Template Features Guided Siamese Keypoints Detection for Visual Object Tracking

Ershen Wang*, Donglei Wang, Yufeng Huang, Pingping Qu, Tao Pang and Song Xu
School of Electronic and Information Engineering, Shenyang Aerospace University, Shenyang, China
*Corresponding author: ershenwang@sau.edu.cn

Abstract. Visual object tracking is one of the challenging tasks in computer vision. Among tracking tasks, accurately state estimation is the main challenge. Most current methods simply adopt multi-scale searches or anchors to estimate the state, which need many hyper-parameters and complex calculations. To face this challenge, we propose an anchor-free tracking framework based on Siamese network. The proposed framework consists of template features guided Siamese subnetwork and keypoints detection subnetwork. We take simplified hourglass network as backbone in Siamese subnetwork to improve the tracking efficiency and the prediction of corners around target instead of the prediction of target in keypoints detection subnetwork. We experiment our approach on OTB-2015 and VOT-2016, and our approach acquire the best precision of 0.841 on OTB-2015 and runs at 39 FPS.

1. Introduction
Research related to visual object tracking has become very popular in recent years, due to visual object tracking has widely application in video surveillance, human-computer interaction and unmanned vehicles [1]. It aims to stably estimate the state of target in subsequent frames only given the target position and size information in the initial frame. Recent years great progress has been made on visual object tracking, however achieving long-term stable target tracking is still a challenging task due to complex factors such as scale changes, rotation, illumination changes in the background, and interference from similar objects during the movement of the target [2].

In recent years, visual tracking methods mainly revolve around two aspects. One is based on discriminative correlation filter (DCF), and the other is based on Siamese network. The first category, thanks to the fast Fourier transform, MOSSE [3] has achieved high efficiency, but it has difficulty coping with complex scenes. With the development of deep learning, combining it with visual tracking improves the tracking accuracy of related filtering algorithms, but the speed is sacrificed.

The Siamese network based trackers have implemented the balance of accuracy and speed. SiamFC [4] as one of the pioneering works, it is essentially to study the similarity between template frame and search frame. CFNet [5] adds Correlation Filter layer to improve the speed without loss of accuracy. Accurately state estimation is a challenge in Siamese network based trackers. Many methods simply adopt a multi-scale search to estimate the target size. SiamRPN [6] and some improved methods around it introduces a region proposal network to effectively obtain an accurate target state estimation. However,
the design of anchors introduced in the region proposal need prior knowledge to define, which introduce many hyper-parameters and improve the computational complexity.

In this paper, we propose an anchor-free Siamese network based tracking framework to face the challenge of state estimation. Inspired by CornerNet [7], our framework gives up the design of anchors and adopts keypoints detection to predict the bounding box. Essentially, we track an object as the top-left corner and bottom-right corner of the bounding box. We achieve the classification and state estimation of target, by using template features guided Siamese network to predict the heatmaps, embeddings and offsets for the top-left corners and bottom-right corners of all instances.

Our main contributions are listed as follows:
- We introduce the corner detection network for tracking framework. The corner detection network needs no prior designed anchors, which avoids complicated hyper-parameters calculating.
- The design of template features guided Siamese network in our approach is used to better extract the features of the search frame to distinguish between foreground and background.
- In tracking, we adopt soft non-maximum suppression (Soft-NMS) [8] to suppress redundant boxes and find the optimal bounding box location to further improve tracking robustness.

2. Method

In the following we demonstrate the proposed method for tracking framework in detail. Figure 1 shows the main architecture of our network framework: the dotted box on the left is the Siamese subnetwork with template features guiding for feature maps extraction. Next to it is the keypoints prediction subnetwork, it consists of top-left branch and right-bottom branch.

![Figure 1. Main framework of our proposed tracking network](image)

2.1. Features guided Siamese Network

2.1.1. Backbone Network. In Siamese network, we adopt the hourglass network framework which in CornerNet-Saccade [19] as backbone network. Hourglass network [9] was first introduced for human pose prediction in computer vision task. It is stacked of 3 hourglass modules with 54 layers of depth. The hourglass module preserves the origin resolution by a series down-sample and up-sample processing. Figure 2 shows the architecture First-order hourglass module. We simply use stride 2 to down sample the feature maps. Before the first stage hourglass network, we apply a 5×5 convolution layer with stride 2 and a residual module to down sample the image feature resolution by 4 times.
2.1.2. Template Features guided Siamese Network. We design the features guided Siamese network to get better feature expression by exploiting the correlation. Specifically, given \( f() \) denotes the feature maps obtained through backbone, then we encode the feature maps of template to \( 1 \times 1 \times \text{Channel} \) ( \( \varphi(f(z)) \in R^{c \times C} \)). Finally, we get template features guided feature by calculating the correlation between the feature maps of search patch \( f(x) \) and \( \varphi(f(z)) \):

\[
f_o(x) = f(x) \otimes \varphi(f(z))
\]

Where \( \otimes \) denotes the correlation, \( f_o(x) \) denotes the correlation result.

2.2. Keypoints Prediction

2.2.1. Corners Prediction. Two corner prediction modules follow the backbone network. The corners which outside the objects often lack of local appearance features. We adopt corner pooling to solve this problem.

We compute the correlation on both the top-left branch and the bottom-right branch, then predict the heatmaps, embeddings and offsets as same as CornerNet [7]. During training, we adopt the focal loss \( \mathcal{L}_{det} \) to detect the corners for heatmap, the smooth L1 loss \( \mathcal{L}_{off} \) to predict the offsets between predict corners and ground-truth corner locations. We apply the “pull” loss \( \mathcal{L}_{pul} \) to group the corners and the “push” loss \( \mathcal{L}_{pus} \) to separate the corners between foreground and background.

2.2.2. Training Loss. We train the network end to end and optimize the full training loss:

\[
\mathcal{L} = \alpha \mathcal{L}_{det} + \gamma \mathcal{L}_{emb} + \lambda \mathcal{L}_{off}
\]

Where \( \mathcal{L}_{emb} = \theta \mathcal{L}_{pul} + \sigma \mathcal{L}_{pus} \) denotes the loss of embedding, \( \alpha, \beta \) and \( \lambda \) denote the weights for balance the full training loss.

2.3. Tracking Details

We adopt the Soft-NMS based on intersection over union (IoU) to remove redundant boxes and improve accuracy. Instead of the direct deletion of NMS algorithm, the core of Soft-NMS is setting an attenuation function to reduce confidence and has the same complexity as traditional NMS which is efficient. Simply, we set a list of detection boxes as \( B = \{b_1, b_2, \cdots, b_n\} \), denote their scores as \( S = \{s_1, s_2, \cdots, s_n\} \), and suppose the maximum score detection box is \( M \). If a currently determined detection box \( b_i \) overlaps with \( M \) for the most part, it will have a very low score; and if \( b_i \) with \( M \) only a small part, then its
original detection score will not be affected too much. We use Gaussian weighting as the attenuation function:

\[
S_i = \begin{cases} 
S_i & \text{IoU}(M, b_i) < N_t \\
\frac{\text{IoU}(M, b_i)}{\sigma}, & \forall b_i \not\in D \text{ IoU}(M, b_i) \geq N_t 
\end{cases} 
\]  

(3)

Where \( N_t \) denotes the threshold, this function will decay the score when IoU exceeds the threshold.

3. Experiments

3.1. Implementation Details

Our method is implemented using PyTorch1.0 with python3.7 and runs on NVIDIA Tesla V100. We apply the hourglass-54 network as the backbone network with no pretraining on any dataset, and the sample image pairs are picked from VID [10] and COCO [11] datasets to train the whole network.

We use stochastic gradient descent (SGD) to train the network with 20 epochs, use a warmup learning rate increasing from 0.001 to 0.005 in the first 5 epochs and a learning rate decays exponentially from 0.005 to 0.00005 for the last 15 epochs. By following SiamFC, we set the input size of template patches as \(127 \times 127\), the size of search patches as \(255 \times 255\).

3.2. Comparisons with the mainstream trackers

We comprehensively compare our approach with the state-of-the-art on OTB-2015 and VOT-2016 datasets. OTB2015: OTB2015 [12] is composed of 100 challenging videos and one of the most widely used visual object tracking benchmark. It has two evaluation metrics in one-pass evaluation (OPE), one is precision score (PS), it is the percentage of frames where the tracking result lies within 20 pixels of the ground-truth center. The other is area under curve (AUC) of success plot, this is the area under the plot which consists of ratios of successfully tracking frames at the thresholds ranged from 0 to 1. We compare our tracker with 9 mainstream trackers including SINT [14], CFNet [5], SiamFC [4], Staple [15], DSST [16], HCF [17] and KCF [18]. Figure 3 shows the precision and success plots. We obtain the best precision of 0.841 and the AUC of 0.613, compared with HCF, it improves the precision scores by 0.4%.

**Figure 3.** Comparation on OTB2015 with the evaluation metrics of precision and success plots in one-pass evaluation
Qualitative analysis results as shown in Figure 4, we compare our approach with SiamFC and SiamRPN which use Siamese network on three challenging sequences including CarScale, MotorRolling and Human6, and shows our approach can deal with complex scenarios well. In the sequence of CarScale, our approach can attack the problem of scale estimation due to the keypoints detection. To face the fast-moving small targets, our approach also shows excellent performance on the sequence of MotorRolling. The tracking results in Human6 verified that our method can distinguish blurred backgrounds.

**Figure 4.** Example tracking results of our approach with other Siamese network based trackers on three challenging sequences (from top to down: CarScale, MotorRolling and Human6.)

VOT2016: VOT2016 [13] are composed of 60 sequences. There are three commonly used evaluation metrics in VOT datasets: accuracy (A), robustness (R), and expected average overlap (EAO). We evaluate our tracker on VOT2016 and compared with representative trackers in VOT2016. Table 1 shows the compared results. Our tracker achieves the top EAO and the second accuracy, and the similar robustness as DNT in VOT2016.

**Table 1.** results on VOT2016 with the evaluation metrics of A↑, R↓, and EAO↑. Respectively, **Red**, **blue** and **green** denote 1st, 2nd, 3rd performance.

| Trackers | SiamFC | MDNet | DNT  | EBT  | Staple | C-COT | Ours |
|----------|--------|-------|------|------|--------|-------|------|
| VOT2016  |        |       |      |      |        |       |      |
| A↑       | 0.532  | 0.541 | 0.515 | 0.465 | 0.544  | 0.539 | 0.541|
| R↓       | 0.461  | 0.337 | 0.329 | 0.252 | 0.378  | 0.238 | 0.311|
| EAO↑     | 0.235  | 0.257 | 0.278 | 0.291 | 0.295  | 0.331 | 0.333|

4. Conclusion
In this paper, we propose a keypoints detection based Siamese network tracking framework. It requires no prior knowledge to design anchors. It directly predicts the classification and state of a pair of corners. By using template features guided Siamese network to get better feature expression and simplified Hourglass-54 as backbone to streamline our method. Extensive experiments on OTB2015 and VOT2016 verified that our method achieves excellent performance and runs at 39 FPS, which demonstrate the valid of our method.
Acknowledgments
This work was financially supported by The National Natural Science Foundation of China (61571309, 61703287); The Key R & D projects of Liaoning Province (2020JH2/10100045); The National Science Foundation of Liaoning Province (2019-MS-251); The Talent Project of Revitalization Liaoning (XL YC1907022); The High-Level Innovation Talent Project of Shenyang (RC190030); The second young and middle-aged talents support program of Shenyang Aerospace University fund.

References
[1] Smeluders A W M, CHU D M, Cucchiara R, et al. Visual tracking: an experimental survey[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 36(7):1442-1468. doi: 10.1109/TPAMI.2013.230.
[2] H. Yang, L. Shao, F. Zheng, L. Wang, Z. Song, Recent advances and trends in visual tracking: A review, Neurocomputing 74 (18) (2011) 3823–3831.
[3] D. S. Bolme, J. R. Beveridge, B. A. Draper, Y. M. Lui, Visual object tracking using adaptive correlation filters, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 2544–2550.
[4] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr. Fully-convolutional siamese networks for object tracking. In ECCV, 2016.
[5] J. Valmadre, L. Bertinetto, J. F. Henriques, A. Vedaldi, and P. H. Torr. End-to-end representation learning for correlation filter based tracking. In CVPR, 2017.
[6] B. Li, J. Yan, W. Wu, Z. Zhu, X. Hu, High performance visual tracking with siamese region proposal network, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2018, pp. 8971–8980.
[7] H. Law, J. Deng, CornerNet: Detecting objects as paired keypoints, in: European Conference on Computer Vision (ECCV), 2018, pp. 734–750.
[8] Bodla, Navaneeth, et al. "Improving Object Detection with One Line of Code." (2017).
[9] Newell, Alejandro, K. Yang, and J. Deng. "Stacked Hourglass Networks for Human Pose Estimation." European Conference on Computer Vision Springer, Cham, 2016.
[10] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. IJCV, 115(3):211–252, 2015.
[11] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In ECCV, 2014.
[12] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. TPAMI, 37(9):1834 –1848, 2015.
[13] Kristan M, Leonardis A, Jiri Matas, et al. The visual object tracking vot2016 challenge results. Proceeding of the European Conference on Computer Vision Workshops, 2016: 1-45.
[14] Tao R, Gavves E, Smeluders A W M. Siamese instance search for tracking. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016: 1420-1429.
[15] Bertinetto L, Valmadre J, Goloedetz S, et al. Staple: Complementary learners for real-time tracking. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016: 1401-1409.
[16] Danelljan M, Hager G, Khan F S, et al. Discriminative Scale Space Tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(8): 1561 - 1575.
[17] C. Ma, J. -B. Huang, X. Yang, and M. -H. Yang. Hierarchical convolutional features for visual tracking. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015: 3074-3082.
[18] Henriques J F, Caseiro R, Martins P, et al. High-speed tracking with kernelized correlation filters. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(3): 583-596.
[19] Law, Hei, et al. "CornerNet-Lite: Efficient Keypoint Based Object Detection." (2019).