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 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 exploit 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.

*Corresponding author.

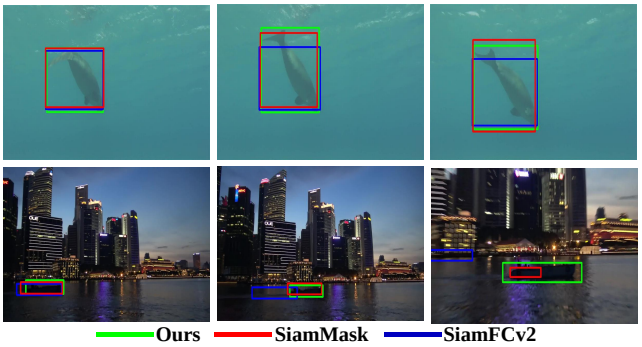

Fig. 1. A comparison of our method with the stage-of-the-art trackers SiamMask [6] and SiamFCv2 [4] in challenging situations. The example frames are from the GOT-10k [7] testing set.

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, 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 [8, 9]. 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 faster RCNN's two stage detection paradigm [10], we propose a siamese tracker based on the global perception mechanism. During the tracking process, our tracker 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.

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 [11]; in the testing phase, we can use the position information of any number of historical frames instead of just the previous frame for prediction.

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 tracking component and a more robust motion model, especially in the long-term scenario.

Specifically, 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 a two-stage tracking component, 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. Tracking component

We build our tracking component based on the popular object detection architecture — faster RCNN [10]. Although the use of object detection modules and siamese-based feature extractor for object tracking is not first proposed in this paper, our tracker has advantages over relevant tracking algorithms [5, 6, 12, 13]. Compared with SiamRPN [5], whose input is always image patches with target located at the center, the input of our tracking component is the entire image, which means the target may appear at any spatial location. This prevents the network from learning bias to the center location, thereby breaking the spatial invariance restriction [14] of the network architecture. As a result, the very deep networks such as ResNet50 [15] can be used as the feature extractor of the tracking component. For detailed explain about spatial invariance restriction and center bias, please refer to [14]. Compared with SiamRPN [5] and SiamMask [6], we use a two stage architecture. The second stage (RoI head [10]) can distinguish the foreground and background more effectively. Compared with SiamMask [6] and ATOM [12], our tracker can search for the target in the whole image, which allows the target to be retrieved in time spatially after the tracking module makes a mistake. Compared with Siam R-CNN [13], whose backbone and RPN are frozen, our tracker generate the target-specific features before RPN, and the whole network can be trained end to end. In object tracking, a target may be foreground in one video and background in another. Therefore, the candidates generated by our RPN can better meet the requirements of general object tracking.

To describe our tracking component, we we first briefly revisit faster RCNN. It consists of a feature extractor followed by two detection stages: the RPN head and the RoI head. The feature extractor ϕ_1 is a variant of ResNet50. The first stage uses a region proposal network (RPN) to slide on the last feature map of the backbone layers and predict whether there is an object or not and also predict the bounding box of those objects. The second stage (*i.e.*, RoI head) is run for each region proposed by the RPN by performing RoI Align [16] to extract deep features from this proposed region. Each RoI (Region of Interest) is classified as a specific category using a classification layer and the bounding box is refined using a regression layer. Please refer to [10] for more detailed information.

For the task of object tracking, there are two inputs to the feature extractor ϕ_1: 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. After generating the template feature $f_z = \phi_1(z)$ and search feature $f_x = \phi_1(x)$, we crop the object feature $f_{obj} \in \mathbb{R}^{1024\times7\times7}$ by the RoI Align operation [16] from the template features according to the ground truth of the target:

$$f_{obj} = \mathcal{R}(b_{obj}, f_z), \tag{1}$$

where $\mathcal{R}$ represents the RoI Align operator and b_{obj} is the ground truth bounding box of the target. Next, the search feature and the object feature are merged via the depth-wise cross-correlation [14]:

$$f_{corr} = f_{obj} * f_x, \tag{2}$$

where f_{corr} is the fusion feature and $*$ is the depth-wise cross-correlation operator. Then f_{corr} is sent to the RPN head to generate candidates $B = \{b^i_{roi}\}^{i=1:N}$. At a second stage, the RoI Align operation is performed on the fusion feature f_{corr}, generating a small feature map with a channel dimension of 2048 and a fixed spatial extent of 7×7 for every RoI:

$$f^i_{roi} = \mathcal{R}(b^i_{roi}, f_{corr}), \tag{3}$$

where b^i_{roi} is the bounding box of candidate region i. Finally, each RoI is classified as foreground/background. During testing, we select top K ranked RoI as the candidate targets, which will be post-processed in the motion model.

2.2. Motion model

After detecting object regions that are visually similar to the given first-frame template object with our tracking component, we 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, which is a 2D heatmap measuring the likelihood that the target is located at each spatial location.

Let $H_t^k = \{h_{t-i}\}_{i=1:k}$ denote the historical trajectory information, where t is the index of the current frame, k is the length of history, and $h_j = \{x_j, y_j\}$ is a two-dimensional coordinate representing the position of the target in frame j. Inspired by the pose estimation task, we present h_j as a two-dimensional heatmap $m_j \in \mathbb{R}^{h \times w}$ with a 2D Gaussian centered on the target position (x_j, y_j). To model the temporal information, The generated k heatmaps are concatenated according to the time order to obtain the trajectory tensor $\mathcal{M} \in \mathbb{R}^{k \times h \times w}$ with channel dimension k:

$$\mathcal{M}_t^k = \mathcal{C}(m_{t-k}, m_{t-k+1}, ..., m_{t-1}), \tag{4}$$

where $\mathcal{C}(\cdot)$ is the concatenation operation.

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 $\mathcal{I} \in \mathbb{R}^{3 \times h \times w}$ and the trajectory tensor are concatenated to obtain the enhanced trajectory tensor $\mathcal{N} \in \mathbb{R}^{(3+k) \times h \times w}$ with channel dimension $(3 + k)$:

$$\mathcal{N}_t^k = \mathcal{C}(\mathcal{I}_t, \mathcal{M}_t^k). \tag{5}$$

Assuming ϕ_2 is the motion model, which is a CNN network, the output of ϕ_2 is calculated as follows:

$$\mathcal{O}_t^k = \phi_2(\mathcal{N}_t^k), \tag{6}$$

where $\mathcal{O}_t^k \in \mathbb{R}^{h \times w}$ is a 2D heatmap reflecting the position distribution of the target in the frame t.

The network of our motion model ϕ_2 is the same as the pose estimation network HRNet [17]. 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 [17] for details of the network structure.

Table 1. Performance of our algorithm with different components on GOT-10k test set.

ROI head	Motion model	AO	$SR_{0.50}$	$SR_{0.75}$
		0.410	0.486	0.162
✓		0.521	0.595	0.440
✓	✓	0.560	0.645	0.457

3. EXPERIMENTS

3.1. Implementation details

Our tracking component is trained on the GOT-10k [7] training set. 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. 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. The tracking component is trained with stochastic gradient descent (SGD). We use a weight decay of 10^{-4} and momentum of 0.9. We train the tracking component for 27k iterations. The learning rate is decreased from 10^{-2} to 10^{-4}.

The motion model is also trained on the GOT-10k training set. The Adam optimizer [18] is adopted for training. The base learning rate is set as 10^{-3}, and is dropped to 10^{-4} and 10^{-5} at the 170th and 200th epochs, respectively. The training process is terminated within 210 epochs.

3.2. Evaluation on GOT-10k Dataset

In this subsection, we evaluate our method on GOT-10k [7] dataset. The evaluation metric of GOT-10k 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 0.5/0.75.

Ablation Studies From Table 1 (the 1^{st} and 2^{nd} row), we see that the AO performance increases by 11.1% by adding the RoI head. The RPN stage rapidly filters out most background samples, and the RoI head adopts a fixed foreground-to-background ratio to maintain a manageable balance between foreground and background. From Table 1 (the 2^{nd} and 3^{rd} row), we see that with the motion model, the AO, the $\text{SR}_{0.50}$ and the $\text{SR}_{0.75}$ increases by 3.9%, 5.0% and 1.7%, respectively. This is because the proposed motion model can effectively predict the position distribution of the target, effectively avoiding the adverse effects of distractors.

Overall Performance We compare our proposed method with 8 trackers including state-of-the-arts, on GOT-10k testing set. Compared to the listed approaches, our approach achieves a superior AO of 0.560 (Table 2). Compared with

Table 2. Comparing the results of our approach against other approaches over the GOT-10k test set. The trackers are ranked by their average overlap (AO) scores.

Method	AO	$SR_{0.50}$	$SR_{0.75}$
Ours	**0.560**[1]	**0.645**[1]	**0.457**[1]
SiamMask	0.459	0.560	0.205
SiamFCv2	0.374	0.404	0.144
SiamFC	0.348	0.353	0.098
GOTURN	0.347	0.375	0.124
CCOT	0.325	0.328	0.107
ECO	0.316	0.309	0.111
CF2	0.315	0.297	0.088
MDNet	0.299	0.303	0.099

Table 3. Performance on subsets with different attributes collected from GOT-10k validation set.

Att.	SiamFC		SiamMask		Ours	
	AO	$SR_{0.5}$	AO	$SR_{0.5}$	AO	$SR_{0.5}$
FM	0.472	0.538	0.526	0.608	0.639	0.715
OC	0.411	0.447	0.494	0.559	0.585	0.659
CU	0.505	0.545	0.595	0.701	0.738	0.837
LO	0.557	0.655	0.643	0.779	0.721	0.807

SiamMask [6], 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.

Performance Analysis by Attributes Every video in GOT-10k training/validation dataset is annotated with multiple attributes including: visible ratios, motion speed, video length and cut by image. To analyze the performance of trackers in various scenarios, we compare our tracker with two state-of-the-art trackers (*i.e.*, SiamFC [4] and SiamMask [6]) on GOT-10k validation set. We collect four subset from the validation set according to the attribute annotations: FM (fast motion) subset, OC (occlusion) subset, CU (cut by image) subset, and LO (long video) subset. The FM subset includes videos in which the target motion speed is fast. The OC subset include videos in which the target is occluded frequently. The CU subset include videos in which the target is cut by the image boundary frequently. The LO subset includes the longest 40 videos in the validation set. Table 3 shows the different performance characteristics of the tracking algorithms. On the FM subset, our method outperforms SiamMask [6] with a relative gain of 21.48% in terms of AO. In terms of AO, our algorithm outperforms SiamMask by relative 18.42% and 24.03% on OC and CU subset, respectively. On the LO subset, the relative improvement of the AO score is 12.13% compared with SiamMask. This result shows that the global perception mechanism allows our tracker to reduce the cumulative error when tracking the long videos.

3.3. Evaluation on UAV20L Dataset

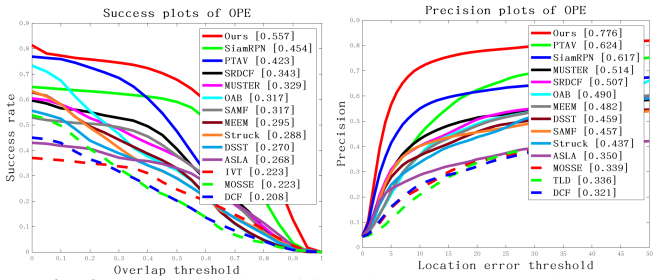

Fig. 2. Success and precision plots on UAV20L dataset.

UAV20L [19] 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. Following the evaluation method of OTB50 [20], 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. 2, the performance comparison of different trackers is visualized by precision plot and success plot.

The proposed method is compared against 13 recent trackers. Fig. 2 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 [21] and SiamRPN [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 SiamRPN [5] and PTAV [21], 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 tracking component and the data-driven motion model. The global perception mechanism allows the tracking component to reduce the cumulative error during the tracking process. The tracking component 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 the tracking component 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," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1328–1338.

[7] Lianghua Huang, Xin Zhao, and Kaiqi Huang, "Got-10k: A large high-diversity benchmark for generic object tracking in the wild," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg, "Atom: Accurate tracking by overlap maximization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4660–4669.

[13] Paul Voigtlaender, Jonathon Luiten, Philip HS Torr, and Bastian Leibe, "Siam r-cnn: Visual tracking by re-detection," *arXiv preprint arXiv:1911.12836*, 2019.

[14] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan, "Siamrpn++: Evolution of siamese visual tracking with very deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4282–4291.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

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

[17] 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.

[18] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[19] 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.

[20] 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.

[21] 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.