Supplementary Material for "MBPTrack: Improving 3D Point Cloud Tracking with Memory Networks and Box Priors"

Tian-Xing Xu[1] Yuan-Chen Guo[1] Yu-Kun Lai[2] Song-Hai Zhang[1] *
[1] Tsinghua University, China [2] Cardiff University, United Kingdom
[1]{xutx21@mails., guoyc19@mails., shz@}tsinghua.edu.cn [2]LaiY4@cardiff.ac.uk

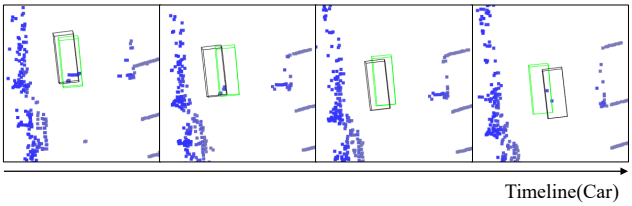

Figure 1: **Failure Cases.**

1. More Implementation Details

Model details. We adopt DGCNN [1] as our backbone, which is commonly used in point cloud-based tasks. It contains 3 EdgeConv layers and 3 downsampling layers, which yield $N = 128$ point features with $C = 128$ channels. We sample $N_p = 64$ predicted centers as the proposal centers for efficiency. For the box-prior sampling, we adopt $n_x = n_y = 3$, while setting $n_z = 5$ to handle objects like pedestrians. We adopt a $3 \times 3 \times 5$ Conv to aggregate 3D feature maps $\mathcal{Z}$. For the point-to-reference feature transformation, we adopt KNN to obtain the neighborhood $\mathcal{N}(r)$ for the reference point $r \in \mathcal{R}$, where K is set to 8 for efficiency.

Data augmentation. We enlarge the ground truth bounding box $\mathcal{B}_{t-1}$ by 2 meters for each dimension to obtain the sub-region in the current frame where tracked targets may appear. To simulate the inaccurate predictions the model might encounter, we also add a slight random shift to the bounding boxes with a range of [-0.3m, 0.3m] along each axis as well as random rotation around the up-axis. Following CXTrack [2], we randomly sample $\dot{N}_t = 1024$ points in each frame to obtain the input point cloud $\mathcal{P}_t$.

2. Failure Cases

The predicted orientation θ_t by MBPTrack relies heavily on the predicted bounding box $\mathcal{B}_{t-1}$, which defines the canonical coordinate system for the point cloud $\mathcal{P}_t$. Supposing the tracked target disappears at timestamp $(t-1)$ and reappears at timestamp t, MBPTrack may fail to predict an accurate orientation of the target if the orientation of the 3D bounding box $\mathcal{B}_{t-1}$ is very inaccurate. Besides, as shown in Fig. 1, even for humans it is difficult to determine whether the points in the fourth image belong to the left or right side of the car without consistent motion information. The overlook of consistent motion information leads to limited performance when the point clouds are too sparse. In the future, we would like to explicitly model the target motion to address this issue.

*corresponding author

References

[1] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 38(5):1–12, 2019. 1

[2] Tian-Xing Xu, Yuan-Chen Guo, Yu-Kun Lai, and Song-Hai Zhang. CXTrack: Improving 3D point cloud tracking with contextual information. *arXiv preprint arXiv:2211.08542*, 2022. 1