Supplementary Material

Know Your Surroundings: Exploiting Scene Information for Object Tracking

Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte

CVL, ETH Zürich, Switzerland

The supplementary material provides additional details about the network architecture and results. In Section S1, we provide details about our tracking architecture. Section S2 contains detailed results on the VOT2018 dataset, while Section S3 provides qualitative comparison of our approach with the baseline tracker DiMP-50 [2]. We also include a **supplementary video** for the qualitative comparison with DiMP-50.

S1 Network details

In this section, we provide more details about our tracking architecture.

State initializer Υ: Given the first frame target annotation B_0 as input, the initializer network Υ first generates a single-channel label map specifying the target center. We use a Gaussian function to generate this label map. The label map is passed through a single convolutional layer with 3×3 kernels. The output is then passed through a tanh activation to obtain the initial state vectors.

State propagation: We use the features from the fourth convolutional block of ResNet-50 [5], having a spatial stride of 16, to construct our cost volume. Our network can process images of any input resolution. However, in all our experiments, we resize the input search region crop to 288×288 for convenience. Thus the features x used for computing the cost volume have the size $W = H = 18$, with $D_m = 1024$ channels. The maximal displacement $d_{\max}$ for cost volume computation is set to 9.

The network architecture used to map the raw cost volume slices to obtain the processed matching costs ϕ is shown in Table S1. Note that the network weights are shared for all cost volume slices. We use an identical network architecture to process the initial correspondence ϕ'.

Target Confidence Score Prediction: The network architecture for our predictor module P is shown in Table S2.

State update: The state update module Φ contain a convolutional gated recurrent unit (ConvGRU) [1] which performs the state updates. The input $f_t \in \mathbb{R}^{W \times H \times 4}$ to the ConvGRU is obtained by concatenating the target confidence scores $\varsigma_t \in \mathbb{R}^{W \times H \times 1}$ and the appearance model output $s_t \in \mathbb{R}^{W \times H \times 1}$, along with their maximum values along the third dimension. The propagated

Table S1. The network architecture used to process cost volume slices. The network takes individual cost volume slices (size $18 \times 18 \times 1$) as input. All convolutional layers use 3×3 kernels. BN denotes batch normalization [6].

Layer	Operation	Output size
1	Conv + BN + ReLU	$18 \times 18 \times 8$
2	Conv + BN	$18 \times 18 \times 1$

Table S2. The network architecture for predictor module P. The input to the network is obtained by concatenating the propagated states $\hat{h}_{t-1}$ ($18 \times 18 \times 8$), reliability score ξ_t ($18 \times 18 \times 1$), and appearance model output s_t ($18 \times 18 \times 1$). All convolutional layers use 3×3 kernels.

Layer	Operation	Output size
1	Conv + ReLU	$18 \times 18 \times 16$
2	Conv + Sigmoid	$18 \times 18 \times 1$

state vectors $\hat{h}_{t-1} \in \mathbb{R}^{W \times H \times S}$ are treated as the hidden states of the ConvGRU from the previous time-step. We use the standard update equations for ConvGRU,

$$z_t = \sigma\big(\text{Conv}(f_t \oplus \hat{h}_{t-1})\big) \tag{S1a}$$

$$r_t = \sigma\big(\text{Conv}(f_t \oplus \hat{h}_{t-1})\big) \tag{S1b}$$

$$\tilde{h_t} = \tanh\big(\text{Conv}(f_t \oplus (r_t \odot \hat{h}_{t-1}))\big) \tag{S1c}$$

$$h_t = (1 - z_t) \odot \hat{h}_{t-1} + z_t \odot \tilde{h_t}\,. \tag{S1d}$$

Here, $\oplus$ denotes concatenation of the feature maps along the third dimension, while $\odot$ denotes element-wise product. σ and tanh denote the sigmoid and hyperbolic tangent activation functions, respectively. We use 3×3 kernels for all the convolution layers, represented by Conv.

S2 Detailed Results on VOT2018

Here, we provide detailed results on the VOT2018 [7] dataset, consisting of 60 challenging videos. The trackers are evaluated using the expected average overlap curve, which plots the expected average overlap between the tracker prediction and groundtruth for different sequence lengths. The average of the expected average overlap values over typical sequence lengths provides the expected average overlap (EAO) score, which is used to rank the trackers. We refer to [8] for more details about EAO score computation.

We compare our approach with the recent state-of-the-art trackers: DRT [10], RCO [7], UPDT [3], DaSiamRPN [12], MFT [7], LADCF [11], ATOM [4], SiamRPN++ [9], and DiMP-50 [2]. Figure S1 shows the expected average overlap curve. The EAO score for each tracker is shown in the legend. Our approach

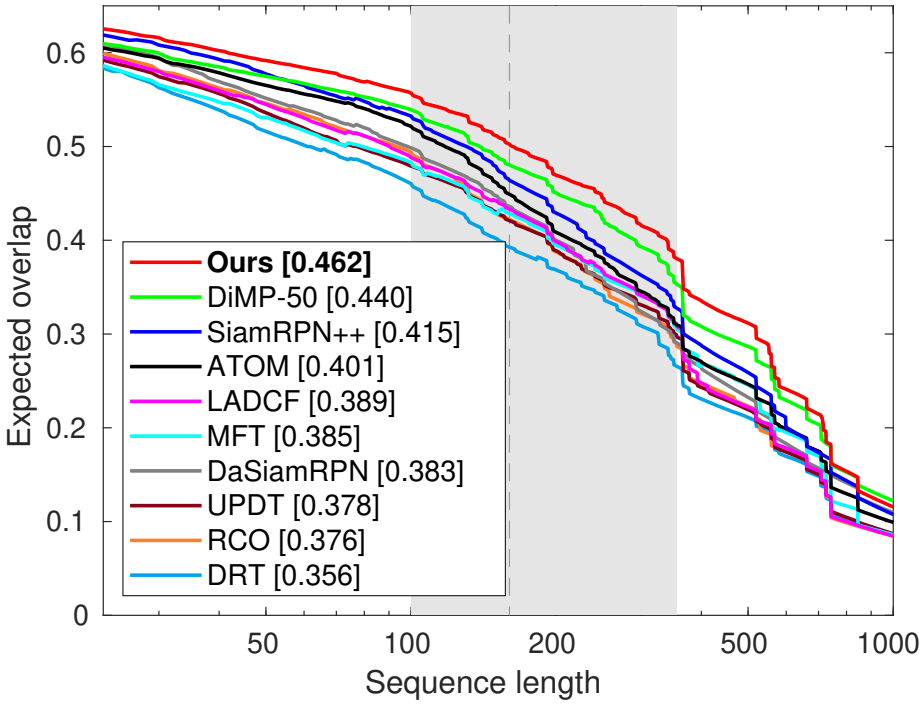

Fig. S1. Expected average overlap curve on the VOT2018 dataset. The plot shows the expected overlap between the tracker prediction and groundtruth for different sequence lengths. The expected average overlap (EAO) score, computed as the average of expected overlap values over typical sequence lengths (shaded region) is shown in the legend. Our tracker obtains the best EAO score, outperforming the previous best method DiMP-50 with a relative improvement of 5% in EAO.

obtains the best results with an EAO score of 0.462, outperforming the previous best method DiMP-50 with a relative improvement of 5%. This demonstrates the benefit of exploiting scene information for tracking.

S3 Qualitative Results

Here, we provide a qualitative comparison of our approach with the baseline tracker DiMP-50 [2], which uses only an appearance model. Figure S2 shows the tracking output for both the trackers on a few example sequences. DiMP-50 struggles to handle distractor objects which are hard to distinguish based on only appearance (second, third, fifth). In contrast, our approach is aware of the distractor objects in the scene and can exploit this scene information to achieve robust tracking. Propagating the scene information is also helpful in case of fast target appearance changes (first and fourth rows). In these cases, keeping track of the background regions can be useful to eliminate target candidate regions,

greatly simplifying target localization. The last row shows a failure case of our approach. Here, the appearance model fails to detect the occlusion caused by the white dog. As a result, the state vectors are updated incorrectly, and the tracker starts tracking the white dog.

References

1. Nicolas Ballas, Li Yao, Chris Pal, and Aaron C. Courville. Delving deeper into convolutional networks for learning video representations. In *ICLR*, 2016.
2. Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *ICCV*, 2019.
3. Goutam Bhat, Joakim Johnander, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Unveiling the power of deep tracking. In *ECCV*, 2018.
4. Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ATOM: Accurate tracking by overlap maximization. In *CVPR*, 2019.
5. 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*, pages 770–778, 2016.
6. Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.
7. Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka Cehovin Zajc, Tomas Vojir, Goutam Bhat, Alan Lukezic, Abdelrahman Eldesokey, Gustavo Fernandez, and et al. The sixth visual object tracking vot2018 challenge results. In *ECCV workshop*, 2018.
8. M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Čehovin, G. Fernández, T. Vojír, G. Nebehay, R. Pflugfelder, and G. Hger. The visual object tracking vot2015 challenge results. In *ICCV workshop*, 2015.
9. Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *CVPR*, 2019.
10. Chong Sun, Dong Wang, Huchuan Lu, and Ming-Hsuan Yang. Correlation tracking via joint discrimination and reliability learning. In *CVPR*, 2018.
11. Tianyang Xu, Zhen-Hua Feng, Xiao-Jun Wu, and Josef Kittler. Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual tracking. *CoRR*, abs/1807.11348, 2018.
12. Zheng Zhu, Qiang Wang, Li Bo, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *ECCV*, 2018.

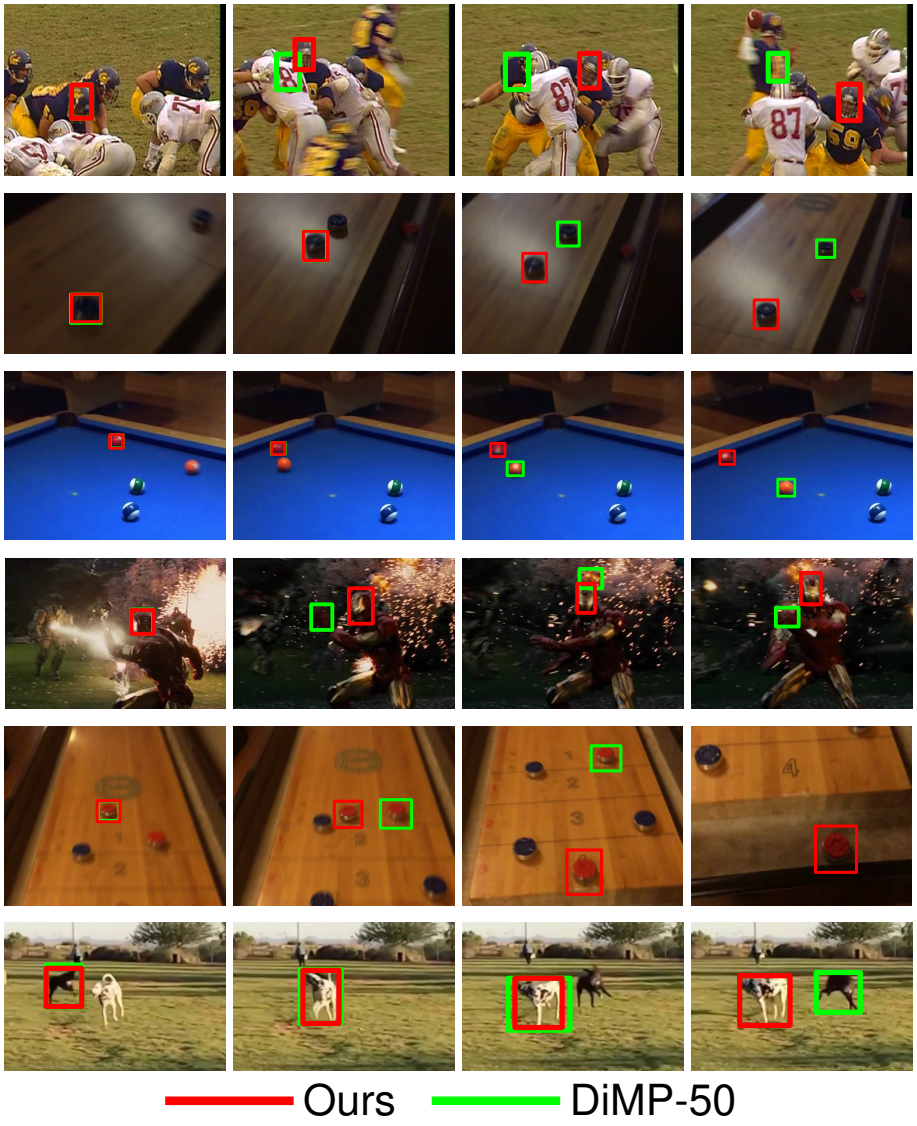

Fig. S2. A qualitative comparison of our approach with the baseline appearance model, DiMP-50. Our tracker extracts information about other objects in the scene and exploits this knowledge to provide scene-aware predictions. Consequently, our approach can handle distractor objects which are hard to distinguish based on appearance only (second, third, and fifth rows). The propagated scene information is also beneficial to eliminate target candidate regions, which can be helpful in case of fast target appearance changes (first and fourth rows). The last row shows a failure case of our approach. Here, the appearance model cannot detect the occlusion caused by the white dog. This results in incorrect state updates, leading to tracking failure.