GLOBALLY SPATIAL-TEMPORAL PERCEPTION: A LONG-TERM TRACKING SYSTEM

Zhenbang Li[a,c], *Qiang Wang*[a,c], *Jin Gao*[a], *Bing Li*[a,*], *Weiming Hu*[a,b,c]

[a]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[b]CAS Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Beijing, China
[c]University of Chinese Academy of Sciences, Beijing, China

ABSTRACT

Although siamese trackers have achieved superior performance, these kinds of approaches tend to favour the local search mechanism and are thus prone to accumulating inaccuracies of predicted positions, leading to tracking drift over time, especially in long-term tracking scenario. To solve these problems, we propose a siamese tracker in the spirit of the faster RCNN's two-stage detection paradigm. This new tracker, namely SiamRCNN, is dedicated to reducing cumulative inaccuracies and improving robustness based on a global perception mechanism, which allows the target to be retrieved in time spatially over the whole image plane. Since the very deep network can be enabled for feature learning in this two-stage tracking framework, the power of discrimination is guaranteed. What's more, we also add a CNN-based trajectory prediction module exploiting the target's temporal motion information to mitigate the interference of distractors. These two spatial and temporal modules exploits both the high-level appearance information and complementary trajectory information to improve the tracking robustness. Comprehensive experiments demonstrate that the proposed Globally Spatial-Temporal Perception-based tracking system performs favorably against state-of-the-art trackers.

***Index Terms*—** Visual object tracking, siamese network, motion model

1. INTRODUCTION

Object tracking [1, 2, 3] is a challenging problem in the field of computer vision, which aims to establish the positional relationship of the object to be tracked in a continuous video sequence. The popular siamese trackers [4, 5, 6] are typically based on the local search mechanism: searching the target within a small neighborhood centered on the target position of the previous frame to determine its current position. This mechanism works well if the target only has a small displacement between two adjacent frames. It also brings benefits in another aspect, which is to avoid interference from the distractors in the background.

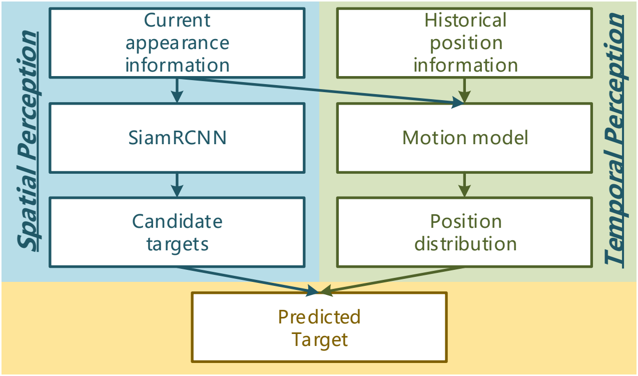

Fig. 1. Architecture of our Globally Spatial-Temporal Perception-based (GSTP) tracking system.

However, the local search mechanism bears some shortcomings. First, it could cause irreversible cumulative errors if the predictions of the target positions in the previous frames drift away due to challenging illumination change variations, motion blur, etc., because the search area generated in the current frame may not cover the target leading to a complete failure in subsequent frames. Second, it is difficult for the local mechanism-based trackers to meet the needs of long-term tracking [7, 8]. Under the long-term scenario, the target frequently re-enters and re-exits the screen. Since the tracker cannot set the correct search area when the target leaves and re-enters the screen, it often fails to retrieve the target due to the wrong search area without the target covered.

Inspired by the faster RCNN's two-stage detection paradigm [9], we propose a siamese tracker based on the global perception mechanism. During the tracking process, our tracker, namely SiamRCNN, is always able to perceive the target over the entire image. Therefore, even if the tracker makes a mistake due to the challenging target appearance variations, the target can still be retrieved in time once its appearance returns to normal. Especially under the long-term out-of-view disappearance scenario, where the tracker cannot find the target in the full image when the target leaves the screen, our tracker can continue to work when the target re-enters the screen from any position.

[*]Corresponding author.

Besides the above globally spatial perception mechanism, we also propose a temporal motion model to mitigate the interference of distractors. It is known that the siamese-based tracking framework is sometimes plagued by distractors because it is difficult for the end-to-end trained siamese matching network to distinguish well between objects that look very similar. Different from the simply designed hand-crafted strategies [4, 5], our proposed CNN-based motion model is end-to-end trained and can predict the target current position distribution using its historical trajectory information and current target appearance information. Specifically, the motion patterns are automatically learned from the trajectory dataset in the training phase instead of the hand-crafted features or rules [10]; in the testing phase, we can use the position information of any number of historical frames instead of just the previous frame for prediction.

To sum up, this paper makes following three main contributions. (1) Based on the global perception mechanism, we introduce a two stage tracking pipeline using a very deep network to reduce cumulative inaccuracies and improve robustness. (2) A novel unified end-to-end convolutional neural network architecture for trajectory prediction is proposed, where historical trajectory information and appearance information of the current frame are used to predict the target position distribution. (3) We demonstrate the effectiveness of our proposed Globally Spatial-Temporal Perception-based tracking system (GSTP) by showing that the tracking performance performs favorably against other state-of-the-art approaches on the GOT10k [11] and UAV20L [12] Long-term Tracking datasets.

2. THE PROPOSED ALGORITHM

Our method explores the key idea that the task of visual object tracking can be tackled by splitting it into first extracting candidate targets (including the real target and the distractors), followed by eliminating distractors using the motion model. By adopting this paradigm, we can achieve better performance by designing a more accurate tracker and a more robust motion model, especially in the long-term scenario.

To solve the cumulative error caused by local search, we propose the Globally Spatial-temporal Perception tracking system, which means: (1) We use an entire image instead of a small image patch as the input to the tracker to provide the global spatial information for it. (2) In order to better perceive the global spatial information, we propose the SiamRCNN tracker, which is able to detect candidate targets that are visually similar to the ground truth target. (3) To perceive the temporal information, we propose a motion model, which is able to exclude the distractors by predicting the location distribution to obtain the final tracking result.

2.1. SiamRCNN

The first component of our tracking framework (i.e., SiamRCNN) is a two-stage tracker (Fig. 2) used to detect object regions that are visually similar to the given first-frame template object. The SiamRCNN tracker consists of four modules: (1) feature extraction module, (2) feature fusion module, (3) RPN head module, and (4) RoI head module.

There are two inputs to the feature extraction module: the template image z and the search image x. According to the design of the siamese architecture, the two inputs share the same network parameters to extract features. The network structure of the feature extraction module is a variant of ResNet50, which is pretrained on the 1000-class ImageNet classification set. Features of the input images are extracted from the final convolution layer of the 4-th stage. The obtained template features $\phi(z)$ and search features $\phi(x)$ with channel dimension 1024 and stride 16 are sent to the subsequent feature fusion module.

In the feature fusion module, object features are obtained by the RoI Align operation from the template features according to the ground truth of the target. The search features and the object features are merged via the depth-wise cross-correlation. Two 1 × 1 convolutions with channel dimension 1024 are added on top of the correlation layer to obtain the fusion feature.

The RPN head includes two sibling 1 × 1 convolutional layers – a classification layer with channel dimension $2k$, and a regression layer with channel dimension $4k$, where k is the number of maximum possible proposals for each location. The RPN head takes the fusion feature as input and simultaneously regress region bounds and objectness scores for each anchor.

The RoI head is run for each region proposed by the RPN by performing RoI Align [13] to extract deep features from this proposed region. The RoI Align operation is performed on the fusion feature, generating a small feature map with a channel dimension of 2048 and a fixed spatial extent of 7 × 7 for every RoI. The RoI Aligned features are fed into the global average pooling layer followed by two sibling output layers: one that produces softmax probability estimates over two classes (foreground or background) and another layer that outputs four real-valued numbers for the foreground class. These four values encode the refined bounding-box position for the RoI. The loss of SiamRCNN is:

$$Loss = L_{cls}^{rpn} + L_{cls}^{roi} + \lambda(L_{reg}^{rpn} + L_{reg}^{roi}),$$

where λ is hyper-parameter to balance the classification loss and the regression loss. L_{cls}^{*} is the cross entropy loss and L_{reg}^{*} is the standard smooth $L1$ loss for regression.

2.2. Motion model

After detecting object regions that are visually similar to the given first-frame template object with our SiamRCNN, we

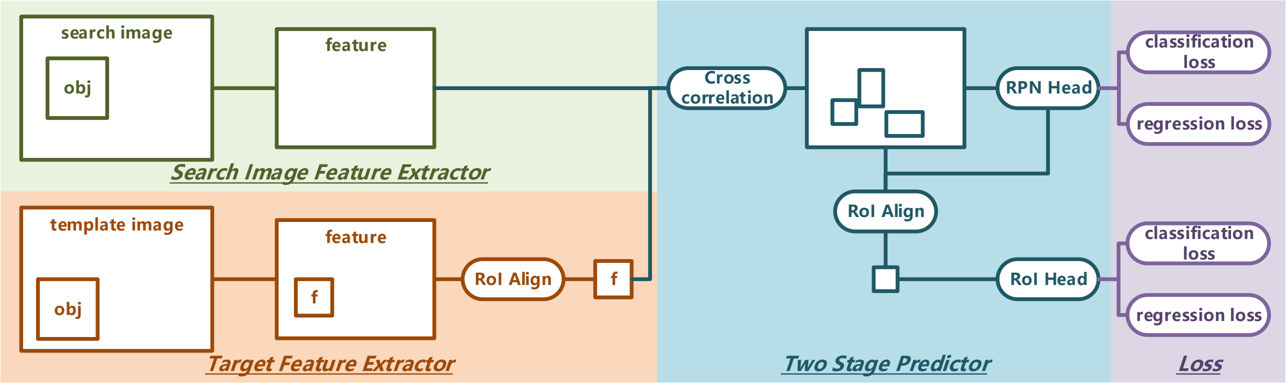

Fig. 2. Architecture of SiamRCNN.

use a motion model to eliminate distractors and obtain the final tracking result. The motion model works in an end-to-end manner by learning the target position distribution using historical trajectory information and appearance information of the current frame. It then rescores the candidate targets according to the position distribution measuring the likelihood that the target is located at each spatial location.

We build our motion model using the pose estimation network such as HRNet [14]. The main reason why the pose estimation network can be used to design the motion model is that the output of the pose estimation network is the position distribution of joint points, and the output of the motion model is the position distribution of the target. This means that there is a great deal of commonality between the two tasks. In order to use HRNet for trajectory estimation, we need to change the input and output of HRNet and keep the network structure unchanged.

During tracking the i-th frame, we utilize the position information of the previous K frames as the input to the network. Specifically, for a historical frame, a heat map is generated by applying 2D Gaussian with stand deviation of 3 pixels centered on the target position in that frame. The generated K heatmaps are concatenated according to the time order to obtain the **trajectory tensor** with channel dimension K. Our motion model not only utilizes the historical trajectory information for prediction, but also considers the appearance information of the current frame. To achieve this, the RGB image of the current frame and the trajectory tensor are concatenated to obtain the tensor with channel dimension $(3+K)$. This tensor is sent to the network, and the output of the network is a heatmap reflecting the position distribution of the target in the current frame. The loss function, defined as the mean squared error, is applied for comparing the predicted heatmap and the groundtruth heatmap. The groundtruth heatmap are generated by applying 2D Gaussian with standard deviation of 3 pixels centered on the target position in the current frame. The network structure of our motion model is the same as HRNet. A brief description is provided here. The first stage of HRNet is a high-resolution subnetwork. Then high-to-low resolution subnetworks are added one by one to form more stages. The multi-resolution subnetworks are connected in parallel. Repeated multi-scale fusions are conducted by exchanging the information across the parallel multi-resolution subnetworks over and over through the whole process. We estimate the position distribution over the high resolution representations output by HRNet. Please refer to [14] for details of the network structure.

2.3. Implementation details

Our SiamRCNN is trained on the GOT10k training set. We perform multi-scale training: the target size varies from 64×64 to 256×256. The image size remains the same during the tracking progress. SiamRCNN is trained with stochastic gradient descent (SGD). We use a weight decay of 0.0001 and momentum of 0.9. We train SiamRCNN for 27k iterations. The learning rate is decreased from 0.01 to 0.0001.

The motion model is also trained on the GOT10k training set. The Adam optimizer is adopted for training. The base learning rate is set as 1e-3, and is dropped to 1e-4 and 1e-5 at the 170th and 200th epochs, respectively. The training process is terminated within 210 epochs.

3. EXPERIMENTS

3.1. Evaluation on GOT10k Dataset

In this subsection, we evaluate our method on GOT10k [11] dataset. It contains more than 10000 video segments of real-world moving objects and over 1.5 million manually labeled bounding boxes, which covers 563 classes of real-world moving objects and 87 classes of motion patterns. The evaluation metric of GOT10k includes average overlap (AO) and success rate (SR). The AO denotes the average of overlaps between all groundtruth and estimated bounding boxes, while the SR measures the percentage of successfully tracked frames where the overlaps exceed 0.5. We compare our proposed method

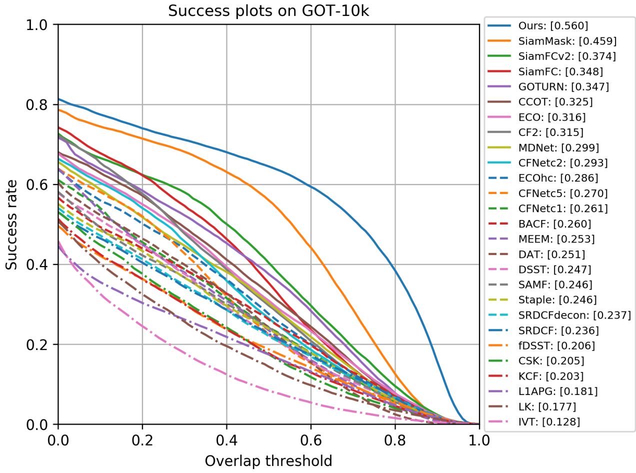

Fig. 3. Overall performance on GOT-10k, ranked by their average overlap (AO) scores.

with 26 trackers including state-of-the-arts, on GOT10k testing set. To reflect the generalization ability, all deep learning based trackers are trained using the GOT10k training set.

The success plot of the evaluated trackers is shown in Fig. 3. Compared to other listed approaches, our approach achieves a superior AO of 0.560. CSK is a typical DCF-based tracker, which uses a well-established theory of Circulant matrices to derive closed-form solutions for training and detection with several types of kernels. In contrast, our tracker is based on powerful CNNs. As a result, our tracker significantly outperforms CSK with a relative gain of 173.17% in terms of AO, which suggests the excellence of the CNN framework as well as the effectiveness of GSTP. MDNet learns shared layers and multiple branches of domain-specific layers, where domains correspond to individual training sequences and each branch is responsible for binary classification to identify target in each domain. In contrast, our tracker learns a general similarity map by cross-correlation between the feature representations learned for the target template and the search region. Compared with MDNet, the relative improvement of the AO score is 87.29%, which suggests that the Siamese architecture and the cross-correlation operation are more suitable for the object tracking task. SiamMask is one of state-of-the-art trackers. It is a Siamese tracker and takes advantage of very deep networks to extract features. SiamMask utilizes the region proposal subnetwork to predict the location of the target. Compared with SiamMask, our two stage tracker is designed based on the global perception mechanism to reducing cumulative inaccuracies. The motion model suppresses distractors and improves the tracking robustness. As a result, our tracker outperforms SiamMask by relative 22.00% in terms of AO, which highlights the importance of the proposed tracker and the motion model.

3.2. Evaluation on UAV20L Dataset

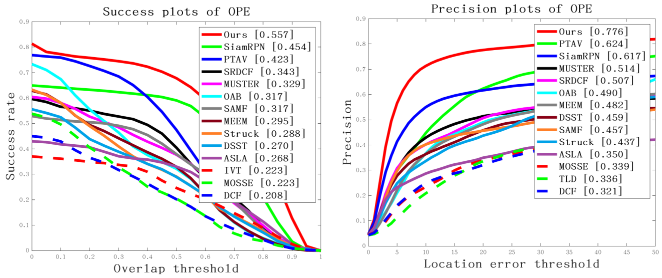

Fig. 4. Success and precision plots on UAV20L dataset.

UAV20L [12] is an aerial video dataset captured from a low-altitude aerial perspective. Designed for long-term tracking, the UAV20L database has 20 videos with an average length of 2934 frames, far exceeding the length of other popular databases: OTB50 – 578 frames, OTB100 – 590 frames, VOT14 – 416 frames and VOT15 – 365 frame.

Following the evaluation method of OTB50 [15], we use precision and success to evaluate the performance of trackers on the UAV20L dataset. Precision refers to the distance from the center point of the predicted bounding box to the center point of the ground truth bounding box. Success refers to the intersection over union (IOU) of the predicted bounding box and the ground truth bounding box. In Fig. 4, the performance comparison of different trackers is visualized by precision plot and success plot.

The proposed method is compared against 13 recent trackers. Fig. 4 clearly shows that our algorithm outperforms all other trackers in terms of success and precision scores. Specifically, in the success plot, our tracker obtains a AUC score of 0.557. Compared with the state-of-art method PTAV [16] and SiameseRPN [5], the proposed tracker outperforms these trackers by relative 31.7% and 22.7%. In the precision plot, the proposed algorithm obtains a score of 0.776. Compared with SiameseRPN [5] and PTAV [16], the proposed tracker outperforms these trackers by relative 25.8% and 24.4%.

4. CONCLUSION

In this paper, we propose a novel tracking architecture including the SiamRCNN tracker and the data-driven motion model. The global perception mechanism allows SiamRCNN to reduce the cumulative error during the tracking process. SiamRCNN uses a very deep network for two-stage tracking, which makes the tracker more discriminative. The motion model is trained end-to-end and is capable of learning the motion patterns of targets from large-scale trajectory datasets. Through the collaborative work of SiamRCNN and the motion model, the proposed method performs favorably against state-of-the-art trackers.

5. REFERENCES

[1] Isabelle Leang, Stéphane Herbin, Benoît Girard, and Jacques Droulez, “On-line fusion of trackers for single-object tracking,” *Pattern Recognition*, vol. 74, pp. 459–473, 2018.

[2] Lingfeng Wang and Chunhong Pan, “Visual object tracking via a manifold regularized discriminative dual dictionary model,” *Pattern Recognition*, vol. 91, pp. 272–280, 2019.

[3] Shunli Zhang, Wei Lu, Weiwei Xing, and Shukui Zhang, “Using fuzzy least squares support vector machine with metric learning for object tracking,” *Pattern Recognition*, vol. 84, pp. 112–125, 2018.

[4] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr, “Fully-convolutional siamese networks for object tracking,” in *European conference on computer vision*. Springer, 2016, pp. 850–865.

[5] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu, “High performance visual tracking with siamese region proposal network,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8971–8980.

[6] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr, “Fast online object tracking and segmentation: A unifying approach,” *arXiv preprint arXiv:1812.05050*, 2018.

[7] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas, “Tracking-learning-detection,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 7, pp. 1409–1422, 2011.

[8] Zhibin Hong, Zhe Chen, Chaohui Wang, Xue Mei, Danil Prokhorov, and Dacheng Tao, “Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 749–758.

[9] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in *Advances in neural information processing systems*, 2015, pp. 91–99.

[10] Irene Anindaputri Iswanto and Bin Li, “Visual object tracking based on mean-shift and particle-kalman filter,” *Procedia computer science*, vol. 116, pp. 587–595, 2017.

[11] Lianghua Huang, Xin Zhao, and Kaiqi Huang, “Got-10k: A large high-diversity benchmark for generic object tracking in the wild,” *arXiv preprint arXiv:1810.11981*, 2018.

[12] Matthias Mueller, Neil Smith, and Bernard Ghanem, “A benchmark and simulator for uav tracking,” in *European conference on computer vision*. Springer, 2016, pp. 445–461.

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick, “Mask r-cnn.,” *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[14] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang, “Deep high-resolution representation learning for human pose estimation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5693–5703.

[15] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, “Online object tracking: A benchmark,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2411–2418.

[16] Heng Fan and Haibin Ling, “Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5486–5494.