined grid structure to represent a target object. Kwon and Lee [33] utilize a number of local image patches to represent a target object with an online scheme to update the appearance and geometric relations. In [34] Liu *et al.* propose a tracking method that represents a target object by the histograms of sparse coding of local patches. However, the local structural information of the target has not been fully exploited [34]. To address this problem, Jia *et al.* [8] present an alignment-pooling method to combine the histograms of sparse coding.

The discriminative methods have been applied to visual tracking in which a binary classifier is learned online to separate a target object from the background. Numerous learning methods have been developed to further improve classifiers rather than image features based on support vector machine (SVM) classifiers [1], structured output SVM [4], online boosting [35], P-N learning [3], multiple instance learning [36], and some efficient hand-crafted features are available off the shelf like the Haar-like features [4], [5], [35], [36], histograms [35], HOG descriptors [11], binary features [3], and covariance descriptors [12].

Our approach for object tracking is biologically inspired from recent findings in neurophysiological studies. First, we leverage predefined convolutional filters (i.e., normalized image patches from the first frame) to extract the high-order features, which is motivated by the hierarchical MAX (HMAX) model proposed by Serre et al. [37] that uses

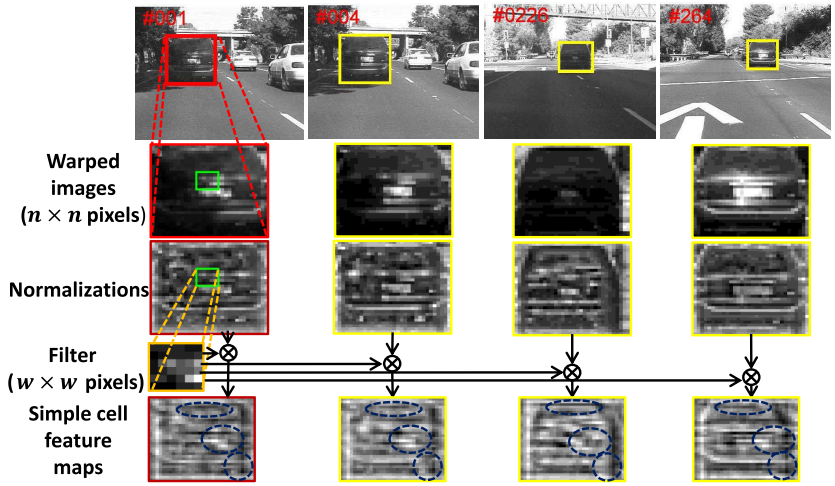


Fig. 2. Although the target appearance changes significantly due to illumination changes and scale variations, the simple cell feature map can well preserve the local structure (e.g., the regions in the dotted ellipses) of the target and maintain its global geometric layout invariant to some degree.

Gabor filters instead. Furthermore, we combine local features without changing their structures and spatial arrangements to generate a global representation, which increases feature invariance while maintaining specificity, thereby satisfying the two essential requirements in cognitive tasks [38]. In contrast, the HMAX model [37] exploits a pooling mechanism with a maximum operation to enhance feature invariance and specificity. Second, our algorithm is based on a feed-forward architecture, which is largely consistent with the standard model of object recognition in the primate cortex [38] that focuses on the capabilities of the ventral visual pathway for immediate recognition without the help of attention or other top-down visual information. The rapid performance of the human visual system suggests humans most likely use feed-forward processing due to its simplicity. Recently, psychophysical experiments show that generic object tracking can be implemented in a low level neural mechanism [39], and hence our method leverages a simple template matching scheme without using a high-level object model.

III. Convolutional Networks for Tracking

A. Image Representation

Given a target template, we develop a hierarchical representation architecture with a convolutional network including two separated layers. Figure 1 summarizes the main components of the proposed algorithm. First, local selective features are extracted from a bank of filters convolving the input image at each position. Second, selective features are stacked together to form a global representation that is robust to appearance variations. In the following, we refer these layers as the simple and complex layers, with analogy to the V1 simple and complex cells discovered by Hubel and Wiesel [40].

1) Preprocessing: Each input image is warped to a canonical size of $n \times n$ pixels and represented by the intensity values, denoted as $\mathbf{I} \in \mathbb{R}^{n \times n}$. We densely sample a set of overlapping local image patches $\mathcal{Y} = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_l\}$ centered at each pixel location inside the input image through sliding a window of size $w \times w$ (w is referred to as the receptive field size), where $\mathbf{Y}_i \in \mathbb{R}^{w \times w}$ is the i-th image patch and $l = (n - w + 1) \times (n - w + 1)$. Each patch $\mathbf{Y}_i$ is preprocessed by subtracting the mean and ℓ_2 normalization that correspond to local brightness and contrast normalization, respectively.

2) Simple Layer: After preprocessing, we employ the k-means algorithm to select a bank of patches $\mathcal{F}^o = \{\mathbf{F}_1^o, \ldots, \mathbf{F}_d^o\} \subset \mathcal{Y}$ sampled from the object region in the first frame as fixed filters to extract our selective features from simple cells. Given the i-th filter $\mathbf{F}_i^o \in \mathbb{R}^{w \times w}$, its response on the input image $\mathbf{I}$ is denoted with a feature map $\mathbf{S}_i^o \in \mathbb{R}^{(n-w+1) \times (n-w+1)}$, where $\mathbf{S}_i^o = \mathbf{F}_i^o \otimes \mathbf{I}$ and $\otimes$ is the convolution operator. As illustrated in Figure 2, the filter $\mathbf{F}_i^o$ is localized and selective that can extract local structural features (e.g., oriented edges, corners, and endpoints), most of which are similar despite significant appearance variation. Furthermore, the simple cell feature maps have a similar geometric layout (see the bottom row of Figure 2), which shows that the local filter can extract useful information across the entire image, and hence the global geometric layout information can also be effectively exploited. Finally, the local filters can be referred as a set of fixed local templates that encode stable visual information in the first frame, thereby handling the drifting problem effectively. Similar strategy has been adopted in [10], [27], and [34], where [27] utilizes the template in the first frame and the tracked results to update the template whereas [10] and [34] exploit a static dictionary learned from the first frame to sparsely represent the tracked target.

The background context surrounding the object provides useful information to discriminate the target from the background. As illustrated in Figure 1, we choose m background samples surrounding the object, and use the k-means algorithm to select a bank of filters $\mathcal{F}_i^b = \{\mathbf{F}_{i,1}^b, \ldots, \mathbf{F}_{i,d}^b\} \subset \mathcal{Y}$ from the i-th background sample. We use the average pooling method to summarize each filter in $\mathcal{F}_i^b$, and generate the background context filter set, $\mathcal{F}^b = \{\mathbf{F}_1^b = \frac{1}{m}\sum_{i=1}^m \mathbf{F}_{i,1}^b, \ldots, \mathbf{F}_d^b = \frac{1}{m}\sum_{i=1}^m \mathbf{F}_{i,d}^b\}$. Given the input image $\mathbf{I}$, the i-th background feature map is defined as $\mathbf{S}_i^b = \mathbf{F}_i^b \otimes \mathbf{I}$. Finally, the simple cell

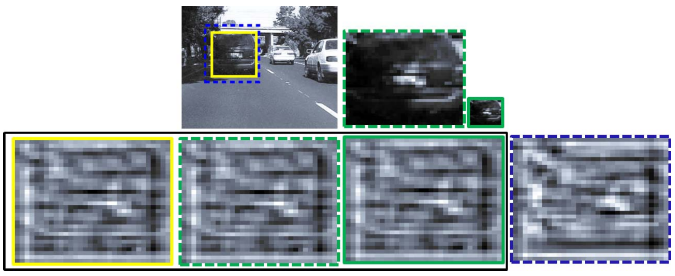


Fig. 3. Illustration of the scale-invariant and shift-variant properties of the complex cell features. Scale invariance: although the scale of the target varies (top row), their corresponding simple feature maps (inside the black rectangle) have similar local structures and geometric layouts due to wrapping and normalization. Shift-variance: the bottom right feature map generated by shifting the tracked target (in blue dotted rectangle) shows much difference from the left ones due to inclusion of numerous background pixels.

feature maps are defined as

$$\mathbf{S}_i = \mathbf{S}_i^o - \mathbf{S}_i^b = (\mathbf{F}_i^o - \mathbf{F}_i^b) \otimes \mathbf{I}, \quad i = 1, \ldots, d. \tag{1}$$

3) Complex Layer: Each simple cell feature map $\mathbf{S}_i$ simultaneously encodes the local structural and global geometric layout information of the target object, thereby generating a good representation to handle appearance variations. To further enhance the strength of this representation, we construct a complex cell feature map that is a 3D tensor $\mathbf{C} \in \mathbb{R}^{(n-w+1)\times(n-w+1)\times d}$, which stacks d different simple cell feature maps constructed with the filter set $\mathcal{F} = \mathcal{F}^o \cup \mathcal{F}^b$. This layer is analogous to the pooling layers in the CNNs [23] and the HMAX model [37] where the local averaging and subsampling operations are used in the CNNs and the local maximum is used in the the HMAX model.

Both the HMAX model and CNNs focus on learning shift-invariant features that are useful for image classification and object recognition [6] which are less effective for visual tracking (which requires higher position precision). As illustrated in Figure 3, if the complex features are shift-invariant, both the blue dotted and the yellow bounding boxes can be treated as the accurate tracking results, thereby leading to the location ambiguity problem. To overcome this problem, in [36] the multiple instance learning approach is developed for visual tracking. In contrast, the shift-variant complex cell features make our method more robust to location ambiguity. Furthermore, the complex cell features are more robust to scale variation, which is validated by the experimental results (see Section IV).

After warping the target at different scales to a canonical size (e.g., 32×32 pixels), the location of each useful part in the target does not vary much in the warped images at this abstract view, and hence the complex cell features can preserve the geometric layouts of the useful parts at different scales as well as their local structures due to normalizing the wrapped target and local filters.

To make the feature map $\mathbf{C}$ more robust to appearance variation, we utilize a sparse vector $\mathbf{c}$ to approximate $\text{vec}(\mathbf{C})$ by minimizing the following objective function

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c}} \lambda \|\mathbf{c}\|_1^1 + \frac{1}{2}\|\mathbf{c} - \text{vec}(\mathbf{C})\|_2^2, \tag{2}$$

where $\text{vec}(\mathbf{C}) \in \mathbb{R}^{(n-w+1)^2 d}$ is a column vector by concatenating all the elements in $\mathbf{C}$. We note that (2) has a closed form solution that can be readily achieved by a soft shrinkage function [41],

$$\hat{\mathbf{c}} = \text{sign}(\text{vec}(\mathbf{C})) \max(0, \text{abs}(\text{vec}(\mathbf{C})) - \lambda), \tag{3}$$

where $\text{sign}(\cdot)$ is a sign function, and λ is set to $\text{median}(\text{vec}(\mathbf{C}))$, i.e., median value of $\text{vec}(\mathbf{C})$, which adapts well to target appearance variations during tracking as demonstrated by the experimental results.

4) Model Update: The sparse representation $\mathbf{c}$ in (2) is used as the target template and updated incrementally to accommodate appearance changes over time for robust visual tracking. We use a temporal low-pass filtering method [13],

$$\mathbf{c}_t = (1 - \rho)\mathbf{c}_{t-1} + \rho\hat{\mathbf{c}}_{t-1}, \tag{4}$$

where ρ is a learning parameter, $\mathbf{c}_t$ is the target template at frame t and $\hat{\mathbf{c}}_{t-1}$ is the sparse representation of the tracked target at frame $t-1$. This online update scheme not only accounts for rapid appearance variations but also alleviates the drift problem due to retaining the local filters in the first frame.

5) Efficient Computation: The computational load for the target or background template $\mathbf{c}$ mainly includes preprocessing the local patches in $\mathcal{Y}$ as well as convolving the input image $\mathbf{I}$ with d local filters in $\mathcal{F}$. However, the operations of local normalization and mean subtraction when preprocessing all patches can be reformulated as convolutions on the input image [42]. Therefore, only the convolution operations are needed when constructing the target template, which can be efficiently computed by the fast Fourier transforms (FFTs). The computational complexity for computing each FFT is $\mathcal{O}(2n^2 \log n)$ for each image patch of $n \times n$ pixels. Furthermore, since the local filters are independent during tracking, the convolution operations can be easily parallelized, thereby further reducing the computational load.

B. Proposed Tracking Algorithm

Our tracking algorithm is formulated within a particle filtering framework. Given the observation set $\mathcal{O}_t = \{\mathbf{o}_1, \ldots, \mathbf{o}_t\}$ up to frame t, our goal is to determine a posteriori probability $p(\mathbf{s}_t|\mathcal{O}_t)$ using the Bayes' theorem:

$$p(\mathbf{s}_t|\mathcal{O}_t) \propto p(\mathbf{o}_t|\mathbf{s}_t) \int p(\mathbf{s}_t|\mathbf{s}_{t-1}) p(\mathbf{s}_{t-1}|\mathcal{O}_{t-1}) d\mathbf{s}_{t-1}, \tag{5}$$

where $\mathbf{s}_t = [x_t, y_t, s_t]^\top$ is the target state with translations x_t, y_t and scale s_t, $p(\mathbf{s}_t|\mathbf{s}_{t-1})$ is the motion model that predicts the state $\mathbf{s}_t$ based on the previous state $\mathbf{s}_{t-1}$, and $p(\mathbf{o}_t|\mathbf{s}_t)$ is the observation model that estimates the likelihood of observation $\mathbf{o}_t$ at the state $\mathbf{s}_t$ belonging to the target category.

We assume that the target state parameters are independent, which are modeled by three scalar Gaussian distributions, and hence it can be formulated as Brownian motion [28], i.e., $p(\mathbf{s}_t|\mathbf{s}_{t-1}) = \mathcal{N}(\mathbf{s}_t|\mathbf{s}_{t-1}, \Sigma)$, where $\Sigma = \text{diag}(\sigma_x, \sigma_y, \sigma_s)$ is a diagonal covariance matrix whose elements are the standard deviations of the target state parameters. In visual tracking, the posterior probability $p(\mathbf{s}_t|\mathcal{O}_t)$ in (5) is approximated

TABLE I
TEST VIDEOS CATEGORIZED WITH 11 ATTRIBUTES

Sequence	LR	IPR	OPR	SV	OCC	DEF	BC	IV	MB	FM	OV
Basketball			✓		✓	✓	✓	✓			
Bolt		✓	✓		✓	✓					
Boy		✓	✓	✓					✓	✓	
Car4				✓				✓			
CarDark							✓	✓			
CarScale		✓	✓	✓	✓					✓	
Coke		✓	✓		✓			✓		✓	
Couple			✓	✓		✓	✓			✓	
Crossing			✓	✓		✓	✓			✓	
David		✓	✓	✓	✓	✓		✓	✓		
David2		✓	✓								
David3			✓		✓	✓	✓				
Deer	✓	✓					✓		✓	✓	
Dog1		✓	✓	✓							
Doll		✓	✓	✓	✓			✓			
Dudek		✓	✓	✓	✓	✓	✓			✓	✓
FaceOcc1					✓						
FaceOcc2		✓	✓		✓			✓			
Fish								✓			
FleetFace		✓	✓	✓		✓			✓	✓	
Football		✓	✓		✓		✓				
Football1		✓	✓				✓				
Freeman1		✓	✓	✓							
Freeman3		✓	✓	✓							
Freeman4		✓	✓	✓	✓						
Girl		✓	✓	✓	✓						
Ironman	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
Jogging1			✓		✓	✓					
Jogging2					✓	✓					
Jumping									✓	✓	
Lemming			✓	✓	✓			✓		✓	✓
Liquor			✓	✓	✓		✓	✓	✓	✓	✓
Matrix		✓	✓	✓	✓		✓	✓			
Mhyang			✓			✓	✓	✓			
MotorRolling	✓	✓		✓			✓	✓	✓	✓	
MountainBik		✓	✓				✓				
Shaking		✓	✓	✓			✓	✓			
Singer1			✓	✓	✓			✓			
Singer2		✓	✓			✓	✓	✓			
Skating1			✓	✓	✓	✓	✓	✓			
Skiing		✓	✓	✓		✓		✓			
Soccer		✓	✓	✓	✓		✓	✓	✓	✓	
Subway					✓	✓	✓				
Suv		✓			✓						✓
Sylvester		✓	✓					✓			
Tiger1		✓	✓		✓	✓		✓	✓	✓	
Tiger2		✓	✓		✓	✓		✓	✓	✓	✓
Trellis		✓	✓	✓			✓	✓			
Walking				✓	✓	✓					
Walking2	✓			✓	✓						
Woman			✓	✓	✓	✓		✓	✓	✓	
Total number	4	31	39	28	29	19	21	25	12	17	6

by a particle filter in which N particles $\{\mathbf{s}_t^i\}_{i=1}^N$ are sampled with corresponding importance weights $\{\pi_t^i\}_{i=1}^N$, where $\pi_t^i \propto p(\mathbf{o}_t|\mathbf{s}_t^i)$.

The optimal state is achieved by maximizing the posteriori estimation over a set of N particles

$$\hat{\mathbf{s}}_t = \arg\max_{\{\mathbf{s}_t^i\}_{i=1}^N} p(\mathbf{o}_t|\mathbf{s}_t^i)\, p(\mathbf{s}_t^i|\hat{\mathbf{s}}_{t-1}). \tag{6}$$

The observation model $p(\mathbf{o}_t|\mathbf{s}_t^i)$ in (6) plays a key role in robust tracking, and the formulation in this work is

$$p(\mathbf{o}_t|\mathbf{s}_t^i) \propto e^{-\|\mathbf{c}_t - \mathbf{c}_t^i\|_2^1}, \tag{7}$$

where $\mathbf{c}_t$ is the target template at frame t,

$$\mathbf{c}_t^i = \mathrm{vec}(\mathbf{C}_t^i) \odot \mathbf{w} \tag{8}$$

is the i-th candidate sample representation at frame t based on the complex cell features, where $\odot$ denotes the element-wise multiplication, and $\mathbf{w}$ is an indicator function whose element is defined as

$$w_i = \begin{cases} 1, & \text{if } \mathbf{c}_t(i) \neq 0 \\ 0, & \text{otherwise,} \end{cases} \tag{9}$$

Algorithm 1 Convolutional Network Based Tracking

Input: Target filter set $\mathcal{F}^o$, background filter set $\mathcal{F}^b_{t-1}$, target state $\hat{\mathbf{s}}_{t-1}$, target template $\mathbf{c}_t$

1) Sample N candidate particles $\{\mathbf{s}^i_t\}^N_{i=1}$ with the motion model $p(\mathbf{s}^i_t|\hat{\mathbf{s}}_{t-1})$ in (5)
2) For each particle $\mathbf{s}^i_t$, extract its corresponding image patch, compute its representation $\mathbf{c}^i_t$ by (8), and compute its observation model $p(\mathbf{o}_t|\mathbf{s}^i_t)$ by (7).
3) Estimate the optimal state $\hat{\mathbf{s}}_t$ by (6)
4) Extract background samples to update their corresponding filter set $\mathcal{F}^b_t$ in (1), and then compute the sparse representation $\hat{\mathbf{c}}_t$ of the target template by (3)
5) Update the target template $\mathbf{c}_{t+1}$ by (4)

Output: Target state $\hat{\mathbf{s}}_t$ and target template $\mathbf{c}_{t+1}$

where $\mathbf{c}_t(i)$ denotes the i-th element of $\mathbf{c}_t$. With the incremental update scheme (4), the observation model is able to adapt to the target appearance variations while alleviating the drift problem. The main steps of the proposed algorithm are summarized in Algorithm 1.

IV. Experimental Results

A. Experimental Setup

The proposed CNT is implemented in MATLAB and runs at 5 frames per second on a PC with Intel i7 3770 CPU (3.4 GHz), which is much faster than the DLT method [6] that runs at 1.5 frames per second with an offline model trained with GPU. The images of each video are converted to grayscale, and the state of the target (i.e., size and location) in the first frame is given by the ground truth. The size of the warped image is set to 32×32 ($n = 32$). The receptive field size is set to 6×6 ($w = 6$) and the number of filters is set to $d = 100$. The learning parameter ρ in (4) is set to 0.95 and the template is updated every frame. The standard deviations of the target state of the particle filter are set as follows: $\sigma_x = 4$, $\sigma_y = 4$, and $\sigma_s = 0.01$, and $N = 600$ particles are used. All the parameters are fixed for all experiments. The source code is available to the public at http://faculty.ucmerced.edu/mhyang/project/cnt/.

B. Evaluation Metrics

For experimental validation, we use the tracking benchmark dataset and code library [7] which includes 29 trackers and 50 fully-annotated videos (more than 29,000 frames). In addition, we also add the results of two state-of-the-art trackers including the KCF [11], TGPR [12], and DLT [6] methods. To better evaluate and analyze the strength and weakness of the tracking approaches, the videos are categorized with 11 attributes based on different challenging factors including low resolution (LR), in-plane rotation (IPR), out-of-plane rotation (OPR), scale variation (SV), occlusion (OCC), deformation (DEF), background clutters (BC), illumination variation (IV), motion blur (MB), fast motion (FM), and out-of-view (OV), which are summarized in Table I.

For quantitative evaluations, we use the success plot and the precision plot [7]. The success plot is based on the overlap ratio, $S = \text{Area}(B_T \cap B_G)/\text{Area}(B_T \cup B_G)$, where B_T is the tracked bounding box and B_G denotes the ground truth. The success plot shows the percentage of frames with $S > t_0$ throughout all threshold $t_0 \in [0, 1]$. The area under curve (AUC) of each success plot serves as the second measure to rank the tracking algorithms. Meanwhile, the precision plot illustrates the percentage of frames whose tracked locations are within the given threshold distance to the ground truth. A representative precision score with the threshold equal to 20 pixels is used to rank the trackers.

We report the results of one-pass evaluation (OPE) [7] based on the average success and precision rate given the ground truth target state in the first frame. For presentation clarity, we only present the top 10 algorithms in each plot. The evaluated trackers include the proposed CNT, KCF [11], TGPR [12], Struck [4], SCM [10], TLD [3], DLT [6], VTD [2], VTS [43], CXT [9], CSK [44], ASLA [8], DFT [45], LSK [34], CPF [46], LOT [47], TM-V [48], KMS [49], L1APG [30], MTT [31], MIL [36], L1APG [30], OAB [35], and SemiT [50]. Table II and Table III summarize the tracking results in terms of success and precision plots. More results and videos are available at http://faculty.ucmerced.edu/mhyang/project/cnt/.

C. Quantitative Comparisons

1) Overall Performance: Figure 4 shows the overall performance of the top 10 performing tracking algorithms in terms of success and precision plots. Note that all the plots are generated using the code library from the benchmark evaluation [7], and the results of KCF [11], TGPR [12], and DLT [6] methods are provided by the authors. The proposed CNT algorithm ranks first based on the success rate while third based on the precision rate. In the success plot, the proposed CNT algorithm achieves the AUC of 0.545, which outperforms the DLT method by 10.9%. Meanwhile, in the precision plot, the precision score of the CNT algorithm is 0.723 which is close to the TGPR (0.766) and KCF (0.740) methods, but outperforms the DLT approach by 14.5%. Note that the proposed CNT algorithm exploits only simple sparse image representation that encodes local structural and geometric layout information of the target, and achieves competitive performance to the Struck and SCM methods that utilize useful background information to train discriminative classifiers. Furthermore, even using only specific target information from the first frame without learning with auxiliary training data, the CNT algorithm performs well against the DLT method (more than 10 percent in terms of both success and precision rates), which shows that the generic features learned offline from numerous auxiliary data may not adapt well to target appearance variations in visual tracking.

2) Attribute-Based Performance: To analyze the strength and weakness of the proposed algorithm, we further evaluate the trackers on videos with 11 attributes [7]. Figure 5 shows the success plots of videos with different attributes and Figure 6 shows the corresponding precision plots. We note that the proposed CNT algorithm ranks within top 3 on 7

TABLE II

SCORE OF SUCCESS PLOT (BEST VIEWED ON A COLOR DISPLAY). THE RED FONTS INDICATE THE BEST PERFORMANCE, THE BLUE FONTS INDICATE THE SECOND BEST ONES, AND THE GREEN FONTS INDICATE THE THIRD BEST ONES

Attribute	**CNT**	TGPR	KCF	SCM	Struck	TLD	DLT	ASLA	CXT	VTS
LR	0.437	0.351	0.312	0.279	0.372	0.309	0.346	0.157	0.312	0.168
IPR	0.495	0.487	0.497	0.458	0.444	0.416	0.411	0.425	0.452	0.416
OPR	0.501	0.507	0.495	0.470	0.432	0.420	0.412	0.422	0.418	0.425
SV	0.508	0.443	0.427	0.518	0.425	0.421	0.455	0.452	0.389	0.400
OCC	0.503	0.494	0.514	0.487	0.413	0.402	0.423	0.376	0.372	0.398
DEF	0.524	0.556	0.534	0.448	0.393	0.378	0.394	0.372	0.324	0.368
BC	0.488	0.543	0.535	0.450	0.458	0.345	0.339	0.408	0.338	0.428
IV	0.456	0.486	0.493	0.473	0.428	0.399	0.405	0.429	0.368	0.429
MB	0.417	0.440	0.497	0.298	0.433	0.404	0.363	0.258	0.369	0.304
FM	0.404	0.441	0.459	0.296	0.462	0.417	0.360	0.247	0.388	0.300
OV	0.439	0.431	0.550	0.361	0.459	0.457	0.367	0.312	0.427	0.443
Overall score	0.545	0.529	0.514	0.499	0.474	0.437	0.436	0.434	0.426	0.416

TABLE III

SCORE OF PRECISION PLOT (BEST VIEWED ON A COLOR DISPLAY). THE RED FONTS INDICATE THE BEST PERFORMANCE, THE BLUE FONTS INDICATE THE SECOND BEST ONES, AND THE GREEN FONTS INDICATE THE THIRD BEST ONES

Attribute	TGPR	KCF	**CNT**	Struck	SCM	TLD	DLT	VTD	VTS	CXT
LR	0.539	0.381	0.557	0.545	0.305	0.349	0.396	0.168	0.187	0.371
IPR	0.706	0.725	0.661	0.617	0.597	0.584	0.548	0.599	0.579	0.610
OPR	0.741	0.729	0.672	0.597	0.618	0.596	0.561	0.620	0.604	0.574
SV	0.703	0.679	0.662	0.639	0.672	0.606	0.590	0.597	0.582	0.550
OCC	0.708	0.749	0.662	0.564	0.640	0.563	0.574	0.545	0.534	0.491
DEF	0.768	0.740	0.687	0.521	0.586	0.512	0.563	0.501	0.487	0.422
BC	0.761	0.753	0.646	0.585	0.578	0.428	0.495	0.571	0.578	0.443
IV	0.687	0.728	0.566	0.558	0.594	0.537	0.534	0.557	0.573	0.501
MB	0.578	0.650	0.507	0.551	0.339	0.518	0.453	0.375	0.375	0.509
FM	0.575	0.602	0.500	0.604	0.333	0.551	0.446	0.352	0.353	0.515
OV	0.431	0.550	0.439	0.459	0.361	0.457	0.367	0.462	0.443	0.427
Overall score	0.766	0.740	0.723	0.656	0.649	0.608	0.587	0.576	0.575	0.575

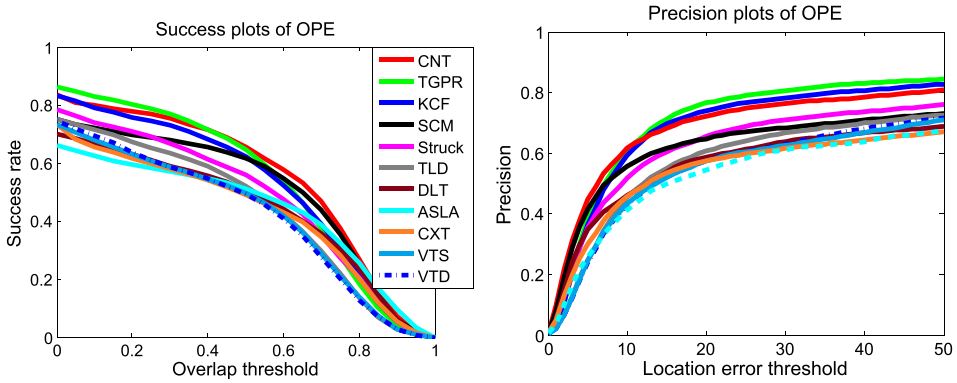


Fig. 4. The success plots and precision plots of OPE for the top 10 trackers. The performance score of success plot is the AUC value while the performance score for each tracker is shown in the legend. The performance score of precession plot is at error threshold of 20 pixels. Best viewed on color display.

out of 11 attributes in success plots, and outperforms the DLT method on all 11 attributes. In the precision plots, the CNT algorithm ranks top 3 on 6 out of 11 attributes, and outperforms the DLT method on all attributes. Since the AUC score of the success plot is more informative than the score at one position in the precision plot, in the following we analyze the results based on these values.

On the image sequences with the *low resolution* attribute, the CNT algorithm ranks first among all evaluated trackers. The low resolution in the videos makes it difficult to extract effective hand-crafted features from the targets. In contrast, the CNT algorithm extracts dense information across the entire target region by convolution operators to separate the target from the background.

For the image sequences with attributes such as *in-plane rotation, out-of-plane rotation, scale variation*, and *occlusion*, the CNT algorithm ranks second among all evaluated algorithms with a narrow margin (about 1 percent) to the best performing methods, such as KCF, TGPR, and SCM. All these methods use local image features as image representations. The KCF method utilizes HOG features to describe the target and its local context region, and the TGPR approach extracts

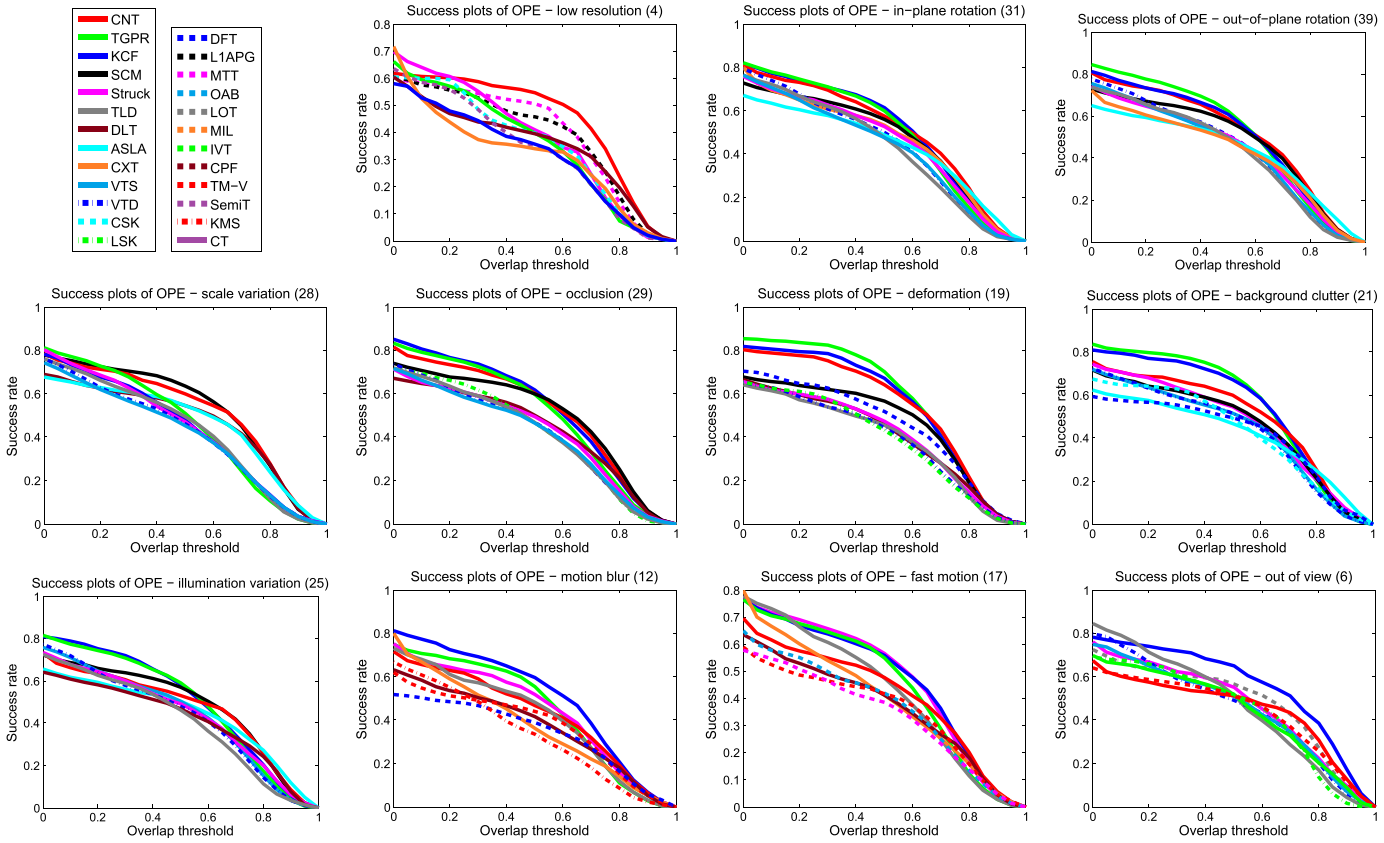


Fig. 5. The success plots of videos with different attributes. The number in the title indicates the number of sequences. Best viewed on color display.

the covariance descriptors from the local image patches as image representations. Furthermore, both CNT and SCM algorithms employ local features extracted from the normalized local image patches. The proposed CNT algorithm exploits useful local features across the target object via filtering while the SCM method learns local features from the target and background with sparse representation. In addition, both CNT and SCM algorithms utilize the target template from the first frame to handle the drift problem.

On the videos with *deformation* and *background clutter* attributes, the CNT algorithm ranks third which follows the KCF and TGPR methods. The proposed CNT algorithm encodes the geometric layout information using multiple simple cell feature maps (see Figure 2), which are stacked together to form a global representation, thereby equipping it to account for deformation. Furthermore, the CNT algorithm uses background context information that is online updated and pooled in every frame, and hence provides helpful information to accurately locate target objects from the background clutters.

On the videos with the *illumination variation* and *motion blur* attributes, the CNT algorithm ranks fourth while the TGPR and KCF methods perform well. All these methods take advantage of normalized local image information, which is robust to illumination variation. Furthermore, when the target appearance changes significantly due to motion blur, the relatively unchanged backgrounds exploited by these methods provide useful information to help localize the target objects.

Finally, for the videos with *fast motion* attribute, the CNT algorithm ranks fifth while the top 4 trackers are Struck, KCF, TGPR, and TLD. The CNT algorithm does not address fast motion well as simple dynamic model based on stochastic search is used (similar to IVT, SCM and ASLA). In contrast, the trackers based on dense sampling (e.g., Struck, KCF, TGPR, and TLD) that detect all samples in a local region surrounding the tracking location in the previous frame perform well in the test set with the fast motion attribute as a large state space is exploited. The performance of the CNT algorithm can be further improved with more complex dynamic models, by reducing the image resolution that equals to increasing the search range, or with more particles in larger ranges. Furthermore, there are 6 videos with the *out of view* attribute and almost all of them contain fast moving objects. The KCF and Struck methods perform well on image sequences with both attributes. Struck employs a budgeting mechanism that maintains useful target samples from the entire tracking sequences, and can re-detect the target when it reappears after out of view. On the other hand, CNT explores the stable visual information from the first frame, which helps re-detect the target objects.

D. Qualitative Comparisons

1) Deformation: Figure 7 shows some screenshots of the tracking results in three challenging sequences where the

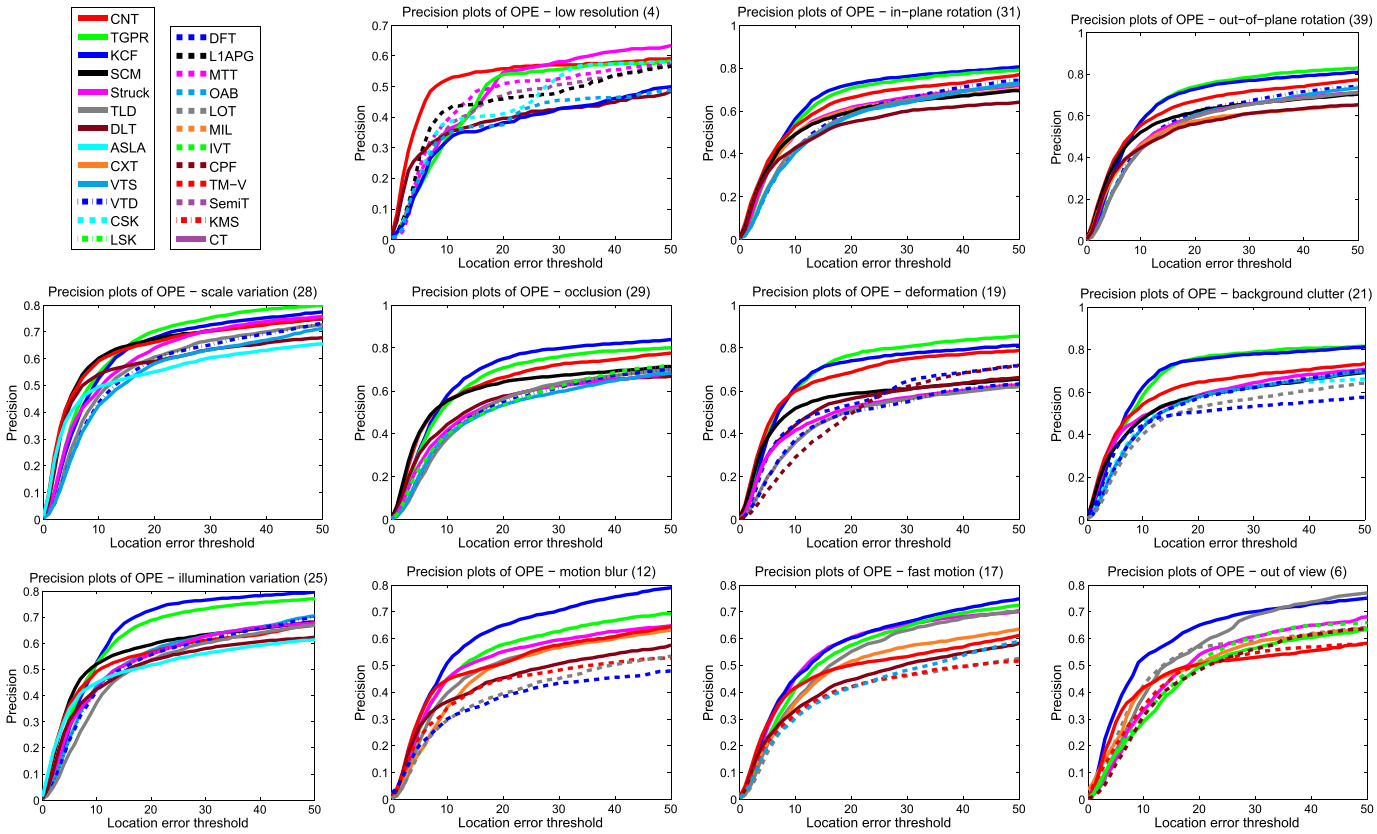


Fig. 6. Precision plots of videos with different attributes. The number in the title indicates the number of sequences. Best viewed on color display.

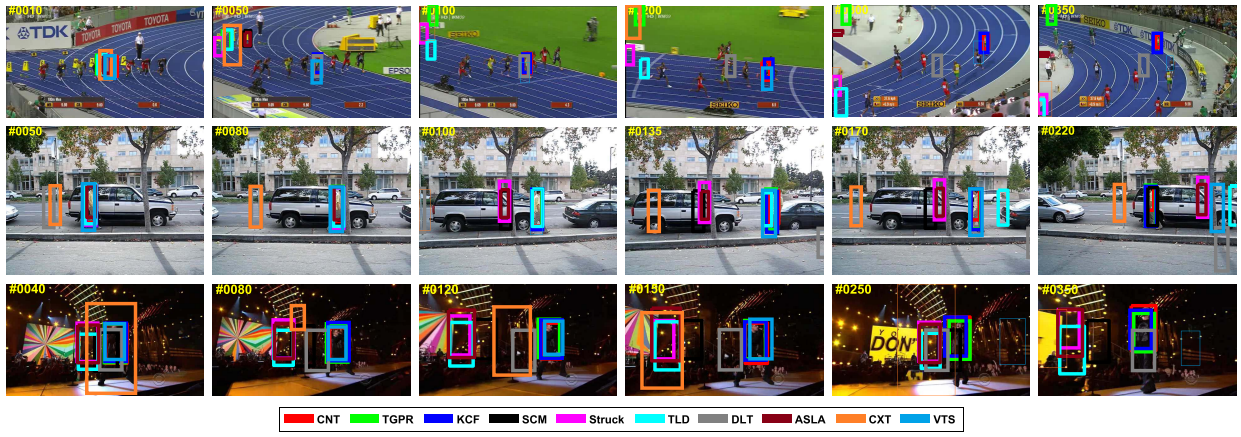


Fig. 7. Qualitative results of the 10 trackers over sequences *bolt*, *david3* and *singer2*, in which the targets undergo severe deformation. Best viewed on color display.

target objects undergo large shape deformation. In the *bolt* sequence, several objects appear in the scenes with rapid appearance changes due to shape deformation and fast motion. Only the CNT and KCF algorithms can track the targets well. The TGPR, SCM, TLD, ASLA, CXT and VTS methods undergo large drift at the beginning of the sequence (e.g.#10, #100). The DLT approach drifts to the background at frame #200. The target object in the *david*3 sequence undergoes significant appearance variations due to non-rigid body deformation. Furthermore, the target appearance changes drastically when the person walks behind the tree and turns around. The DLT and CXT methods lose track of the target object after frame #50. The SCM, ALSA and VTS methods lock on some to parts of background when the person walks behind the tree (e.g., #100, #135, and #170). The TLD algorithm loses the target when the man turns around at frame #135, and the Struck method locks on to the background when the person walks behind the tree again (e.g., #220). Only the CNT, TGPR and KCF methods perform well at all frames. The target in the *singer*2 sequence undergoes both

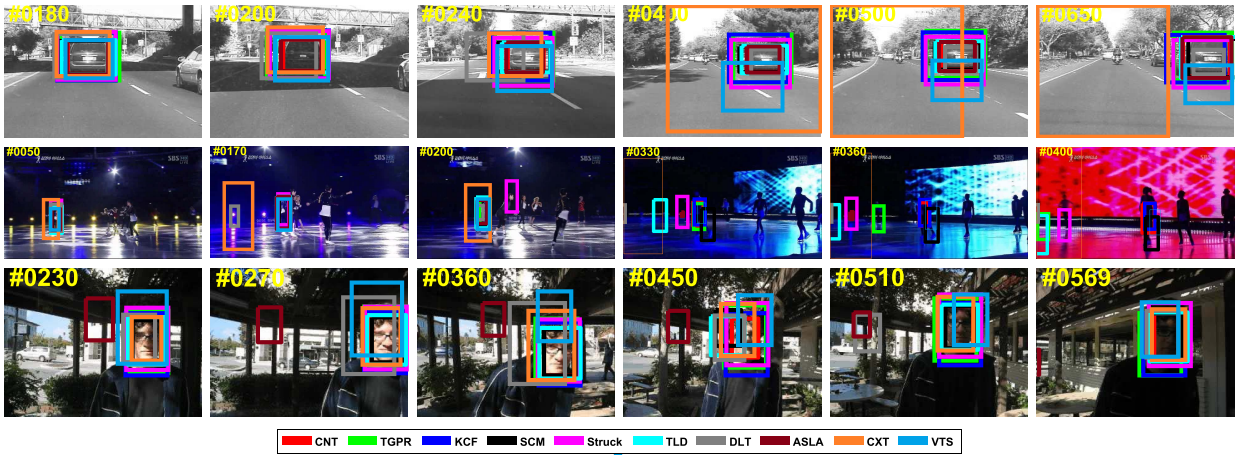


Fig. 8. Qualitative results of the 10 trackers over sequences *car4*, *skating1* and *trellis*, in which the targets undergo severe illumination changes. Best viewed on color display.

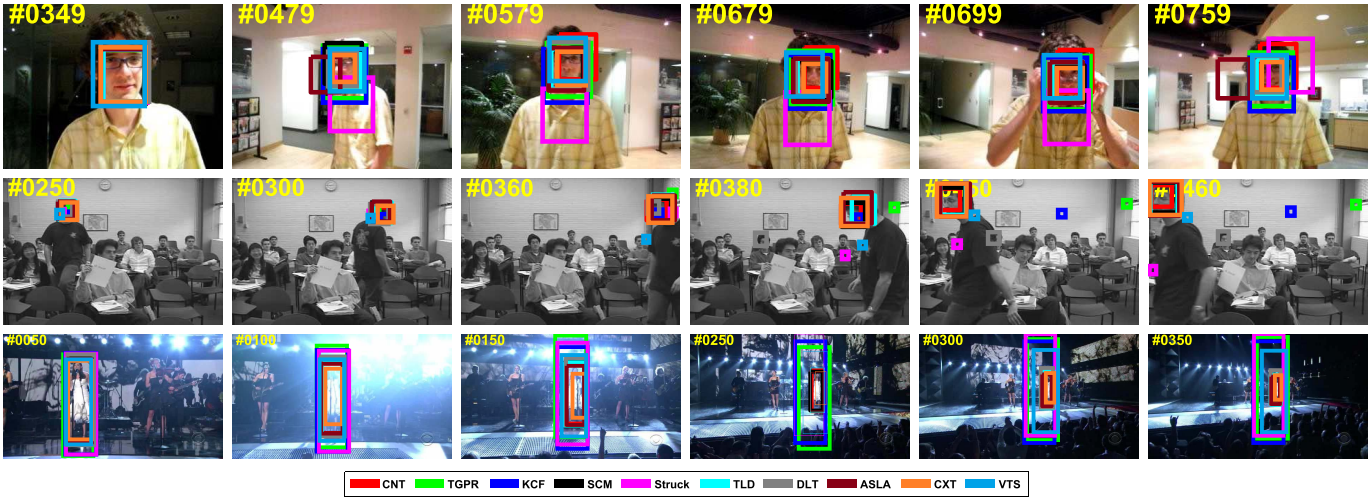


Fig. 9. Qualitative results of the 10 trackers over sequences *david*, *freeman3* and *singer1*, in which the targets undergo scale variations. Best viewed on color display.

deformation and illumination variations. Only the CNT, TGPR and KCF algorithms perform well in the entire sequence.

2) Illumination Changes: Figure 8 shows some sampled results in three sequences in which the target objects undergo large illumination variations. In the *car4* sequence, a moving vehicle passes underneath a bridge and trees. Despite large illumination variations at frames #180, #200, and #240, the CNT algorithm is able to track the object well. The DLT, CXT and VTS methods drift away from the target objects when sudden illumination change occurs at frame #240. Furthermore, the target object also undergoes scale variations (e.g. #500 and #650). Although the TGPR and KCF methods are able to successfully track the target objects, they do handle scale variations well (e.g., #500 and #650). The target object in the *skating*1 sequence undergoes rapid pose variations and drastic light changes (e.g., #170, #360, and #400). Only the CNT and KCF algorithms persistently track the object from the beginning to the end. In the *trellis* sequence, the object appearance changes significantly due to variations in variations and pose. The DLT and ASLA methods drift away to background (e.g., #510). The CNT, TLD and Struck algorithms are able to stably track the target with much more better accuracy than the TGPR, KCF and CXT methods.

3) Scale Variations: Figure 9 demonstrates some results over three challenging sequences with targets undergoing significant scale variations. In the *david* sequence, a person moves from a dark room to a bright area while his appearance changes much due to illumination variation, pose variation, and a large scale variation of the target object with respect to the camera. The ASLA and VTS algorithms drift away to the background (e.g. #479 and #759). The KCF and Struck methods do not handle scale well with lower success rate than the CNT algorithm. In the *freeman3* sequence, a person moves towards the camera with a large scale variation in his face appearance. Furthermore, the appearance also changes significantly due to pose variation and low resolution. The TGPR, KCF, Struck, DLT and VTS methods drift away to the background regions (e.g., #380, #450, and #460) whereas the CNT, SCM, TLD and CXT algorithms perform well. In the *singer1* sequence, the target object moves far away

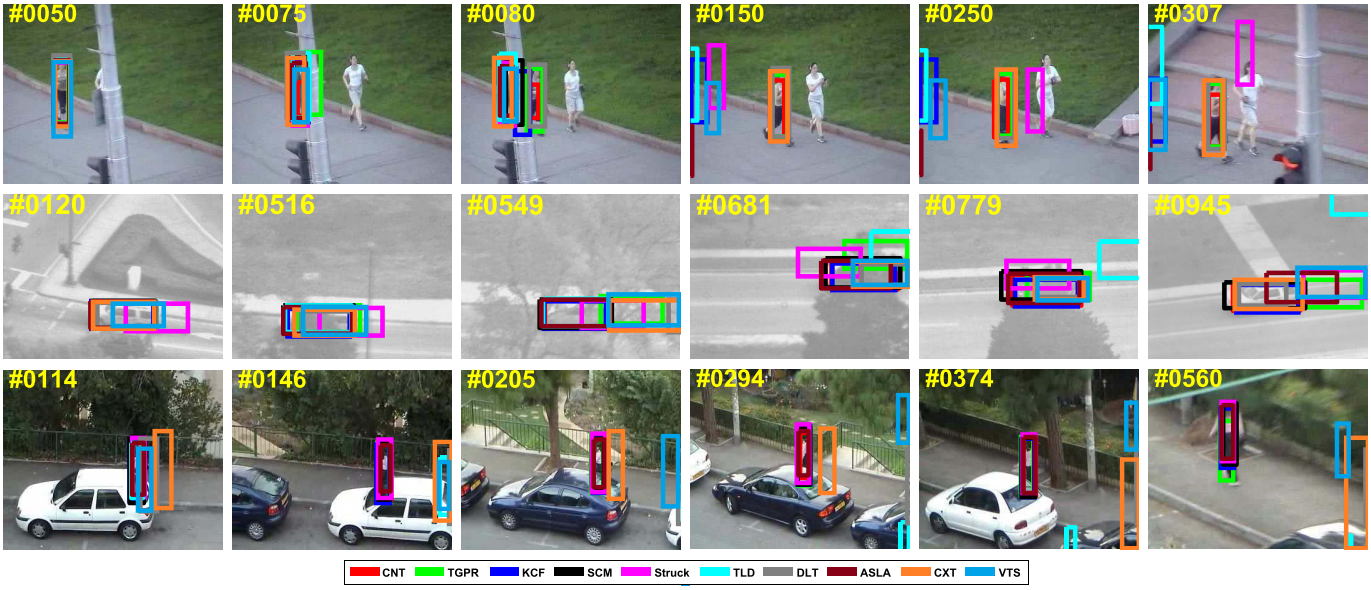


Fig. 10. Qualitative results of the 10 trackers over sequences *jogging-1*, *suv* and *woman*, in which the targets undergo heavy occlusion. Best viewed on color display.

Fig. 11. Qualitative results of CNT with and without model update over six long sequences *sylvester*, *mhyang*, *dog1*, *lemming*, *liquor*, and *doll*.

from the camera with large scale change. The TGPR, KCF, Struck and VTS methods do not perform well while the CNT, SCM, ASLA and CXT approaches achieve better performance. The CNT algorithm handles scale variation well because its representation is built on scale-invariant complex cell features (see Figure 3).

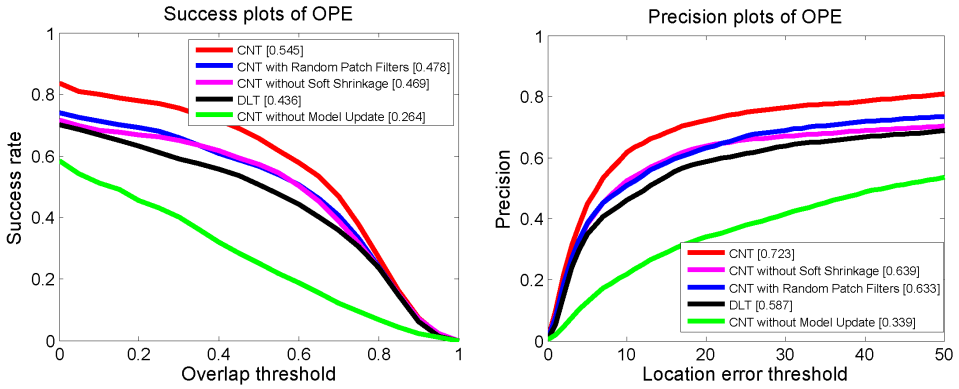


Fig. 12. Success plots and precision plots of OPE for CNT with different components. The DLT is taken as a baseline.

4) Heavy Occlusion: Figure 10 shows sampled results of three sequences where the targets undergo heavy occlusions. In the *jogging-1* sequence, a person is almost fully occluded by the lamp post (e.g., #75 and #80). Only the CNT, TGPR, DLT and CXT algorithms are able to re-detect the object when the person reappears in the screen (e.g., #80, #150, #250). In the *suv* sequence, the target vehicle is frequently occluded by dense tree branches (e.g., #516, #549, #681, and #799). In addition, there are several shot changes in this video. The TGPR, Struck, TLD, ASLA and VTS methods do not perform well (e.g., #945) Although the target person is occluded for a long duration in the *woman* sequence (e.g., #114 and #374), the CNT, TGPR, KCF, SCM, Struck and ASLA algorithms achieve favorable results. All these methods use local features that are robust to occlusions.

E. Analysis of CNT

To validate the effectiveness of key components of CNT, we propose three variants of CNT: one utilizes random patch filters to replace the filters learned by k-means algorithm, one does not involve the soft shrinkage process, and another one does not employ the model update scheme (4). Figure 12 shows the quantitative results on the benchmark dataset. The results show that with random patch filters, the AUC score of success rate reduces by 7%. Meanwhile, the CNT without soft shrinkage can only achieve AUC score of 0.469, which is lower than the original CNT method with 0.545 by a large margin. However, both variants perform better than the DLT method. Furthermore, the results for the CNT method without model update are worse than the proposed algorithm. Figure 11 shows some sampled results over six long sequences. In the *mhyang* and *dog1* sequences, the CNT is able to track the targets stably over all frames, which performs much better than the CNT without model update. In the other four sequences *sylvester*, *lemming*, *liquor*, and *doll*, the CNT performs favorably for most frames in each sequence while the CNT without model update undergoes severe drift after a few frames. These results show that all the filers, soft shrinkage, and model update components play key roles in the proposed algorithm for robust visual tracking.

V. Concluding Remarks

In this paper, we propose a two-layer feed-forward convolutional network that generates an effective representation for robust tracking. The first layer is constructed by a set of simple cell feature maps defined by a bank of filters, in which each filter is a normalized patch extracted from the first frame with simple k-means algorithm. In the second layer, the simple cell feature maps are stacked to a complex cell feature map as the target representation, which encodes the local structural and geometric layout information of the target. A soft shrinkage strategy is employed to de-noise the target representation. In addition, an effective online scheme is adopted to update the representation, which adapts to the target appearance variations during tracking. Extensive evaluation on a large benchmark dataset demonstrates the proposed tracking algorithm achieves favorable results against some state-of-the-art methods.

References

[1] S. Avidan, "Support vector tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 8, pp. 1064–1072, Aug. 2004.

[2] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1269–1276.

[3] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N learning: Bootstrapping binary classifiers by structural constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 49–56.

[4] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 263–270.

[5] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 864–877.

[6] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 809–817.

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

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

[9] T. B. Dinh, N. Vo, and G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 1177–1184.

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

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

[12] J. Gao, H. Ling, W. Hu, and J. Xing, "Transfer learning based visual tracking with Gaussian processes regression," in *Proc. 13th Eur. Conf. Comput. Vis.*, 2014, pp. 188–203.

[13] K. Zhang, L. Zhang, and M. Yang, "Fast compressive tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 10, pp. 2002–2015, Oct. 2014.

[14] H. Song, "Robust visual tracking via online informative feature selection," *Electron. Lett.*, vol. 50, no. 25, pp. 1931–1933, Dec. 2014.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[16] J. Donahue *et al.*, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 647–655.

[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[18] H. Li, Y. Li, and F. Porikli, "Robust online visual tracking with a single convolutional neural network," in *Proc. 12th Asian Conf. Comput. Vis.*, 2014, pp. 194–209.

[19] X. Zhou, L. Xie, P. Zhang, and Y. Zhang, "An ensemble of deep neural networks for object tracking," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2014, pp. 843–847.

[20] J. Fan, W. Xu, Y. Wu, and Y. Gong, "Human tracking using convolutional neural networks," *IEEE Trans. Neural Netw.*, vol. 21, no. 10, pp. 1610–1623, Oct. 2010.

[21] L. Wang, T. Liu, G. Wang, K. L. Chan, and Q. Yang, "Video tracking using learned hierarchical features," *IEEE Trans. Image Process.*, vol. 24, no. 4, pp. 1424–1435, Apr. 2015.

[22] Y. LeCun *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.

[23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[24] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2559–2566.

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

[26] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework," *Int. J. Comput. Vis.*, vol. 56, no. 3, pp. 221–255, 2004.

[27] I. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 810–815, Jun. 2004.

[28] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 125–141, 2008.

[29] X. Mei and H. Ling, "Robust visual tracking and vehicle classification via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2259–2272, Nov. 2011.

[30] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust L1 tracker using accelerated proximal gradient approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1830–1837.

[31] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2042–2049.

[32] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 798–805.

[33] J. Kwon and K. M. Lee, "Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping Monte Carlo sampling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1208–1215.

[34] B. Liu, J. Huang, L. Yang, and C. Kulikowsk, "Robust tracking using local sparse appearance model and K-selection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 1313–1320.

[35] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proc. Brit. Mach. Vis. Conf.*, 2006, pp. 47–56.

[36] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, Aug. 2011.

[37] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, "Robust object recognition with cortex-like mechanisms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 411–426, Mar. 2007.

[38] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nature Neurosci.*, vol. 2, no. 11, pp. 1019–1025, Nov. 1999.

[39] V. Mahadevan and N. Vasconcelos, "On the connections between saliency and tracking," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1664–1672.

[40] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *J. Physiol.*, vol. 148, no. 3, pp. 574–591, 1959.

[41] M. Elad, M. A. T. Figueiredo, and Y. Ma, "On the role of sparse and redundant representations in image processing," *Proc. IEEE*, vol. 98, no. 6, pp. 972–982, Jun. 2010.

[42] S. Ben-Yacoub, B. Fasel, and J. Luettin, "Fast face detection using MLP and FFT," in *Proc. 2nd Int. Conf. AVBPA*, 1999, pp. 31–36.

[43] J. Kwon and K. M. Lee, "Tracking by sampling trackers," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1195–1202.

[44] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 702–715.

[45] L. Sevilla-Lara and E. Learned-Miller, "Distribution fields for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1910–1917.

[46] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2002, pp. 661–675.

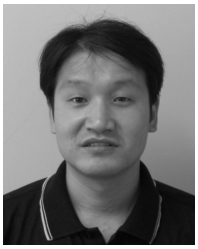
[47] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan, "Locally orderless tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1940–1947.

[48] R. Collins, X. Zhou, and S. K. Teh, "An open source tracking testbed and evaluation Web site," in *Proc. PETS*, 2005, pp. 17–24.

[49] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564–577, May 2003.

[50] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 234–247.

Kaihua Zhang received the B.S. degree in technology and science of electronic information from the Ocean University of China, in 2006, the M.S. degree in signal and information processing from the University of Science and Technology of China, in 2009, and the Ph.D. degree from the Department of Computing, The Hong Kong Polytechnic University, in 2013. From 2009 to 2010, he was a Research Assistant with the Department of Computing, The Hong Kong Polytechnic University. He is a Professor with the School of Information and Control, Nanjing University of Information Science and Technology, Nanjing, China. His research interests include image segmentation, level sets, and visual tracking.



Qingshan Liu received the M.S. degree from the Department of Auto Control, Southeast University, Nanjing, China, in 2000, and the Ph.D. degree from the National Laboratory of Pattern Recognition, Chinese Academic of Science, Beijing, China, in 2003. He was an Assistant Research Professor with the Department of Computer Science, Computational Biomedicine Imaging and Modeling Center, Rutgers, The State University of New Jersey, New Brunswick, NJ, USA, from 2010 to 2011. Before, he joined Rutgers University. He was an Associate Professor with the National Laboratory of Pattern Recognition, Chinese Academic of Science, and an Associate Researcher with the Multimedia Laboratory, Chinese University of Hong Kong, Hong Kong, from 2004 to 2005. He is a Professor with the School of Information and Control Engineering, Nanjing University of Information Science and Technology, Nanjing. He was a recipient of the President Scholarship of the Chinese Academy of Sciences in 2003. His current research interests are image and vision analysis, including face image analysis, graph and hypergraph-based image and video understanding, medical image analysis, and event-based video analysis.

Yi Wu received the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, China, in 2009. Since Fall 2009, he has been an Assistant Professor with the Nanjing University of Information Science and Technology. From 2010 to 2012, he was a Post-Doctoral Fellow with Temple University, USA, and the University of California, Merced, from 2012 to 2014. His research interests include computer vision, multimedia analysis, and machine learning.

Ming-Hsuan Yang (M'92–SM'06) received the Ph.D. degree in computer science from the University of Illinois at Urbana–Champaign, in 2000. He joined University of California at Merced (UC Merced) in 2008. He was a Senior Research Scientist with the Honda Research Institute working on vision problems related to humanoid robots. He is an Associate Professor in electrical engineering and computer science with UC Merced. He is a Senior Member of the ACM. He received the NSF CAREER Award in 2012, the Senate Award for Distinguished Early Career Research at UC Merced in 2011, and the Google Faculty Award in 2009. He served as an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE from 2007 to 2011, and is an Associate Editor of the *International Journal of Computer Vision, Image and Vision Computing* and the *Journal of Artificial Intelligence Research*.