Computational Visual Media
DOI 10.1007/s41095-017-0087-3 Vol. 3, No. 4, December 2017, 325–335

Research Article

Object tracking using a convolutional network and a structured output SVM

Junwei Li[1], Xiaolong Zhou[1], Sixian Chan[1], and Shengyong Chen[2](✉)

Abstract Object tracking has been a challenge in computer vision. In this paper, we present a novel method to model target appearance and combine it with structured output learning for robust online tracking within a tracking-by-detection framework. We take both convolutional features and hand-crafted features into account to robustly encode the target appearance. First, we extract convolutional features of the target by kernels generated from the initial annotated frame. To capture appearance variation during tracking, we propose a new strategy to update the target and background kernel pool. Secondly, we employ a structured output SVM for refining the target's location to mitigate uncertainty in labeling samples as positive or negative. Compared with existing state-of-the-art trackers, our tracking method not only enhances the robustness of the feature representation, but also uses structured output prediction to avoid relying on heuristic intermediate steps to produce labelled binary samples. Extensive experimental evaluation on the challenging OTB-50 video sequences shows competitive results in terms of both success and precision rate, demonstrating the merits of the proposed tracking method.

Keywords object tracking; convolutional network; structured learning; feature extraction

1 Introduction

Visual tracking is a fundamental research problem in computer vision and robotics, with wide applications such as intelligent video surveillance, transportation monitoring, robot–human interaction, etc. In recent years, many excellent tracking algorithms have been proposed, but it remains a challenging problem for a tracker to handle occlusion, abrupt motion, appearance variation, and background clutter.

In this paper, we propose a novel tracking method which utilizes a discriminative convolutional network [1] and HOG descriptors [2] to encode target appearance, together with a structured output support vector machine (SO-SVM) to jointly estimate target appearance. In the proposed method, tracking is formulated as binary classification and structured output tasks, to select the most likely target candidate and reject background patches. It uses an online trained structural output classifier within a particle filter framework. The convolutional filters for modeling target appearance are generated from the initial frame (annotated manually). We perform a soft-shrink operation on the output convolutional feature maps to enhance their robustness. One of the most significant advantages is that the convolutional filters are generated from both the target and its surrounding area, so fully exploiting local structure and internal geometric layout of the target. Additionally, our method employs an SO-SVM to overcome the drawback that samples used for training the classifier are all equally weighted, meaning that a negative example which overlaps significantly with the tracker's bounding box is treated the same as one which overlaps very little. Another advantage of SO-SVM is that the labeler no longer labels the samples as positive or negative based on intuition and heuristics.

The remainder of this paper is organized as follows. Section 2 gives a brief overview of related work. Details of our tracking method are described in Section 3. Section 4 reports qualitative

1 Zhejiang University of Technology, Hangzhou, 310023, China.

2 Tianjin University of Technology, Tianjin, 300384, China. E-mail: csy@tjut.edu.cn (✉).

Manuscript received: 2017-02-27; accepted: 2017-04-27

and quantitative experimental results. We finally conclude this paper in Section 5.

2 Related work

Most existing tracking methods belong to two categories: generative models and discriminative models. The generative approach formulates the tracking problem as minimizing reconstruction error, while the discriminative model method considers the tracking problem as a binary classification task to separate the target from the background. From another perspective, a tracker can be decomposed into two components: an online updated appearance model for feature extraction and an observation model to find the most probable target transformation.

Recent tracking methods mainly focus on designing a robust appearance model [3] to capture target appearance variation. The most popular approach based on the discriminative model casts tracking as a foreground and background separation problem, performing tracking by learning a classifier using multiple instance learning [4], P-N learning [5], online boosting [6, 7], SVM [8], structured output SVMs [9], CRFs [10], probability hypothesis density methods [11, 12], etc. These tracking methods first train a classifier online, inspired by statistical machine learning methods, to separate the target from the background surrounding the target location in the previous frame. Generative methods describe the target's appearance using generative models and search for target regions that best fit the model. Various generative target appearance modeling algorithms have been proposed using sparse representation [13, 14], density estimation [15, 16], and incremental subspace learning [17]. In Ref. [18], generative and discriminative models were combined for more accurate online tracking. Some efficient trackers have been proposed using hand-crafted features, including Haar-like feature histograms [4, 6, 9, 19], HOG descriptors [2], binary features [20], and covariance descriptors [21]. However, such trackers do not adapt well to target appearance variation.

To overcome the shortcomings of handcrafted features in modeling object appearance, deep networks have been employed to directly learn features from raw data without resorting to manual intervention. Convolutional features have been used in many applications such as Ref. [22]. In Ref. [23], Li et al. used a convolutional neural network (CNN) for visual tracking with multiple image cues as inputs. In Ref. [24], Zhou et al. used an ensemble of deep networks in combination with an online boosting method for visual tracking. Reference [25] presented a human tracking algorithm that learns a specific feature extractor with CNNs. Numerous auxiliary data are required for offline training the deep networks; the pre-trained model is then used for online visual tracking. Wang and Yeung [26] developed a deep learning tracking method that uses stacked de-noising auto-encoders to learn generic features from a large number of auxiliary images. Reference [27] used a two-layer CNN to learn hierarchical features from auxiliary video sequences; it takes into account complicated motion transformations and appearance variations in visual tracking. A drawback of all the above frameworks is that they need a large amount of auxiliary data to pre-train a deep network model; such models can be highly specific and have poor adaptive ability.

In Ref. [1] Zhang et al. incorporated convolutional networks (convolutional filters defined as normalized image patches from the first frame) which do not require auxiliary data to train filters. They achieved state-of-the-art precision. Several tracking algorithms based on hand-crafted features have been developed within a multiple instance learning framework, aiming to improving the poor ability of hand-crafted features to represent semantic level features. Grabner et al. proposed an online boosting algorithm to select features for tracking. However, these trackers [28, 29] use one positive sample (i.e., the current tracker location) and a few negative samples when updating the classifier. As the appearance model is updated with noisy and potentially misaligned examples, this often leads to the problem of tracking drift. A semi-supervised learning approach can be used in which positive and negative samples are selected via an online classifier with structural constraints. Yang et al. [30] present a discriminative appearance model based on superpixels. It is able to handle heavy occlusion and recover from drift. In Ref. [9], Hare et al. used an online structured output support vector

machine (SVM) for robust tracking; it can mitigate the effect of wrongly labeled samples. Reference [31] introduced a fast tracking algorithm which exploits the circulant structure of the kernel matrix in SVM classifiers so that it can be efficiently computed by the fast Fourier transform algorithm.

3 Method

In this section, we describe our proposed tracking method in detail. The tracking problem is formulated as a detection task, and the pipeline of the proposed approach is shown in Fig. 1. We assume that the tracking target is manually annotated in the first frame. To model target appearance, we sample various background and foreground convolutional kernels to encode target and background structural information. When a new frame arrives, we first extract its convolutional feature map to estimate the target transformation. Secondly, we incorporate structured output learning and HOG descriptors to predict the target location and scale variation. Lastly, the tracking results are combined to jointly determine the target transformation and scale variation.

3.1 Feature extraction by convolutional network

The convolutional network includes two separate layers. Firstly, a set of background and foreground convolutional kernels are generated from a bank of filters which sample the input frame using a sliding window. Secondly, to enhance the robustness of the convolutional feature representation, all feature maps are stacked together, and the final feature vector is determined by solving a sparse representation equation.

In the initial frame, a set of samples, denoted $\boldsymbol{I} \subset \boldsymbol{R}^{n\times n}$, is warped to a canonical size of $n \times n$ in grayscale color space. Then each sample is pre-processed by subtracting the mean and L_2 normalization is performed, aiming to modify local brightness differences and achieve contrast normalization, respectively. A sliding window strategy is employed to generate a bank of patches with the field size $w \times w$. This results in a total of $l = (n-w+1) \times (n-w+1)$ image patches sampled from the initial frame.

Following the pre-processing step, the k-means algorithm is used to select multiple convolution filter kernels $F_1^{\mathrm{o}} = \{F_{1,1}^{\mathrm{o}}, \ldots, F_{1,d}^{\mathrm{o}}\} \subseteq \mathcal{Y}_1$ from the filter bank using the initial frame as the representative target filters. The remaining object filter kernels $F_t^{\mathrm{o}} = \{F_{t,1}^{\mathrm{o}}, \ldots, F_{t,d}^{\mathrm{o}}\} \subseteq \mathcal{Y}_t$ are selected dynamically to capture target appearance variation, and are generated by clustering the target filter bank of the tth frame. $\mathcal{Y}_1$ and $\mathcal{Y}_t$ are filter banks obtained

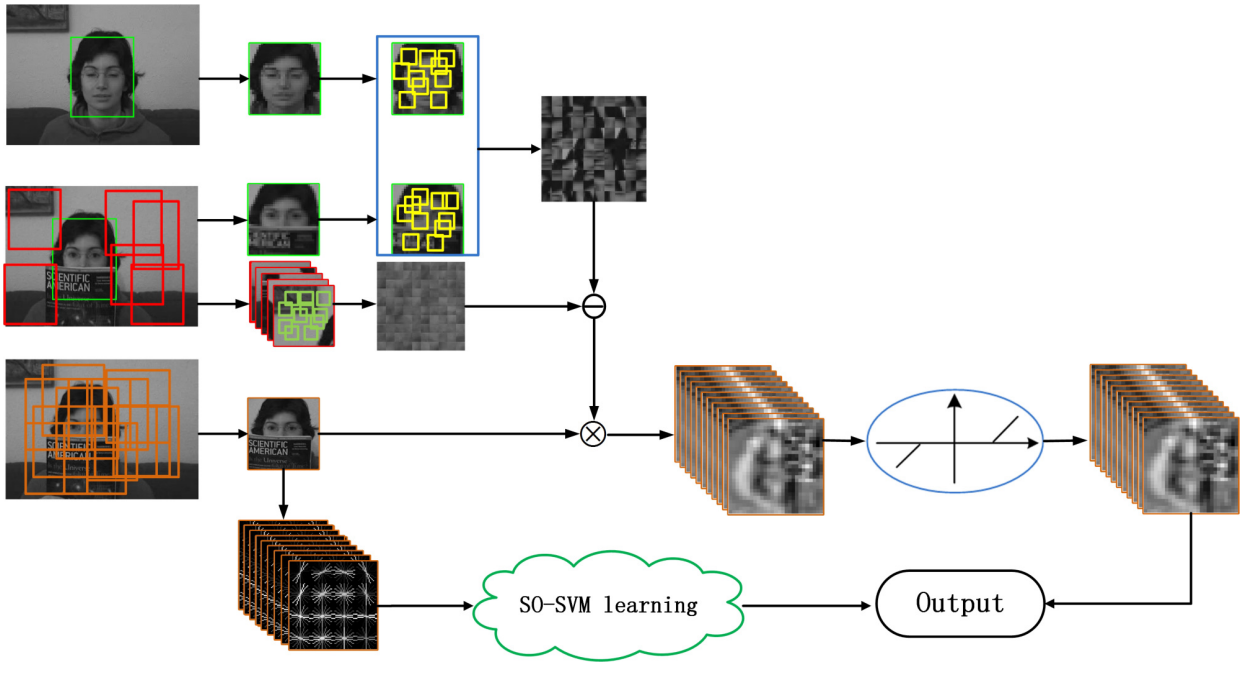


Fig. 1 Architecture of the proposed tracking algorithm.

from the initial and tth frame, respectively. The strategy of generating a dynamic convolutional filter is the most significant difference from Ref. [1]. One obvious advantage is that our filters have the ability to adapt in the face of target occlusion and deformation, and illumination changes, which cause target appearance variation. In other words, the proposed convolutional filters are more robust in dynamic environments.

Given the target candidate image $\boldsymbol{I}$, and the ith object filter kernels $F_i^{\mathrm{o}} \in \boldsymbol{R}^{w\times w}$, the convolution of $\boldsymbol{I}$ and the filters is denoted by $S_i^{\mathrm{o}} \in \boldsymbol{R}^{(n-w-1)\times(n-w-1)}$, where $S_i^{\mathrm{o}} = \left\{F_{1,i}^{\mathrm{o}}, F_{t,i}^{\mathrm{o}}\right\} \otimes \boldsymbol{I}$, and $\otimes$ denotes the convolution operator. The local filters encode stable object visual information from both the initial frame and the previous frame even though the object may experience significant appearance change from the initial frame. Thereby, we can extract more discriminative features and effectively handle the drift problem.

The background context surrounding the object provides useful information to discriminate the target. The convolutional networks select m background samples surrounding the object, and then the same cluster algorithm is used to generate background filters $F_i^{\mathrm{b}} = \{F_{i,1}^{\mathrm{b}}, \ldots, F_{i,d}^{\mathrm{b}}\} \subseteq \mathcal{Y}_{\mathrm{b}}$ from the ith background sample. An average pooling strategy is operated to summarize each filter in F_i^{b}. Next, the background kernels F^{b} that encode the visual information and geometric layout surrounding the object are generated:

$$F^{\mathrm{b}} = \left\{F_1^{\mathrm{b}}, F_2^{\mathrm{b}}, ..., F_d^{\mathrm{b}}\right\} \tag{1}$$

$$F_i^{\mathrm{b}} = \frac{1}{m}\sum_{i=1}^{m} F_{i,1}^{\mathrm{b}} \tag{2}$$

Given the input image $\boldsymbol{I}$, the ith background feature map is defined as $S_i^{\mathrm{b}} = S_i^{\mathrm{b}} \otimes \boldsymbol{I}$. The final feature map is

$$S_i = S_i^{\mathrm{o}} - S_i^{\mathrm{b}} = \left\{[F_{1,i}^{\mathrm{o}}, F_{t,i}^{\mathrm{o}}] - F_i^{\mathrm{b}}\right\} \otimes \boldsymbol{I} \tag{3}$$

To further enhance the strength of this representation and eliminate the influence of noise, a complex cell feature map that is a 3D tensor $\boldsymbol{C} \in \boldsymbol{R}^{(n-w-1)\times(n-w-1)\times d}$ is constructed, which stacks d different simple cell feature maps constructed with the filter set $F = \{F_1^{\mathrm{o}}, F_t^{\mathrm{o}}\} \bigcup F_i^{\mathrm{b}}$. A sparse vector $\boldsymbol{c}$ is set to approximate vec($\boldsymbol{C}$) by minimizing the following objective function:

$$\hat{\boldsymbol{c}} = \arg\min \lambda\|\boldsymbol{c}\|_1^1 + \frac{1}{2}\|\boldsymbol{c} - \mathrm{vec}(\boldsymbol{C})\|_2^2 \tag{4}$$

where vec($\boldsymbol{C}$) is a column vector concatenating all the elements in $\boldsymbol{C}$, of length $(n-w+1)^2 d$. The optimization problem in Eq. (4) has a closed form solution, as explained in Ref. [32]:

$$\hat{\boldsymbol{c}} = \mathrm{sign}(\boldsymbol{C})\max(0, \mathrm{abs}(\mathrm{vec}(\boldsymbol{C}))) - \mathrm{median}(\boldsymbol{C}) \tag{5}$$

where median(vec($\boldsymbol{C}$)) is robust to target appearance variation and noise interference.

3.2 HOG feature extraction

Hand-crafted features are morphological, shape, statistical, or textural based representations that attempt to encode object appearance at low-level, and are the fundamental elements of object representation. Contrary to the high-level semantic features found by convolutional networks, which can be treated as a black-box object representation, hand-crafted features encode object appearance and effectively preserve structure information, which is very important in object tracking. In this paper, we use HOG (histograms of oriented gradients) features [33] as complementary features to jointly encode target appearance. HOG descriptors have several advantages. For example, their gradient structure is very characteristic of local shape, they are computed in a local cell with an easily controllable degree, and they are invariant to local geometric and photometric transformations. In other words, translations or rotations make little difference even if they are much smaller than the local spatial or orientation bin size. All of these advantages of HOG features play a key role in target location and scale estimation.

3.3 Structured output learning

Traditional tracking-by-detection approaches employ a classifier trained online to distinguish the target object from its surrounding background. In the tracking process, the classifier is used to estimate the object's transformation by searching for the maximum classification score amongst a set of target candidates around the target's location in the previous frame, typically using a sliding window or another motion model to generate target candidates. Given the estimated target location, traditional tracking methods generate a set of binary labelled training samples to update the classifier online.

This tracking framework raises a number of issues. Firstly, it is not clear how to label the training samples in a principled manner. One popular way is to utilize predefined rules such as the distance between a sample and the estimated target candidate to determine whether a sample should be labelled as positive or negative. Secondly, the goal of a classifier is to predict a binary label instead of a structured output. However, the objective for a tracker is to estimate the object's transformation accurately. In Ref. [6], Ma et al. formulated the tracking problem as structured output prediction to mitigate the gap between binary classification and accurate target transformation determination.

When a new frame arrives, the ultimate goal for a tracker is to estimate the target position $\boldsymbol{p}$ (a 2D rectangle) in the current frame. To capture target appearance variation, the classifier is updated online based on the newly estimated target appearance around $\boldsymbol{p}$ and the corresponding samples $\boldsymbol{x}_t^{\boldsymbol{p}} \in \mathcal{X}$. The classifier is trained on the example pairs $(\boldsymbol{x}, z)$, where $z = \pm 1$ is a binary label, and makes its prediction according to $z = \text{sign}(h(\boldsymbol{x}))$, where $h : \mathcal{X} \to R$ is the classification confidence function which maps from feature space $\mathcal{X}$ to a real target confidence value R. Let $\boldsymbol{p}_{t-1}$ denote the estimated bounding box at time $(t-1)$. The objective is to estimate a transformation $\boldsymbol{p}_t \in \mathcal{Y}$, so the new position of the object is approximated by the composition $\boldsymbol{p}_t = \boldsymbol{p}_{t-1} \circ \boldsymbol{y}_t$. $\mathcal{Y}$ is the search space. Mathematically, the estimation process is converted to searching for the position change relative to the previous frame by solving:

$$\boldsymbol{y}_t = \arg\max_{\boldsymbol{y} \in \mathcal{Y}} (\boldsymbol{x}^{\boldsymbol{p}_{t-1} \circ \boldsymbol{y}}) \tag{6}$$

To overcome the above two issues arising from traditional classifiers, we utilize the structured output SVM (SO-SVM) framework to estimate object location changes. The output space is thus the space of all transformations $\mathcal{Y}$ instead of the confidence labels. Thus we introduce an SO-SVM based discriminant function F:

$$\boldsymbol{y}_t = f(\boldsymbol{x}_t^{\boldsymbol{p}_t{}^{-1}}) = \arg\max_{\boldsymbol{y} \in \mathcal{Y}} F(\boldsymbol{x}_t^{\boldsymbol{p}_{t-1}}, \boldsymbol{y}) \tag{7}$$

SO-SVM performs a maximization step in order to predict the object transformation, while the discriminant function F includes the label $\boldsymbol{y}$ explicitly, meaning it can be incorporated into the learning algorithm. The model update procedure is performed on a labelled example pair $(\boldsymbol{x}_t^{p_t}, \boldsymbol{y}^0)$. In Eq. (6), $\boldsymbol{y}$ is generated by a motion model, and the objective is to estimate the object confidence of each candidate sample instead of the target transformation.

Function F measures the compatibility between (x, y) pairs, and gives high scores to those which are well matched. We restrict F to be of the form $F(\boldsymbol{x}, \boldsymbol{y}) = \langle \boldsymbol{w}, \phi(\boldsymbol{x}, \boldsymbol{y}) \rangle$, where $\phi(\boldsymbol{x}, \boldsymbol{y})$ is a joint kernel map. The parameters can be learned in a large-margin framework from a set of example pairs $\{(\boldsymbol{x}_1, \boldsymbol{y}_1), ..., (\boldsymbol{x}_n, \boldsymbol{y}_n)\}$ by minimizing a convex objective function:

$$\begin{aligned} &\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n} \varepsilon_i \\ &\text{such that} \qquad \forall i : \varepsilon_i \geqslant 0, \\ &\forall i, \forall \boldsymbol{y} \neq \boldsymbol{y}_i : \langle \boldsymbol{w}, \delta(\phi_i(\boldsymbol{y})) \rangle \geqslant \Delta(\boldsymbol{y}_i, \boldsymbol{y}) - \varepsilon_i \end{aligned} \tag{8}$$

where $\delta(\phi_i(\boldsymbol{y})) = \phi(\boldsymbol{x}_i, \boldsymbol{y}_i) - \phi(\boldsymbol{x}_i, \boldsymbol{y})$, and $\Delta(\boldsymbol{y}_i, \boldsymbol{y})$ is a loss function. The value of $\Delta(\boldsymbol{y}_i, \boldsymbol{y})$ decreases towards 0 as $\boldsymbol{y}$ and $\boldsymbol{y}_i$ become more similar. The optimization aims to ensure the value of $F(\boldsymbol{x}_i, \boldsymbol{y}_i)$ is greater than $F(\boldsymbol{x}_i, \boldsymbol{y})$ for any $\boldsymbol{y} \neq \boldsymbol{y}_i$, by a margin which depends on a loss function Δ. The loss function plays an important role in our approach, as it allows us to address the issue raised previously of all samples being treated equally.

Using standard Lagrangian duality techniques, Eq. (8) can be converted into an equivalent dual form:

$$\begin{aligned} &\max_{\alpha} \sum_{i, \boldsymbol{y} \neq \boldsymbol{y}_i} \Delta(\boldsymbol{y}, \boldsymbol{y}_i)\alpha_i^{\boldsymbol{y}} - \frac{1}{2}\sum_{i;j} \alpha_j^{\boldsymbol{y}} \alpha_j^{\bar{\boldsymbol{y}}} \langle \delta\phi_i(\boldsymbol{y}), \delta\phi_j(\bar{\boldsymbol{y}}) \rangle \\ &\text{such that} \quad \forall \boldsymbol{y} \neq \boldsymbol{y}_i : \alpha_i^{\boldsymbol{y}} \geqslant 0; \forall i : \sum_{\boldsymbol{y} \neq \boldsymbol{y}_i} \alpha_i^{\boldsymbol{y}} \leqslant C \end{aligned} \tag{9}$$

The discriminant function then becomes $F(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i, \bar{\boldsymbol{y}} \neq \boldsymbol{y}_i} \alpha_i^{\bar{\boldsymbol{y}}} \langle \delta\phi_j(\bar{\boldsymbol{y}}), \phi(\boldsymbol{x}, \boldsymbol{y}) \rangle$. This dual problem can be considerably simplified by reparametrizing it with $n * k$ variables $\beta_i^{\boldsymbol{y}}$ defined by

$$f(x) = \begin{cases} -\alpha_i^{\boldsymbol{y}}, & \text{if} \quad \boldsymbol{y} \leqslant \boldsymbol{y}_i \\ \sum\limits_{\bar{\boldsymbol{y}} \neq \boldsymbol{y}}^{\alpha^{\bar{\boldsymbol{y}}}}, & \text{otherwise} \end{cases} \tag{10}$$

which leads to a much simpler expression for the dual problem in Eq. (11) and corresponding discriminant function Eq. (12):

$$\begin{aligned} &\max_{\alpha} \sum_{i, \boldsymbol{y} \neq \boldsymbol{y}_i} \Delta(\boldsymbol{y}, \boldsymbol{y}_i)\beta_i^{\boldsymbol{y}} - \\ &\qquad \frac{1}{2}\sum_{i;j} \beta_j^{\boldsymbol{y}} \beta_j^{\bar{\boldsymbol{y}}} \langle \delta\phi_i(\boldsymbol{x}_i, \boldsymbol{y}), \delta\phi_j(\boldsymbol{x}_j, \bar{\boldsymbol{y}}) \rangle \end{aligned}$$

$$\text{such that} \quad \forall i, \forall \boldsymbol{y} : \beta_i^{\boldsymbol{y}} \leqslant \delta \phi_i(\boldsymbol{y}, \boldsymbol{y}_i) \mathcal{C}$$
$$\forall i : \sum_{\boldsymbol{y}} \beta_i^{\boldsymbol{y}} = 0 \tag{11}$$

$$F(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i, \bar{\boldsymbol{y}}} \beta_i^{\bar{\boldsymbol{y}}} \langle \phi_i(\boldsymbol{x}_i, \boldsymbol{y}), \phi_j(\boldsymbol{x}, \boldsymbol{y}) \rangle \tag{12}$$

3.4 Tracking algorithm

The proposed tracking algorithm is formulated within a particle filter framework. Given the observation set $\mathcal{O}_t = \{\boldsymbol{o}_1, \boldsymbol{o}_2, ..., \boldsymbol{o}_t\}$, the goal is to determine the maximize the a posteriori probability $p(\boldsymbol{s}_t|\mathcal{O}_t)$ using Bayes Theorem:

$$p(\boldsymbol{s}_t|\mathcal{O}_t) \propto p(\mathcal{O}_t|\boldsymbol{s}_t) \int p(\boldsymbol{s}_t|\boldsymbol{s}_{t-1}) p(\boldsymbol{s}_{t-1}|\mathcal{O}_{t-1}) \mathrm{d}\boldsymbol{s}_{t-1} \tag{13}$$

where $\boldsymbol{s}_t = [x_t, y_t, s_t]$ denotes the target state translation (x_t, y_t) and scale s_t, and $p(\boldsymbol{s}_t|\boldsymbol{s}_{t-1})$, $p(\mathcal{O}_t|\boldsymbol{s}_t)$ are the motion model that predicts state $\boldsymbol{s}_t$ based on the previous state $\boldsymbol{s}_{t-1}$ and the likelihood of observation respectively.

The sparse feature vector $\boldsymbol{c}$ in Eq. (5) is used as the object feature template. It is updated incrementally to accommodate appearance changes over time for robust visual tracking. We use temporal low-pass filtering to update the tracking model:

$$\boldsymbol{c}_t = (1 - \rho)\boldsymbol{c}_{t-1} + \rho \hat{\boldsymbol{c}}_{t-1} \tag{14}$$

where ρ is a learning parameter, $\boldsymbol{c}_t$ is the target template in the tth frame, and $\hat{\boldsymbol{c}}_{t-1}$ is the sparse representation of the tracked object in frame $t-1$. Note that a significant innovation compared with the strategy in Ref. [1] is that the convolutional filters for extracting the object template are updated based on the newly tracked target:

$$F^{\mathrm{o}} = F_1^{\mathrm{o}} \bigcup \mathrm{cluster}_d\left(p_t\right), p_t \in \boldsymbol{R}^{w \times w} \tag{15}$$

where $\mathrm{cluster}_d(\cdot)$ denotes a clustering operation with d clssses, and p_t are the image patches generated by a sliding window within the tracked object region. One of the advantages of this operation is that we can both preserve the original target appearance as well as capture new object variation, preventing target drift.

4 Experiments

We have evaluated our proposed tracking algorithm on a public dataset [34] which includes 50 video sequences categorized with 11 attributes based on different challenging factors including illumination variation (IV), out-of-plane rotation (OPR), scale variation (SV), occlusion (OCC), deformation (DEF), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of-view (OV), background clutter (BC), and low resolution (LR). We compared the proposed tracking algorithm on OTB-50 against other methods including SCM [35], Struck's method [9], TLD [20], MIL [3], and CT [19]. In addition, we also compared our method with a state-of-the-art method convolutional network based tracker (CNT). For quantitative evaluation, we used a success plot and a precision plot for one-pass evaluation (OPE) protocols. All 50 videos were processed using the same parameter values during the tracking process, without modification.

The results in Fig. 2 compare our tracking framework and CNT, Struck's method, TLD, MIL, SCM, CT. It is clear that the combination of convolutional features and HOG features plays an important role in robust object tracking. For both success rate and precision rate for the OTB-

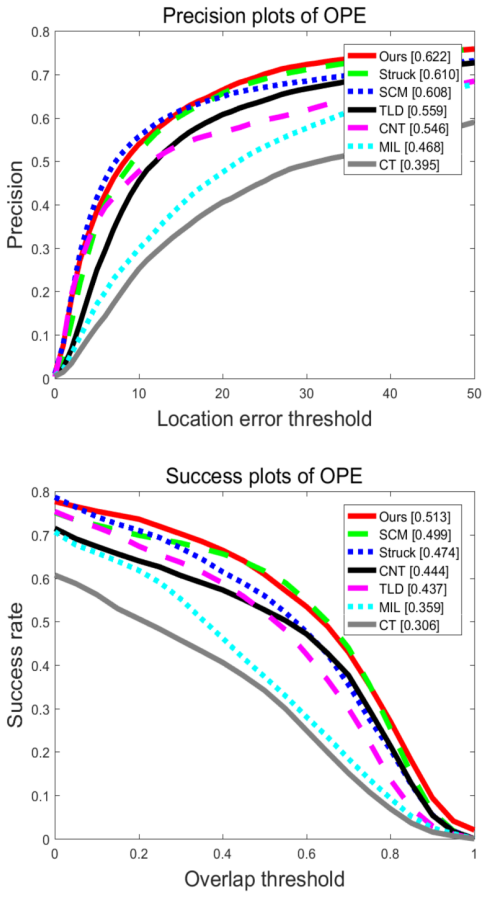


Fig. 2 One-pass evaluation. Top: average precision plot. Bottom: success rate plot.

50 dataset, our method achieves the maximum area under the curve. Unlike the CNT tracker, our tracking method updates the convolutional filter during the tracking process to capture target appearance variation, taking into account both the original target appearance and any target variation. In addition, the combination of HOG features and a structural output SVM improves the success and precision rates of our tracker by 13.4% and 12.2% respectively. In contrast, by adding convolutional features, our tracker enhances the performance of Struck's method by 7.6% and 2.0% in terms of overall success rate and precision rate respectively.

To analyze the strengths and weaknesses of the proposed algorithm, we further evaluate the trackers on videos with 11 attributes. Figure 3 shows success rate plots for videos with different attributes, while Fig. 4 shows the corresponding precision plots. In the success rate evaluation, our tracking algorithm ranks first in 8 out of 11 attributes. Meanwhile, for the video sequences with occlusion, deformation, and fast motion, our tracking method is ranked second, with the SCM and Struck trackers achieving the best performance—they employe useful background information to train discriminative classifiers. In the precision plots in Fig. 4, our tracking algorithm is ranked first in 6 out of 11 attributes, namely scale variation, out-of-plane rotation, in-plane rotation, illumination variation, motion blur, and background cluster, while our tracker is ranked second for the other 5 attributes.

Figures 5 and 6 show some tracking results on some challenge image sequences for 7 trackers. The *basketball*, *deer*, and *soccer* video sequences contain illumination change, pose variation, and fast motion. In the above 3 sequences, the CNT tracker fails around frames 134, 6, and 79, respectively. At *basketball* frame 531, all other trackers (CT, MIL, SCM, Struck, TLD) lose the target. The *coke* and *freeman4* sequences contain significant out-of-plane rotation, occlusion, and pose variation. Tracking results on the *freeman4* sequence show that most trackers drift away from the target when it is heavily occluded. These tracking results prove the effectiveness and robustness of the proposed feature representation (a combination of HOG and convolutional features) and structural output learning. The proposed tracking method can cope

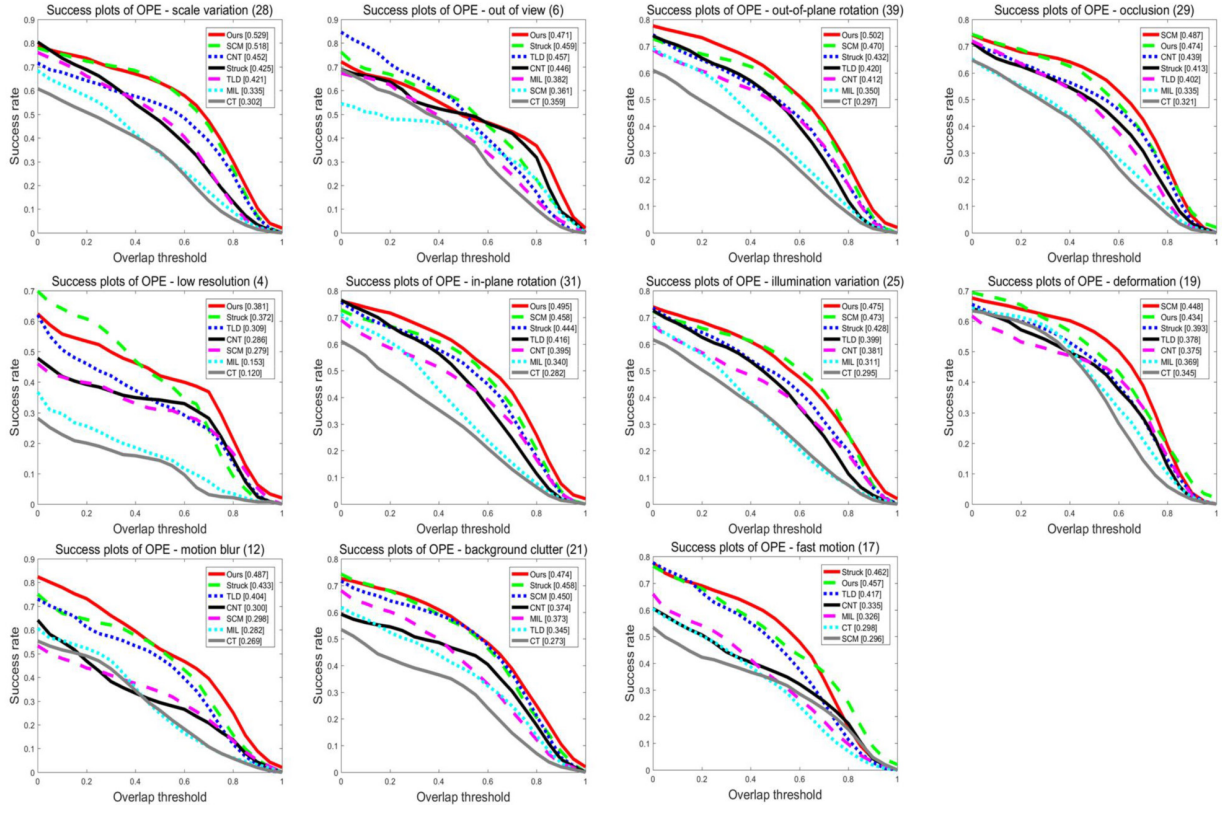


Fig. 3 Tracker success rates for videos with different attributes, annotated with the area under the curve. The number in each title indicates the number of video sequences with a given attribute.

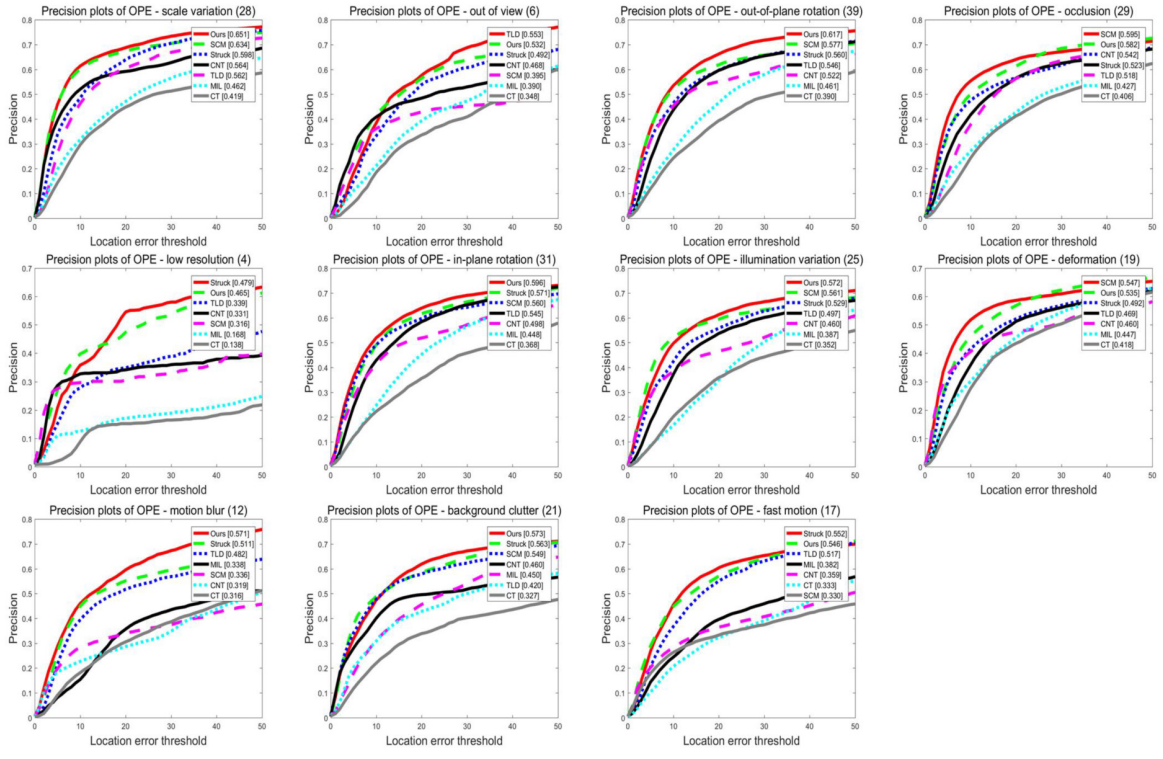


Fig. 4 Tracker precision for videos with different attributes, annotated with the area under the curve. The number in each title indicates the number of video sequences with a given attribute.

Fig. 5 Qualitative results using the proposed method on various challenging sequences (basketball, deer, liquor, soccer) having illumination variation. Frame number is shown at the top left of each frame in green.

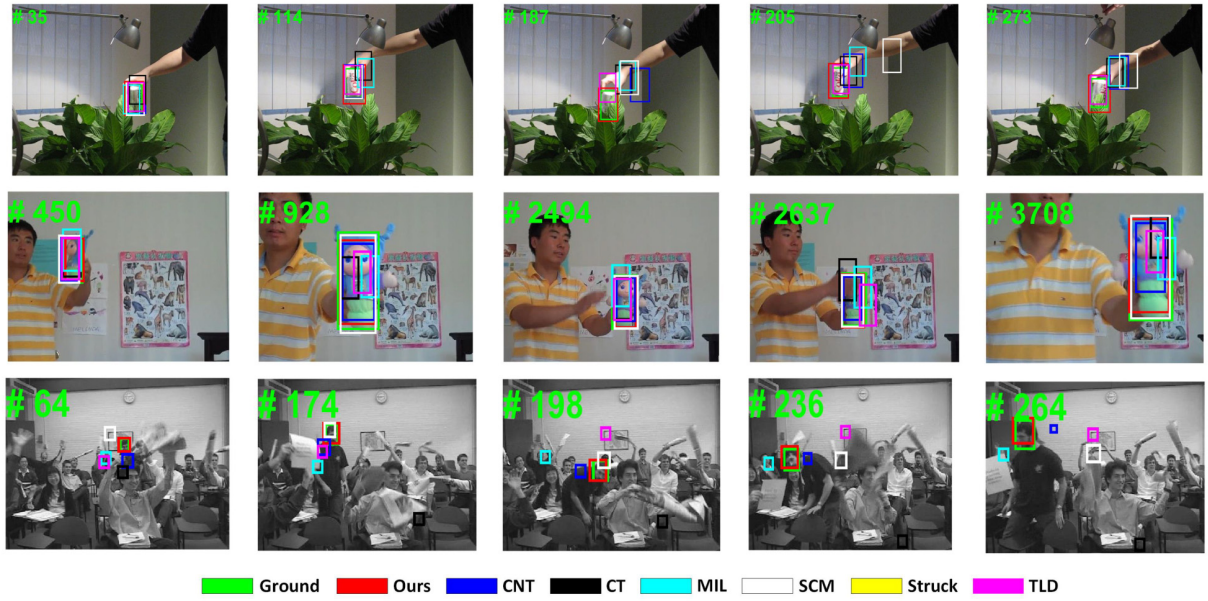


Fig. 6 Qualitative results using the proposed method on various challenging sequences (coke, doll, freeman4) having out-of-plane rotation. Frame number is shown at the top left of each frame in green.

with target appearance variation in the tracking process by updating the object kernels over time. To make the tracker robust to target scale variation, we employ a combination of HOG descriptors and SO-SVM to capture mid-level object cues. However, the time consumed is only 1.2 times greater than that used by the CNT tracker, and runs at 4.1 fps.

5 Conclusions

In this paper, we have proposed a novel method to model target appearance with background and foreground convolutional filters for online tracking. To further improve tracking performance, we exploit the combination of hand-crafted features and structured output learning within a particle filter framework to jointly estimate target transformation and scale variation. Experimental results show that the proposed tracking method achieves excellent results in terms of both success rate and precision when compared to several state-of-the-art methods on public datasets. In the future, we hope to further exploit the convolutional feature representation at super-pixel level and use sparse representation to encode target appearance.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Nos. 61403342, 61273286, U1509207, 61325019), and Hubei Key Laboratory of Intelligent Vision Based Monitoring for Hydroelectric Engineering (No. 2014KLA09).

References

[1] Zhang, K.; Liu, Q.; Wu, Y.; Yang, M.-H. Robust visual tracking via convolutional networks without training. *IEEE Transactions on Image Processing* Vol. 25, No. 4, 1779–1792, 2016.

[2] Henriques, J. F.; Caseiro, R.; Martins, P.; Batista, J. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 37, No. 3, 583–596, 2015.

[3] Smeulders, A. W. M.; Chu, D. M.; Cucchiara, R.; Calderara, S.; Dehghan, A.; Shah, M. Visual tracking: An experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 36, No. 7, 1442–1468, 2014.

[4] Babenko, B.; Yang, M.-H.; Belongie, S. Robust object tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 33, No. 8, 1619–1632, 2011.

[5] Kalal, Z.; Mikolajczyk, K.; Matas, J. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 34, No. 7, 1409–1422, 2012.

[6] Grabner, H.; Grabner, M.; Bischof, H. Real-time tracking via on-line boosting. In: Proceedings British Machine Vision Conference, Vol. 1, 47–56, 2006.

[7] Grabner, H.; Leistner, C.; Bischof, H. Semi-supervised on-line boosting for robust tracking. In: *Computer Vision–ECCV 2008*. Forsyth, D.; Torr, P.; Zisserma, A. Eds. Springer Berlin Heidelberg, 234–247, 2008.

[8] Ma, Y.; Chen, W.; Ma, X.; Xu, J.; Huang, X.; Maciejewski, R.; Tung. A. K. H. EasySVM: A

visual analysis approach for open-box support vector machines. *Computational Visual Media* Vol. 3, No. 2, 161–175, 2017.

[9] Hare, S.; Golodetz, S.; Saffari, A.; Vineet, V.; Cheng, M.-M.; Hicks, S. L.; Torr, P. H. Struck: Structured output tracking with kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 38, No. 10, 2096–2109, 2016.

[10] Ren, X.; Malik, J. Tracking as repeated figure/ground segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1–8, 2007.

[11] Zhou, X.; Li, Y.; He, B.; Bai, T. GM-PHD-based multi-target visual tracking using entropy distribution and game theory. *IEEE Transactions on Industrial Informatics* Vol. 10, No. 2, 1064–1076, 2014.

[12] Zhou, X.; Yu, H.; Liu, H.; Li, Y. Tracking multiple video targets with an improved GM-PHD tracker. *Sensors* Vol. 15, No. 12, 30240–30260, 2015.

[13] Mei, X.; Ling, H. Robust visual tracking using $l1$ minimization. In: Proceedings of the IEEE 12th International Conference on Computer Vision, 1436–1443, 2009.

[14] Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint* arXiv:1409.1556, 2014.

[15] Han, B.; Comaniciu, D.; Zhu, Y.; Davis, L. S. Sequential kernel density approximation and its application to real-time visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 30, No. 7, 1186–1197, 2008.

[16] Jepson, A. D.; Fleet, D. J.; El-Maraghi, T. F. Robust online appearance models for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 25, No. 10, 1296–1311, 2003.

[17] Ross, D. A.; Lim, J.; Lin, R.-S; Yang, M.-H. Incremental learning for robust visual tracking. *International Journal of Computer Vision* Vol. 77, Nos. 1–3, 125–141, 2008.

[18] Zhong, W.; Lu, H.; Yang, M.-H. Robust object tracking via sparse collaborative appearance model. *IEEE Transactions on Image Processing* Vol. 23, No. 5, 2356–2368, 2014.

[19] Zhang, K.; Zhang, L.; Yang, M.-H. Real-time compressive tracking. In: *Computer Vision–ECCV 2012*. Fitzgibbon, A.; Lazebnik, S.; Perona, P.; Sato, Y.; Schmid, C. Eds. Springer Berlin Heidelberg, 864–877, 2012.

[20] Kalal, Z.; Matas, J.; Mikolajczyk, K. P-N learning: Bootstrapping binary classifiers by structural constraints. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 49–56, 2010.

[21] Gao, J.; Ling, H.; Hu, W.; Xing, J. Transfer learning based visual tracking with Gaussian processes regression. In: *Computer Vision–ECCV 2014*. Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars. Eds. Springer Cham, 188–203, 2014.

[22] Zhu, Z.; Liang, D.; Zhang, S.; Huang, X.; Li, B.; Hu, S. Traffic-sign detection and classification in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2110–2118, 2016.

[23] Li, H.; Li, Y.; Porikli, F. Robust online visual tracking with a single convolutional neural network. In: *Computer Vision–ACCV 2014*. Cremers, D.; Reid, I.; Saito, H.; Yang, M.-H. Eds. Springer Cham, 194–209, 2014.

[24] Zhou, X.; Xie, L.; Zhang, P.; Zhang, Y. An ensemble of deep neural networks for object tracking. In: Proceedings of the IEEE International Conference on Image Processing, 843–847, 2014.

[25] Fan, J.; Xu, W.; Wu, Y.; Gong, Y. Human tracking using convolutional neural networks. *IEEE Transactions on Neural Networks* Vol. 21, No. 10, 1610–1623, 2010.

[26] Wang, N.; Yeung, D.-Y. Learning a deep compact image representation for visual tracking. In: Proceedings of the Advances in Neural Information Processing Systems, 809–817, 2013.

[27] Wang, L.; Liu, T.; Wang, G.; Chan, K. L.; Yang, Q. Video tracking using learned hierarchical features. *IEEE Transactions on Image Processing* Vol. 24, No. 4, 1424–1435, 2015.

[28] Avidan, S. Support vector tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 26, No. 8, 1064–1072, 2004.

[29] Collins, R. T.; Liu, Y.; Leordeanu, M. Online selection of discriminative tracking features. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 27, No. 10, 1631–1643, 2005.

[30] Yang, F.; Lu, H.; Yang, M.-H. Robust superpixel tracking. *IEEE Transactions on Image Processing* Vol. 23, No. 4, 1639–1651, 2014.

[31] Henriques, J. F.; Caseiro, R.; Martins, P.; Batista, J. Exploiting the circulant structure of tracking-by-detection with kernels. In: *Computer Vision–ECCV 2012*. Fitzgibbon, A.; Lazebnik, S.; Perona, P.; Sato, Y.; Schmid, C. Eds. Springer Berlin Heidelberg, 702–715, 2012.

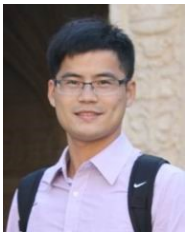
[32] Elad, M.; Figueiredo, M. A. T.; Ma, Y. On the role of sparse and redundant representations in image processing. *Proceedings of the IEEE* Vol. 98, No. 6, 972–982, 2010.

[33] Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1, 886–893, 2005.

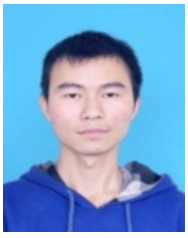
[34] Wu, Y.; Lim, J.; Yang, M.-H. Online object tracking: A benchmark. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2411–2418, 2013.

[35] Zhong, W.; Lu, H.; Yang, M.-H. Robust object tracking via sparsity-based collaborative model. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1838–1845, 2012.

Junwei Li, Ph.D., is with the College of Computer Science and Technology, Zhejiang University of Technology. He is a member of the China Computer Federation. His main research interests include object tracking, machine learning, convolutional neural networks, and object detection.



Xiaolong Zhou, Ph.D. and associate professor, is with the College of Computer Science and Technology, Zhejiang University of Technology. He is a member of the China Computer Federation, IEEE, and ACM. His main research interests are in visual tracking, gaze estimation, and pattern recognition.



Sixian Chan, Ph.D., is with the College of Computer Science and Technology, Zhejiang University of Technology. His main research interests include visual tracking, image processing, pattern recognition, robotics, and image understanding.

Shengyong Chen, Ph.D., professor. He is an IET fellow, an IEEE senior member, and a senior member of the China Computer Federation. His main research interests include computer vision, pattern recognition, and robotics.