Verified by the Organizing Committee.

SparseOcc4D: a multi-scale sparse occupancy and flow prediction system

Ning Li
Independent Researcher
Beijing
ning.h.li@tum.de

Abstract

The Task of Occupancy and Flow in CVPR 2024 Autonomous Grand Challenge, is to estimate, not only the likelihood of an area being occupied by an object (such as a vehicle, pedestrian, or cyclist), but also the motion patterns (velocity and direction) of objects. To solve the problem, a sparse occupancy and flow prediction system is proposed during this competition, multi-scale supervision and combined loss is used in this work. The best result ranks 7th place on the leaderboard.

1. Introduction

Multi-camera based occupancy networks are playing an increasingly important role in autonomous driving and are mainly used to know the occupancy and semantic information of the environment around the self-driving car. A lot of excellent work has emerged in this area in recent years, such as SurroundOcc and SparseOcc.

1.1. SurroundOcc

SurroundOcc [6] leverages a combination of techniques to predict the status of surrounding voxels. As in most cases of vision based object detection network, firstly SurroundOcc extracts multi-scale features from multi-view images , then multi-scale features are lifted to the 3D volume space by using spatial cross attention. To progressively upsample the volume features, SurroundOcc applies 3D convolutions, then imposes supervision at multiple levels to enhance system performance.

1.2. SparseOcc

Traditional occupancy maps divide the environment into dense grid cells, where each single cell represents the probability of an object being present. Instead of densely representing the entire environment, SparseOcc [4] focus on areas, where voxels are likely to be occupied by any object. SparseOcc only predicts the occupancy of sparse voxels in surroundings, since the most part of the grid cells are empty, resulting in a more efficient representation.

1.3. RayIOU

RayIoU [4] is a ray-wise evaluation metric designed for occupancy prediction tasks, used as main metric in this challenge. Compared with voxel-based metrics, RayIoU accounts for this inconsistency by evaluating occupancy along Lidar-rays, providing a more consistent and precise measurement of occupancy. RayIoU avoids taking unscanned voxels into consideration and handles depth inconsistencies more effectively.

1.4. Dataset

The OpenOcc dataset [5] is used in this track, which is based on the Nuscenes Dataset. It contains 17 classes, including free class. For each grid cell, there is a label, stands for the class id, and a 2d-vector, stands for the flow.
To predict occupancy and flow, based on cameras, following SparseOcc, a simple yet effective flow predictor is designed. Secondly to speed up training process, multi-stage training process is proposed, thus to save time. Experiments have demonstrate the efficiency and effectiveness of our model.

2. Approach

2.1. System Design

Following the baseline design, SparseOcc, the Resnet-50 [2] is used as backbone to extract multi-scale features. Then multi-scale features are fed into FPN network, number of stages is set to 4, finally all features are processed in SparseOccHead, in which, a sparse voxel decoder to predict class-agnostic occupancy status of sparse grid cells, and a mask transformer decoder to predict semantics and instances of those occupied cells [4].
Photometric augmentation techniques such as random brightness, random contrast, random saturation and random color space conversion, are used in this approach. To accelerate data augmentation process, this module is designed to be running on GPU.

Method	w_{flow}	RayIOU	mAVE	Score
SparseOcc-8	2.0	0.316	0.4094	0.3434
SparseOcc-8-f	2.0	0.3286	0.3909	0.3567

Table 1. Results.

To predict occupancy flow of the voxels, a linear flow predictor is designed in the sparse voxel decoder. Same to the occupancy voxel decoder, there are 3 levels of flow prediction, directly connected with each level of query features, input dimension $[B, K, C]$, where B is the batch size, K is the number of queries, C is the embedding dimension. The hidden size of neurons is set to $2 * C$.

2.2. Loss Design

There are 6 type of losses, BCE occupancy loss L_{occ}, MSE flow loss L_{flow}, CE semantic loss $L_{semantic}$, focal loss L_{focal}, mask loss L_{mask}, dice loss L_{dice}.

The total loss is defined as:

$L = w_{occ} * L_{occ} + w_{flow} * L_{flow} + w_{semantic} * L_{semantic} + w_{focal} * L_{focal} + w_{mask} * L_{mask} + w_{dice} * L_{dice}$

Note that there are several levels of losses, such as occupancy loss, semantic loss and etc.

3. Experiments

3.1. Implementation Details

One piece of RTX4080 is used for the model training and testing, with a batch size of 2. The Resnet50 is chosen as backbone, pretrained on Coco dataset [3], provided by MMDetection [1]. The total number of data samples for training reaches 28139, and 6019 for validation. No extra data is used in the training process.

During first training stage, the overall framework is optimized end-to-end with the AdamW optimizer for 12 epoches, learning rate is set 2e-4, batch size is set 2, fp16 is enabled. The image size is set $[256, 704]$, number of frames is set 8.

For the second training stage, or fine-training stage, there are 2 possible methods to improve system performance. One is to use more frames, such as set number of frames to 16, as described in SparseOcc, the other is to use larger image sizes, such as set resolution to $[512, 1408]$.

3.2. Results

The submission results are listed in Tab. 1. The best result ranks the 7th place on the public leaderboard.

SparseOcc-8-f means SparseOcc with 8 frames and fine tuned from model SparseOcc-8, learning rate set to 1e-4.

4. Conclusion

In this technical report, a flow predictor is designed with multi-scale supervision to achieve better results. Combined with SparseOccHead the system is proved to be effective and efficient. Because of limited time and resource, no more ablation study is conducted, such as stronger backbones to replace Resnet50, more frames are used, test time augmentation and etc.

References

[1] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 2

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 1

[3] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll'a r, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. 2

[4] Haisong Liu, Yang Chen, Haiguang Wang, Zetong Yang, Tianyu Li, Jia Zeng, Li Chen, Hongyang Li, and Limin Wang. Fully sparse 3d occupancy prediction, 2024. 1

[5] Chonghao Sima, Wenwen Tong, Tai Wang, Li Chen, Silei Wu, Hanming Deng, Yi Gu, Lewei Lu, Ping Luo, Dahua Lin, and Hongyang Li. Scene as occupancy. 2023. 1

[6] Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Jie Zhou, and Jiwen Lu. Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. *arXiv preprint arXiv:2303.09551*, 2023. 1