

VoxelTrack: Exploring Voxel Representation for 3D Point Cloud Object Tracking

Anonymous Authors

1 MORE IMPLEMENTATION DETAILS

1.1 Model Input

We crop two consecutive frames of the point cloud (p_t and p_{t-1}) on the timestamp by centering on the previous ground truth box. Due to the subsequent voxelization operations, we do not need to down-sample the point cloud to the same number of points per frame. Following, we then set W_l , L_l and H_l to 64, 64 and 10 for large voxel branch, and set W_s , L_s and H_s to 128, 128 and 20 for small voxel branch. Finally, we concatenate the previous frame and the current frame along the channel dimension as input.

1.2 Data Augmentation

we first crop the point cloud P_{t-1} and P_t with a range of $[(4.8, -4.8), (4.8, -4.8), (1.5, -1.5)]$ for Car category and $[(1.92, -1.92), (1.92, -1.92), (1.5, -1.5)]$ for Pedestrian category. To demonstrate the feasibility of our VoxelTrack in real scenarios, we augmented the original dataset to simulate the uncertainties that may exist in the real world. Specifically, we randomly shifted the cropping range to some extent. For each epoch, we first duplicate the dataset in four copies, after which one copy keeps the original cropping range, and the cropping ranges of each of the remaining three copies will be added with random offsets.

1.3 Network Architecture

The specific network components can be found in Tab. 2. We employ the VoxelNext as backbone of our VoxelTrack to extract point spatial information. The backbone contains three stages, which encode different scale of features. Specifically, each stage is consisted of three 3D sparse convolution layers, which aggregate spatial information and enhance feature representation. Moreover, we employ ReLU and SyncBatchNorm as the activation function and batch normalization function throughout our network. The eps and momentum parameters of the SyncBatchNorm layer are set respectively to 1e-3 and 0.01.

2 MORE ANALYSIS EXPERIMENTS

2.1 Computational Cost Comparison

The FLOPs of a model is a important metric for evaluating the computational coast and complexity of a model. We analyze the computational coast for VoxelTrack by FLOPs and compare it with other trackers. The results are showed in Tab. 1. Our VoxelTrack mainly employed the 3D sparse convolution layer to reduce FLOPs and achieved the best mean Success on both KITTI and NuScenes dataset.

REFERENCES

- [1] Silvio Giancola, Jesus Zarzar, and Bernard Ghanem. 2019. Leveraging shape completion for 3d siamese tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 1359–1368.

Table 1: Backbone architectures of our VoxelNet. The output size means width \times length \times height \times channel. “[$k \times k \times k, c, s$] $\times n$ ” means n 3D sparse convolution layers with kernel size $k \times k \times k$, output channel c and stride s .

Stage	Output Size of branch I	Output Size of branch II	Voxel. Embed
1	$64 \times 64 \times 10 \times 32$	$32 \times 32 \times 5 \times 32$	$\begin{bmatrix} 3 \times 3 \times 3, 32, 2 \\ 3 \times 3 \times 3, 32, 1 \end{bmatrix} \times 1$ $\begin{bmatrix} 3 \times 3 \times 3, 32, 1 \\ 3 \times 3 \times 3, 64, 2 \end{bmatrix} \times 2$
2	$32 \times 32 \times 5 \times 64$	$16 \times 16 \times 3 \times 64$	$\begin{bmatrix} 3 \times 3 \times 3, 64, 2 \\ 3 \times 3 \times 3, 64, 1 \end{bmatrix} \times 1$ $\begin{bmatrix} 3 \times 3 \times 3, 128, 2 \\ 3 \times 3 \times 3, 128, 1 \end{bmatrix} \times 2$
3	$16 \times 16 \times 3 \times 128$	$8 \times 8 \times 2 \times 128$	$\begin{bmatrix} 3 \times 3 \times 3, 128, 2 \\ 3 \times 3 \times 3, 128, 1 \end{bmatrix} \times 1$ $\begin{bmatrix} 3 \times 3 \times 3, 128, 1 \\ 3 \times 3 \times 3, 128, 1 \end{bmatrix} \times 2$

Table 2: Comparison of computational coast with different trackers

Tracker	FLOPs	KITTI	NuScenes
SC3D [1]	20.07 G	31.2 / 48.5	20.7 / 20.2
P2B [3]	4.28 G	42.4 / 60.0	36.4 / 45.0
BAT [4]	5.53 G	51.2 / 72.8	38.1 / 45.7
PTTR [6]	2.61G	57.9 / 78.2	44.5 / 52.0
GLT-T [2]	3.87 G	60.1 / 79.3	44.4 / 54.3
M ² Track [5]	2.54 G	62.9 / 83.4	49.2 / 62.7
VoxelTrack	1.18 G	70.4 / 88.3	59.0 / 71.4

- [2] Jiahao Nie, Zhiwei He, Yuxiang Yang, Mingyu Gao, and Jing Zhang. 2023. GLT-T: Global-Local Transformer Voting for 3D Single Object Tracking in Point Clouds. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 1957–1965.
- [3] Haozhe Qi, Chen Feng, Zhiguo Cao, Feng Zhao, and Yang Xiao. 2020. P2b: Point-to-box network for 3d object tracking in point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 6329–6338.
- [4] Chaoda Zheng, Xu Yan, Jiantao Gao, Weibing Zhao, Wei Zhang, Zhen Li, and Shuguang Cui. 2021. Box-aware feature enhancement for single object tracking on point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 13199–13208.
- [5] Chaoda Zheng, Xu Yan, Haiming Zhang, Baoyuan Wang, Shenghui Cheng, Shuguang Cui, and Zhen Li. 2022. Beyond 3d siamese tracking: A motion-centric paradigm for 3d single object tracking in point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8111–8120.
- [6] Changqing Zhou, Zhipeng Luo, Yueru Luo, Tianrui Liu, Liang Pan, Zhongang Cai, Haiyu Zhao, and Shijian Lu. 2022. Pttr: Relational 3d point cloud object tracking with transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8531–8540.