LETTER

Combining Siamese Network and Regression Network for Visual Tracking

Yao GE†, Rui CHEN††, Ying TONG††, Xuehong CAO††, Nonmembers, and Ruiyu LIANG††, Member

SUMMARY We combine the siamese network and the recurrent regression network, proposing a two-stage tracking framework termed as SiamReg. Our method solves the problem that the classic siamese network can not judge the target size precisely and simplifies the procedures of regression in the training and testing process. We perform experiments on three challenging tracking datasets: VOT2016, OTB100, and VOT2018. The results indicate that, after offline trained, SiamReg can obtain a higher expected average overlap measure.

1. Introduction

Visual object tracking can save computing resources in various tasks of computer vision, such as automatic driving, video surveillance, and pose estimation. Compared with other vision problems like object detection and instance segmentation, visual tracking has less prior knowledge of the target, so it is not easy for an offline-trained model to attain excellent performance when tracks generic objects. Furthermore, the appearance of the target needs updating due to some factors like occlusions and illumination change. All these difficulties make visual tracking a challenging topic to research.

With the growth of deep learning, neural networks have shown show their powerful capabilities in this field. SiamFC[1] proposed a real-time tracking approach, converting the candidate matching problem to the similarity learning problem by using fully-convolutional siamese networks for the first time. The conciseness and expansibility of this structure inspire some methods with greater accuracy and robustness. In [2]–[5], the Region Proposal Network [6], which is widely used in the object detection task, was combined with the siamese network for better tracking performance. It was also proved that the features extracted by the siamese structure have enough information to produce the segmentation mask [7].

In most vision applications, regression networks, usually follow the modules consist of convolution layers and pooling layers, regroup the abstract features to calculate the final decision. GOTURN [8] fused pivotal features of two adjacent frames by fully-connected layers and can track at 100 fps. Re3 [9] replaced fully-connected layers with peephole connection LSTM [10] and joined skip-layer connection to capture information from different convolution layers. The rich features from appearance and sequence make Re3 output the corner of the bounding box more precisely than GOTURN [11].

We use LSTM to build a regression network and append it to the siamese network, presenting a two-stage tracking framework termed as SiamReg. The siamese part of our method determines the key area centered at the target, and then, the regressive part fixes the boundary by outputting the scale of the rectangular box. In the tracking process, SiamReg has the ability to take sequential features into account, which can give the precise regressive result more concisely. On the Visual Object Tracking (VOT) challenge datasets[12], SiamReg obtains a better Expected Average Overlap and Robustness measure than trackers solely relay on the siamese or regression network and keeps the high tracking speed.

The main contributions can be summarized as follows:

• We propose the siamese regression network (SiamReg) with a clear structure, which can directly address the invariance problem of the target scale in the classic siamese tracking framework.
• Through the combination of the siamese structure, the way we apply regression networks is more efficient than other methods.

2. Proposed Approach

As shown in Fig. 1, we consider visual tracking as a two-task problem: determining the center position of the target
and regressing the scale of the rectangular box surrounding the object. The centroid matching network is responsible for locating the point which represents the center of the area occupied by the target. Based on the picked point’s position, it is easy for the bounding regression network to make a judgment on which features are likely useful in the regression step for outputting height and width. SiamReg combines the outputs of two networks and completes the tracking process on the present frame.

2.1 Centroid Matching Network

It is convenient for CNN to focus on the center of the area with a potential object, rather than determines the object’s boundary. From this perspective, we take the SiamFC architecture as our Centroid Matching Network (CMN). SiamFC utilizes the same CNN to make the representation $F_z$ and $F_x$ for an exemplar image $z$ and a search image $x$, respectively. After that, a cross-correlation operation is performed on them to get the 2D response map $R$:

$$R(z, x) = F_z * F_x + W$$  \tag{1}$$

where $W$ denotes a window which has the same size as $R$ to penalize large displacements. The peak in $R$ indicates the relative position of the tracking object’s centroid. SiamFC adopts candidate images of several scales as input to resist the loss of the size estimation in its convolutional process and chooses the best scale to specify the bounding box. However, this method raises the disadvantage that the ratio between length and width is always fixed. After appending the bounding regression network, multiple-scale inputs are no longer needed, and rectangular bounding boxes with variable aspect ratios can be generated directly.

2.2 Bounding Regression Network

Rather than use the RPN like [2], we draw the rectangular boundary through our Bounding Regression Network (BRN), which is composed of regression layers. Figure 2 gives detail about the bounding regression network.

According to the location of the peak in response map, the features given to BRN are distilled from the feature map of the search image produced by CNN, which can significantly reduce the computation burden. As shown at the right of Fig. 2, we opt for a fully-connected layer with 1024 units to refine features and reduce the dimensions. In our experiments, a two-layer LSTM has less possibility to lose the boundary feature than a single-layer LSTM. We need only initialize the LSTM state at the beginning of the tracking, and the LSTM’s internal updates are sufficient for most scenarios. After LSTM merges the temporal information, the fully-connected layer with two units will give the scales of height and width. When testing, SiamReg needs only a single-scale image input, which enhances the computing efficiency.

The siamese network in CMN uses the classic AlexNet [13] as the backbone network. The corresponding template feature map of a $127 \times 127$ size exemplar image has the size $6 \times 6$. For more robustness, we assign $8 \times 8$ the size of the critical area in 5th Conv layer’s outputs. It can be inferred that an $8 \times 8$ critical area represents a $143 \times 143$ area in $255 \times 255$ search image which is resized from key area in original frame. The way we take the key area is the same as SiamFC. When the CMN locates the centroid, the selected $143 \times 143$ area will contain the whole tracking object with different aspect ratio.

2.3 Training

We choose the ImageNet Video dataset 2015 (ImageNet VID) [14] as the training dataset to compare SiamReg with similar work—SiamFC and Re3. There are 3862 training snippets, 555 validation snippets, and 937 test snippets in this dataset with a total size of 86GB. The training process takes about four days.

The following binary cross-entropy loss function is...
used to train the CMN:

\[
L_{CMN} = \frac{1}{RC} \sum_{j=1}^{C} \sum_{i=1}^{R} (y_{ij} \log(p_{ij}) + (1 - y_{ij}) \log(1 - p_{ij})) \tag{2}
\]

where, in response map with size \( R \times C \), label \( y \) indicates whether the point at spatial coordinates \((i, j)\) is in the target correspondence area, (i.e., \( y_{ij} = 1 \)), or it is in the background correspondence area, (i.e., \( y_{ij} = 0 \)), and \( p_{ij} \) denotes the softmax probability of \( y_{ij} \) = 1. For BRN, we use an L1 loss to limit potential drift, improving the robustness:

\[
L_{BRN} = \frac{|h_p - h| + |w_p - w|}{2} \tag{3}
\]

where \( h_p \) and \( w_p \) denote the predicted scale of height and width, and \( h, w \) are the corresponding ground truth.

Due to the join of LSTM, our training approach is different from that of other multi-task neural networks which train every task using the same input. During training, CMN adopts pairs that each consists of an exemplar image and a corresponding search image, but, BRN takes sequences with different lengths. Because of this inconsistency, the end-to-end training method will hurt the performance of the shared CNN and result in CMN failing to converge.

To address the above problem, we first train CMN and freeze it, then use sequences to train BRN. It is unnecessary to select the critical features under the guidance of CMN in training. Similar to the inputs of SiamFC, the input training sequences can always be centered on the target. As a result, BRN takes the center area of 5th Conv layer’s output directly. This design speeds up the training process. We also find the distillation makes it easy to train the regressive part with a considerable sequence length like 32, solving the convergence problem in [9].

3. Experiments

We use PyTorch to build our networks. The experiments are carried out on a machine equipped with an Intel Core i7-7800X CPU and 2 NVIDIA GeForce GTX 1080 Ti GPUs. We test trackers on benchmarks VOT2016 [15], OTB100 [16] and VOT2018 [17]. To achieve exact measures, we analyze trackers’ results through the official toolkit (on VOT dataset) and tracking platform PySOT (on OTB dataset). For the comparability, we rebuild the SiamFC network and implement its training process. The final weights are loaded in SiamReg as mentioned in the previous Freezing Method subsection. The Re3 model used in our experiment is the newest version, which makes some improvements on [9].

Both VOT2016 and VOT2018 contain 60 sequences. Each sequence is per-frame annotated by the following visual attributes: occlusion, illumination change, motion change, size change, camera motion, and unassigned. We take three estimators in VOT benchmark: Accuracy, Robustness and Expected Average Overlap, to compare trackers.

### Table 1 Comparison results on VOT2016

| Tracker  | Accuracy | Robustness | EAO  | FPS  |
|----------|----------|------------|------|------|
| ASMS*    | 0.498    | 0.522      | 0.212| 57.79|
| sKCF*    | 0.470    | 0.816      | 0.153| 63.74|
| FoT*     | 0.381    | 0.820      | 0.142| 74.00|
| BDF*     | 0.367    | 0.792      | 0.136| 96.69|
| SiamFC   | 0.517    | 0.573      | 0.211| 93.54|
| Re3      | 0.517    | 0.508      | 0.227| 191.59|
| SiamReg  | 0.494    | 0.443      | 0.236| 131.87|

The accuracy estimator calculates the average overlap between the predicted and ground truth bounding boxes when a tracking period is successful. The robustness estimator counts the number of times a tracker loses the target. In our experiments, the robustness values \( R \) represents the average number of failures per 100 frames. For robustness, lower is better. The expected average overlap (EAO), which takes both the raw values of per-frame accuracies and failures into account and has a clear practical interpretation [18], is usually used as the primary estimator for ranking trackers.

In Table 1, * denotes that the results are taken directly from the VOT 2016 results report [15]. Our SiamReg method is competitive with other fast methods and achieves better robustness score and EAO score, i.e., 0.443 and 0.236. The accuracy score of SiamReg is not the best among these trackers (0.494 vs. 0.517), mainly because of the absence of the end-to-end training. Without the multiple-scale inputs, SiamReg outperforms SiamFC in speed (131.87 vs. 93.54) as expected.

OTB100 is another challenging dataset that consists of 100 video sequences. Different from VOT datasets, which focus on short-term tracking and will restart the tracker after its failure, OTB100 emphasizes the one pass evaluation (OPE). Re3 gives the point that while performing the long-term tracking, it is necessary to reset the LSTM state to the value from the first forward pass to avoid booming outputs. Under the instruction of CMN, the LSTM in SiamReg can give the equally excellent outputs without the reset. We only use the real data from ImangNet VID and get the results as good as Re3, which extra uses synthetic data from the same dataset. The comparison of SiamReg, SiamReg with reset, and Re3 on OTB100 are shown in Fig. 3.

In [8] and [9], merging feature maps from the current frame and the previous frame will make noise accumulated,
that have a more complex regression structure. The experiments also demonstrated that the deep features of the classic siamese network have rich information to infer the target's scale. We tried some variants of our network, hoping further studies can find inspiration from it.

References

[1] L. Bertinetto, J. Valmadre, J.F. Henriques, A. Vedaldi, and P.H.S. Torr, “Fully-convolutional siamese networks for object tracking,” ECCV2016, vol.9914, pp.850–865, 2016.

[2] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, “High Performance Visual Tracking with Siamese Region Proposal Network,” CVPR2018, pp.8971–8980, 2018.

[3] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, “Distractor-Aware Siamese Networks for Visual Object Tracking,” ECCV2018, vol.11213, pp.103–119, 2018.

[4] H. Fan and H. Ling, “Siamese Cascaded Region Proposal Networks for Real-Time Visual Tracking,” CVPR2019, pp.7944–7953, 2019.

[5] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, “SiamRPN++: Evolution of Siamese Visual Tracking With Very Deep Networks,” CVPR2019, pp.4277–4286, 2019.

[6] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol.39, no.06, pp.1137–1149, 2017.

[7] Z. Wang, L. Zhang, L. Bertinetto, W. Hu, and P.H.S. Torr, “Fast Online Object Tracking and Segmentation: A Unifying Approach,” CVPR2019, pp.1328–1338, 2019.

[8] D. Held, S. Thrun, and S. Savarese, “Learning to Track at 100 FPS with Deep Regression Networks,” ECCV2016, vol.9905, pp.749–765, 2016.

[9] D. Gordon, A. Farhadi, and D. Fox, “Re3: Real-Time Recurrent Regression Networks for Visual Tracking of Generic Objects,” IEEE Robotics and Automation Letters, vol.3, no.2, pp.788–795, April 2018.

[10] K. Greff, R.K. Srivastava, J. Koutnık, B.R. Steunebrink, and J. Schmidhuber, “LSTM: A Search Space Odyssey,” IEEE Trans. Neural Netw. Learning Syst., vol.28, no.10, pp.2222–2322, Oct. 2017.

[11] R. Chen, Y. Tong, and R. Liang, “Real-time Generic Object Tracking via Recurrent Regression Network,” IEICE Trans. Inf.& Syst., vol.E103-D, no.3, pp.602–611, March 2020.

[12] M. Kristan, J. Matas, A. Leonardis, T. Vojir, R. Pfugfelder, G. Fernandez, G. Nebehay, F. Porikli, and L. Cehovin, “A Novel Performance Evaluation Methodology for Single-Target Trackers,” IEEE Trans. Pattern Anal. Mach. Intell., vol.38, no.11, pp.2137–2155, 2016.

[13] A. Krizhevsky, I. Sutskever, and G.E. Hinton, “ImageNet classification with Deep convolutional neural networks,” NIPS2012, pp.1106–1114, 2012.

[14] 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,” International Journal of Computer Vision, vol.115, no.3, pp.211–252, 2015.

[15] M. Kristan et al., “The visual object tracking VOT2016 Challenge results,” ECCV Workshops, vol.9914, pp.777–823, 2016.

[16] Y. Wu, J. Lim, and M.-H. Yang, “Object Tracking Benchmark,” IEEE Trans. Pattern Anal. Mach. Intell., vol.37, no.9, pp.1834–1848, 2015.

[17] M. Kristan et al., “The sixth visual object tracking VOT2018 Challenge results,” ECCV Workshops, vol.11129, pp.3–53, 2018.

[18] M. Kristan et al., “The visual object tracking VOT2015 Challenge results,” ICCV Workshops, pp.564–586, 2015.