IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 21, 2021, accepted January 29, 2021, date of publication February 2, 2021, date of current version March 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3056469

Two-Step Affine Transformation Prediction for Visual Object Tracking

WEIWEI ZHENG[1], HUIMIN YU[1,2,4], (Member, IEEE), AND ZHAOHUI LU[2,3]
[1]College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310007, China
[2]ZJU-League Research & Development Center, Zhejiang 310007, China
[3]Zhejiang Lijia Electronic Technology Company Ltd., Zhejiang 311800, China
[4]Zigong Innovation Center, Zhejiang University, Hangzhou 310007, China

Corresponding author: Huimin Yu (yhm2005@zju.edu.cn)

This work was supported in part by the Research Project of ZJU-League Research & Development Center, Zhejiang Lab under Grant 2019KD0AB01, and in part by the Sichuan Science and Technology Project under Grant 2019YJ0680.

ABSTRACT The main drawback of correlation filter scheme is that it can only predict translation of the object, ignoring the other affine transformations such as rotation, aspect ratio change, and scale change. This paper tries to address this problem by a two-step affine transformation prediction method. The first step is to predict coarse target translation by the correlation filter model with adaptive model update rate generation. Compared with fixed update rate in existing correlation filter based trackers, the presented method generates the update rate by Long Short-Term Memory model with inputs of historical target templates. The second step maps well-designed features to affine transformation parameters. The designed features contain the information of affine transformation which makes the learning of non-linear mapping function possible. The training samples for on-line filter model update are transformed to the same pose, and they help the learnt filter to represent the object better than non-aligned samples. The proposed network is trained from end to end and achieves competitive performance when comparing with state-of-the-arts trackers on four benchmarks, OTB100, UAV123, VOT2018, and VOT2020.

INDEX TERMS Affine transformation, correlation filter, feature alignment, visual object tracking.

I. INTRODUCTION

There are mainly three schemes on visual object tracking problem, consisting of key-point tracking, contour tracking and bounding box tracking. The bounding box tracking scheme is more popular because of its high efficiency and accuracy. So the paper focuses on it. The commonly used tracking-by-detection framework [21] considers the tracking process as feature matching, in which the core problems are feature design and matching method.

Correlation filter model [22] is one kind of matching method that measures the similarities of two image regions by cyclic correlation operation. It has the advantage of high efficiency compared with dense search since it handles all cyclic shifts of search region by Discrete Fourier Transform in one step. But this is also its pain since only translation prediction is completed. In reality, the object in the image space is projected from three-dimensional space, and its movement is one kind of affine transformation.

Most correlation filter based trackers, such as KCF [1], ECO [10], and RPCF [23], only predict the translation and scale change (by simple scale search) of the target object. They work well on videos with few target rotations and aspect ratio changes, however they are easy to fail on more challenging situations. SiamBM [24] and SiamOS [25] handle the rotation and scale change problem by sampling several search regions with different rotation angles and scales, and parallel process these regions. The one with highest score is chosen to update target angle and scale. This kind of method is limited in predicting enumerated target changes and needs more computing resources. So the paper handles this problem by learning a non-linear mapping function from well-designed features to the affine transformation parameters. This kind of regression approach relies on good features and the ability to learn the mapping function from large amounts of data. So we design our network architecture by introducing correlation filter and Log-Polar transformation into it (see Fig.1 for details).

It is found that the output of correlation operation between the object template and search region contains information

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

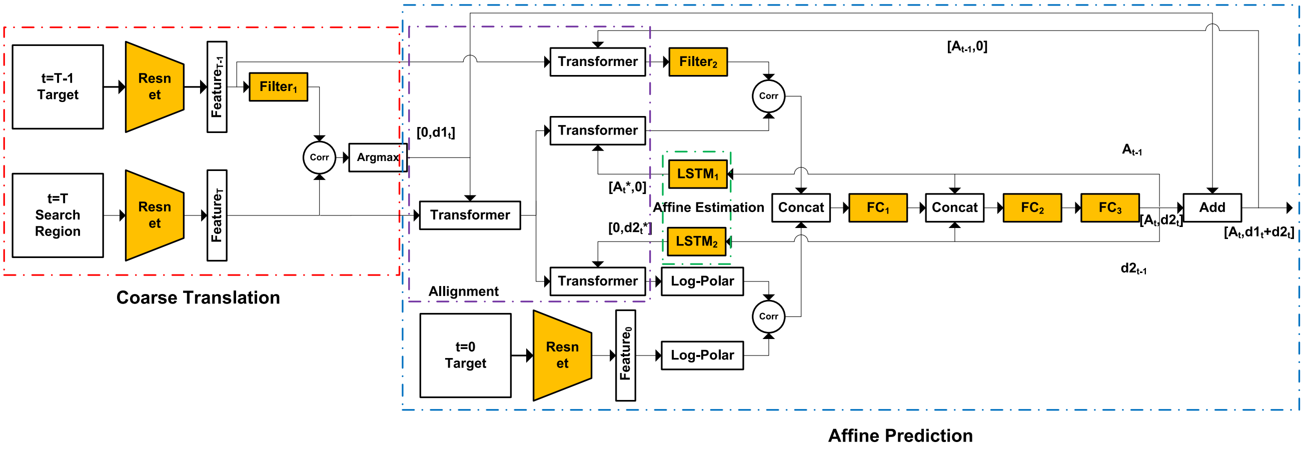


FIGURE 1. Network structure of the proposed method. Recurrent neural network structure is used for our method. A coarse-to-fine scheme is applied to improve tracking performance. Coarse translation is predicted by correlation filter based model, and the search region is transformed for subsequent affine transformation prediction. The correlation filtering output between Log-Polar features of the search region and target object is used to predict the scale change and rotation of the object. And the aligned correlation filtering output is used to predict the translation, shear, and aspect ratio change. These two outputs are concatenated and taken as the feature for final affine transformation prediction. All ResNet blocks share parameters with each other. The parameters of orange net blocks are trainable.

of translation, shear, and aspect ratio change. Besides, the Log-Polar coordinate system takes the logarithm of central distance and angle as two axes, so the scale change and rotation of the object cause the Log-Polar image to shift along the corresponding axes. It means that the correlation between Log-Polar features of the target and search region contains information of scale change and rotation. The concatenation of outputs from the two correlation operations are considered as the feature for affine transformation prediction.

In addition to the aforementioned issues, model update is also important since visual object tracking is a on-line process. Fixed model update rate is not suitable for all tracking situations, and as we know, there is still no good approach to address this problem. Here the paper tries to handle it by adaptively inferring the update rate from deep feature. In Fig.3, Long Short-Term Memory (LSTM) model is applied to map the Resnet50 feature to the update rate. With the help of end-to-end training, the update rate is automatically determined by the current scene and historical target states.

The proposed method follows a two-step scheme that predicts the coarse translation first and then the affine transformation. The main reason is that direct affine transformation prediction is difficult when the target movement is large. The success of coarse-to-fine scheme in TLD [26] and CTFT [27] demonstrates the benefits of this approach.

The main contributions of this paper are three-fold:

1) The paper proposes a novel network structure to map well-designed features to target affine transformations.

2) The paper improves the traditional correlation filter model with adaptive model update rate generation.

3) The proposed method addresses the main problem of correlation filter model and can successfully track the object with complex affine transformations.

II. RELATED WORK

Since the proof of high efficiency and performance of correlation filter by MOSSE [22] and KCF [1], many variants of it have been proposed to handle tracking problems. Sparse representation based trackers have low speed because of $L1$ norm in the optimization equation. CST [28] reduces the computational cost by introducing circulant data structure into the sparse representation and addressing the optimization in the Fourier Domain. Traditional tracking precision is limited to single-resolution in the feature map. C-COT [29] addresses this problem by learning continuous convolution filters with the interpolation model. ECO [10] follows this work and further improves its performance by two innovations. The first one is a factorized convolution operator that reduces the number of filters by removing those with low energy. The second innovation is to represent the training templates as a Gaussian mixture model which alleviates the over-fitting problem. Because of the existence of background clusters, learned correlation filters may focus on unexpected background regions. DRT [30] addresses this problem by modeling the filter as element-wise product of a base filter and a reliability term. The reliability term encourages the final filter to focus on more reliable regions. Similar thoughts occur in attention based feature extraction methods in RASNet [31] and DARL [32]. They try to learn weight matrix generation nets that catch the attention region of object. The number of training samples is important for filter learning. Following this instruction, RPCF [23] adopts the ROI pooling method from Fast R-CNN [33] and applies it to increase the sample size of correlation filter learning. MDIAAN [37] and ATOM [39] both introduce IoU-Net into the correlation filter tracking model. The IoU-Net is widely used for target localization in object detection approaches. The difference is that ATOM uses IoU-Net for region proposal generation and MDIAAN uses it for final object localization.

III. METHOD

The proposed method consists of two tracking steps, coarse translation and affine prediction. The first step follows the traditional correlation filter tracking mode [1]. A common problem in this kind of method is the fixed update rate of tracking model. In reality, target appearance changes in different ways. Low update rate will not catch up with rapid appearance change and high update rate is not suitable for some challenging scenes, such as occlusion. So a better way is to change the model update rate according to the tracking scene. Here, LSTM model is adopted for generating adaptive ratio for model update. For more details, see section III-B.

The second step predicts affine transformation by mapping the output of aligned filter model and log-Polar model to a 1 × 6 affine parameter. The training samples for on-line filter model update are transformed by the predicted affine parameter. In this way, each training sample has the same pose. The aligned filter model provides information of translation, shear, and aspect ratio change. While the log-Polar model provides information of scaling and rotation. Detailed explanations are presented in III-C.

The reason of applying two-step tracking scheme is that it is hard to directly estimate the affine transformation parameters when the target movement is large. The effect of the first step is like coarse alignment and makes the second step easier to success.

A. PRIOR KNOWLEDGE

1) CORRELATION FILTER

Correlation filter based trackers measure similarities between the search region and target region by correlation operation. Maximum point on the detection output generated by the correlation operation is taken as the translation of object between successive frames. Different correlation filter methods design corresponding optimization equations for specific purposes. Basic correlation filter optimization equation is

$$\min_{w} ||x \star w - y||_F^2 + \lambda ||w||_F^2, \tag{1}$$

where w is the optimized filter. x and y are the target feature and pre-designed ground truth. The symbol $\star$ denotes correlation operation. The subscript F denotes Frobenius norm.

According to Parseval's theorem, the Frobenius norms of a matrix before and after Fourier transform are equal. So the optimization equation (1) can be transformed to

$$\min_{\hat{w}} ||\hat{x} \circ \hat{w^*} - \hat{y}||_F^2 + \lambda ||\hat{w}||_F^2, \tag{2}$$

where the hat denotes the Discrete Fourier Transform. The symbol $*$ denotes conjugation. The symbol $\circ$ denotes element-wise multiplication. Since (2) is convex differentiable with respect to $\hat{w}$, a closed form expression for the optimal $\hat{w}$ is found by setting partial derivative of (2) to zero. It is as follows:

$$\hat{w} = \frac{\hat{x} \circ \hat{y^*}}{\hat{x} \circ \hat{x^*} + \lambda}. \tag{3}$$

The final detection output is

$$f(z) = \mathcal{F}^{-1}(\hat{z} \circ \hat{w^*}), \tag{4}$$

where z is the feature of search region. In order to handle border effect, all features are multiplied by Hanning windows.

KCF tracker [1] maps the input feature to a non-linear feature space with the kernel trick, and generates the filter model

$$f(z) = \mathcal{F}^{-1}(\hat{k}^{xz} \circ \hat{\alpha}), \tag{5}$$

where

$$\alpha = \frac{\hat{y^*}}{\hat{k}^{xx} + \lambda}. \tag{6}$$

Here, k^{xz} is the kernel matrix between x and z.

2) AFFINE TRANSFORMATION

The movement of target object in the imaging plane can be taken as some kind of affine transformation. As is shown in Fig.2, basic affine transformations consist of translation, scaling, aspect ratio change, rotation, shear, and reflection. It takes the form of

$$g(p) = Ap + d, \tag{7}$$

$$\begin{array}{ccc} \text{Translation} & \text{Scaling} & \text{Aspect ratio change} \\ \begin{bmatrix} 1, & 0, & dx \\ 0, & 1, & dy \end{bmatrix} & \begin{bmatrix} s, & 0, & 0 \\ 0, & s, & 0 \end{bmatrix} & \begin{bmatrix} r, & 0, & 0 \\ 0, & 1, & 0 \end{bmatrix} \\ \text{Shear} & \text{Reflect} & \text{Rotation} \\ \begin{bmatrix} 1, & c, & 0 \\ d, & 1, & 0 \end{bmatrix} & \begin{bmatrix} \text{-1(1)}, & 0, & 0 \\ 0, & 1(-1), & 0 \end{bmatrix} & \begin{bmatrix} \cos(\theta), & \text{-}\sin(\theta), & 0 \\ \sin(\theta), & \cos(\theta), & 0 \end{bmatrix} \end{array}$$

FIGURE 2. Basic affine transformations.

where $p = [x, y]$ is a pixel coordinate in image. $[A, d]$ is the 2 × 3 affine parameters.

Affine transformation has been used in classification [2] and object detection [3] for image alignment.

B. COARSE TRANSLATION

Fig.3 displays the filter block of this paper. In our method, the update ratio of the filter model (5) is generated by the LSTM model.

$$\begin{aligned} \tilde{\alpha}_t &= (1 - l_t)\tilde{\alpha}_{t-1} + l_t \alpha_t, \\ \tilde{x}_t &= (1 - l_t)\tilde{x}_{t-1} + l_t x_t, \end{aligned} \tag{8}$$

where l_t is the update ratio.

Compared to traditional correlation filter methods with fixed update ratios, the proposed method is more flexible and easier to adapt to target appearance change.

The correlation operation between the filter and feature of the search region outputs a hot map. The maximum point of this map is taken as the translation vector $d1$.

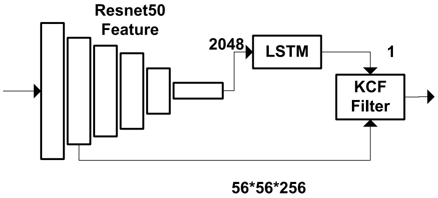


FIGURE 3. Filter block in Fig.1. LSTM net generates the rate to update the KCF filter model. The input of LSTM comes from the last convolution block of Resnet50. The input feature of the KCF filter comes from the second convolution block of Resnet50.

C. AFFINE PREDICTION

Among the six basic affine transformations, the proposed method focuses on translation, scaling, aspect ratio change, shear, and rotation, since common target object movement is a combination of them. Fig.1 shows the affine prediction module. After the coarse translation step, the feature of the search region is translated and then input to the affine prediction module. The translated search region is now close to the target region and makes it easier to predict the affine transformation. The affine prediction module consists of two feature extraction blocks and a mapping block.

The first feature extraction block is also a correlation filtering process. The difference is that the target feature template used for updating filters is aligned by the affine transformation $[A_{t-1}, 0]$ of previous moment. So all target feature templates have the same pose with the template of the first frame. Also the search region is aligned by the affine transformation $[A_t^*, 0]$, which is generated by the affine estimation block. The motivation of all the alignments is to make the search region and filter have the same pose. So the filtering result contains the information of target translation, shear, and aspect ratio change. The same correlation filter method with the coarse translation module is used for feature extraction. We found that the detection output has some specific patten when aspect ratio change or shear occurs. Fig.4 displays this phenomenon.

The second feature extraction block transforms the feature of the object in first frame and the search region into Log-Polar space. The Log-Polar transformation is

$$\begin{aligned} \rho &= log_a \frac{a^{N-1}}{R}\sqrt{i^2+j^2}, \\ \theta &= arctan(\frac{y}{x}), \end{aligned} \tag{9}$$

where R is the radius of minimum circumscribed circle outside the target bounding box. $a = 1.02$ is the scaling step. And the inverse transformation is

$$\begin{aligned} r &= a^{\rho}\frac{R}{a^N}, \quad \rho = 0, 1, \ldots, N-1; \\ i &= rcos(\frac{2\pi}{K}\theta), \quad \theta = 0, 1, \ldots, K-1; \\ j &= rsin(\frac{2\pi}{K}\theta). \end{aligned} \tag{10}$$

The origin $H \times W$ feature is transformed to a $K \times N$ Log-Polar feature. It is interesting to find that the Log-Polar feature is right shifted by one pixel when the target scale increases by $a - 1$. In the same way, when the target rotates by $2\pi/K$, the Log-Polar feature is vertical shifted by one pixel. So the Log-Polar space contains the information of rotation and scaling. The correlation operation between the Log-Polar features of the target in first frame and the search region is performed. In order to avoid border effect, both Log-Polar features are multiplied by Hanning windows along the ρ axis.

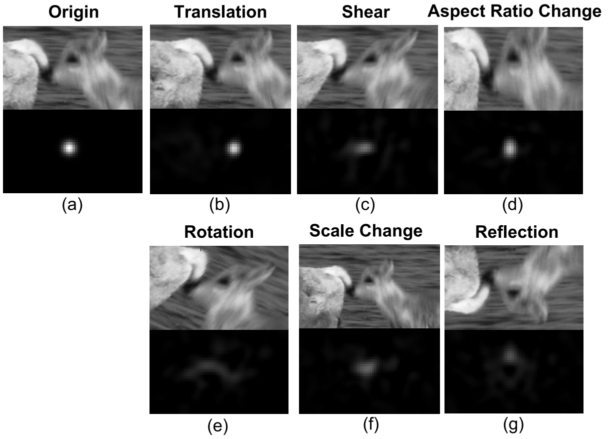


FIGURE 4. The filtering results about different affine transformations. On the upper images with target translation, shear, and aspect ratio change, the filtering results show specific patterns that the central highlights undergo similar affine transformations with the object. While on the lower images with target rotation, scale change, and reflection, the filtering results are chaotic. It means that the correlation filtering output contains information of these affine transformations.

Then the outputs of two correlation operations are flattened and concatenated to become a one-dimensional feature. This feature contains the information of translation, scaling, aspect ratio change, shear, and rotation. It is input to three fully connected layers (Fc), and the output is a 1×6 affine transformation parameter. This parameter is reshaped to be a 2×3 matrix $[A, d2]$. The final affine transformation parameter is $[A, d1 + d2]$.

D. OFF-LINE TRAINING

1) LOSS

There are five losses in the proposed network. The first one is a MSE loss between the detection output of coarse translation module and ground truth,

$$l_1 = \frac{1}{W * H}\sum_{i}^{W-1}\sum_{j}^{H-1}(M_{i,j} - max(M)G_{i-dx,j-dy})^2, \tag{11}$$

where M is the detection output and G is a Gaussian-shaped ground truth. $[dx, dy]$ is the ground truth translation vector.

The second loss is a $L2$ loss between the predicted translation and ground truth,

$$l_2 = ||d1 - d_{xy}||_F^2. \tag{12}$$

The third loss is a cropped MSE loss between the target feature of first frame and the search region after affine transformation,

$$l_3 = \frac{1}{W * H * C} \sum_{c}^{C-1} \sum_{i}^{W-1} \sum_{j}^{H-1} P^2(g(F^s)_{i,j,c} - F^0_{i,j,c})^2, \quad (13)$$

where P is the cropping matrix whose none-zero area has the same size with the target object. F^s is the feature of search region.

The forth loss is a $L2$ loss between the final predicted translation and ground truth,

$$l_4 = ||d1 + d2 - d_{xy}||_F^2. \quad (14)$$

The fifth loss is a Iou loss between the output bounding box and ground truth.

$$l_5 = Iou(b_t, b_t^{gt}). \quad (15)$$

2) THREE-STEP TRAINING

The whole off-line training process consists of three steps. (i) Fix the parameters of Resnet50, and train the coarse translation module with l_1 and l_2. (ii) Fix the parameters of coarse translation module and Resnet50, and train the affine prediction module with l_3, l_4, and l_5. (iii) Train the whole network with l_1, l_3, l_4, and l_5.

E. ON-LINE TRACKING

The on-line tracking process consists of forward affine parameter prediction and model update. The first step predicts the affine parameter and transforms the target bounding box of the first frame to a current one. The search region for next frame is $3\sqrt{WH} \times 3\sqrt{WH}$, where $H \times W$ is the size of axis-aligned circumscribed rectangle of the current target bounding box. Then the filter parameters on (8) are updated by the current target features.

IV. EXPERIMENTS

The proposed method is implemented on python with Tensorflow. All training and evaluation are executed on a Intel(R) Core(TM) I7-6900K CPU @ 3.2GHz and a NVIDIA TITAN Xp GPU.

A. TRAINING AND EVALUATION DATASETS

1) TRAINING

The feature extraction network Resnet50 was pre-trained on ImageNet2012 [4] for classification tasks. The proposed network was trained on ImageNet2015 [4], with Adam Optimizer. The exponential decay rate for the 1st and 2nd moment estimates are 0.9 and 0.999 for all training steps. The learning rate is set to 0.001 for training step (i) and (ii), and 0.0001 for training step (iii). Training step (i) runs for 10 epochs with mini-batches of size 32. Training step (ii) runs for 20 epochs with mini-batches of size 32. Training step (iii) runs for 40 epochs with mini-batches of size 24.

The input of the network is obtained by reshaping the extracted image region to the size of 224 × 224 × 3. The size of the search region is $3\sqrt{WH} \times 3\sqrt{WH}$ supposing the object size is $H \times W$. The sizes of features are 56 × 56 × 256 (from Resnet50 conv2) for the correlation filters and 112 × 112 × 64 (from Resnet50 conv1) for the Log-Polar transformation. The size of the Log-Polar feature is 90 × 40 × 64. In the filter block, the number of hidden units in LSTM is 20, and a Fc layer is connected to the LSTM to output the update ratio. In the affine estimation block, the numbers of hidden units in the two LSTM are 40 and 20 correspondingly, and also one Fc layer is connected to each of them. The sizes of outputs in the final three Fc layers are 24, 300, and 6. Tanh activation function is added to the first two Fc layers.

2) EVALUATION

Since the proposed method focuses on the single object short-term tracking problem. The evaluation datasets are OTB100 [5], UAV123 [7], VOT2018 [6] and VOT2020 [40]. OTB100 contains 100 videos with 11 attributes, *e.g.* illumination variation, out-of-plane rotation, and scale variation. On this dataset, the performance of a tracker is evaluated by two criteria that focus on central location error and overlap ratio. The first one is distance precision (DP) which is the percentage of frames where the central location error between tracking result and ground truth is below 20. And the second one is area under curve (AUC) which is the average value of the success rate curve.

The second dataset UAV123 evaluates the tracking performance on unmanned aerial vehicle scenes. It contains 123 sequences with 12 attributes. The evaluation criteria of it are the same with OTB100's.

VOT2018 contains 60 challenging public videos. This challenge has been held every year since 2013, and a lot of trackers are tested on it. A difference of testing process between VOT2018 and the other two datasets is that VOT2018 resets the tracker with the ground truth when a tracking failure occurs. And the number of tracking failures is used for measuring the robustness of a tracker. The criteria used for comparing tracking performance are Expected Average Overlap (EAO), Accuracy(A) and Robustness(R). A summary of all the metrics is showed on Table 1.

TABLE 1. Metrics.

Metrics	Description
DP ↑	Distance precision. A metric of central distance error.
AUC ↑	Area under curve. Average overlap between the results and ground truth.
EAO ↑	Expected average overlap.
A ↑	Accuracy. The average overlap between the periods of successful tracking.
R ↓	Robustness. The number of times the tracker is reset.

B. ABLATION STUDIES

Table 2 displays the ablation results of the proposed tracker. Our filter model is adapted from KCF, so we want to find out how much improvement is obtained by adding adaptive model update rate to it. Our coarse translation module outperforms the baseline KCF by 4.5% and 5.7% on OTB100 and UAV123.

Since the correlation filter can only handle target translation, our original intention is to adapt it to predict translation,

TABLE 2. Ablation results.

Method	Speed (FPS)	OTB100 (AUC)	UAV123 (AUC)
KCF	148	0.478	0.331
Coarse translation	52.4	0.523	0.388
Affine prediction	32.4	0.602	0.498
Coarse translation + Affine prediction(conv2)	20.2	0.672	0.611
Coarse translation + Affine prediction(conv1)	20.0	0.690	0.627

scale change, rotation, shear, and aspect ratio change in the meantime. The proposed method is inspired by the spatial transformer networks [2]. We use the Log-Polar transformation and correlation filter to extract related features for affine transformation prediction. The performance of the affine prediction module is better than the coarse translation module, with gains of 7.9% and 11.0% on the two datasets. We analyzed the performance of this method on every video and found that it is bad on those sequences with fast target motion.

It turns out that direct rotation matrix prediction is difficult if the target movement between adjacent frames is large. So we decided to construct a two-step tracking framework with a coarse translation prediction module and a fine affine prediction module. The performance is improved by the combination.

In our Log-Polar transformation (10), the sampling step in the Cartesian coordinate system is exponentially increasing. So the feature size is important cause it will affect the cartesian pixel distance of two neighbor Log-Polar pixels. Conv2 feature ($56 \times 56 \times 256$) and conv1 feature ($112 \times 112 \times 64$) of Resnet50 are used for evaluation. It turns out that conv1 feature is more suitable for the Log-Polar transformation part.

The influence of hyper parameters variation on the tracking performance was tested since some tracking methods are sensitive to parameters variation. The results are showed on Table 3. In the proposed approach, the hyper parameters of the correlation filter block are the same as the parameters of the baseline tracker KCF for fair comparison. For the scaling step a of the Log-polar transformation, the best accuracy on OTB100 is achieved when the value of a is 1.020. On UAV123, 1.015 and 1.020 are both suitable values for the scaling step. When this parameter ranges from 1.010 to 1.030, the performance changes of DP and AUC are below 5%. K and N are the other two hyper parameters in our method, they represent the width and height of the Log-Polar feature. When their values are changed from 80 to 100 and 30 to 50 with a step of 5, the maximum performance gaps of DP and AUC are 0.014 and 0.008. The experimental results prove that the proposed method is robust to hyper parameters variation and does not need to be excessively fine-tuned in various scenarios.

C. COMPARISON WITH CORRELATION FILTER BASED METHODS

1) OTB100

On OTB100, the proposed method is compared with seven correlation filter based trackers, including TADT [11], UDT [12], ECO [10], Staple [13], CFNet [14], SiamFC [15], and KCF [1]. Fig.5 shows the precision and success plots. Among all correlation filter based trackers, our ACT is the best one with DP of 90.3 and AUC of 69.0. ECO achieves competitive performance with the proposed method, but the speed of ECO is 6.5 frames per second (FPS), which is slower than ours (20 FPS).

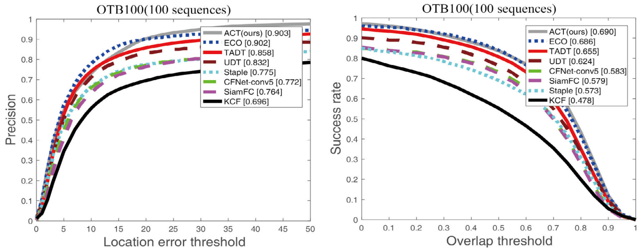


FIGURE 5. Precision and success plots of OTB100. All the compared trackers are based on correlation filter. The trackers are ranked by DP and AUC.

Common correlation filter based trackers only predict translation and scale change of the object. They are easy to fail on videos with other affine transformations. As mentioned before, the proposed method tries to infer the affine parameters of translation, scale change, rotation, shear, and aspect ratio change from the designed features. The performance on videos with these attributes should not be bad. Fig.6 shows the precision and success plots on videos with scale variation, deformation, in-plane rotation, and out-of-plane rotation. ACT performs best on tracking sequences with these attributes. Especially on videos with deformation, ACT outperforms the second best tracker ECO by 6.8% on DP and 3.3% on AUC.

The main reason of our better results on these test videos is that traditional correlation filter models lack the ability to predict other target affine transformations than translation. On videos with target scale change, the other compared trackers adopt scale pyramid to search for the target scale after object localization, but they ignore the other affine transformations. The scale change may be caused by rotation, shear, and aspect ratio change since these methods use axis-aligned circumscribed rectangles. The proposed ACT handles this problem successfully by predicting the target affine transformations simultaneously. By transforming the target features into the Log-Polar space, the target scale change and rotation are reflected by the change of Log-Polar features. That is why the proposed method can handle complex target appearance changes.

D. STATE-OF-THE-ARTS COMPARISON

1) UAV123

On UAV123, eleven trackers are used for comparison, including MDIAAN [37], HROM [38], SiamRPN++ [8], SiamR-cnn [9], ECO [10], SRDCF [16], MEEM [17], SAMF [18], DSST [19], Struck [20], and KCF [1]. Fig.7 and Table 4 show the experimental results on UAV123. ACT achieves better DP

TABLE 3. Influence of hyper parameters variation. DP and AUC are adopted to measure the tracking accuracy. *a* is the scaling step of the Log-Polar transformation. *K* and *N* are the width and height of the Log-Polar feature. The proposed tracker is robust to hyper parameters variation.

	Hyper Parameters	a					K					N				
	value	1.010	1.015	1.020	1.025	1.030	80	85	90	95	100	30	35	40	45	50
OTB100	DP	0.894	0.901	0.903	0.895	0.887	0.893	0.899	0.903	0.900	0.896	0.889	0.899	0.903	0.897	0.895
	AUC	0.681	0.685	0.690	0.681	0.677	0.683	0.686	0.690	0.688	0.685	0.677	0.686	0.690	0.685	0.680
UAV123	DP	0.815	0.823	0.824	0.819	0.810	0.815	0.819	0.824	0.822	0.818	0.812	0.817	0.824	0.819	0.815
	AUC	0.619	0.628	0.627	0.619	0.615	0.621	0.625	0.627	0.626	0.622	0.619	0.622	0.627	0.623	0.620

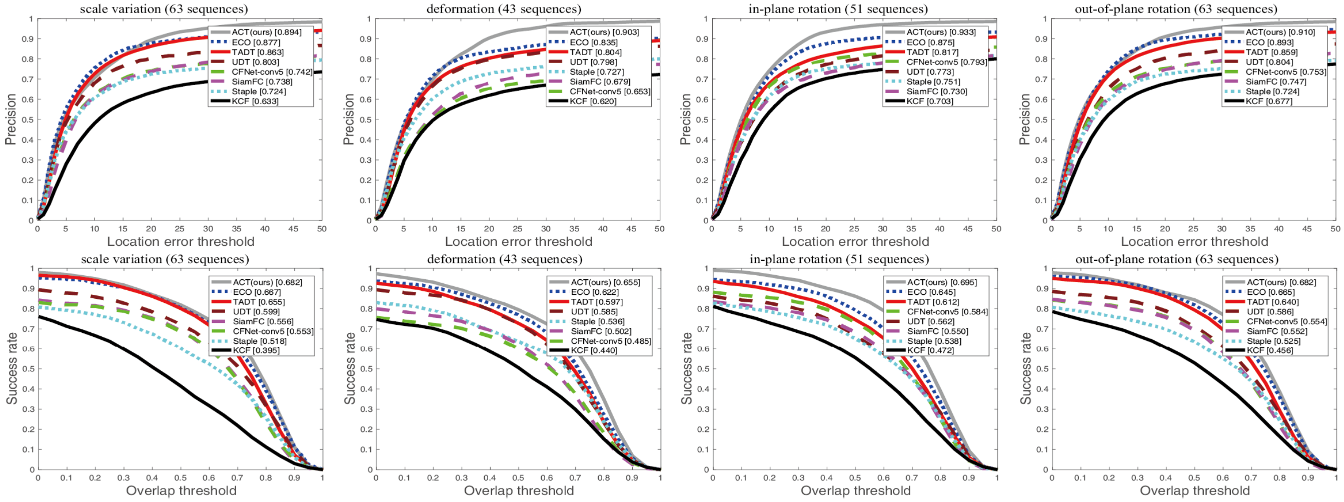


FIGURE 6. Attribute-based evaluation on OTB100. Four attributes are used, including scale variation, deformation, in-plane rotation, and out-of-plane rotation. The proposed tracker performs best on videos with these attributes.

TABLE 4. Results on UAV123.

UAV123	KCF	DSST	Struck	MEEM	SAMF	SRDCF	ECO	SiamRPN++	SiamRcnn	MDIAAN	HROM	ACT(ours)
DP	0.523	0.586	0.578	0.627	0.592	0.676	0.741	0.840	0.834	0.821	0.834	0.824
AUC	0.331	0.356	0.381	0.392	0.396	0.464	0.525	0.642	0.649	0.610	0.636	0.627

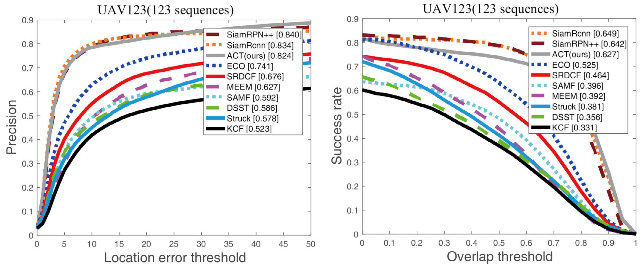


FIGURE 7. Precision and success plots of UAV123.

and AUC than the correlation-based tracker ECO by gains of 8.3% and 10.2%. ECO is one of the best correlation filter based trackers in recent years. It achieves worse performance on UAV123 than OTB100. One of the reasons may be that it lacks off-line training with large data. MDIAAN combines Discriminative Correlation Filter (DCF) and instance-aware IoU-Net into a two-stage tracking scheme. Our ACT performs slightly better than this method on both DP and AUC. HROM is based on SiamRPN which is the precursor of SiamRPN++. These siamese tracking approaches performs joint classification and regression with region proposal networks. HROM, SiamRPN++, and SiamRcnn achieve better performance than ACT.

2) VOT2018

On VOT2018, ten state-of-the-art methods are compared, including ATOM [39], SiamRPN++ [8], SiamRcnn [9], ECO [10], LADCF [34], MFT [36], UPDT [35], RCO [6], SiamFC [15], and KCF [1]. ATOM decomposes the tracking problem into a target estimation module (region proposal generation) and a target classification module. IoU predictor and correlation filter model are used for the two modules. Table 5 shows the experimental results on VOT2018. SiamRPN++ achieves the best EAO of 0.417, while its accuracy and robustness are lower than our ACT. SiamRcnn owes the highest accuracy of 0.609 and has better EAO than ACT. However, our ACT is more robust and faster than SiamRcnn, since the speed of SiamRcnn is 2.9 FPS. In the meantime, the sizes of training data for SiamRcnn and SiamRPN++ are much bigger than ours. The proposed tracker should achieve higher performance with more training data.

3) VOT2020

On VOT2020, nine tracking approaches are compared, including RPT [41], OceanPlus [42], AlphaRef [43], AFOD [40], LWTL [44], UPDT [35], ATOM [39], SiamFC [15], and KCF [1]. Among them, RPT, OceanPlus,

TABLE 5. Results on VOT2018.

VOT2018	KCF	SiamFC	RCO	UPDT	MFT	LADCF	ECO	SiamRPN++	SiamRcnn	ATOM	ACT(ours)
EAO	0.135	0.188	0.376	0.378	0.386	0.389	0.281	0.417	0.408	0.401	0.401
Accuracy	0.447	0.498	0.505	0.530	0.501	0.502	0.476	0.596	0.609	0.590	0.602
Robustness	0.773	0.585	0.155	0.184	0.140	0.159	0.276	0.220	0.234	0.204	0.178

TABLE 6. Results on VOT2020.

VOT2020	KCF	SiamFC	ATOM	UPDT	LWTL	AFOD	AlphaRef	OceanPlus	RPT	ACT(ours)
EAO	0.154	0.179	0.271	0.278	0.463	0.472	0.482	0.491	0.530	0.479
Accuracy	0.407	0.418	0.462	0.465	0.719	0.713	0.754	0.685	0.700	0.708
Robustness	0.432	0.502	0.734	0.755	0.798	0.795	0.777	0.842	0.869	0.855

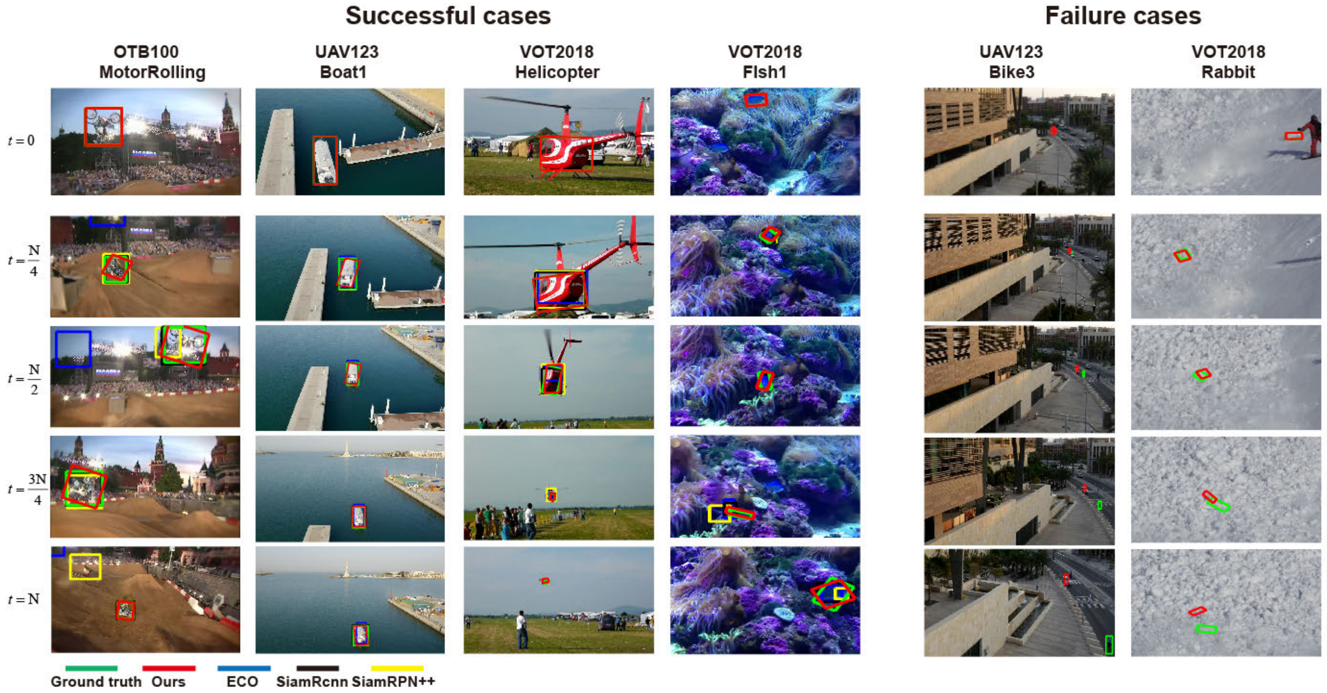


FIGURE 8. Tracking results on six videos. The compared methods are ECO, SiamRcnn, and SiamRPN++. Our ACT performs well on the first four sequences. The last two videos are failure cases.

AlphaRef, AFOD, and LWTL are top five trackers reported in the VOT2020 paper [40]. Table 6 shows the comparative results on VOT2020. RPT achieves the best EAO and robustness, and AlphaRef achieves the best accuracy. The proposed tracker achieves the second best robustness and the other two criteria, EAO and accuracy, are competitive with the other compared methods.

4) QUANTITATIVE ANALYSIS

Fig.8 shows the tracking results of four comparing methods on six challenging videos. In the video named MotorRolling, the target object experiences a lot of in-plane rotation. SiamRcnn and our ACT perform well on this video. Moreover, tracking window of the proposed tracker can adapt to the rotation of the object. In the video Helicopter of VOT2018, the trackers are reset if tracking failures occur. All the other three trackers are reset once or more in this video, while the proposed ACT successfully tracks the object from the beginning to the end. In the video named Fish1, the target fish undergoes complex affine transformations and occlusion, and its color is similar with the background color. The occlusion happens at $t = \frac{3N}{4}$. Before that, the object undergoes a lot of affine transformations. Our method handles this condition by the Affine Prediction module. During the occlusion, the tracking bounding boxes drift slightly from the ground truth boxes because most parts of the target features are replaced by the background features. After the occlusion, the object returns to the normal condition and the proposed method can successfully track the object again since the target features of the first frame are used for the affine transformation prediction.

The small target size (the initial size is 17 × 10) is the main cause of our tracking failures in the video Bike3. Although all inputs to the network are resized to the same size, the feature

TABLE 7. Computational efficiency comparison by the metric of frame per second (FPS). The symbol $*$ means the calculating speeds are reported in the their papers. Other results are all calculated on the same hardware.

	KCF	SiamFC	CFNet	Staple	UDT	TADT	SAMF	DSST	Struck
FPS	148.0	55.4	45.4	89.2	56.2	37.3	4.7	19.3	17.2
	MEEM	ECO	SRDCF	MDIAAN	HROM	RCO	UPDT	MFT	LADCF
FPS	9.7	6.5	5.2	25.6*	40.0*	2.7	1.4	2.5	19.2
	ATOM	SiamRPN++	SiamRCNN	LWTL	AFOD	AlphaRef	Oceanplus	RPT	ACT(ours)
FPS	33.2	34.6	2.9	12.4	63.8	62.4	66.8	18.3	20.0

of a small object is not discriminative enough to distinguish the object from the backgrounds. In the video named Rabbit, the color of the target rabbit is similar with the background color and the object is frequently occluded by the snow, so tracking failures occur after frame 100.

5) COMPUTATIONAL EFFICIENCY

Table 7 shows the calculating speeds of all the compared trackers. KCF, Staple, SAMF, DSST, Struck, MEEM, and SRDCF run on CPU, while the other tracking methods run on GPU. The most efficient approach is KCF which accelerates the solution of kernelized correlation filters by Discrete Fourier Transform (DFT). DFT is the reason of high efficiency for correlation filter based trackers since it reduces the time complexity of matrix inversion from $O(n^3)$ to $O(nlogn)$. The proposed approach runs at 20 FPS with no code optimization and the computational complexity is 14.1G FLOPs.

V. CONCLUSION

This paper addresses the issue of traditional correlation filter based trackers that they cannot predict target rotation, aspect ratio change, shear, and scale change. The proposed method predicts coarse target translation and affine transformation parameters in a two-step scheme. It is found that the information of target affine transformations is hidden in the correlation filtering output. In order to predict the rotation and scale change, Log-Polar features are adopt and turn out to be useful. The mapping function is learnt by the off-line training process and makes it possible to predict complex target transformations. One direction for further work is to add tracking failure detection into the network and it will be helpful for long-term tracking scenarios.

REFERENCES

[1] 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.

[2] M. Jaderberg, K. Simonyan, and A. Zisserman, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 28, Dec. 2015, pp. 2017–2025.

[3] X. Wang, A. Shrivastava, and A. Gupta, "A-fast-RCNN: Hard positive generation via adversary for object detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2606–2615.

[4] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[5] 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.

[6] M. Kristan *et al.*, "The sixth visual object tracking VOT2018 challenge results," in *Proc. Eur. Conf. Comput. Vis. (ECCV), Workshops*, Sep. 2018, pp. 3–53.

[7] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for UAV tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2016, pp. 445–461.

[8] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN++: Evolution of Siamese visual tracking with very deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4282–4291.

[9] P. Voigtlaender, J. Luiten, P. H. S. Torr, and B. Leibe, "Siam R-CNN: Visual tracking by re-detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6578–6588.

[10] 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.

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

[12] N. Wang, Y. Song, C. Ma, W. Zhou, W. Liu, and H. Li, "Unsupervised deep tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1308–1317.

[13] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1401–1409.

[14] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. Torr, "End-to-end representation learning for correlation filter based tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017*, pp. 2805–2813.

[15] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional Siamese networks for object tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2016, pp. 850–865.

[16] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4310–4318.

[17] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: Robust tracking via multiple experts using entropy minimization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2014, pp. 188–203.

[18] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2014, pp. 254–265.

[19] M. Danelljan, G. Hager, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Nottingham, U.K., Sep. 2014, pp. 1–11. [Online]. Available: http://www.bmva.org/bmvc/2014/papers/paper038/index.html

[20] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S. L. Hicks, and P. H. S. Torr, "Struck: Structured output tracking with kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2096–2109, Oct. 2016.

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

[22] 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. (CVPR)*, Jun. 2010, pp. 2544–2550.

[23] Y. Sun, C. Sun, D. Wang, Y. He, and H. Lu, "Roi pooled correlation filters for visual tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5783–5791.

[24] A. He, C. Luo, X. Tian, and W. Zeng, "Towards a better match in Siamese network based visual object tracker," in *Proc. Eur. Conf. Comput. Vis. (ECCV), Workshops*, Sep. 2018, pp. 132–147.

[25] D.-H. Lee, "One-shot scale and angle estimation for fast visual object tracking," *IEEE Access*, vol. 7, pp. 55477–55484, Apr. 2019.

[26] 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.

[27] D. Li, G. Wen, and Y. Kuai, "Collaborative convolution operators for real-time coarse-to-fine tracking," *IEEE Access*, vol. 6, pp. 14357–14366, Feb. 2018.

[28] T. Zhang, A. Bibi, and B. Ghanem, "In defense of sparse tracking: Circulant sparse tracker," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3880–3888.

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

[30] C. Sun, D. Wang, H. Lu, and M. H. Yang, "Correlation tracking via joint discrimination and reliability learning," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 489–497.

[31] Q. Wang, Z. Teng, J. Xing, J. Gao, W. Hu, and S. Maybank, "Learning attentions: Residual attentional Siamese network for high performance online visual tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 4854–4863.

[32] S. Pu, Y. Song, C. Ma, H. Zhang, and M. H. Yang, "Deep attentive tracking via reciprocative learning," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Jun. 2018, pp. 1931–1941.

[33] R. Girshick, "Fast R-CNN," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1440–1448.

[34] T. Xu, Z.-H. Feng, X.-J. Wu, and J. Kittler, "Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual object tracking," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5596–5609, Nov. 2019.

[35] G. Bhat, J. Johnander, M. Danelljan, F. S. Khan, and M. Felsberg, "Unveiling the power of deep tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 483–498.

[36] S. Bai, Z. He, Y. Dong, and H. Bai, "Multi-hierarchical independent correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2020, pp. 1–6.

[37] S. Zhang, L. Zhuo, H. Zhang, and J. Li, "Object tracking in unmanned aerial vehicle videos via multifeature discrimination and instance-aware attention network," *Remote Sens.*, vol. 12, no. 16, p. 2646, Jul. 2020.

[38] D. Zhang, Z. Zheng, T. Wang, and Y. He, "HROM: Learning high-resolution representation and object-aware masks for visual object tracking," *Sensors*, vol. 20, no. 17, p. 4807, Aug. 2020.

[39] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "Atom: Accurate tracking by overlap maximization," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4660–4669.

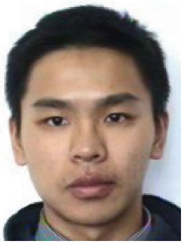
[40] M. Kristan, J. Matas, A. Leonardis, and M. Felsberg, "The eighth visual object tracking VOT2020 challenge results," in *Proc. 16th Eur. Conf. Comput. Vis. Workshop*, Aug. 2020, pp. 547–601. [Online]. Available: https://prints.vicos.si/publications/384/the-eighth-visual-object-tracking-vot2020-challenge-results

[41] Z. Ma, L. Wang, H. Zhang, W. Lu, and J. Yin, "RPT: Learning point set representation for Siamese visual tracking," Aug. 2020, *arXiv:2008.03467*. [Online]. Available: http://arxiv.org/abs/2008.03467

[42] Z. Zhang, H. Peng, J. Fu, B. Li, and W. Hu, "Ocean: Object-aware anchor-free tracking," Jun. 2020, *arXiv:2006.10721*. [Online]. Available: http://arxiv.org/abs/2006.10721

[43] B. Yan, X. Zhang, D. Wang, H. Lu, and X. Yang, "Alpha-refine: Boosting tracking performance by precise bounding box estimation," Dec. 2020, *arXiv:2012.06815*. [Online]. Available: http://arxiv.org/abs/2012.06815

[44] G. Bhat, F. J. Lawin, M. Danelljan, A. Robinson, M. Felsberg, L. Van Gool, and R. Timofte, "Learning what to learn for video object segmentation," in *Proc. 16th Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 777–794. [Online]. Available: https://arxiv.org/abs/2003.11540



WEIWEI ZHENG received the B.S. degree from Zhejiang University, in 2013. He is currently pursuing the Ph.D. degree in information science and electronic engineering. His main research interests include visual object tracking, video fire and smoke detection, and multi-object tracking.

HUIMIN YU (Member, IEEE) worked as the Director of the Department of Information and Electronic Engineering, Zhejiang University, and the Director of the Research Center on Network Media Cloud Processing and Analysis Engineering Technology of Zhejiang Province. He is currently a Professor and a Doctoral Supervisor with Zhejiang University, where he is also associated with the Zigong Innovation Center. He is active in the study of image/video intelligent processing and analysis, computer vision, and multimedia information processing. He has made internationally advanced research achievements in computer vision, multimedia information processing, and other related research fields. He has successively published articles in the top international academic conferences and journals in the field of artificial intelligence, such as CVPR, ICCV, AAAI, the IEEE TRANSACTIONS ON IMAGE PROCESSING, and *Pattern Recognition*.

ZHAOHUI LU received the B.S. degree from Zhejiang University, in 1999, and the M.S. degree from Xi'an Jiaotong University, in 2003. He is a Senior Engineer with the ZJU-League Research & Development Center and with Zhejiang Lijia Electronic Technology Company Ltd. His main research interests include video surveillance and intelligent traffic monitoring.