IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 24, 2020, accepted April 4, 2020, date of publication April 8, 2020, date of current version April 22, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2986498

Affine Geometrical Region CNN for Object Tracking

YINGHONG XIE[1], **JIE SHEN**[2], **AND CHENGDONG WU**[3]

[1]College of Information Science and Engineering, Shenyang University, Shenyang 110044, China
[2]College of Engineering and Computer Science, University of Michigan, Dearborn, MI 48128, USA
[3]Faculty of Robot Science and Engineering, Northeastern University, Shenyang 110819, China

Corresponding author: Yinghong Xie (xieyinghong@163.com)

This work was supported in part by the National Science Foundation of China under Grant 61503274 and Grant 61703285, in part by the Shenyang Science and Technology Program under Project 18-013-0-15, and in part by the Double Hundred Project of Shenyang City under Grant Z18-5-013.

ABSTRACT The state-of-the-art trackers using deep learning technology have little special strategy to gain the bounding box well when the target suffers drastic geometric deformation. In this paper, we take full use of the convolutional neural network (CNN) features of the deepest layer to represent the semantic feature model, and affine transformation to be as the space information model. A tracking method based on geometrical transformation region CNN is proposed. Firstly, affine transformation is applied to predict possible locations of a target, and the candidate bounding boxes obtained by affine transformation sampling can locate the possible geometric regions of the target more effectively before extracting features from CNN. Furthermore, RoI pooling with different sizes and shapes are designed to describe the geometric deformation region of the target. Then, multi-tasks loss function including the affine transformation regression is designed to refine the affine bounding box. Finally, the affine transformation NMS (Non-maximum suppression) is used to ensure the tracking bounding box having the largest IoU value. Extensive experimental results show that the proposed algorithm performs favorably against the compared methods in the public benchmarks.

INDEX TERMS Object tracking, CNN, affine manifold, affine transformation NMS, geometric deformation.

I. INTRODUCTION

Visual tracking is one of the basic problems of computer vision, which has a wide range of applications, such as video surveillance, autonomous driving and behavior analysis. Given the bounding box of the image target in the first frame, the tracking task is to locate the position of the target correctly in subsequent frames. But the target in the subsequent video frames may suffer complex situations such as deformation, occlusion, illumination change and background change. This makes the tracking problem more difficult.

The key parts for a tracking method are semantic feature modeling and spatial feature modeling, which could be generation based or discrimination based method. The generation based algorithm tracks the target by finding the best matching image region with the template or the appearance model, while the method based on discrimination regards target tracking as a binary classification problem in the local image region, with the purpose of separating the tracking target from the background.

For semantic feature modeling, deep convolution neural network (CNN) has shown its superior performance in various visual tasks [1]–[4]. Nam *et al.* [5], the target appearance is represented by CNNs, and by managing the appearance models in a tree to design the tracking frame. Nam and Han [6] applies a large set of image videos with ground-truths for CNN to compute an object representation. Some other trackers [7], [8] also directly use CNNs as classifiers and take full advantage of end-to-end training. While Ma *et al* [9] extracted features from deep CNNs to gain more accurate tracking results. And papers [10]–[13] also integrate deep features into traditional tracking methods, which benefit from the expression ability of CNN features. As it is verified in [9], the highest convolutional layer has the closest relationship to category-level semantics information while spatial resolution is gradually reduced with the increase of the depth of convolutional layers, we take full use of the CNN features of the deepest layer to design semantic feature model and design a

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Wei.

separate model to represent the spatial feature in the proposed tracking method.

For spatial feature modeling, it is meaningful to distinguish object classification and object estimation as two independent but related subtasks. Object classification is basically to determine the existence of the target in an image location. However, only part of the object status information is obtained, for example, the image coordinates. The object estimation aims to find the full state which is not only including the coordinates but also including axis aligned or rotated [14], [15] and a bounding box [16] of an object. The aim of tracking is to compute and gain the bounding box that fits the object best. For a rigid object with simple movement, the object classification algorithm is enough for tracking task. But in most real tracking applications, the object may suffer drastic deformation in pose and viewpoint, which brings more difficulty for predicting the bounding box.

In the past few years, target classification methods are mainly realized by discriminatively on-line training classifier [6], [17], [18]. Correlation filters [19] also have been developed. For example, paper [20] designs a minimum output sum of squared error filter for the target appearance for fast visual tracking. And some other correlation trackers also be designed for visual tracking, including context learning [21], kernelized correlation filters [22], scale estimation [23], multiple dimensional features [24], [18], re-detection [25], and spatial regularization [17] etc. These method using the reliable confidence scores to determine the target classification, but no strategy is considered to improve target estimation accuracy [26], [27]. The full state estimation of a target is completed only depending on the classifier.

However, the target classifier is not sensitive to some aspects of the object state, for example. The width and height of the target. Accurate bounding box estimation is a complicated question in object tracking field, which often needs advanced prior knowledge. Many recent methods have therefore integrated prior knowledge in the form of offline learning [6], [28], [29]. In particular, paper [28]–[30] have been shown capable of bounding box regression. Yet, the shape of the bounding box is always a rectangle that can't adapt to the geometric deformation of the target. Many classic tracking methods [31] use affine transformation to describe the geometric deformation of the target. Considering that affine transformation manifold can describe the deformation of the target more accurately when the target experiences drastic geometric deformation, we apply affine transformation manifold to design the separate spatial feature model for geometric deformation of the target.

In this paper, CNN is combined with affine transformation manifold for the tracker design. Inspired by RPN and considering the bounding box for consecutive frames being closely related, instead of RPN, we use affine transformation bounding box of the former frame to predict the possible target bounding boxes. For every frame, we compute the affine transformation samples according to the affine transformation vectors of the former frame, the samples are input to CNN network to abstract feature maps. Moreover, a multi-tasks loss function is designed to regress the affine transformation parameters. Furthermore, different RoI pooling sizes and affine transformation NMS are adopted in our tracker. As shown in Figure 1, when the target has significant geometric transformation, our tracker still has high performance.

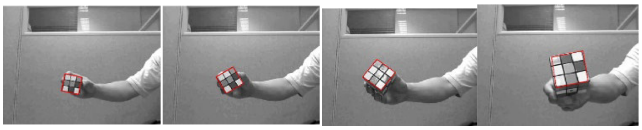

FIGURE 1. Example tracking bounding boxes of the proposed tracker, when the object suffers drastic geometric deformation.

The main contribution of the proposed tracker are as follows.

(1) Affine transformation is applied to predict possible locations and geometric deformation of a target, which makes the tracked bounding box more accurately.
(2) Compared with region proposal networks (RPN), the candidate boundary box obtained by affine transformation sampling can locate to the effective range of the target more accurately before extracting CNN features.
(3) Features from the deepest CNN layers being semantic information model combines with affine manifold being spatial information model, which forms complementary advantages.
(4) Multi-tasks loss function including the affine transformation regression is designed to optimize CNN networks performance.
(5) Different Region of Interest (RoI) pooling sizes are adopted to assist in describing the deformation of the target.
(6) Affine transformation NMS is applied to ensure the tracking bounding box having the largest IoU value.

The rest of this paper is organized as follows: In section2 we build affine manifold and the geometric transformation model. In section 3 we propose the geometrical region CNN tracking approach. Then, we design the implementation details and evaluate the experimental effectiveness by comparing with other state-of-the-art trackers in section 4. Finally, conclusions are drawn in section 5.

II. AFFINE MANIFOLD

A. AFFINE MANIFOLD AND ITS METRIC

In this paper, we apply affine transformation manifold to represent the object full state estimation. Let $I(X)$ represent the gray value of the template image position $X = (x, y)'$. The Cartesian coordinate system is established with the center of the target as the coordinate origin. The gray value of the target in the input image after affine transformation is $I(W(x : r))$, where $W(x : r)$ represents the affine transformation of the object in the input image with respect to the template,

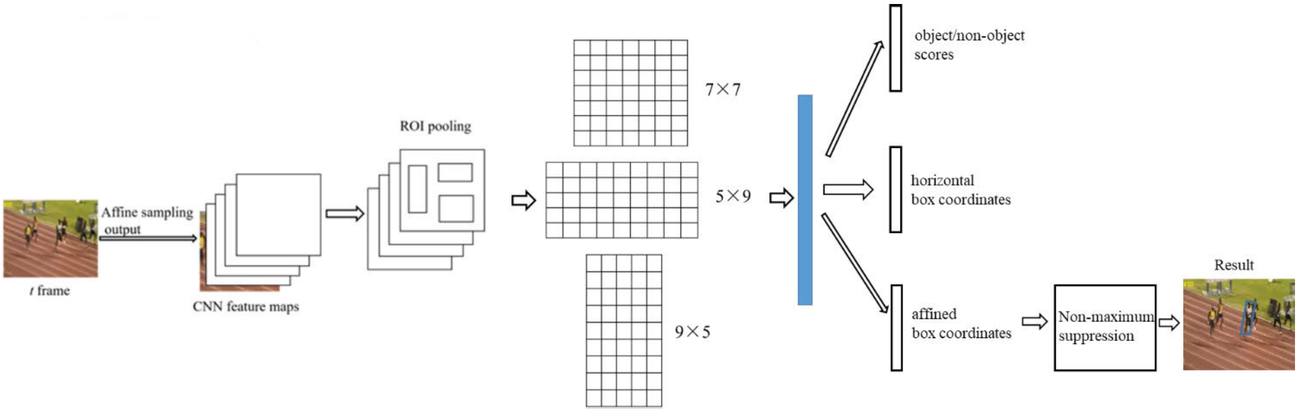

FIGURE 2. Schematic overview of the proposed framework based on Affine Transformation and Convolutional Features. It consists of the following six stages: (1) affine sampling (2) CNN feature extraction (3) RoI pooling with kernels of different sizes (4) multi-tasks loss (5) Non-maximum suppression (6) tracking result.

$r = (r_1, r_2, r_3, r_4, r_5, r_6)'$ is a parameter vector.

$$W(x:r) = \begin{bmatrix} r_1x + r_2y + r_5 \\ r_3x + r_4y + r_6 \end{bmatrix} \tag{1}$$

The transformation matrix is represented with homogeneous coordinates as:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_5 \\ r_3 & r_4 & r_6 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{2}$$

The affine transformation matrix

$$T(r) = \begin{bmatrix} r_1 & r_2 & r_5 \\ r_3 & r_4 & r_6 \\ 0 & 0 & 1 \end{bmatrix}$$

has the structure of Lie group $GA(2)$, and $ga(2)$ is Lie algebra corresponding to affine Lie group $GA(2)$. And matrix $G_i(\forall i \in \{1, 2, \cdots 6\})$ is the generators of $GA(2)$ and the basis of matrix $ga(2)$. For matrix $GA(2)$, the generators are

$$G_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad G_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad G_3 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$
$$G_4 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad G_5 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad G_6 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}. \tag{3}$$

For Lie group matrix, Riemann distance is defined by matrix logarithmic operation:

$$d'(X, Y) = \left\| \log(YX^{-1}) \right\|. \tag{4}$$

where X and Y are the elements of Lie group matrix. Given N symmetric positive definite matrices $\{X_i\}_{i=1}^{N}$,the intrinsic mean is defined as

$$\boldsymbol{\mu}^* = \exp(\frac{1}{N}\sum_{i=1}^{N}\log(\boldsymbol{X_i})). \tag{5}$$

For more knowledge of the Lie group, please refer to the reference [32].

B. DESIGN THE GEOMETRIC TRANSFORMATION MODEL

In this paper, affine transformation is applied for designing the special information model. It can reflect the geometric transformation properly. And the geometrical change between two adjacent frames is equivalent to the movement of corresponding points of affine matrices on Riemann manifold, because affine transformation matrix is a positive definite symmetric manifold, which belongs to Lie group structure and doesn't obey Euclidean space. The basic idea for establishing the model of the target deformation is to find the transformation relationship between two adjacent points on the manifold. The tangent vector of the point on the manifold can be used to describe this relationship. The affine transformation model is designed in Riemannian manifold and its tangent space, respectively,

$$S_t = S_{t-1}\exp(v_t) \tag{6}$$
$$v_t = av_{t-1} + \mu_{t-1}, \tag{7}$$

where the vector $S_t = [x_1, x_2, x_3, x_4, x_5, x_6]^{\mathrm{T}}$ is the affine transformation parameter of the target geometric deformation. v_t represents as the velocity vector from point S_{t-1} to point S_t on the tangent space, and it describes the movement of the target, which is the tangent vector from point S_t on manifold. Suppose v_t obeys the Gauss distribution, $\mu_{1:t}$ is Gauss white noise, and a is autoregressive coefficient.

III. THE PROPOSED AFFINE GEOMETRICAL REGION CNN APPROACH

The main strategy for the current effective deep learning algorithm for classification and recognition is two-stage object detection strategy. They use region proposal network (RPN) and region classification network. In the proposed algorithm, we also adopt a two-stage strategy, but we use affine transformation region to replace RPN region. Based on Faster R-CNN [2] and R2CNN [3], we design our tracking framework. As shown in figure 2. According to the results of t-1 frame, M transformation vector samples are computed, which is instead of the function of RPN. Then reshape the image region corresponding to the samples to rectangle regions with the same size. For each sample, deep learning network is used to compute feature maps. Then, three different RoI poolings are built to make the most of object appearance characteristics. And multi-task Loss function is designed to optimize CNN networks. Finally, affine transformation NMS is applied to ensure the tracking bounding box having the largest IoU value.

A. AFFINE MANIFOLD SAMPLING

For image tracking, there is a close relationship between two consecutive images. Therefore, we preliminarily determine the approximate target positions of the current frame according to the bounding box tracked to the previous image frame. By using the principle of affine transformation, the possible positions are sampled by affine transformation sampling. Compared with RPN, the candidate bounding box obtained by affine transformation sampling can more accurately sample the effective region of the target.

For the tracking bounding box of frame t, the initial value is set as the same with the tracked bounding box of frame t-1, which is $S_{t-1} = [r_1, r_2, r_3, r_4, r_5, r_6]$. Affine transformation sampling is to compute M samples of S_{t-1}. For each sample, it corresponds to a bounding box. Random walk model were applied to sample v_t in equation (7). The steps are as follows:

(1) Initialize k = 1, while k <= M, randomly generate a 6-dimensional vector between(−1, 1) as $\mu = (\mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6)(-1 < \mu_{\mathrm{i}} < 1, i = 1, 2, \ldots\ldots, 6)$.

(2) Standardize u as

$$\mu' = \mu / \sqrt{\sum_{i=1}^{6} \mu_i^2}. \tag{8}$$

(3) Compute $v_t^k = av_{t-1} + \mu'$, where v_{t-1} is the velocity vector from frame t-1.

(4) Let $k = k + 1$, Repeat the above steps, until $k =$ M. $v_t = [v_t^1, v_t^2, \ldots, v_t^M]$.

(5) According to $v_t = [v_t^1, v_t^2, \ldots, v_t^M]$, the M samples of affine transformation parameters are generated as $\{S_{\mathrm{t}}^1, S_{\mathrm{t}}^2, \ldots\ldots, S_{\mathrm{t}}^M\}$ according to equation (6).

After all the affine samples are obtained, we draw each candidate region on the current image frame. Then, the size of each candidate target region is normalized, and the normalized results are input into the trained deep learning network to obtain the feature maps.

B. RoI POOLINGS WITH DIFFERENT SIZES

The RoI pooling layer uses maximum pooling to transform the features in any effective region of interest into a small feature map with a fixed spatial extent of $H \times W$. Where H and W are layer height and width parameters independent of any specific RoI.

According to the different tracking targets, different shapes of the RoI pooling may be designed aiming to capture more feature information. For most deformable targets, the deformation of horizontal and vertical directions is not particularly severe in the adjacent two frames. So three RoI pools with different sizes (7 × 7, 5 × 9, 9 × 5) are executed on the convolution feature maps, and the pooled features are concatenated for further classification and regression. With concatenated features and fully connected layers, we predict object/non-object scores, the horizontal rectangle bounding box and the affine transformation bounding box. After that, the affine transformation parameters are post-processed by affine transformation non-maximum suppression to get the best tracking results.

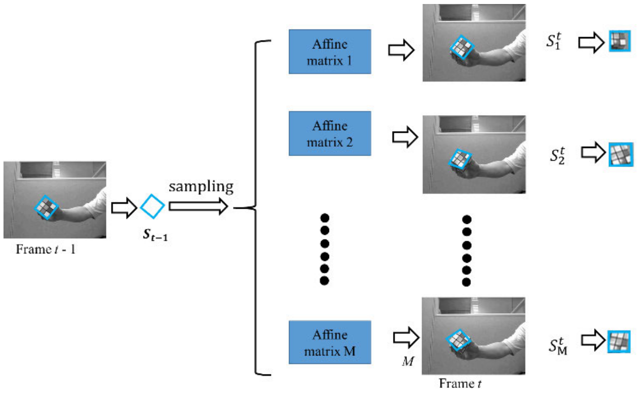

FIGURE 3. The flowchart for affine transformation sampling. Firstly, generating affine matrix samples according to the affine matrix of frame t-1. Then, drawing the bounding box on image of frame t. Finally, computing the rectangular area before input into CNN network.

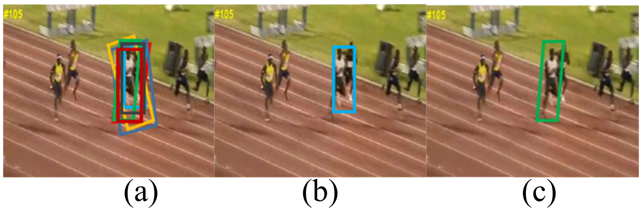

FIGURE 4. Illustrates the detection results after horizontal rectangle NMS and affine transformation NMS are performed. (a) The candidate horizontal rectangle boxes and affine transformation boxes; (b) the detection result based on horizontal rectangle NMS on horizontal rectangle bounding boxes;(c)the detection results based on affine transformation NMS on affine transformation bounding boxes.

C. AFFINE TRANSFORMATION NON-MAXIMUM SUPPRESSION

In the current object detection methods, Non-Maximum suppression (NMS) is widely used in post-processing detection candidates. While estimating both the horizontal rectangle bounding box and the affine transformation bounding box, we can perform normal NMS on the horizontal rectangle bounding box or affine transformation NMS on the affine transformation bounding box. In affine transformation NMS, the calculation of traditional intersection (IoU) is modified to IoU between two affine transformation bounding boxes. The IoU calculation method used in [33] is used.

D. MULTI-TASKS LOSS

The loss function we define on each proposal is the sum of appearance classification loss and box regression loss. Box regression loss consists of two parts: loss of horizontal rectangle bounding box and loss of affine transformation bounding box. The multi-tasks loss function for each proposal is defined as:

$$\begin{aligned} &\mathrm{L}\left(p, t, v_i, v_i^*, u_i, u_i^*\right) \\ &= L_c\left(p, t\right) + \lambda_1 t \sum\nolimits_{i\in(x,y,w,h)} L_{reg}\left(v_i, v_i^*\right) \\ &\quad + \lambda_2 t \sum\nolimits_{i\in(r1,r,2,r3,r4,r5,r6)} L_{reg}\left(u_i, u_i^*\right), \end{aligned} \tag{9}$$

where λ_1 and λ_2 are the balancing parameters that control the trade-off between three terms. And the parameter p = (p0, p1) is the probability over target and background classes calculated by the softmax function. The logarithm loss of true class t is

$$L_c\left(p, t\right) = -logp_t. \tag{10}$$

And the parameter $v = (v_x, v_y, v_w, v_h)$ is a tuple of true rectangle bounding box regression targets including the coordinates of the center point and its width and height, $v^* = \left(v_x^*, v_y^*, v_w^*, v_h^*\right)$ is the predicted tuple for the targets.

$u = (u_{r1}, u_{r2}, u_{r3}, u_{r4}, u_{r5}, u_{r6})$ the predicted tuple for the affine transformation bounding box, while $u^* = (u_{r1}^*, u_{r2}^*, u_{r3}^*, u_{r4}^*, u_{r5}^*, u_{r6}^*)$ is the predicted tuple for the affine transformation bounding box.

Let $(w, w*)$ denotes (v_i, v_i^*) or (u_i, u_i^*), $L_{reg}\left(w, w*\right)$ is defined as:

$$L_{reg}\left(w, w*\right) = smooth_{L1}(w - w*), \tag{11}$$

$$smooth_{L1}\left(w - w*\right) = \begin{cases} 0.5x^2 & if \ |x| < 1 \\ |x| - 0.5 & else. \end{cases} \tag{12}$$

E. OBJECT TRACKING ALGORITHM

On the observation that the last layer of CNNs encodes the semantic abstraction of objects and their outputs are robust to appearance variations, the proposed method uses the highest layer information to predict the target appearance. Meanwhile, we apply affine transformation to predict possible locations of a target instead of RPN. In order to catch more features of the target, 3 different sizes of RoI poolings are made on the output of CNN features.

The main steps for the proposed tracking algorithm based on affine transformation and CNN feature map is as follows.

Initialization: when t = 1, initialize the affine transformation parameter $S_1 = [r_1, r_2, r_3, r_4, r_5, r_6]$.

Step 1: draw the bounding box for t-th image frame, according to the target position of the t-1 frame. That's the initial bounding box in t frame has the same position and the same shape with that is in t-1 frame.

Step 2: According to formulas (6)-(8), M candidate affine transformation vectors are generated.

Step 3: The candidate image regions determined by M affine parameters are transformed into rectangular regions, and input into deep convolution neural networks.

Step 4: The RoI pooling kernels of three different sizes are designed in consideration of the deformation of the target. Then on the output feature maps of the convolution neural network, the RoI (Region of Interest) pooling operation is performed.

Step 5: With concatenated features and fully connected layers, we predict object appearance scores.

Step 6: After Non-maximum suppression is computed, the tracking result for frame t is obtained.

Step 7: t = t + 1, if t is smaller than the total frame of the video to track, return to step one.

Step 8: it ends until all the frames have been tracked.

IV. EXPERIMENTS

A. IMPLEMENTATION DETAILS

1) CNN FRAMEWORK

The proposed method uses Faster RCNN network as the base network, which applies 13 convolutional layers, 13 Relu non-linear function of the hidden layers, and 4 max pooling layers to generate feature maps before different size RoI pooling.

2) TRAINING

In our implementation, the target image has a dimension of 127*127*3. And the model is trained offline on the video dataset ImageNet [34]. The training consists of more than 50 epochs, each consisting of 50000 sampling pairs. The gradients of each iteration are estimated by 10 mini-batches size, and the learning rate is from 10^{-2} to 10^{-5} at each period from.

The proposed tracker is implemented in TensorFlow 2.0 framework on the computer with a single NVIDA GTX 1080 and an Intel Core i7 at 4.0 GHz CPU and 256GB memory. Furthermore, the parameters of each compared methods is set in accordance with the original of the respective method.

B. DATASETS AND EVALUATION METRICS

1) THE OTB BENCHMARKS

The benchmarks OTB-2013 [35] and OTB-2015 [15] are most widely used in visual tracking, which contains 50 and 100 image sequences with various challenging factors respectively. They are divided into eleven attributes, such as illumination, deformation and scale change.

The metrics standards on the OTB benchmark include two aspects: the average per-frame success rate and precision. On the one hand, if the intersection-over-union (IoU) between its estimation and the truth is beyond a certain threshold, the tracker is successful in a given frame. It includes success of spatial robustness evaluation (SRE), success of temporal robustness evaluation (TRE) and success of one-pass evaluation (OPE). Normally, the area-under-curve (AUC) of the

TABLE 1. Results of the proposed method under different settings on some image sequences in VOT2015.

serial number of settings	Horizontal rectangle box (λ_1)	Affine transformation box (λ_2)	Pooling size(s)	Affine transformation NMS	Precision	Time
1(baseline)	$\lambda_1 = 1$	$\lambda_2 = 0$	7×7	No	54.78%	1.98 s
2	$\lambda_1 = 0$	$\lambda_2 = 1$	7×7	No	60.25%	1.98s
3	$\lambda_1 = 1$	$\lambda_2 = 1$	7×7	No	65.76%	2.25s
4	$\lambda_1 = 1$	$\lambda_2 = 1$	7×7	Yes	74.28%	2.25s
5	$\lambda_1 = 1$	$\lambda_2 = 1$	7×7 5×9 9×5	No	80.12%	2.28s
6	$\lambda_1 = 1$	$\lambda_2 = 1$	7×7 5×9 9×5	Yes	91.30%	2.28s

TABLE 2. Comparison of state-of-the-art trackers on OTB-2013 and OTB-2015 benchmarks.

		Ours	SRDCFdecon	Staple	DSST	SCM	Struck	TLD
OTB-2013	SUCCESS	0.678	0.653	0.600	0.554	0.499	0.474	0.437
	PRECISION	0.943	0.870	0.793	0.739	0.649	0.656	0.608
OTB-2015	SUCCESS	0.644	0.627	0.581	0.524	0.488	0.457	0.419
	PRECISION	0.896	0.825	0.784	0.712	0.598	0.613	0.572

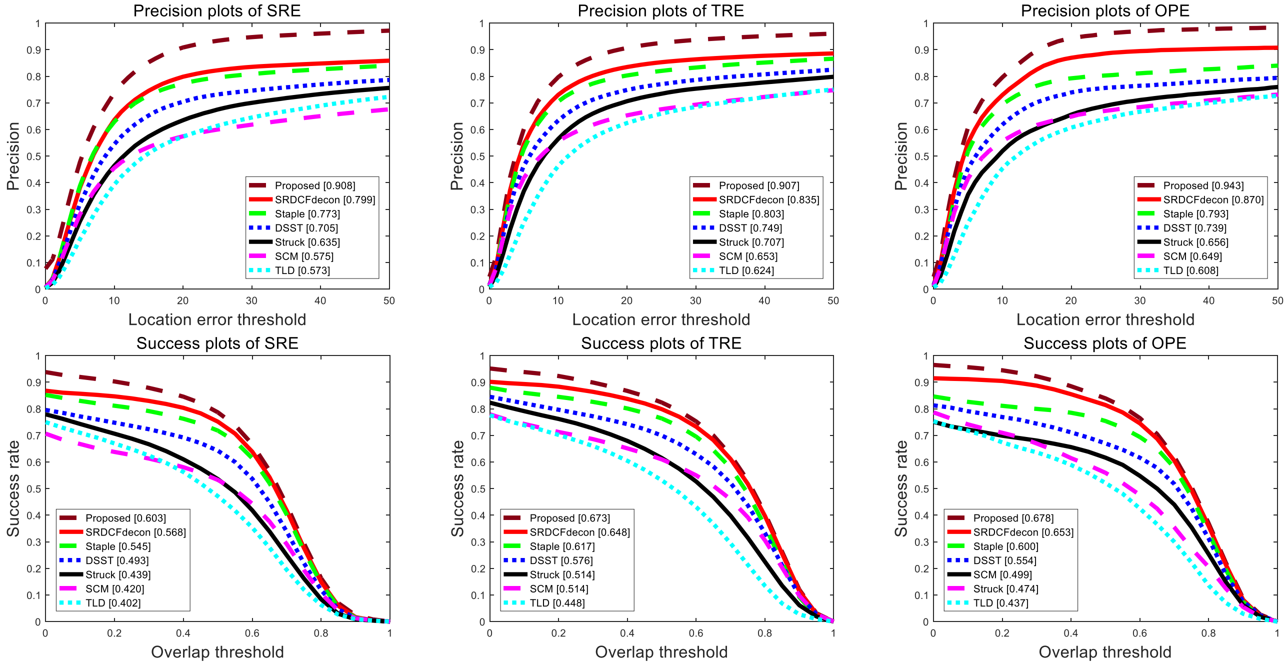

FIGURE 5. Bounding box overlap success rate plots and the center location error precision plots under SRE, OPE and TRE over benchmark sequences. The overlap success contains AUC score for each tracker, and the distance precision contains threshold scores at 20 pixels.

success plot is reported. On the other hand, the precision plot can be gained in the same way. In most papers reports, the threshold for the precision plot is set to 20.

2) THE VOT BENCHMARKS

For our experiments, we use the latest versions of the Visual Object Tracking (VOT) toolkits. They are VOT2015 [36], VOT2016 [37] and VOT2017 [38]. VOT2015 and VOT2016 contain the same sequences, while the ground truth labels in VOT2016 are more accurate than those in VOT2015. The 60 image sequences contained in VOT2016 including illumination change, camera motion, motion change, occlusion and scale change. And ten sequences are different between the former versions and VOT2017.

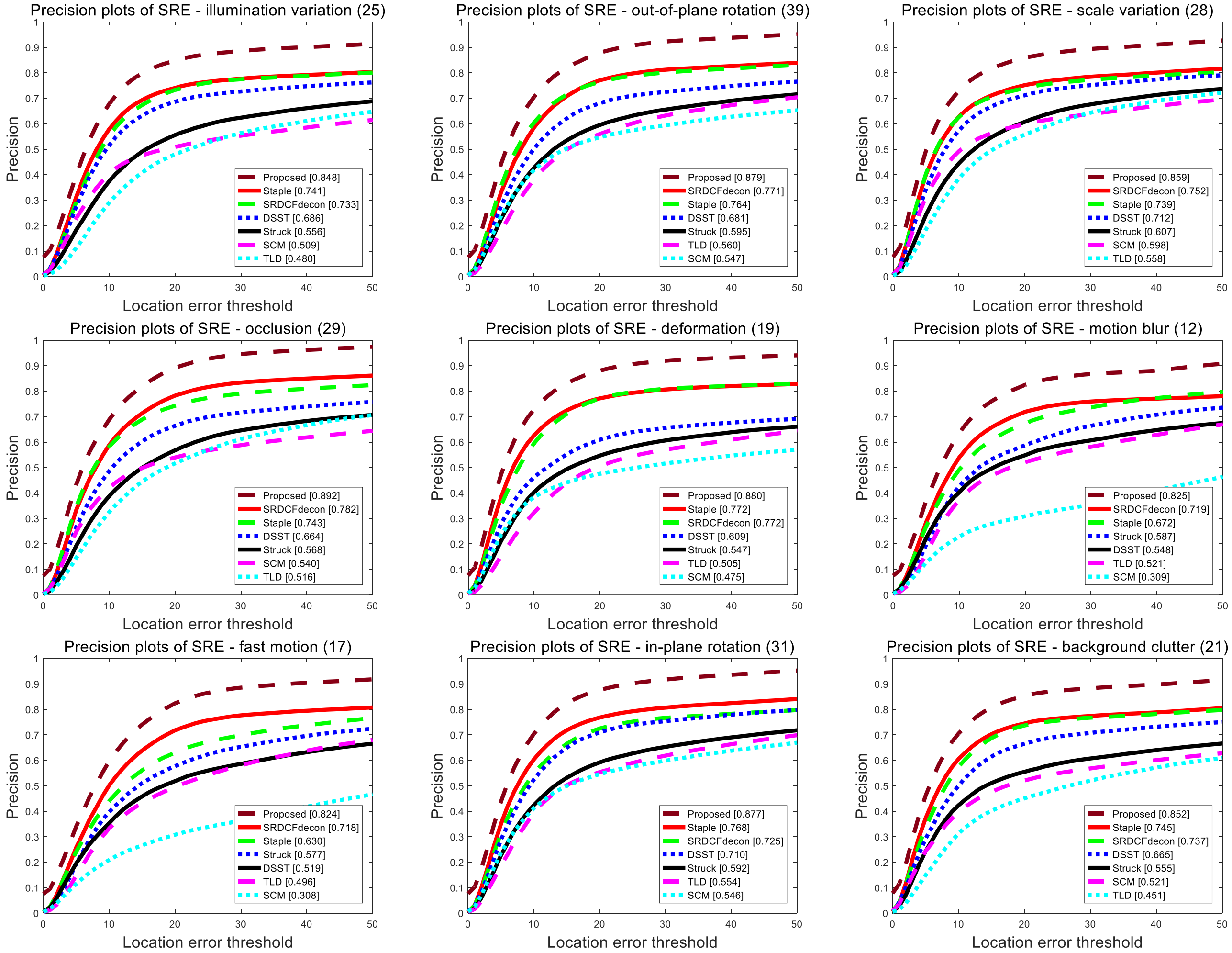

FIGURE 6. The center location error precision plots under SRE, TRE and OPE over sequences with scale variation and in-plane rotation respectively, and the distance precision contains threshold scores at 20 pixels.

The evaluation on the VOT benchmark is based on the re-initialized methodology that a tracker will be reset after five frames of no overlap with the ground truth. The evaluation focuses on the short-term effectiveness, and the metrics standards on the VOT benchmark include accuracy (A), robustness (R) and expected average overlap (EAO). The higher A and EAO scores and the lower R scores, the better the tracker's performance is. For the tracking speed evaluation, to reduce the influence of the hardware, the VOT2014 [39] proposes a new unit called equivalent filter operations (EFO) that reports the tracker speed in terms of a predefined filtering operation that the toolkit automatically carries out prior to running the experiments. The same tracking speed evaluation is used in VOT2016 [37]. At the same time, the value of raw frames per second (FPS) is given in the VOT benchmark reports, which is the speed at which the algorithm runs on an individual computer.

C. ABLATION ANALYSIS

In this section, the function of different parts in the proposed method is analyzed. The Results under different settings on some image sequences in VOT2015 are shown in table 1.

1) HORIZONTAL RECTANGLE BOX AND AFFINE TRANSFORMATION BOX

In the proposed method, there are three kinds of regress function according to the different values of λ_1 and λ_2 in formula (9).

The first setting is $\lambda_1 = 1$ and $\lambda_2 = 0$, the detection outputs are horizontal rectangle box. The second setting is $\lambda_1 = 0$ and $\lambda_2 = 1$, it only regresses affine transformation box, which leads to about 6% performance improvement over the first setting. The reason is that the outputs of the first setting are only the horizontal rectangle boxes, however, the geometrical transformation information is ignored. The third setting is $\lambda_1 = 1$ and $\lambda_2 = 1$, the detection outputs are the both, which leads to another 5% performance improvement over the second settings. This means that learning the additional horizontal rectangle box could help to detect the affine transformation box.

2) SINGLE POOLING SIZE VS. MULTIPLE POOLING SIZES

The third and fifth settings are used to analyze the function of multi-scale pooling. As shown in Table 1, with three pooled sizes (7 × 7, 5 × 9, 5 × 91), the performance of the fifth

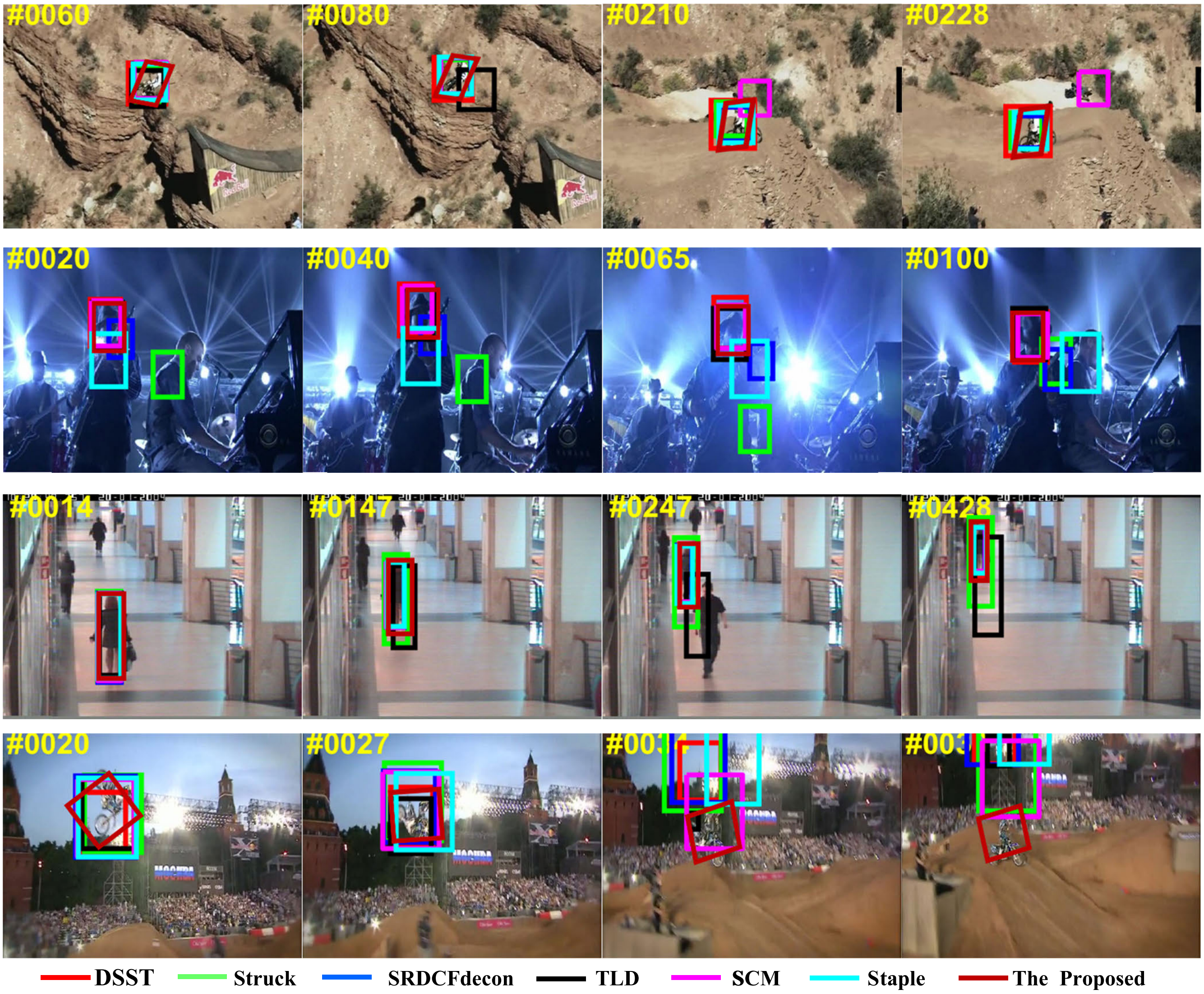

FIGURE 7. Bounding box results for the proposed algorithm and the compared algorithms.

setting is much better than that of the third setting. This confirms that utilizing more features is helpful for transformation target detection.

3) NORMAL NMS ON HORIZONTAL RECTANGLE BOXES VS. AFFINE TRANSFORMATION NMS ON AFFINE BOXES

By comparison between the third setting and the fourth setting, and the comparison between the fifth setting and the sixth setting, we analyze the function of the affine transformation NMS. Because we regress both the horizontal rectangle box and the affine transformation box, each horizontal rectangle box is associated with an affine transformation box. We can conclude that affine transformation NMS under both single pooling size test and multi-scale pooling size test consistently perform better than their counterparts.

4) TEST TIME

The running environment is consistent with the training environment described earlier. Under single-scale pooling size test, our method only increases little affine transformation sampling time compared to the baseline.

D. COMPARISON WITH STATE-OF-THE-ARTS

The widely used benchmarks for object tracking are OTB benchmarks [34], [35], and VOT benchmarks [36]–[38]. Our tracker is evaluated with state-of-the-art trackers on the benchmark datasets OTB-2013 [34], OTB2015 [35], and VOT2015 [36], VOT2016 [37] respectively.

1) OTB BENCHMARKS

We compare our tracker with the 29 default trackers in OTB-2013 and OTB-2015 benchmarks and 6 more popular trackers

TABLE 3. Performance comparison of state-of-the-art trackers on VOT-2016 benchmarks.

	Ours	C-COT	SSAT	MLDF	Staple	DDC	EBT
EAO	0.464	0.331	0.321	0.311	0.295	0.293	0.291
R	0.147	0.238	0.291	0.233	0.378	0.345	0.252
A	0.582	0.539	0.577	0.490	0.544	0.541	0.465
Raw FPS	38.3865	82.1821	0.8041	2.1997	14.4329	0.1649	2.8697
Normalized (EFO)	31.7303	50.9975	0.4629	1.4007	10.9754	0.1928	2.9669

including SRDCFdecon [40], Staple [41], DSST [23], SCM [42],Struck [43],and TLD [44].

The tracking results of all the algorithms to be compared are obtained through the files published by the authors. All the success plots and precision plots on OPE evaluation are summarized in table 2. For more results on bounding box overlap success rate plots and the center location error precision plots under SRE, OPE and TRE over benchmark sequences are illustrated in Figure 6. Figure 7 shows the tracking bounding boxes results on some challenging video sequences. The main challenges of sequence 1 is geometric transformation. For sequence 2, the object suffers illumination variation and background clutter. For sequence 3, the object experiences temporary occlusion. And the main challenge of sequence 4 are drastic geometric transformation, motion blur and fast motion. All the tracking results suggest that our affine transformation strategy can bound the target more accurately than the rectangle boxes, when the target suffers from large geometric deformation.

2) VOT BENCHMARKS

VOT benchmarks include VOT2015 [36], VOT2016 [37] and VOT2017 [38]. They are also the most popular object tracking benchmarks. In our experiments, we choose VOT2016 [37] to evaluate the tracking performance. The proposed method is compared with 6 state-of-the-art trackers including C-COT [12], SSAT [37], MLDF [37], Staple [41], DDC [37], EBT [45]. The performance comparison of these trackers are shown in Table 3. The EAO value of our tracker is 0.464, which out-performs all the compared trackers. Figure 8 shows the EAO curve of all the compared trackers, which also verify the high performance of our tracker. And the speed report for experiment baseline is also shown in Table 3. Except C-COT tracker whose accuracy is worse than our tracker, our tracker is faster than all the other trackers. The reason is that we further improve the implementation method of the second step (CNN feature extraction) in Figure 2. The algorithm first extracts feature maps on the whole image, and then locates M feature maps corresponding to M affine sampling regions, so that the feature maps are extracted only once for M affine sampling regions. This greatly improves the speed efficiency of the algorithm.

Moreover, we choose other more trackers given on the VOT2016 benchmark, which are compared with our tracker on the Accuracy-robustness plot for experiment baseline (mean), the results are shown in Figure 9, and best trackers are closer to the top right corner, which also demonstrates the effectiveness of our tracker.

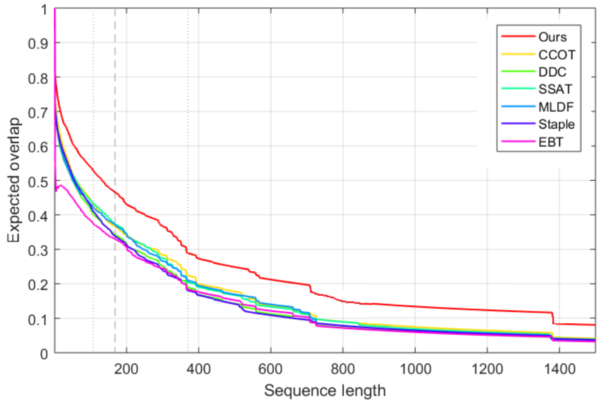

FIGURE 8. Expected Average Overlap(EAO) curve for 6 state-of-the-art trackers on the VOT-2016 dataset. Our tracker has much better performance than the compared trackers.

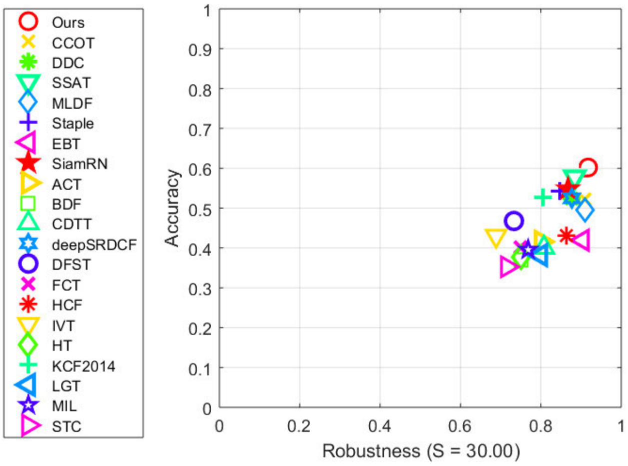

FIGURE 9. Accuracy-robustness plot. Best trackers are closer to the top right corner.

V. CONCLUDING REMARKS

The key to a tracking algorithm is its semantic and spatial models. We take full use of the CNN features of the deepest layer to design semantic feature model, and apply affine transformation as the space information model. Combing CNN with affine transformation manifold, we have proposed a geometrical region CNN based method for object tracking. Affine transformation is applied to predict possible locations and geometric deformation of a target. And the candidate bounding box obtained by affine transform sampling can more accurately locate to the effective region of the target before extracting CNN features. Furthermore, different RoI pooling sizes can assist in describing the deformation of the target. Then, multi-tasks loss function including the affine transformation regression is designed to optimize CNN networks performance. Finally, the affine transformation NMS is applied to ensure the tracking bounding box having the largest IoU value. All the analysis results show an outstanding performance of the proposed trackers.

REFERENCES

[1] R. Girshick, "Fast R-CNN," in *Proc. ICCV*, 2015, pp. 1440–1448.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. NIPS*, vol. 1, 2015, pp. 91–99.

[3] Y. Jiang, X. Zhu, X. Wang, S. Yang, W. Li, H. Wang, P. Fu, and Z. Luo, "R2CNN: Rotational region CNN for orientation robust scene text detection," in *Proc. Comput. Vis. Pattern Recognit.*, 2017, pp. 1–8.

[4] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. ICCV*, 2017, pp. 2961–2969.

[5] H. Nam, M. Baek, and B. Han, "Modeling and propagating CNNs in a tree structure for visual tracking," 2016, *arXiv:1608.07242*. [Online]. Available: http://arxiv.org/abs/1608.07242

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

[7] B. Han, J. Sim, and H. Adam, "BranchOut: Regularization for online ensemble tracking with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, vol. 2, no. 3, pp. 3356–3365.

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

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

[10] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Dec. 2015, vol. 2, no. 7, pp. 58–66.

[11] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M.-H. Yang, "Hedged deep tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4303–4311.

[12] M. Danelljan, A. Robinson, F. Shahbaz Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 472–488.

[13] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jul. 2017, pp. 6638–6646.

[14] H. K. Galoogahi, A. Fagg, C. Huang, D. Ramanan, and S. Lucey, "Need for speed: A benchmark for higher frame rate object tracking," in *Proc. ICCV*, vol. 2, 2017, pp. 1125–1134.

[15] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.

[16] M. Kristan *et al.*, "The sixth visual object tracking vot2018 challenge results," in *Proc. ECCV Workshop*, vol. 1, 2018, pp. 3–53.

[17] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, vol. 1, no. 2, pp. 4310–4318.

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

[19] C. Sun, D. Wang, H. Lu, and M.-H. Yang, "Correlation tracking via joint discrimination and reliability learning," in *Proc. CVPR*, 2018, pp. 489–497.

[20] D. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2544–2550.

[21] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang, "Fast visual tracking via dense spatio-temporal context learning," in *Proc. ECCV Target Attending Tracking CVPR*, 2016, pp. 127–141.

[22] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of Tracking-by-Detection with kernels," in *Proc. ECCV*, 2012, pp. 702–715.

[23] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "'Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 1–11.

[24] M. Danelljan, F. S. Khan, M. Felsberg, and J. V. D. Weijer, "Adaptive color attributes for real-time visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1090–1097.

[25] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Long-term correlation tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5388–5396.

[26] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Discriminative scale space tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1561–1575, Aug. 2017.

[27] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proc. ECCV Workshop*, vol. 2, 2014, pp. 254–265.

[28] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8971–8980.

[29] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "Distractor-aware siamese networks for visual object tracking," in *Proc. ECCV*, 2018, pp. 103–119.

[30] A. He, C. Luo, X. Tian, and W. Zeng, "A twofold siamese network for real-time object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4834–4943.

[31] Z. H. Khan and I. Y.-H. Gu, "Tracking visual and infrared objects using joint Riemannian manifold appearance and affine shape modeling," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, Nov. 2011, pp. 1847–1854.

[32] B. C. Hall, *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*. Springer, 2003.

[33] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue, "Arbitrary-oriented scene text detection via rotation proposals," 2017, *arXiv:1703.01086*. [Online]. Available: http://arxiv.org/abs/1703.01086

[34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, vol. 3, no. 5.

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

[36] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernandez, T. Vojir, G. Hager, G. Nebehay, R. Pflugfelder, A. Gupta, A. Bibi, A. Lukezic, A. Garcia-Martin, A. Saffari, A. Petrosino, and A. S. Montero, "The visual object tracking VOT2015 challenge results," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, vol. 6, Dec. 2015, pp. 564–586.

[37] M. Kristan *et al.*, "The visual object tracking VOT2016 challenge results," in *Proc. Eur. Conf. Comput. Vis. Workshop*, vol. 6, 2016, pp. 777–823.

[38] M. Kristan *et al.*, "The visual object tracking VOT2017 challenge results," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, vol. 6, Oct. 2017, pp. 1949–1972.

[39] M. Kristan *et al.*, "The visual object tracking VOT2014 challenge results," in *Proc. Workshop Visual Object Tracking Challenge (ECCV Workshops)*, 2014, pp. 191–217.

[40] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016.

[41] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016.

[42] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparse collaborative appearance model," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2356–2368, May 2014.

[43] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *Proc. ICCV*, Nov. 2011, pp. 263–270.

[44] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, Jul. 2012.

[45] G. Zhu, F. Porikli, and H. Li, "Beyond local search: Tracking objects everywhere with instance-specific proposals," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 943–951.

YINGHONG XIE received the Ph.D. degree in pattern recognition and artificial intelligence from Northeastern University, China, in 2014. Since 2005, she has been with Shenyang University, and an Associate Professor with the Information and Engineering Institute. From 2014 to 2016, she held a postdoctoral position at Tianjin University. She was a Scholar with the University of Michigan, Dearborn, in 2017. She is the first author of more than 20 articles. She hosted the Natural Science Foundation of China, in 2015. Her main research interests include artificial intelligence, video image processing, and pattern recognition.

JIE SHEN served as an editorial board member for two international journals, an organizer for eight international conferences, an associate editor of two international conference proceedings, a program committee member for 20 international conferences, the session chair for the 13 international or national conferences, a board member for three international- or national-level technical committees, and a member for various committees at department and campus levels within the University of Michigan, Dearborn. He is currently the Editor-in-Chief of the textitInternational Journal of Modelling and Simulation, which is an EI-indexed, peer-reviewed research journal in the field of modeling and simulation.

Dr. Shen received the author's awards and honors, include the Frew Fellowship (Australian Academy of Science), the I. I. Rabi Prize (APS), the European Frequency and Time Forum Award, the Carl Zeiss Research Award, the William F. Meggers Award, and the Adolph Lomb Medal (OSA).

CHENGDONG WU is currently the Vice President of the Faculty of Robot Science and Engineering, Northeastern University, Shenyang, China, where he is also the Director of the Institute Artificial Intelligence, a Professor, and a Ph.D. Tutor. He is an Expert of the Chinese Modern Artificial Intelligence and Robot Navigation. He is also a Special Allowance of the State Council. His research interests include automation engineering, artificial intelligence, and teaching and researching in robot navigation.