IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 5, 2020, accepted March 17, 2020, date of publication March 20, 2020, date of current version April 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2982215

Convolutional Features Combining SL(3) Group for Visual Tracking

YINGHONG XIE[1], **JIE SHEN**[2], **XIAOWEI HAN**[3], **AND CHENGDONG WU**[4]
[1]College of Information Science and Engineering, Shenyang University, Shenyang 110044, China
[2]College of Engineering and Computer Science, University of Michigan–Dearborn, Dearborn, MI 48128, USA
[3]Institute of Scientific and Technological Innovation, Shenyang University, Shenyang 110044, China
[4]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 71601126, in part by the Science and Technology Program Project of Shenyang City under Grant 18013015, and in part by the Double Hundred Project of Shenyang City under Grant Z18-5-013.

ABSTRACT For visual tracking, key factors that affect the performance of trackers are related to whether it can effectively extract the appearance information and spatial information of a target. And most of state-of-the-art trackers either do not model the appearance information and spatial information separately or do not design special strategies to deal with the strong geometric deformation of the target. In this paper, we design an appearance information model and a spatial information model separately, and then combine them to obtain complementary benefits. Firstly, because the features from deeper layers of a convolutional neural network (CNN) can better describe the semantic information of a target while the spatial information becomes less, we adopt the features from the deepest layer as the appearance information model. Secondly, we focus on tracking the target with drastic geometric deformation through utilizing a projection transformation group (SL(3) group) to model the geometric transformation of the target, where SL(3) group can describe the geometric deformation more accurately. Furthermore, a standard discriminative correlation filter is used to develop the effect of convolutional features and is more efficient than other methods used for CNN. Extensive experiments results show that our tracker outperforms all the compared trackers.

INDEX TERMS Object tracking, convolutional neural network, SL(3) group, projection transformation, convolutional features.

I. INTRODUCTION

The purpose of video object tracking is to locate the object in the video continuously and accurately. It is becoming more and more important in various fields. There are two difficulties in this task. Firstly, a target in subsequent video frames may suffer complex situations such as deformation, occlusion, illumination change, background change, and abrupt motion [1]. This makes the difference of target appearance and position between the consecutive frames too large, which leads to the failure of tracking. Secondly, in some application scenarios, it is necessary to track the target in real time, which requires high time efficiency. It is even more difficult to design a real-time high performance tracker. Many traditional algorithms have made some breakthroughs, such as IVT [2], SCM [3],TLD [4], STRUCK [5], MIL [6], APGL1 [7], ASLAS [8], and KCF [16]. However, the performance of these algorithms is far from satisfying all kinds of complex scenarios in real applications [9].

In fact, two important parts of a tracking method for a non-rigid target are appearance representation modeling and geometrical transformation modeling. A robust appearance model can be invariant to illumination change and background clutter, while a better geometrical transformation model can describe the shape and scale deformation of a target more accurately.

For appearance representation modeling, numerous hand-crafted features have been used to describe the target appearance such as color histogram and subspace representation, or different data association models [10] have been applied. Paper [11] applies key patch sparse representation (KPSR) to reduce the disturbance of partial occlusion or unavoidable background information. In recent years, CNNs are an outstanding choice in solving the problem of

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

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

target recognition. They also greatly promote the development of target tracking technology. Some trackers [22]–[25] directly use CNNs as classifiers and take full advantage of end-to-end training. Some other trackers [26]–[31] integrate deep features into traditional tracking methods, and benefit from the expression ability of CNN features.

For the feature map from a deeper convolutional layer that corresponds to a larger receptive field, it can be understood that CNN does feature extraction of the image from a more global perspective. Therefore, the outputs of the last convolutional layer encode the highest semantic information of the target and such representations are robust to significant appearance variations [31]. We use the features from the deepest convolutional layer to design appearance representation model in this paper.

For spatial feature modeling, the classic methods include translation transformation, isometric transformation, Euclidean transformation, affine transformation, and projection transformation. Considering that affine transformation is an approximation of projection transformation (SL(3)), we choose the more accurate imaging model (projection transformation) to represent the deformation of the target. For every frame, we compute the projection transformation samples according to the projection transformation vectors of the former frame.

In order to locate the deformable target more accurately, we design a novel tracking method based on the fact that the projection transformation can better capture the target deformation and that the features from the deepest level of CNN can better represent the appearance of the tracking target. The proposed tracker learns correlation filters over the deepest features. The main differences between papers [16], [17] are the correlation filters based on the CNN features rather than hand-crafted features. On the other hand, we feed the projection transformation region samples into CNN instead of the whole image so that we can draw a non-rectangular bounding box when it needs to adapt to the geometric deformation of the target.

The main contributions of the proposed tracker include

(1) The features from the deepest CNN layer are used to design the appearance model of the target, which is robust to significant appearance variations.
(2) The projection transformation manifold is utilized to estimate possible locations of the target and is more accurate for drastic geometric deformation.
(3) A hybrid strategy offers complementary benefits. In this approach, features from the deepest CNN layer are used as appearance representation model and projection manifold is applied as spatial information model.
(4) The bounding boxes predicted by projection transformation can more effectively locate the regions of the target before extracting CNN features.

The rest of the paper is organized as follows: In section 2 the related work is discussed. In section 3, our approach is described. SL(3) manifold and its Lie algebra are introduced, and then the geometric transformation model is designed on SL(3) group. We also model a discriminative correlation filter for our tracker to compute the correlation score of the semantic information model. In addition, the details of a tracking algorithm are designed. Section 4 provides the details of implementation and the results of numerical evaluations in comparison with other state-of-the-art trackers. Finally, we summarize the design ideas of our tracker and provide some concluding remarks in section 5.

II. RELATED WORK

In this section, we first discuss some tracking methods [63], [64] related closely to our work.

A. TRACKING BY CORRELATION FILTERS

In recent years, correlation filters have been widely applied for visual tracking algorithms because of their high computational efficiency with Fourier transform. Tracking algorithms using correlation filters [38] don't need hard-threshold samples of target appearance because they regress all the circular-shifted versions of input features to a Gaussian function. Correlation filters [35]–[37] were also developed. For example, Ross *et al.* [2] designed a minimum output sum of squared error filter for the target appearance with fast visual tracking. Lu *et al.* [32] proposed a novel shrinkage loss to penalize the importance of easy training data. Lu *et al.* [33] developed an effective channel-aware learning algorithm by analyzing the channel-wise information of convolution features in deep regression tracking. Zhang *et al.* [34] designed a tracking framework based on convolutional net with semantics estimation and region proposals. The context learning methods [13], [62] describe a spatial-temporal relationship between tracking objects and their local dense context in a Bayesian framework, and adopt Fast Fourier Transform (FFT) to adjust the target scale whenever it changes. Danelljan *et al.* [14] modeled a scale estimation filter to estimates the target scale by learning discriminative correlation filters using a scale pyramid representation. Henriques *et al.* [15] designed kernelized correlation filters for training and detection by using circulant matrices. Ma *et al.* [18] presented a long-term correlation tracking, and an online random classifier was trained for objects redetection. Papers [16], [17] utilized multiple dimensional features for tracking. And Danelljan *et al.* [19] introduced a spatial regularization component for penalizing correlation filter coefficients based on their spatial locations. But, the shape of the bounding box is still a rectangle that can't adapt to the geometric deformation of the target.

B. TRACKING BY CNNs

In recent years, CNNs are an outstanding choice in solving the problem of target recognition. Wang and Yeung [20] proposed learning a deep compact representation for visual tracking. Wang *et al.* [21] developed adaptation in two layers of deep features learning module for including the appearance information of specific target. Nam and Han [22] utilized a large image set with ground-truth values for extracting appearance representation from CNN. Nam *et al.* [23] presented the appearance by CNNs and managed the appearance models in a tree for tracking. And trackers [24], [25] utilized CNNs as classifiers and take advantage of end-to-end training. Danelljan *et al.* [26] used an implicit interpolation model for solving the learning problem in continuous space domains. The formula can effectively integrate multi-resolution deep features map. And researchers in [27], [28], [30] integrated deep features into traditional tracking algorithms, using the benefit from the expression ability of CNN features. Li *et al.* [29] integrated target-aware features with a Siamese matching network for visual tracking. Chu *et al.* [62] proposed a CNN-based framework for online multi-object tracking, which utilized the merits of single object trackers in adapting appearance models and searching for target in the next frame. As it is shown in [31] the features from deepest convolutional layer have more semantics information and less space information, we make full use of the CNN features of the deepest layer to model appearance features. The difference from the existing CNN trackers mentioned above is that we feed the projection transformation region samples into the CNN network instead of the whole image.

C. TRACKING ON MANIFOLD

Many classical tracking methods [39], [40] used affine manifold for describing the geometric deformation of the target. Wu *et al.* [41] utilized affine transformation in describing the transformation process of the target, in combination with a particle filter framework in realizing the tracking. Liu *et al.* [42] used the fusion of color and shape as the main features for target tracking under affine manifold, based on a particle filtering tracking framework. Khan and Gu [43], [44] applied affine transformation and proposed a target tracking algorithm by using Riemannian Manifold geometry structure. Considering that affine transformation is an approximation of projection transformation (SL(3)), we choose a more accurate imaging model (projection transformation) to represent the deformation of the target.

III. THE PROPOSED METHOD

A. SL(3) MANIFOLD AND THE GEOMETRIC TRANSFORMATION MODEL

1) SL(3) GROUP AND ITS ALGEBRA

The following theory comes mainly from differential geometry. Reference [45] provides more knowledge.

A Lie group is a group with an analytic manifold structure, which makes the following maps is analytic:

$$\begin{aligned} G \times G \to G \quad (X, Y) \to XY \\ G \to G \quad X \to X^{-1}. \end{aligned} \tag{1}$$

The local neighborhood of any group element G can be described by its tangent space. The tangent space for the identity element forms its Lie algebra.

The set of nonsingular $n \times n$ square matrices forms a Lie group where the group product is computed by matrix multiplication, GL(n, R) denotes the general linear group of order n, where R is an n-dimensional real space. Lie groups are differentiable manifolds on which we can do calculus. Being a sub-group of GL(n, R), the special linear group SL(n,R) is the space of all real $n \times n$ matrices H satisfies detH = 1. Its Lie algebra denoted by sl(n,R) consists of real matrices of trace zero.

In this paper, we use 3*3 dimensional projection group (SL(3)) to represent the geometric transformation model, we normalize the matrices to have determinant 1. Its corresponding Lie algebra is sl(3). The exponential map is a homeomorphism between a neighborhood of $I \in$ SL(3) and a neighborhood of the null matrix $0 \in$ sl(3).

Let $A_i (i \in \{1, 2, \ldots, 8\})$ be a basis of the Lie algebra sl(3). Any matrix $A \in$ sl(3) can be written as a linear combination of the matrices A_i, $A(x) = \sum_{i=1}^{8} x_i A_i$, where $x = (x_1, x_2, \ldots, x_8)$ is an 8*1 dimensional vector and x_i is the i-th element of the base field. The basis matrix of sl(3) is as follows:

$$\begin{aligned} A_1 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad A_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ A_3 &= \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad A_4 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \\ A_5 &= \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad A_6 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \\ A_7 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad A_8 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \end{aligned} \tag{2}$$

2) GEODESIC ON SL(3) GROUP

Because projection group is definite symmetric manifold and belongs to Riemannian manifolds, the distance between projection groups can be computed on Riemannian manifolds.

SL(3) group is non-compact Lie group. Non-compact Lie groups do not have bi-invariant Riemannian metric, so the exponential map on Lie group is not consistent with the geodesic. In order to calculate the geodesic on SL(3) group, a metric structure needs to be defined on SL(3) group to calculate the new exponential map Expp, which is also known as Riemannian exponential map.

The common method for defining the metric structure on manifold M is that an inner product $\langle \cdot, \cdot \rangle$ is given on the tangent space $T_p M$ for each point $p \in M$, which is Riemannian metric, and the length of a tangent vector $U \in T_p M$ is: $\|U\| = \langle U, U \rangle^{\frac{1}{2}}$. Therefore, we can define exponential map Expp as

$$Expp(U) = \exp(-U^T) \exp(U + U^T). \tag{3}$$

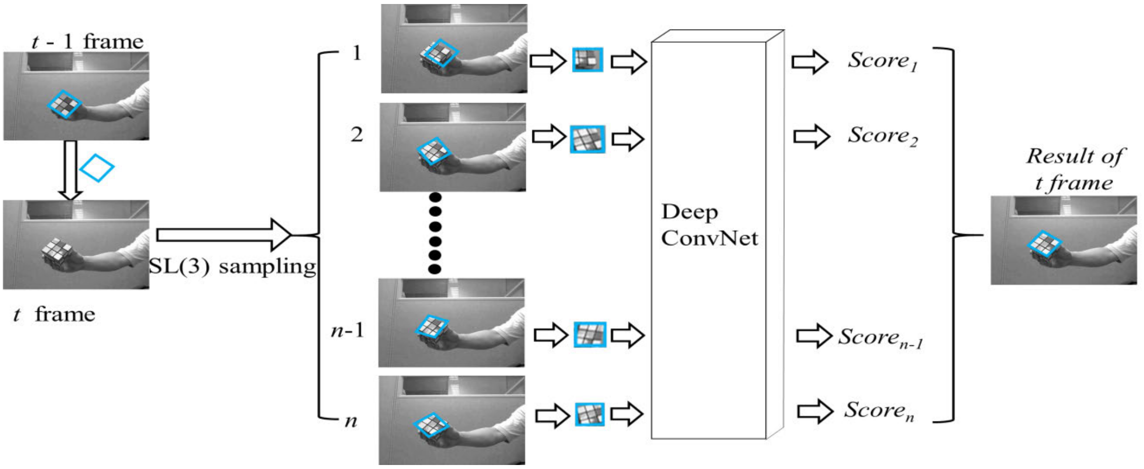


FIGURE 1. Schematic overview of the proposed framework based on projection transformation and convolutional Features. It consists of the following four stages: (1) sampling projection transformation(2)computing appearance CNN features (3) inputting into correlative filter (4) obtaining the tracking result.

3) THE GEOMETRIC TRANSFORMATION MODEL

We use projection transformation to represent the model of the target geometrical deformation. And the geometrical transformation between two adjacent frames can be considered as one point moving to another point on Riemann manifold, because projection transformation matrix is a positive definite symmetric manifold, which is a Lie group and doesn't obey Euclidean space.

In this paper, we make use of the relationship between the two adjacent points on Riemannian manifold to establish a geometric transformation model. And this relationship can be described by the tangent vector of the point on the manifold, so the object geometric transformation model is designed on Riemannian manifold and its tangent space, respectively:

$$S_t = S_{t-1} \exp(v_{t-1}), \tag{4}$$

$$v_t = a(v_{t-1} - v_{t-2}) + \mu_{t-1}, \tag{5}$$

where vector $S_t = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8]^{\mathrm{T}}$ is the projection transformation parameter. v_t is defined as the velocity vector from point S_{t-1} to point S_t on the tangent space, and it represents the movement of the target. Suppose v_t follows a Gauss distribution, and $\mu_{1:t}$ is denoted as Gauss white noise. a is an autoregressive coefficient.

The algorithm makes full use of the Lie group structure of projection transformation parameters space, with the geometric transformation information being described on Riemannian manifold and tangent space.

B. DISCRIMINATIVE CORRELATION FILTER

Compared with other costly methods used for CNNs training, the discriminative filter is more efficient for it is trained by computing a linear least-square and using Fast Fourier Transform (FFT). In this paper, we use a standard discriminative correlation filter to compute the scores of each candidate convolutional features for object tracking. The features of the deepest layer are applied as the input of the discriminative filter.

Let x_k denotes one input sample at frame k, where $k = 1, 2, \ldots, t$. t denotes the current frame number. And y_k is the Gaussian function label which denotes the desired correlation output at frame k. In order to get a minimum loss, we learn a correlation filter w as

$$w^* = \arg\min_{w} \sum_{k=1}^{t} \|w_t \cdot x_k - y_k\|^2 + \lambda \|w_t\|^2, \tag{6}$$

where λ is a regularization parameter ($\lambda >= 0$), x_k^i denotes the *ith* feature channel of x_k.We utilize the online update rule [61] to gain the efficient solution for equation (6). At frame t, we update the numerator M_t^i and denominator N_t of the DFT(discrete Fourier transformed) filter w^i as follows:

$$M_t^i = (1-\delta)M_{t-1}^i + \delta \bar{Y}_t \cdot X_t^i, \tag{7}$$

$$N_t = (1-\delta)N_{t-1} + \delta \sum_{i=1}^{d} \bar{X}_t^i \cdot X_t^i + \lambda, \tag{8}$$

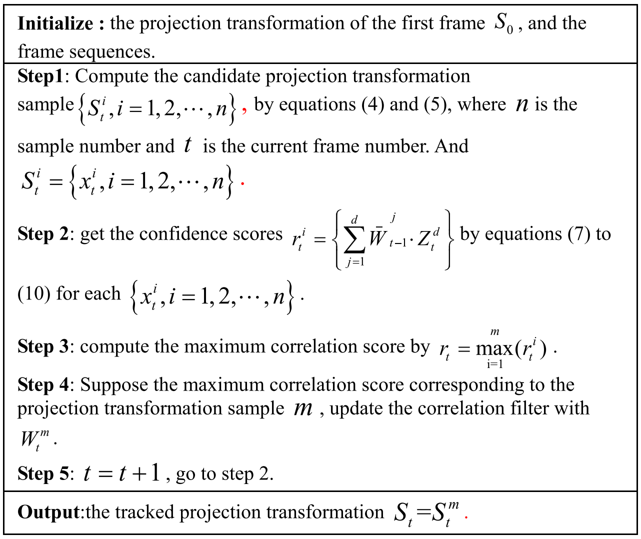
where the capital letter represents the 2-dimensional Fourier transform from the corresponding lowercase, the operator $\cdot$ is element-wise multiplication, and δ is the learning rate. We build the learned filter as

$$W_t^i = \frac{M_t^i}{N_t}, \tag{9}$$

$$r_t = \left\{ \sum_{i=1}^{d} \bar{W}_{t-1}^i \cdot X_t^i \right\}, \tag{10}$$

where d denotes the sample number. In this paper, we use the correlative filter as equation (10), and the correlation scores are gained in the Fourier domain. For the appearance features of each candidate patch at frame t, we input them into the correlative filter to get the correlation scores.

TABLE 1. The details for the proposed algorithm.



Initialize : the projection transformation of the first frame S_0, and the frame sequences.
Step1: Compute the candidate projection transformation sample $\left\{S_t^i, i=1,2,\cdots,n\right\}$, by equations (4) and (5), where n is the sample number and t is the current frame number. And $S_t^i = \left\{x_t^i, i=1,2,\cdots,n\right\}$. **Step 2**: get the confidence scores $r_t^i = \left\{\sum_{j=1}^{d} \bar{W}_{t-1}^{j} \cdot Z_t^d\right\}$ by equations (7) to (10) for each $\left\{x_t^i, i=1,2,\cdots,n\right\}$. **Step 3**: compute the maximum correlation score by $r_t = \max_{i=1}^{m}(r_t^i)$. **Step 4**: Suppose the maximum correlation score corresponding to the projection transformation sample m, update the correlation filter with W_t^m. **Step 5**: $t=t+1$, go to step 2.
Output:the tracked projection transformation $S_t = S_t^m$.

C. TRACKING ALGORITHM FRAMEWORK

For the geometric space information model, we adopt projection transformation to represent the deformation of the target and to predict possible locations of a target. On the other hand, we use the features extracted from the deepest layer to represent the appearance information model, because the features extracted from the deeper layer in CNN contain more appearance information and less space information. The CNN may be an ALexNet [65] or a VGG-Net. We delete the fully connected layers because they have little geometric spatial information.

As it is shown in Figure 1, we first locate the bounding box by hand for the first frame; otherwise, according to the position and shape of the target bounding box of the previous frame, we draw the same target bounding box at the same position of the current frame. Then, the method generates M projection transformation samples according to equations (4) and (5), and resizes these samples to the same size. Next, these sample patches are input into the CNN to extract the feature maps, and the features are input into the correlative filters to compute the confidence scores using equation (10). Finally, the projection transformation sample corresponding to the maximum correlation score is the output as the tracking result of the current frame. The detail is shown in Table 1.

IV. DETAILS AND EXPREIMENTAL EVALUATION

A. IMPLEMENTATION DETAILS

1) CNN FRAMEWORK

We use ImageNet [46] as the data set, randomly select 200 classes of ImageNet [46] as training data set, and divide these 200 classes into 80% training data set, 10% validating data set and 10% testing data set. We adopt the VGG-Net-19 [47] trained on ImageNet [46] to extract appearance features for each candidate projection patch. We delete the fully connected layers and use the outputs extracted from last layer as our features.

2) CNN 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 [46]. The training consists of more than 80 epochs, each consisting of 1000 sampling pairs. The gradients of each iteration are estimated by the size of 10 mini batches, and the learning rate is from 10^{-2} to 10^{-5} at each period from.

3) CORRELATION FILTERS TRAINING

The regularization parameter of equation (6) is set to $\lambda = 10^{-4}$, and the kernel width is 0.1 to generate the Gaussian labels. The learning rate δ in equations (7) and (8) is set to 0.01.

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

B. DATASETS AND EVALUATION METRICS

1) THE OTB BENCHMARKS

The OTB-201 [48] and OTB-2015 [49] are one of the most popular benchmarks in visual tracking field and contain 50 and 100 image sequences with various challenging factors. They are divided into eleven attributes, such as illumination, deformation, scale change and others.

The metrics standards on the OTB benchmark include two aspects: average per-frame success rate and precision. On 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 success plot is reported. On the other hand, the precision plot can be obtained in a similar way. In most of existing papers, the threshold for the precision plot is set to 20.

2) THE VOT BENCHMARKS

The VOT benchmarks are also widely used as tools for the performance evaluation. The VOT benchmarks include VOT2015 [50], VOT2016 [51], VOT2017 [52], VOT2018 [53] and VOT2019 [54]. VOT2015 and VOT2016 contain 60 same video sequences, and the targets in these two are divided into 6 challenging factors including camera motion, motion change, scale change, illumination change, and occlusion. VOT2017 includes 10 different sequences from VOT2016. The evaluation of VOT2019 [54] includes the standard VOT and other popular methodologies for

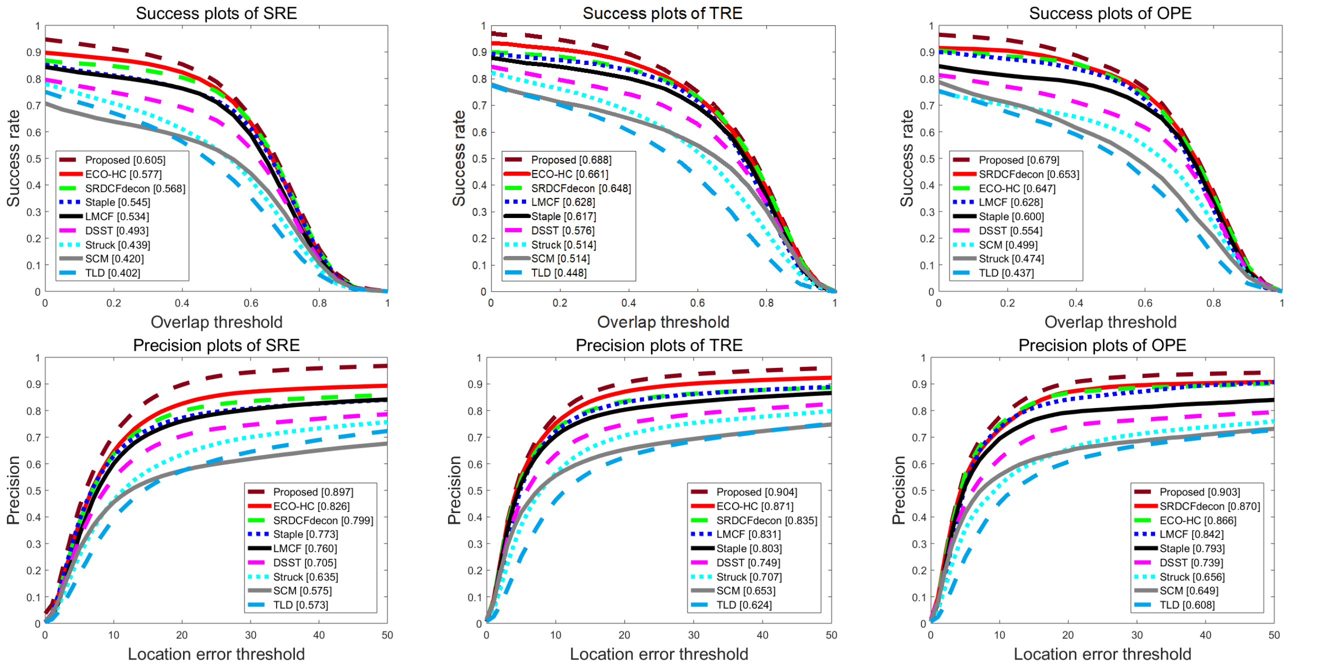


FIGURE 2. Overlap success plots and distance precision plots using SRE, TRE and OPE. The legend of overlap success contains AUC score while the legend of distance precision contains threshold scores at 20 pixels for each tracker.

TABLE 2. Results of different trackers on OTB benchmarks.

		Ours	ECO-HC	SRDCFdecon	Staple	LMCF	DSST	SCM	Struck	TLD
OTB-2013	SUCCESS	0.679	0.652	0.653	0.600	0.628	0.554	0.499	0.474	0.437
	PRECISION	0.903	0.874	0.870	0.793	0.842	0.739	0.649	0.656	0.608
OTB-2015	SUCCESS	0.648	0.592	0.627	0.581	0.533	0.524	0.488	0.457	0.419
	PRECISION	0.872	0.814	0.825	0.784	0.730	0.712	0.598	0.613	0.572

short-term tracking analysis as well as the standard VOT methodology for long-term tracking analysis.

The evaluation on the VOT benchmark is based on the re-initialized methodology in which a tracker will be reset after five frames of no overlap with the ground truth. The evaluation emphasizes the short-term effectiveness, and the metrics standards on the VOT benchmark include accuracy (A), robustness (R), and expected average overlap (EAO). A better tracker has higher A and EAO scores and lower R scores.

C. EXPERIMENTS ON OTB BENCHMARKS AND VOT BENCHMARKS

Our tracker is evaluated with state-of-the-art trackers on the benchmark datasets OTB-2013 [48], OTB2015 [49], VOT2015 [50], and VOT2016 [51], respectively.

1) THE OTB BENCHMARKS

our tracker is compared with 8 popular trackers: ECO-HC [30], SRDCFdecon [56], LMCF [57], Staple [58], Struck [5], DSST [14], SCM [3] and TLD [4]. Table 2 compares the results of different trackers on OPE evaluation. And Figure 2 illustrates the overlap success plots and distance precision plots using SRE, TRE and OPE. The legend of overlap success contains AUC score while the legend of distance precision contains threshold scores at 20 pixels for each tracker. Then, the tracker performance under different video attributes is analysed in Figure 3. The results in the table and figures mentioned above suggest that our tracker has better performance than other trackers. To make the tracking result more intuitive, Figure 4 shows the results of tracking bounding boxes on some challenging video sequences. The main challenges of the first two sequences is geometric transformation. All the algorithms except SCM and Struck can track the target well, but our tracker can gain the deformation of the target and the resulting bounding box is a tilted parallelogram because our tracker uses projection transformation to locate the target. For sequence 3, the object suffers illumination variation and background clutter. The tracking results of all the compared algorithms are chaotic and eventually lost the target. Our tracker can still recognize the target and track it well because it builds the semantic information model and the space information model, respectively. As to sequence 4, the object suffers fast motion and temporary occlusion. The

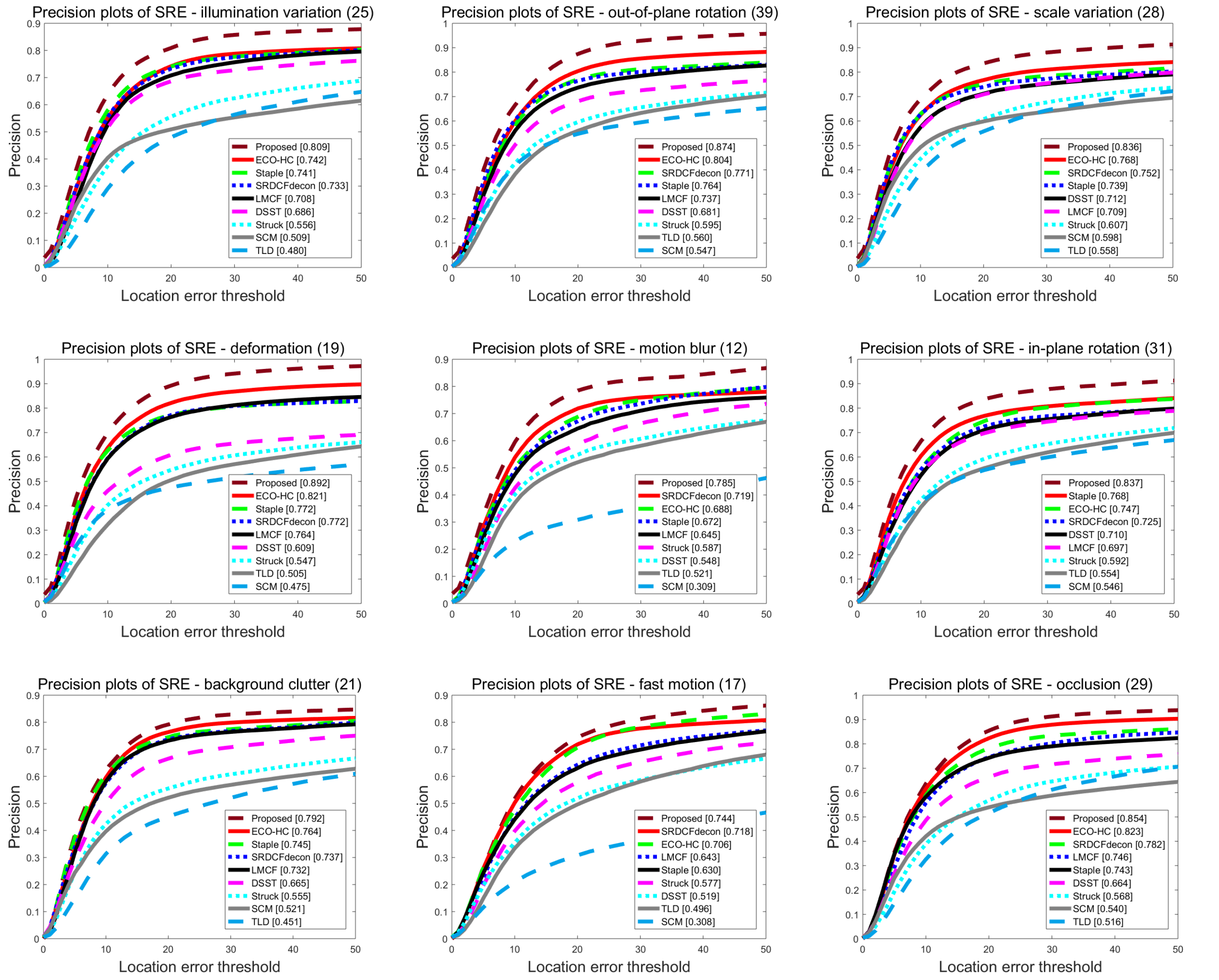


FIGURE 3. Distance precision plots over 9 tracking challenges of illumination variation, out-of-plane rotation, scale variation, deformation, motion blur, in-plane rotation, background clutter, fast motion and occlusion.

TABLE 3. Accuracy values under different challenging sequences.

	tag_camera_motion	tag_empty	tag_illum_change	tagmMotion_change	tag_occlusion	tag_size_change	mean	weighted mean	pooled
Ours	0.5536	0.6021	0.6422	0.5460	0.5104	0.5275	0.5671	0.5593	0.5676
HCF	0.4383	0.4928	0.4497	0.4255	0.4337	0.3458	0.4310	0.4336	0.4464
SRDCF	0.5306	0.5745	0.6891	0.4798	0.4153	0.4662	0.5259	0.5176	0.5248
SCT4	0.4748	0.5331	0.4591	0.4411	0.4451	0.3675	0.4535	0.4619	0.4751
EBT	0.4767	0.4869	0.4007	0.4275	0.3777	0.3465	0.4193	0.4374	0.4480
Staple	0.5284	0.5741	0.7200	0.4989	0.4311	0.5037	0.5427	0.5290	0.5326
IVT	0.4098	0.5089	0.5554	0.4046	0.3111	0.3913	0.4302	0.4272	0.4347

Stuck algorithm has misidentified the target in the tracking process. All the tracking results suggest that our tracker can bound the target more accurately than the rectangle boxes when the target suffers from large geometric deformation, and our tracker performs better than the compared algorithms.

2) VOT BENCHMARKS

In our experiments, we evaluate the tracking performance on VOT2016 [51] and VOT2019 [54], respectively.

For VOT2016 [51], our tracker is compared with 6 state-of-the-art trackers: HCF [31], SRDCF [56], SCT4 [60],

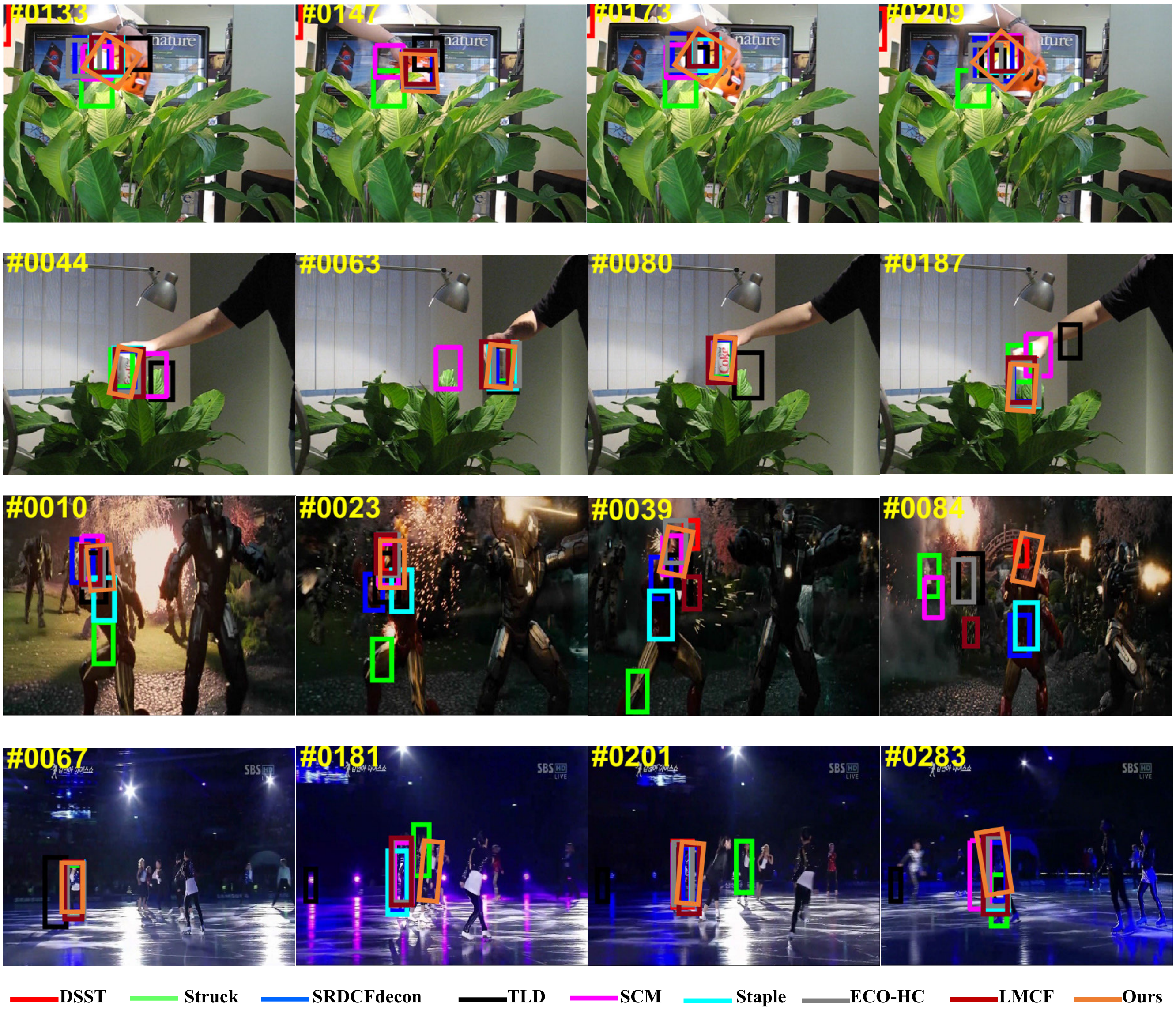


FIGURE 4. Bounding box results for the proposed algorithm and the compared algorithms.

TABLE 4. Robustness values under different challenging sequences.

	tag_camera_motion	tag_empty	tag_illum_change	tagmMotion_change	tag_occlusion	tag_size_change	mean	weighted mean	pooled
Ours	27.8333	8.5667	3.1333	21.6333	16.3333	13.7333	15.2055	17.7248	57.0000
HCF	27.0000	25.0000	5.0000	30.0000	19.0000	17.0000	20.5000	23.8569	85.0000
SRDCF	43.0000	16.0000	8.0000	36.0000	22.0000	21.0000	24.3333	28.3167	90.0000
SCT4	48.0000	29.0000	8.0000	44.0000	19.0000	33.0000	30.1667	36.1918	117.0000
EBT	20.0000	11.0000	3.0000	19.0000	17.0000	11.0000	13.5000	15.1935	54.0000
Staple	34.0000	13.0000	7.0000	35.0000	24.0000	15.0000	21.3333	23.8950	81.0000
IVT	103.0000	54.0000	12.0000	92.0000	34.0000	47.0000	57.0000	70.3371	238.0000

EBT [59], Staple [58], and IVT [2]. Table 3 illustrates the accuracy values (A) under different challenging sequences. Table 4 shows the robustness values (R) under different challenging sequences, where the red, blue and green fonts indicate the first, second and third places, respectively. On average, our tracker ranks first, which is also verified in Figure 5 that is an accuracy-robustness plot with best trackers closer to the upper right corner. Figure 7 illustrates the

TABLE 5. The expected average overlap (EAO) as well as accuracy and robustness raw values (A, R) for the baseline and the real time experiments. For the unsupervised experiment the no-reset average overlap AO [48] is used.

		baseline		realtime		unsuperized	
tracker	EAO	A	R	EAO	A	R	AO
Ours	0.267	0.548	0.456	0.215	0.503	0.412	0.463
RankingR	0.252	0.548	0.417	0.091	0.288	0.783	0.435
SSRCCOT	0.234	0.495	0.507	0.081	0.360	1.505	0.380
TCLCF	0.170	0.480	0.843	0.170	0.480	0.843	0.338
RSiamFC	0.163	0.470	0.958	0.163	0.470	0.958	0.285
Struck	0.094	0.417	1.726	0.088	0.428	1.926	0.174
IVT	0.087	0.391	2.002	0.039	0.366	0.331	0.110

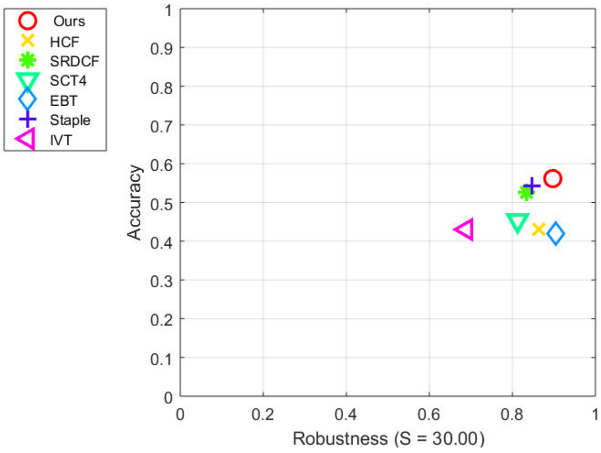


FIGURE 5. Accuracy-robustness plot. Best trackers are closer to the upper right corner.

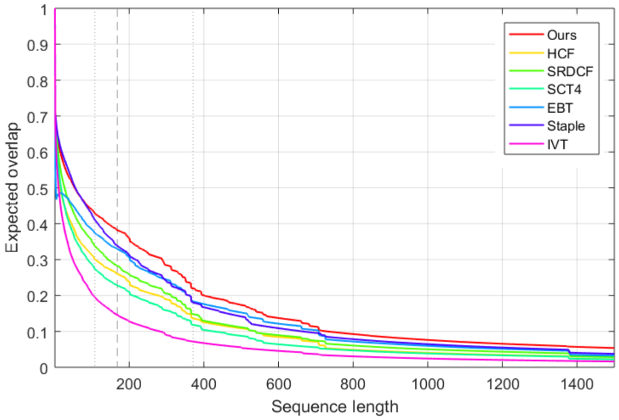


FIGURE 7. Expected average overlap(EAO) curves for 6 state-of-the-art trackers. Our tracker has much better performance than the compared trackers.

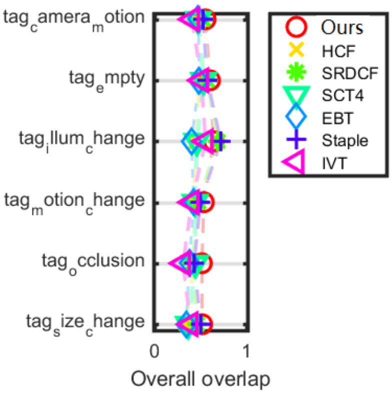


FIGURE 6. Overall overlap plot under different challenges.

expected average overlap (EAO) curves for these 7 trackers. From all of the above, we can conclude that our tracker has much better performance than the compared trackers. The reason is that we design the appearance representation model and spatial information model separately, and use the output from the deepest CNN layer as the appearance representation model and projection manifold as the spatial information model, which forms complementary advantages. Moreover, Figure 6 shows the overall overlap plot under different challenges, which also demonstrates the effectiveness of our tracker.

Furthermore, the raw FPS of the proposed method is 112.354 under speed report for experiment baseline. It is much slower than HCF tracker the value of which is 328.7264, and it is also slower than SRDCF tracker (503.1796 fps) because the projection sampling and feature extraction from CNN take up most of the running time. But, the computation efficiency of correlative filter makes our tracker much faster than other compared trackers, and it can track the object in real time.

For VOT2019 [54], our tracker is compared with 6 state-of-the-art trackers: SSRCCOT [54] that adds a selective spatial regularization to the CCOT [26] tracker, TCLCF [54], RSiamFC that is an extension of SiamFCtracker [55], RankingR [54] that uses a light weight deep neural network, Struck [5], and IVT [2].

Table 5 illustrates the expected average overlap (EAO) as well as the accuracy and robustness raw values (A,R) for the baseline and the real time experiments. And the no-reset average overlap AO [48] is used for the unsupervised experiment. The results show that our method outperforms the compared algorithms.

V. CONCLUDING REMARKS

In real application scenarios, the target suffers more complicated challenges, such as illumination change, background blur, fast motion, various deformation, and others. In order to design a robust tracker, it is necessary to control two dominant factors (appearance representation and spatial information), which significantly affect the performance of the algorithm. In this paper, we put emphasis on tracking the target with drastic geometric deformation and design an appearance representation model and a spatial information model, respectively; then, the two models are combined to achieve complementary benefits. In detail, based on the observation that the features extracted from a deeper layer of CNN can better describe the semantic information of a target while the spatial information becomes less, and because the semantic information is robust to appearance variations, we adopt the features from the deepest layer as the appearance representation model. Then, since the projection group (SL(3)) is used for describing the imaging process of geometric transformation in mathematics field, we utilize SL(3) group to model the geometric transformation of a target, leading to a space information model for our tracking method. Next, a discriminative correlation filter is used to compute the scores for each candidate tracking patch. Finally, by combining the information from both the appearance model and space model, the bounding box is located for each frame.

Extensive experiment results on OBT benchmarks and VOT benchmarks show that our tracker outperforms all the compared trackers. Furthermore, taking the advantage of the high computational efficiency of the discriminative filter using FFT, our tracker also has a higher speed report.

REFERENCES

[1] H. Zhang, Z. Gao, X. Ma, J. Zhang, and J. Zhang, "Hybridizing teaching-learning-based optimization with adaptive grasshopper optimization algorithm for abrupt motion tracking," *IEEE Access*, vol. 7, pp. 168575–168592, 2019.

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

[3] 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.

[4] 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.

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

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

[7] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, p. 13, 2006.

[8] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via coarse and fine structural local sparse appearance models," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4555–4564, Oct. 2016.

[9] P. Li, D. Wang, L. Wang, and H. Lu, "Deep visual tracking: Review and experimental comparison," *Pattern Recognit.*, vol. 76, pp. 323–338, Apr. 2018.

[10] Z. He, X. Li, X. You, D. Tao, and Y. Y. Tang, "Connected component model for multi-object tracking," *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3698–3711, Aug. 2016.

[11] Z. He, S. Yi, Y.-M. Cheung, X. You, and Y. Y. Tang, "Robust object tracking via key patch sparse representation," *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 354–364, Feb. 2017.

[12] 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. 13–18.

[13] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang, "Fast visual tracking via dense spatio-temporal context learning," in *Proc. ECCV*, 2014, pp. 127–141.

[14] M. Danelljan, G. Häger, F. Shahbaz Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2014, pp. 1–6.

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

[16] 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.

[17] 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.

[18] 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.

[19] 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, pp. 4310–4318.

[20] N. Wang and D. Yeung, "Learning a deep compact image representation for visual tracking," in *Proc. NIPS*, 2013, pp. 1–8.

[21] L. Wang, T. Liu, G. Wang, K. L. Chan, and Q. Yang, "Video tracking using learned hierarchical features," in *Proc. ECCV*, 2015, pp. 1–12.

[22] 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.

[23] H. Nam, M. Baek, and B. Han, "Modeling and propagating CNNs in a tree structure for visual tracking," in *Proc. ECCV*, Feb. 2016, pp. 1–10.

[24] 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, pp. 3356–3365.

[25] 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.

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

[27] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proc. ICCV*, Dec. 2015, pp. 58–66.

[28] 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.

[29] X. Li, C. Ma, B. Wu, Z. He, and M.-H. Yang, "Target-aware deep tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1369–1378.

[30] 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)*, Jul. 2017, pp. 6638–6646.

[31] 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.

[32] X. Lu, C. Ma, B. Ni, X. Yang, I. Reid, and M.-H. Yang, "Deep regression tracking with shrinkage loss," in *Proc. ECCV*, 2018, pp. 1–17.

[33] X. K. Lu, F. H. Tang, H. Huo, and T. Fang, "Learning channel-aware deep regression for object tracking," *Pattern Recognit. Letters.*, vol. 2, no. 7, pp. 1–7, Jul. 2018.

[34] H. L. Zhang, J. Chen, G. H. Nie, and S. Q. Hu, "Uncertain motion tracking based on convolutional net with semantics estimation and region proposals," *Pattern Recognit.*, vol. 102, pp. 1–11, Jun. 2020.

[35] X. K. Lu, C. Ma, B. B. Ni, and X. K. Yang, "Adaptive region proposal with channel regularization for robust object tracking," in *Proc. ECCV*, 2018, pp. 1–17.

[36] X. Lu, B. Ni, C. Ma, and X. Yang, "Learning transform-aware attentive network for object tracking," *Neurocomputing*, vol. 349, pp. 133–144, Jul. 2019.

[37] X. P. Dong, J. B. Shen, D. M. Wu, K. Guo, X. G. Jin, and F. Porikli, "Uadruplet network with one-shot learning for fast visual online learning," in *Proc. ECCV*, 2017, pp. 1–12.

[38] X. Lu, W. Wang, C. Ma, J. Shen, L. Shao, and F. Porikli, "See more, know more: Unsupervised video object segmentation with co-attention Siamese networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3623–3632.

[39] T. Wakahara, Y. Kimura, and A. Tomono, "Affine-invariant recognition of gray-scale characters using global affine transformation correlation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 4, pp. 384–395, Apr. 2001.

[40] Y. Yamashita and T. Wakahara, "Affine-transformation and 2D-projection invariant k-NN classification of handwritten characters via a new matching measure," *Pattern Recognit.*, vol. 52, pp. 459–470, Apr. 2016.

[41] Y. Wu, J. Cheng, J. Wang, H. Lu, J. Wang, H. Ling, E. Blasch, and L. Bai, "Real-time probabilistic covariance tracking with efficient model update," *IEEE Trans. Image Process.*, vol. 21, no. 5, pp. 2824–2837, May 2012.

[42] L. Liu, D. Jing, and J. Ding, "Adaptive extraction of fused feature for panoramic visual tracking," in *Proc. IEEE 3rd Int. Conf. Image, Vis. Comput. (ICIVC)*, Jun. 2018, pp. 21–25.

[43] 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)*, Barcelona, Spain, Nov. 2011, pp. 1847–1854.

[44] Z. H. Khan and I. Y.-H. Gu, "Bayesian online learning on Riemannian manifolds using a dual model with applications to video object tracking," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, Barcelona, Spain, Nov. 2011, pp. 1402–1409.

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

[46] 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. (CVPR)*, Jun. 2009, pp. 248–255.

[47] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015, pp. 1–14.

[48] 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.

[49] 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.

[50] M. Kristan *et al.*, "The visual object tracking VOT2015 challenge results," in *Proc. ECCV Workshops*, Jun. 2015, pp. 1–23.

[51] M. Kristan *et al.*, "The visual object tracking VOT2016 challenge results," in *Proc. ECCV Workshops*, Jun. 2016, p. 7.

[52] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. C. Zajc, T. Vojir, G. Hager, A. Lukezic, A. Eldesokey, and G. Fernandez, "The visual object tracking VOT2017 challenge results," in *Proc. ECCV Workshops*, Oct. 2017, pp. 1949–1972.

[53] M. Kristan *et al.*, "The visual object tracking VOT2018 challenge results," in *Proc. Workshop Vis. Object Tracking Challenge (ECCV Workshops)*, 2018, pp. 1–52.

[54] M. Kristan *et al.*, "The seventh visual object tracking VOT2019 challenge results," in *Proc. Workshop Vis. Object Tracking Challenge (ECCV Workshops)*, 2019, pp. 1–36.

[55] L. Bertinetto, J. Valmadre, J. Henriques, P. H. S. Torr, and A. Vedaldi, "Fully convolutional Siamese networks for object tracking," in *Proc. ECCV Workshops*, 2016, pp. 850–865.

[56] 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, pp. 1430–1438.

[57] M. Wang, Y. Liu, and Z. Huang, "Large margin object tracking with circulant feature maps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4021–4029.

[58] 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, pp. 1401–1409.

[59] 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.

[60] J. Choi, H. J. Chang, J. Jeong, Y. Demiris, and J. Y. Choi, "Visual tracking using attention-modulated disintegration and integration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4321–4330.

[61] V. N. Boddeti, T. Kanade, and B. V. K. V. Kumar, "Correlation filters for object alignment," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2291–2298.

[62] Q. Chu, W. Ouyang, H. Li, X. Wang, B. Liu, and N. Yu, "Online multi-object tracking using CNN-based single object tracker with spatial-temporal attention mechanism," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4836–4845.

[63] G. Olague, D. E. Hernandez, E. Clemente, and M. Chan-Ley, "Evolving head tracking routines with brain programming," *IEEE Access*, vol. 6, pp. 26254–26270, 2018.

[64] G. Olague, D. E. Hernández, P. Llamas, E. Clemente, and J. L. Briseño, "Brain programming as a new strategy to create visual routines for object tracking," *Multimedia Tools Appl.*, vol. 78, no. 5, pp. 5881–5918, Mar. 2019.

[65] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. IPS*, 2012, pp. 1–7.

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, where she is currently an Associate Professor with Information and Engineering Institute. From 2014 to 2016, she was a Postdoctoral Researcher with Tianjin University. She was a Scholar with the University of Michigan–Dearborn, in 2017. She is the first author of more than 20 articles, and the Host Natural Science Foundation of China, in 2015. Her main research interests include artificial intelligence, video image processing, and pattern recognition.

JIE SHEN has 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; a Session Chair for 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. His 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). He is currently the Editor-in-Chief of the *International Journal of Modelling and Simulation*, which is an EI-indexed, peer-reviewed research journal in the field of modeling and simulation.

XIAOWEI HAN received the Ph.D. degree in control theory and control engineering from Northeastern University, in 2005. He is currently a Professor and the President of Scientific and Technological Innovation Institute, Shenyang University. He has presided over or undertaken more than ten research projects supported by national, provincial and municipal funds, completed a number of horizontal engineering projects, compiled two monographs, published more than 40 articles, and obtained more than 50 invention patents, utility model patents. His current research interests include computer vision, artificial intelligence, and wireless sensor networks.

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 the Doctoral Tutor. He has long been involved in automation engineering, artificial intelligence, and teaching and researching in robot navigation. He is also an Expert of Chinese Modern Artificial Intelligence and Robot Navigation. He is also a Special Allowance of the State Council.