

Robust Visual Tracking via Convolutional Networks

Kaihua Zhang, Qingshan Liu, Yi Wu, and Ming-Hsuan Yang

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 develop a robust representation for visual tracking. In the first frame, we employ the k -means algorithm to 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 the useful local intensity patterns across the target, thereby encoding its local structural information. Furthermore, all the maps form together a global representation, which is built on mid-level features, thereby remaining close to image-level information, and hence the inner geometric layout of the target is also well preserved. A simple soft shrinkage method with an adaptive threshold is employed to de-noise the global representation, resulting in a robust sparse representation. The representation is updated via a simple and effective online strategy, allowing it to robustly adapt to target appearance variations. Our convolution networks have surprisingly lightweight structure, yet perform favorably against several state-of-the-art methods on the CVPR2013 tracking benchmark dataset 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 (refer to [7]). To address

Kaihua Zhang, Qingshan Liu and Yi Wu are with Jiangsu Key Laboratory of Big Data Analysis Technology (B-DAT), Nanjing University of Information Science and Technology. E-mail: {cshkzhang, qsliu, ywu}@nuist.edu.cn.

Ming-Hsuan Yang is with Electrical Engineering and Computer Science, University of California, Merced, CA, 95344. E-mail: mhyang@ucmerced.edu.

these challenges for robust tracking, recent state-of-the-art approaches [2]–[4], [8]–[10] focus on exploiting robust representations with hand-crafted features (e.g., local binary patterns [3], Haar-like features [4], 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 some sophisticated learning techniques to improve their representative capabilities.

Deep networks can directly learn features from raw data without resorting to manual tweaking, which have gained much attention with state-of-the-art results in some complicated tasks, such as image classification [13], object recognition [14], detection and segmentation [15]. 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 at disposal. Li *et al.* [16] incorporated a convolutional neural network (CNN) to visual tracking with multiple image cues as inputs. In [17] an ensemble of deep networks have been combined by online boosting method for visual tracking. However, due to the lack of sufficient training data, both methods have not demonstrated competitive results compared to 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.* [18] proposed 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 proposed a deep learning tracking method that uses stacked denoising autoencoder to learn the generic features from a large number of auxiliary images (1 million images). Recently, Wang *et al.* [19] employed 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 particular attention to offline learning an effective feature extractor with a large amount of auxiliary data, yet 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 geometric layout information of the target. The proposed CNT has a surprisingly simple architecture, yet effectively constructs a robust representation. Figure 1 presents an overview of our method. Different from the traditional CNNs [20], [21] that combine three architectural ideas (i.e., local receptive fields, shared weights, and pooling with local average and subsampling) to address shift, scale, and distortion variance, our method does not include the pooling process due to the reasons as: first, high spatial resolution is needed to preserve the local structure of the target in visual tracking; second, the precise positions of the features lost by pooling play an important role to preserve the geometric layout of the target. The final image representation is global and sparse, which is a combination of some local feature maps. Such global image representation is built on the mid-level features [22], which extract low-level information, but remain close to image-level information without any need of

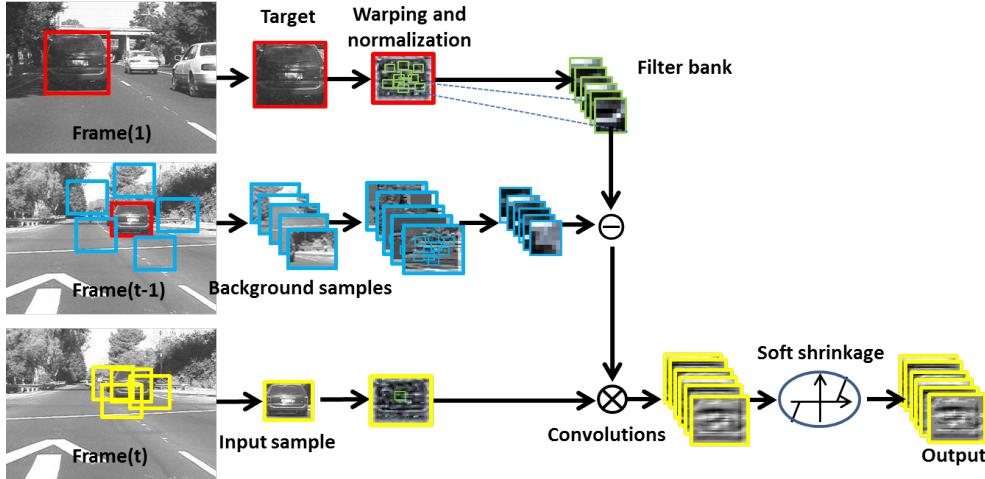


Fig. 1: Overview of the proposed representation. Input samples are warped into a canonical 32×32 images. We first employ the k -means algorithm to extract a set of normalized local patches from the warped target region in the first frame, and then extract a set of normalized local patches from the context 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.

high-level structured image description. In [23] the simple k -means algorithm has also been employed to generate mid-level feature, which achieves promising performance.

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 fast speed for online tracking even on a CPU.
- 2) Our method directly exploits the local structural and inner geometric layout information from data without manual tweaking, which provides additional useful information besides appearance for visual tracking.
- 3) Our method achieves very competitive results through evaluating on the CVPR2013 tracking benchmark dataset with 50 challenging videos [7] among 32 tracking algorithms including the state-of-the-art KCF [11] and TGPR methods [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 the AUC score of success rate), which shows the power of convolutional networks.

II. RELATED WORK

Our approach for object tracking is biologically inspired from recent findings in neurophysiological studies. First, we leverage convolution with predefined local filters (i.e., the normalized image patches from the first frame) to extract the high-order features, which is motivated by the HMAX model proposed by Riesenhuber and Poggio [24] that uses Gabor filters instead. Furthermore, we simply combine the 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 task [25]. In contrast, the HMAX model [24] exploits a new pooling mechanism with a maximum operation to enhance feature invariance and specificity. Second, our method owns a purely feed-forward architecture, which is largely consistent with the standard model of object recognition in primate cortex [25] that focuses on the capabilities of the ventral visual pathway in an immediate recognition without the help of attention or other top-down effects. 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 [26], and hence our method leverages a simple template matching scheme without using a high-level object model.

Most tracking methods emphasize on designing effective object representations [27]. The holistic templates (i.e., raw image intensity) have been widely used in visual tracking [28], [29]. Subsequently, the online subspace-based method has been introduced to visual tracking that handles appearance variations well [30]. Recently, Mei and Ling [31] utilize a sparse representation of templates to deal with the corrupted appearance of the target, which has been further improved recently [32], [33]. Meanwhile, the local templates have attracted much attention in visual tracking due to their robustness to partial occlusion and deformation. Adam *et al.* [34] use a set of local image patch histograms in a predefined grid structure to represent a target object. Kwon and Lee [35] utilize a number of local image patches to represent the target with an online scheme to update their appearances and geometric relations. Liu *et al.* [36] proposed a tracking method that represents a target object by the histograms of sparse coding of local patches. However, in [36] the local structural information of the target has not been fully exploited. To address this problem, Jia *et al.* [8] proposed an alignment-pooling method to combine the histograms of sparse coding. Recently, the discriminative methods have been applied to visual tracking due to the performance 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 [37], P-N learning [3], multiple instance learning [38], and some efficient hand-crafted features are available off the shelf like the Haar-like features [4], [5], [37], [38], histograms [37], HOG descriptors [11], binary features [3], and covariance descriptors [12].

III. CONVOLUTIONAL NETWORKS FOR TRACKING

A. Image Representation

Given the target template, we develop a hierarchical representation architecture with convolutional networks, including two separated layers. Figure 1 summarizes our approach. First, the local selective features are extracted with a bank of filters convolving the input image at each position. Second, the selective features are stacked together into a global representation that is robust to appearance variations. We refer these layers as the simple layer and the complex layer, respectively, with analogy to the V1 simple and complex cells discovered by Hubel and Wiesel [39].

1) Preprocessing: We convert the input image to grayscale and warp it to a fixed size with $n \times n$ pixels, denoted as $\mathbf{I} \in \mathbb{R}^{n \times n}$. We then densely sample a set of overlapping local image patches $\mathcal{Y} = \{\mathbf{Y}_1, \dots, \mathbf{Y}_l\}$ centered at each pixel location inside the input image through sliding a window with size $w \times w$ (w is referred as the

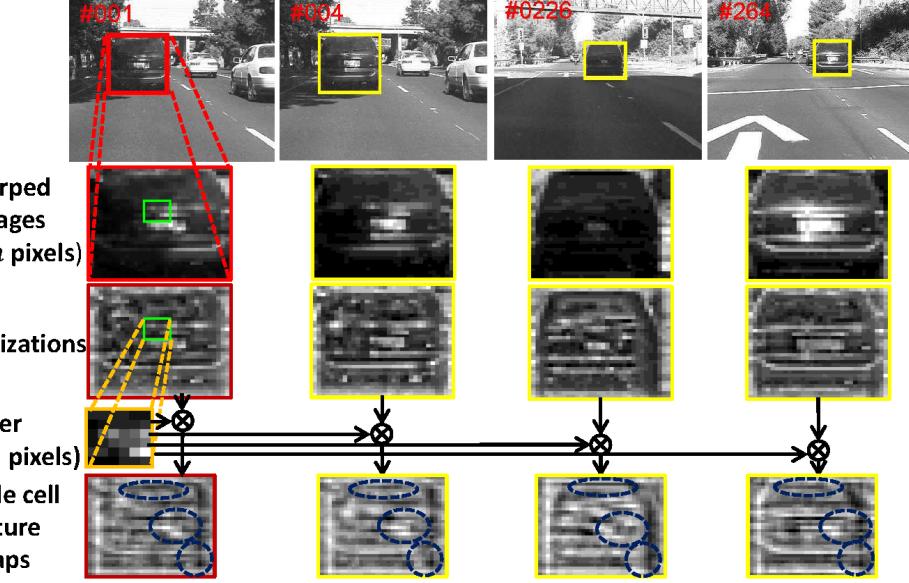


Fig. 2: Although the target appearance changes significantly due to illumination changes and scale variations, the simple cell feature map can not only well preserve the local structure (e.g., the regions in the dotted ellipses) of the target, but also maintain its global geometric layout invariant to some degree.

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 $\mathcal{F}^o = \{\mathbf{F}_1^o, \dots, \mathbf{F}_d^o\} \subset \mathcal{Y}$ sampled from the object region in the first frame as fixed filters to extract our selective simple cell features. 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 $\mathbf{S}_i^o = \mathbf{F}_i^o \otimes \mathbf{I}$. As illustrated by Figure 2, the filter \mathbf{F}_i^o is localized and selective that can extract the local structural features (e.g., oriented edges, corners, endpoints), most of which are similar despite the target appearance changing greatly. Furthermore, the simple cell feature maps have a similar geometric layout (see the bottom row of Figure 2), which illustrates 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 information in the first frame, thereby handling the drifting problem effectively. Similar strategy has been adopted in [10], [29], [36], where [29] utilizes the template in the first frame and the tracked result to update the template while [10], [36] 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 background. As illustrated by Figure 1, we choose m background samples surrounding the object, and use the k -means algorithm to select a bank of filters $\mathcal{F}_i^b = \{\mathbf{F}_{i,1}^b, \dots, \mathbf{F}_{i,d}^b\} \subset \mathcal{Y}$ from the i -th background sample. Then,

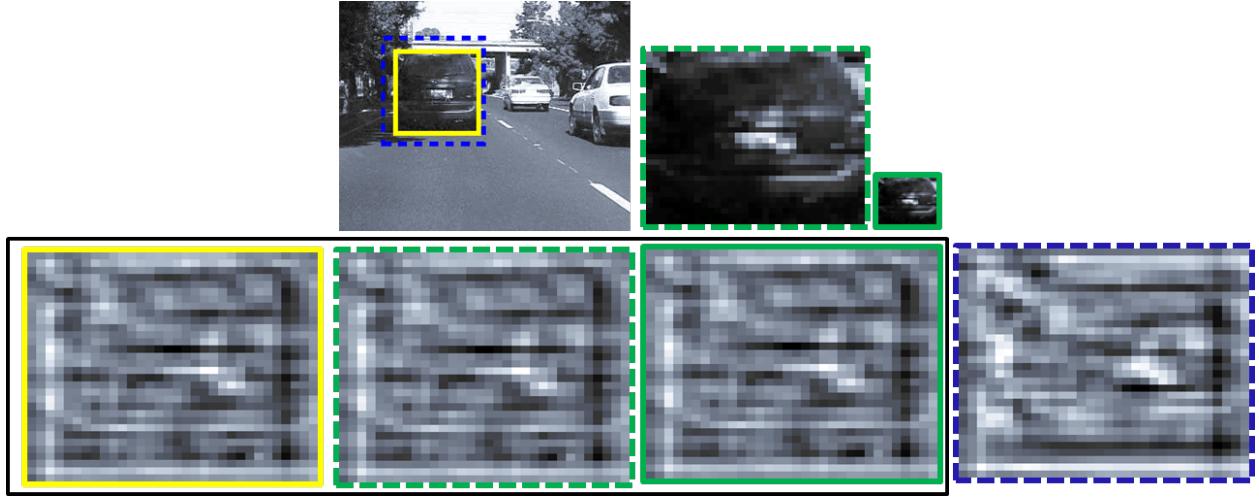


Fig. 3: Illustration of the scale-invariant and shift-variant properties of the complex cell features. Scale invariance: although the scale of the target varies (top row), their corresponding simple feature maps (inside the black rectangle) have similar local structures and geometric layouts due to wrapping 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.

we employ the average pooling method to summarize each filter in \mathcal{F}_i^b , which results in the background context filter set defined as $\mathcal{F}^b = \{\mathbf{F}_1^b = \frac{1}{m} \sum_{i=1}^m \mathbf{F}_{i,1}^b, \dots, \mathbf{F}_d^b = \frac{1}{m} \sum_{i=1}^m \mathbf{F}_{i,d}^b\}$. 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 feature maps are defined as

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

3) *Complex Layer*: The simple cell feature map \mathbf{S}_i simultaneously encodes the local structural and the global geometric layout information of the target, thereby equipping it with 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 simply stacks d different simple cell feature maps constructed with the filter set \mathcal{F} . This layer is analogous to the pooling layers in the CNNs [21] and the HMAX model [24]: the CNNs utilize the local average and subsampling operations while the HMAX model leverages the local maximum scheme.

Both the CNNs and the HMAX model focus on learning shift-invariant features that are useful for image classification and object recognition [6], yet less effective for visual tracking. 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, leading to the location ambiguity problem. To overcome this problem, in [38] the multiple instance learning method is introduced to visual tracking. In contrast, the shift-variant complex cell features make our method robust to location ambiguity. Furthermore, the complex cell features are more robust to scale variation. After warping the target at different scales to a fixed 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 abstracted 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 the local filters.

To make the map \mathbf{C} robust to noise introduced by appearance variations, we utilize a sparse vector \mathbf{c} to approximate $\text{vec}(\mathbf{C})$, which is obtained by minimizing the following objective function

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

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} . (2) has a closed form solution that can be readily achieved by a soft shrinkage function [40]

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

where $\text{sign}(\cdot)$ is a sign function, and we set $\lambda = \text{median}(\text{vec}(\mathbf{C}))$ that is the median value of $\text{vec}(\mathbf{C})$ in our experiments, which can well adapt to target appearance variations during tracking.

4) *Model Update*: The sparse representation \mathbf{c} in (2) serves as the target template, which should be updated incrementally to accommodate appearance changes over time for robust visual tracking. We use a simple temporal low-pass filtering method [41],

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

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 simple online update scheme not only accounts for rapid appearance variations, but also alleviates drift problem due to retaining the local filters in the first frame.

5) *Efficient Computation*: The cost to compute 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). Furthermore, since the local filters are independent during tracking, the convolutions can be easily parallelized, thereby largely reducing the computational cost.

B. Proposed Tracking Algorithm

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

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

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

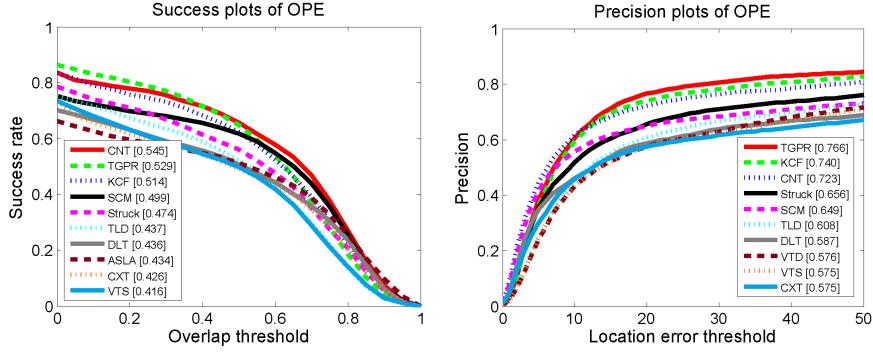


Fig. 4: The success plots and precision plots of OPE for the top 10 trackers. The performance score of success plot is the AUC value while the performance score for each tracker is shown in the legend. The performance score of precession plot is at error threshold of 20 pixels while. Best viewed on color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

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 the motion model can be formulated as a Brownian motion [30], 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 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

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

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

$$p(\mathbf{o}_t|\mathbf{s}_t^i) \propto e^{-\|\mathbf{c}_t - \mathbf{c}_t^i\|_2^2}, \quad (7)$$

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

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

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

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

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.

IV. EXPERIMENTS

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). Each color video is 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 $n \times n$ is set to $n = 32$. The receptive field size $w \times w$ is set to $w = 6$. 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 to $\sigma_x = 4$, $\sigma_y = 4$, and $\sigma_s = 0.01$, and $N = 600$ particles are used. The parameters are fixed for all experiments. The source code will be made available to the public.

B. Evaluation Metric

We use the CVPR2013 tracking benchmark dataset and code library [7], which includes 29 trackers and 50 fully-annotated videos (more than 29,000 frames). Furthermore, we also add the results of two state-of-the-art trackers including the KCF [11] and TGPR methods [12], and one representative deep learning based tracker DLT [6]. 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 illumination variation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, out-of-view, background clutters, and low resolution.

For quantitative comparison, we employ the success plot and the precision plot used in [7]. The success plot is based on the overlap ratio that is $S = \text{Area}(B_T \cap B_G)/\text{Area}(B_T \cup B_G)$, 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 demonstrated 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 [36], CPF [46], LOT [47], TM-V [48], KMS [49], L1APG [32], MTT [33], MIL [38], OAB [37], and SemiT [50].

C. Quantitative Comparisons

1) *Overall Performance:* Figure 4 illustrates the overall performance of the top 10 performing tracking algorithms in terms of success plot and precision plot. Note that all the plots are automatically generated by the code library supported by the benchmark evaluation [7], and the results of KCF [11], TGPR [12], and DLT [6] are provided

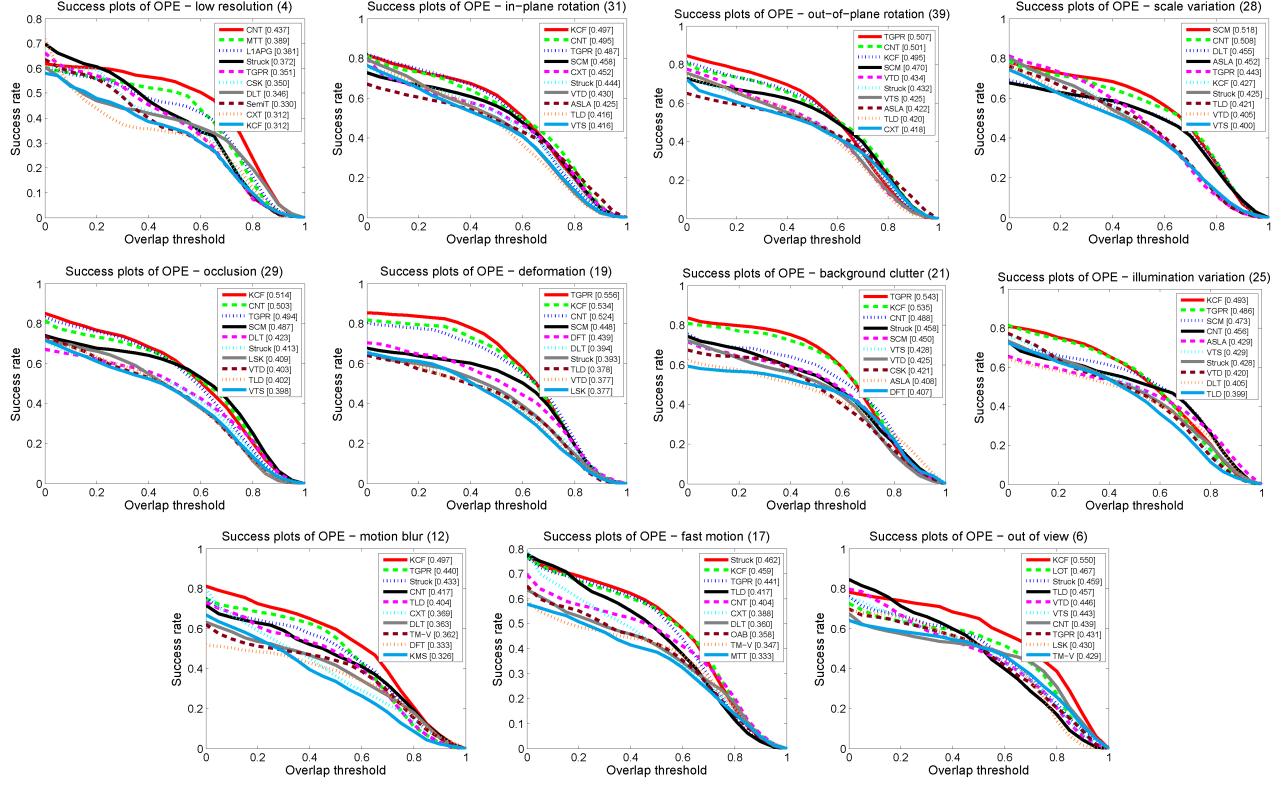


Fig. 5: The success plots of videos with different attributes. Best viewed on color display.

by the authors. The proposed CNT ranks 1st based on the success rate while 3rd based on the precision rate: in the success plot, the proposed CNT achieves the AUC of 0.545, which outperforms DLT by 10.9%. Meanwhile, in the precision plot, its precision score is 0.723, closely following TGPR 0.766 and KCF 0.740, but outperforms significantly DLT by 14.5%. Note that the proposed CNT exploits only the simple sparse image representation that encodes the local structural and geometric layout information of the target, yet achieves competitive performance to Struck and SCM that utilize useful background information to train discriminative classifiers. Furthermore, even using only specific target information from the first frame without further learning with auxiliary training data, CNT can still outperform DLT by a wide margin (more than 10 percents in terms of both success and precision rates), showing that the generic features offline learned from numerous auxiliary data may not adapt well to target appearance variations in visual tracking.

2) *Attribute-based Performance:* To facilitate analyzing the strength and weakness of the proposed algorithm, following [7], we further evaluate the trackers on videos with 11 attributes. Figure 5 shows the success plots of videos with different attributes, while Figure 6 shows the corresponding precision plots. We note that the proposed CNT ranks within top 3 on 7 out of 11 attributes in success plots, which outperforms DLT on all 11 attributes. In the precision plots, CNT ranks top 3 on 6 out of 11 attributes, and again outperforms DLT on all attributes. Since

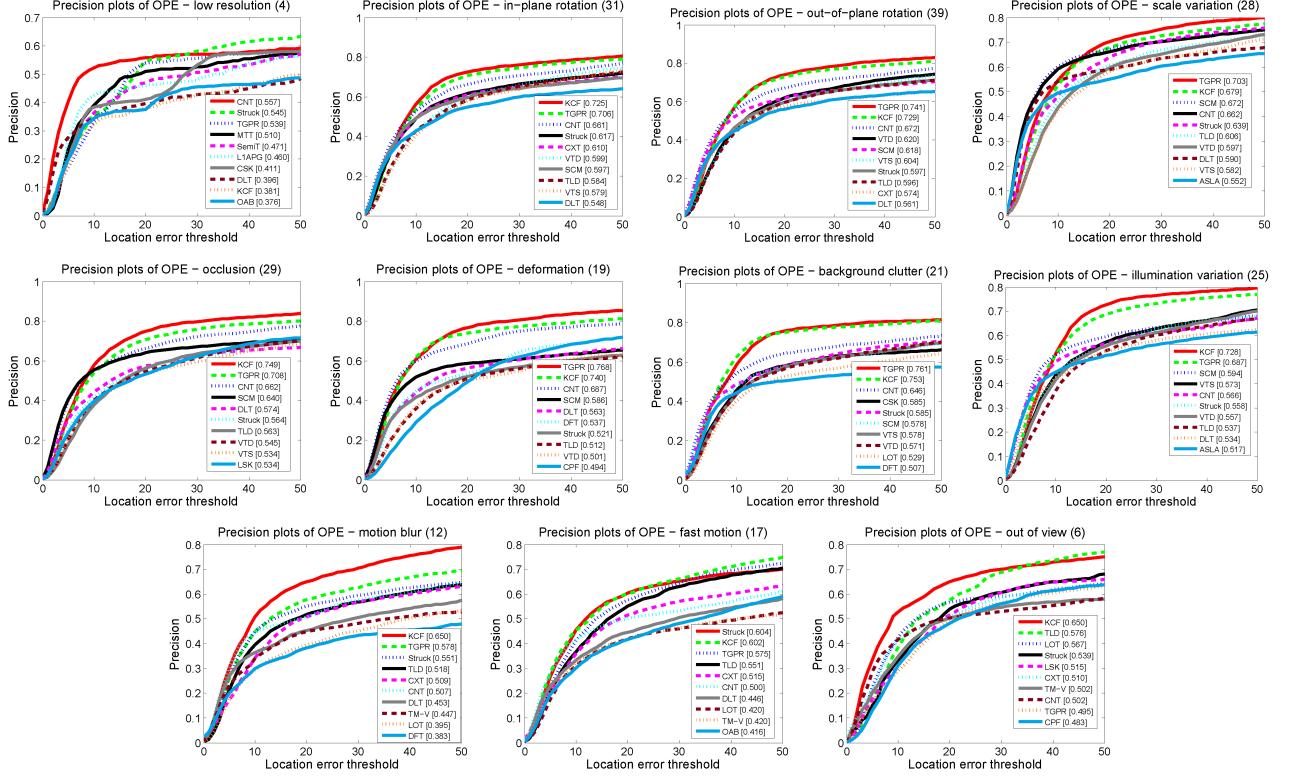


Fig. 6: The precision plots of videos with different attributes. Best viewed on color display.

the AUC score of the success plot is more accurate than the score at one position in the precision plot, as in [7], in the following we mainly analyze the rankings based on the success plots.

On the videos with attribute of *low resolution*, CNT ranks 1st among all evaluated trackers, which outperforms the 2nd counterpart MTT by 4.8%. The low resolution in the videos makes it difficult to extract effective hand-crafted features from the targets, thereby leading to undesirable results. On the contrary, CNT can extract dense useful information across the entire target region by convolution operators, and hence provides enough discriminative information to accurately separate the target from background.

For the videos with attributes such as *in-plane rotation*, *out-of-plane rotation*, *scale variation*, and *occlusion*, CNT ranks 2nd on all evaluated algorithms with a narrow margin (about 1 percent) to the 1st trackers, such as KCF, TGPR, and SCM. All these methods employ the local image features as image representations. KCF utilizes the HOG features to describe the target and its local context region, and TGPR extracts the covariance descriptors from the local image patches as image representations. Furthermore, both CNT and SCM employ local features extracted from the normalized local image patches. CNT exploits the useful local features across the target via filtering while SCM learns the local features from the target and background with sparse representation. Furthermore, both CNT and SCM utilize the target template from the first frame to handle drift problem. The results indicate that the local representation of the target and the target template in the first frame have a positive effect on handling the

above-mentioned attributes.

On the videos with *deformation* and *background clutter* attributes, CNT ranks 3rd which follows KCF and TGPR. KCF exploits dense HOG descriptors with predefined spatial structures to represent the target while TGPR explores similar spatial structures in which covariance descriptors are extracted, which encode them the local geometric layout information of the target, thereby rendering them capability to effectively handle deformation. CNT encodes the geometric layout information to multiple simple cell feature maps (see Figure 2), which are stacked together to a global representation, thereby equipping it with tolerance to deformation. Furthermore, CNT employs the useful background context information that is online updated and pooled in every frame, and hence provides helpful information to accurately locate the target from background clutter.

On the videos with the *illumination variation* and *motion blur* attributes, CNT ranks 4th while TGPR and KCF rank top 2. All these methods take advantage of normalized local image information, which is robust to illumination variation. Furthermore, when the target appearance changes greatly due to motion blur, the relatively unchanged background exploited by these methods can provide useful information to help localize the target.

Finally, for the videos with *fast motion* attribute, CNT ranks 5th while the top 4 trackers are Struck, KCF, TGPR, and TLD. CNT does not address fast motion well due to the simple dynamic model based on stochastic search, and so do SCM and ASLA. On the contrary, the trackers based on dense sampling (e.g., Struck, KCF, TGPR, and TLD) perform much better than others in the subset of fast motion due to their large search ranges. The performance of our CNT 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. The fast motion attribute also affects the performance of trackers on other attribute, such as the *out of view* attribute. There are 6 videos with out of view attribute but 5 videos out of them also have the fast motion attribute. Thus, KCF and Struck which work very well on the fast motion attribute also perform favorably with the out of view attribute. Furthermore, Struck employs a budgeting mechanism that can maintain the useful target samples from the entire tracking sequences, thereby it can redetect the target when it reappears after out of view, and hence results in a favorable result. Meanwhile, CNT explores the stable target information from the first frame, which helps redetect the object.

D. Qualitative Comparisons

1) *Deformation*: Figure 7 illustrates some screenshots of the tracking results in three challenging sequences where the target appearances undergo severe deformation. In the *bolt* sequence, several objects appear in the screen with rapid appearance changes due to shape deformation and fast motion. Only the CNT and KCF algorithms can track the target stably. The TGPR, SCM, TLD, ASLA, CXT and VTS methods undergo severe drift at the beginning of the sequence (e.g. #10, #100). The DLT algorithm drifts to the background at frame #200. The target in the *david3* sequence suffers from significant appearance variations due to non-rigid body deformation. Furthermore, the target appearance changes drastically when the person walks behind the tree and when he turns back, thereby increasing difficulty to robustly tracking the target. The DLT and CXT algorithms lose tracking the target after frame #50. The SCM, ALSA and VTS methods snap to some parts of background when the man walks behind the



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.



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

tree (e.g., #100, #135, #170). The TLD algorithm loses the target when the man turns back at frame #135. The Struck method snaps 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 *singer2* sequence undergoes both deformation and illumination variations. Only the CNT, TGPR and KCF algorithms perform well in the entire sequence. The proposed CNT handles deformation well because it employs the sparse local features with an adaptive threshold in (2) that can effectively filter out the varying parts in the appearance.

2) *Illumination Changes*: Figure 8 shows some sampled results in three sequences in which the targets undergo severe illumination variations. In the *car4* sequence, a moving vehicle passes beneath a bridge and under trees. Although the target undergoes drastic illumination variations at frames #180, #200, #240, the CNT method is able to track the object well. The DLT, CXT and VTS algorithms suffer from drift when the target undergoes a

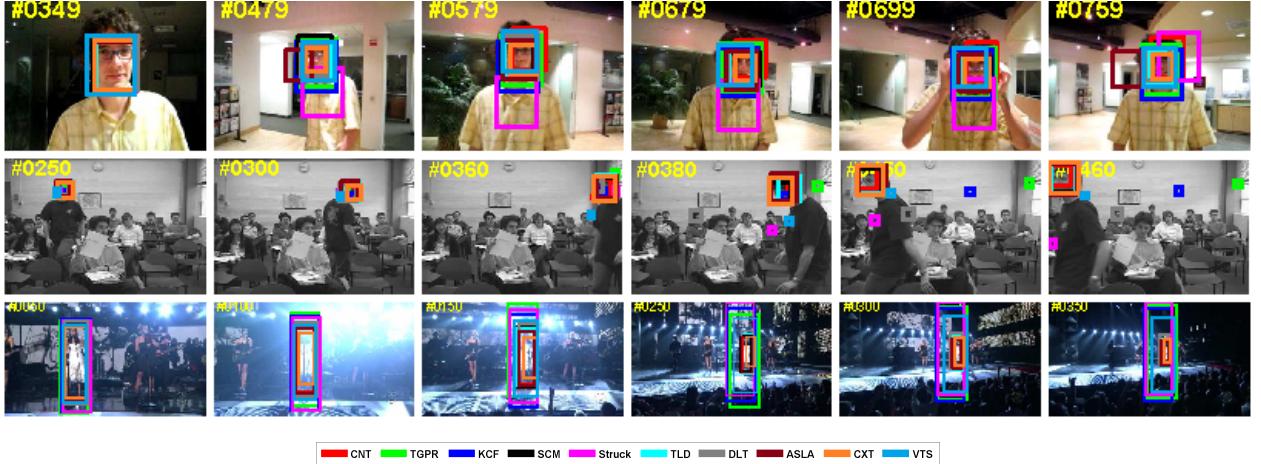


Fig. 9: Qualitative results of the 10 trackers over sequences *david*, *freeman3* and *singer1*, in which the targets undergo scale variations. Best viewed on color display.

sudden illumination change at frame #240. Furthermore, the target also undergoes obvious scale variations (e.g. #500, #650). Although the TGPR and KCF methods are able to successfully track the target, they cannot handle scale variations well (e.g., #500, #650). The target in the *skating1* sequence undergoes rapid pose variations and drastic light changes (e.g., #170, #360, #400). Only the CNT and KCF can persistently track the object from the beginning to the end. In the *trellis* sequence, a person moves underneath a trellis with large illumination change and cast shadows while changing his pose, resulting in a significant variation in appearance. The DLT and ASLA methods drift away to background (e.g., #510). The CNT, TLD and Struck methods are able to stably track the target, with much more accurately results than the TGPR, KCF and CXT methods that can persistently track the target. The CNT algorithm deals with illumination variations well because it extracts features via the normalized local filters with local brightness and contrast normalization.

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 in the target relative to the camera. The ASLA and VTS algorithms drift away to background (e.g. #479, #759). The KCF and Struck methods do not track the target scale, resulting in smaller success rate of their results on the attribute of scale variation than the CNT method. In the *freeman3* sequence, a person moves toward the camera, leading to a large scale variation in his face appearance. Furthermore, the appearance also changes much due to pose variation and low resolution, thereby increasing difficulty to accurately estimate its scale variation. The TGPR, KCF, Struck, DLT and VTS methods suffer from severe drift (e.g., #380, #450, #460). The CNT, SCM, TLD and CXT algorithms perform well. In the *singer1* sequence, the target moves far away from the camera, resulting in a large scale variation. The TGPR, KCF, Struck and VTS methods cannot perform well while the CNT, SCM,

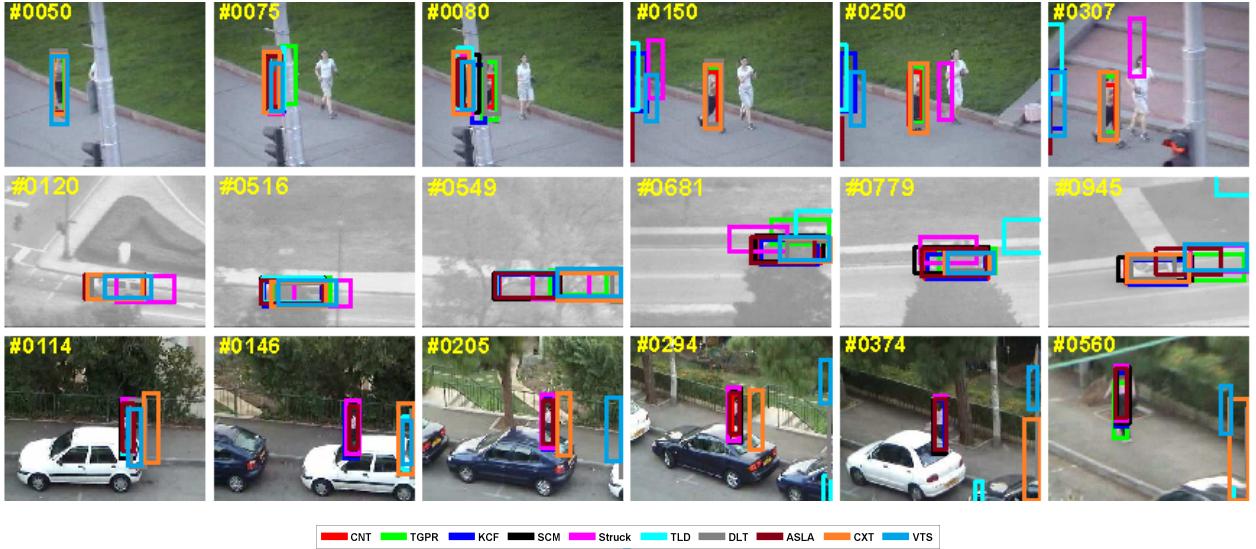


Fig. 10: Qualitative results of the 10 trackers over sequences *jogging-1*, *suv* and *woman*, in which the targets undergo scale variations. Best viewed on color display.

ASLA and CXT algorithm achieve much better performance. The CNT handles scale variation well because its representation is built on scale-invariant complex cell features (See Figure 3).

4) *Heavy Occlusion*: Figure 10 shows some sampled results of three sequences where the targets undergo heavy occlusion. In the *jogging-1* sequence, a person is almost completely occluded by the lamppost (e.g., #75, #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, a vehicle undergoes heavy occlusions several times from dense tree branches (e.g., #516, #549, #681, #799), which makes it very challenging to accurately track the object. The TGPR, Struck, TLD, ASLA and VTS methods cannot perform well (e.g., #945). In the *woman* sequence, almost half body of a person is occluded by cars several times (e.g., #114, #374). The CNT, TGPR, KCF, SCM, Struck and ASLA algorithms achieve favorable results. All these methods employ the local features that are robust to occlusions.

E. Analysis of CNT

To verify the effectiveness of some key components of CNT, we propose two variants of CNT: one is to utilize random patch filters to replace the filters learned by k -means algorithm in CNT, and the other is without the soft shrinkage component in CNT. Figure 11 shows their quantitative results on the benchmark dataset. We can observe that with random patch filters, the AUC score of success rate reduces about 7%. Meanwhile, the CNT without soft shrinkage can only achieve AUC score 0.469, following the original CNT method 0.545 by a large margin. Notwithstanding, both variants perform better than the DLT method. We can conclude that both the filters and the soft shrinkage components play a key role in determining the performance of CNT.

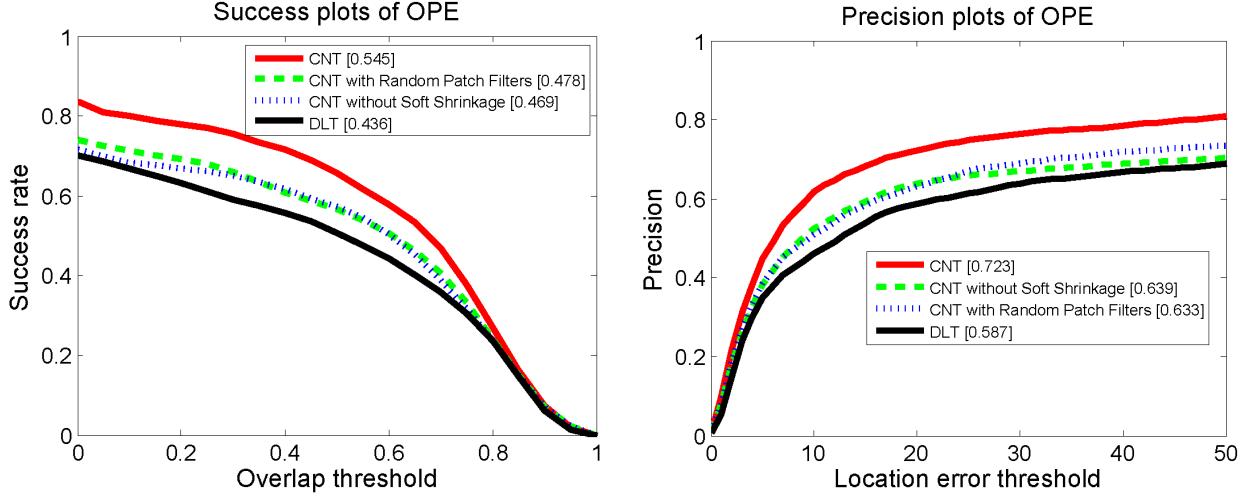


Fig. 11: The success plots and precision plots of OPE for CNT with different components. The DLT is taken as a baseline.

V. CONCLUDING REMARKS

In this paper, we have proposed a simple two-layer feed-forward convolutional network that is powerful enough to produce 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, and then 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 simple soft shrinkage strategy is employed to de-noise the target representation. A simple and 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.

There are several possible directions to extend this work. First, some more effective selection strategies can be exploited to choose a set of more informative filters. Second, it is interesting to take into account a discriminative tracking framework, which employs the proposed two-layer convolutional network as a feature extractor.

REFERENCES

- [1] S. Avidan, "Support vector tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1064–1072, 2004. [1](#), [4](#)
- [2] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1269–1276, 2010. [1](#), [2](#), [9](#)
- [3] Z. Kalal, J. Matas, and K. Mikolajczyk, "Pn learning: Bootstrapping binary classifiers by structural constraints," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 49–56, 2010. [1](#), [2](#), [4](#), [9](#)
- [4] S. Hare, A. Saffari, and P. H. Torr, "Struck: Structured output tracking with kernels," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 263–270, 2011. [1](#), [2](#), [4](#), [9](#)

- [5] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Proceedings of European Conference on Computer Vision*, pp. 864–877, 2012. [1](#), [4](#)
- [6] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Advances in Neural Information Processing Systems*, pp. 809–817, 2013. [1](#), [2](#), [3](#), [6](#), [9](#)
- [7] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2411–2418, 2013. [1](#), [3](#), [9](#), [10](#), [11](#)
- [8] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1822–1829, 2012. [2](#), [4](#), [9](#)
- [9] T. B. Dinh, N. Vo, and G. Medioni, "Context tracker: Exploring supporters and distractors in unconstrained environments," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1177–1184, 2011. [2](#), [9](#)
- [10] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1838–1845, 2012. [2](#), [5](#), [9](#)
- [11] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015. [2](#), [3](#), [4](#), [9](#)
- [12] J. Gao, H. Ling, W. Hu, and J. Xing, "Transfer learning based visual tracking with gaussian processes regression," in *Proceedings of European Conference on Computer Vision*, pp. 188–203, 2014. [2](#), [3](#), [4](#), [9](#)
- [13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012. [2](#)
- [14] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *International Conference on Machine Learning*, pp. 647–655, 2014. [2](#)
- [15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014. [2](#)
- [16] H. Li, Y. Li, and F. Porikli, "Robust online visual tracking with a single convolutional neural network," in *Proceedings of Asian Conference on Computer Vision*, pp. 194–209, 2014. [2](#)
- [17] X. Zhou, L. Xie, P. Zhang, and Y. Zhang, "An ensemble of deep neural networks for object tracking," in *IEEE International Conference on Image Processing*, pp. 843–847, 2014. [2](#)
- [18] J. Fan, W. Xu, Y. Wu, and Y. Gong, "Human tracking using convolutional neural networks," *IEEE Transactions on Neural Networks*, vol. 21, no. 10, pp. 1610–1623, 2010. [2](#)
- [19] L. Wang, T. Liu, G. Wang, K. L. Chan, and Q. Yang, "Video tracking using learned hierarchical features," *IEEE Transactions on Image Processing*, vol. 24, no. 4, pp. 1424–1435, 2015. [2](#)
- [20] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989. [2](#)
- [21] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. [2](#), [6](#)
- [22] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2559–2566, 2010. [2](#)
- [23] A. Coates, A. Y. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Advances in Neural Information Processing Systems*, pp. 215–223, 2011. [3](#)
- [24] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, "Robust object recognition with cortex-like mechanisms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 411–426, 2007. [3](#), [4](#), [6](#)
- [25] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nature neuroscience*, vol. 2, no. 11, pp. 1019–1025, 1999. [4](#)
- [26] V. Mahadevan and N. Vasconcelos, "On the connections between saliency and tracking," in *Advances in Neural Information Processing Systems*, pp. 1664–1672, 2012. [4](#)
- [27] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. V. D. Hengel, "A survey of appearance models in visual object tracking," *ACM Transactions on Intelligent Systems and Technology*, vol. 4, no. 4, p. 58, 2013. [4](#)

- [28] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004. [4](#)
- [29] I. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 810–815, 2004. [4](#), [5](#)
- [30] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125–141, 2008. [4](#), [8](#)
- [31] X. Mei and H. Ling, "Robust visual tracking and vehicle classification via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2259–2272, 2011. [4](#)
- [32] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust l1 tracker using accelerated proximal gradient approach," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1830–1837, 2012. [4](#), [9](#)
- [33] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2042–2049, 2012. [4](#), [9](#)
- [34] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 798–805, 2006. [4](#)
- [35] 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 *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1208–1215, 2009. [4](#)
- [36] B. Liu, J. Huang, L. Yang, and C. Kulikowsk, "Robust tracking using local sparse appearance model and k-selection," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1313–1320, 2011. [4](#), [5](#), [9](#)
- [37] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proceedings of British Machine Vision Conference*, pp. 47–56, 2006. [4](#), [9](#)
- [38] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1619–1632, 2011. [4](#), [6](#), [9](#)
- [39] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *The Journal of physiology*, vol. 148, no. 3, p. 574, 1959. [4](#)
- [40] M. Elad, M. A. Figueiredo, and Y. Ma, "On the role of sparse and redundant representations in image processing," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 972–982, 2010. [7](#)
- [41] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang, "Fast visual tracking via dense spatio-temporal context learning," in *Proceedings of European Conference on Computer Vision*, pp. 127–141, 2014. [7](#)
- [42] S. Ben-Yacoub, B. Fasel, and J. Luettin, "Fast face detection using mlp and fft," in *AVBPA*, 1999. [7](#)
- [43] J. Kwon and K. M. Lee, "Tracking by sampling trackers," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1195–1202, 2011. [9](#)
- [44] J. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proceedings of European Conference on Computer Vision*, pp. 702–715, 2012. [9](#)
- [45] L. Sevilla-Lara and E. Learned-Miller, "Distribution fields for tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1910–1917, 2012. [9](#)
- [46] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Proceedings of European Conference on Computer Vision*, pp. 661–675, 2002. [9](#)
- [47] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan, "Locally orderless tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1940–1947, 2012. [9](#)
- [48] R. Collins, X. Zhou, and S. K. Teh, "An open source tracking testbed and evaluation web site," in *PETS*, pp. 17–24, 2005. [9](#)
- [49] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003. [9](#)
- [50] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Proceedings of European Conference on Computer Vision*, pp. 234–247, 2008. [9](#)