SurroundOcc: Multi-Camera 3D Occupancy Prediction for Autonomous Driving

Authors: Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Jie Zhou, Jiwen Lu

Affiliations: Beijing National Research Center for Information Science and Technology, Tsinghua University, Tianjin University, PhiGent Robotics

Presenter: Neil Zarghami, MSEE, UCR

Introduction

Subject

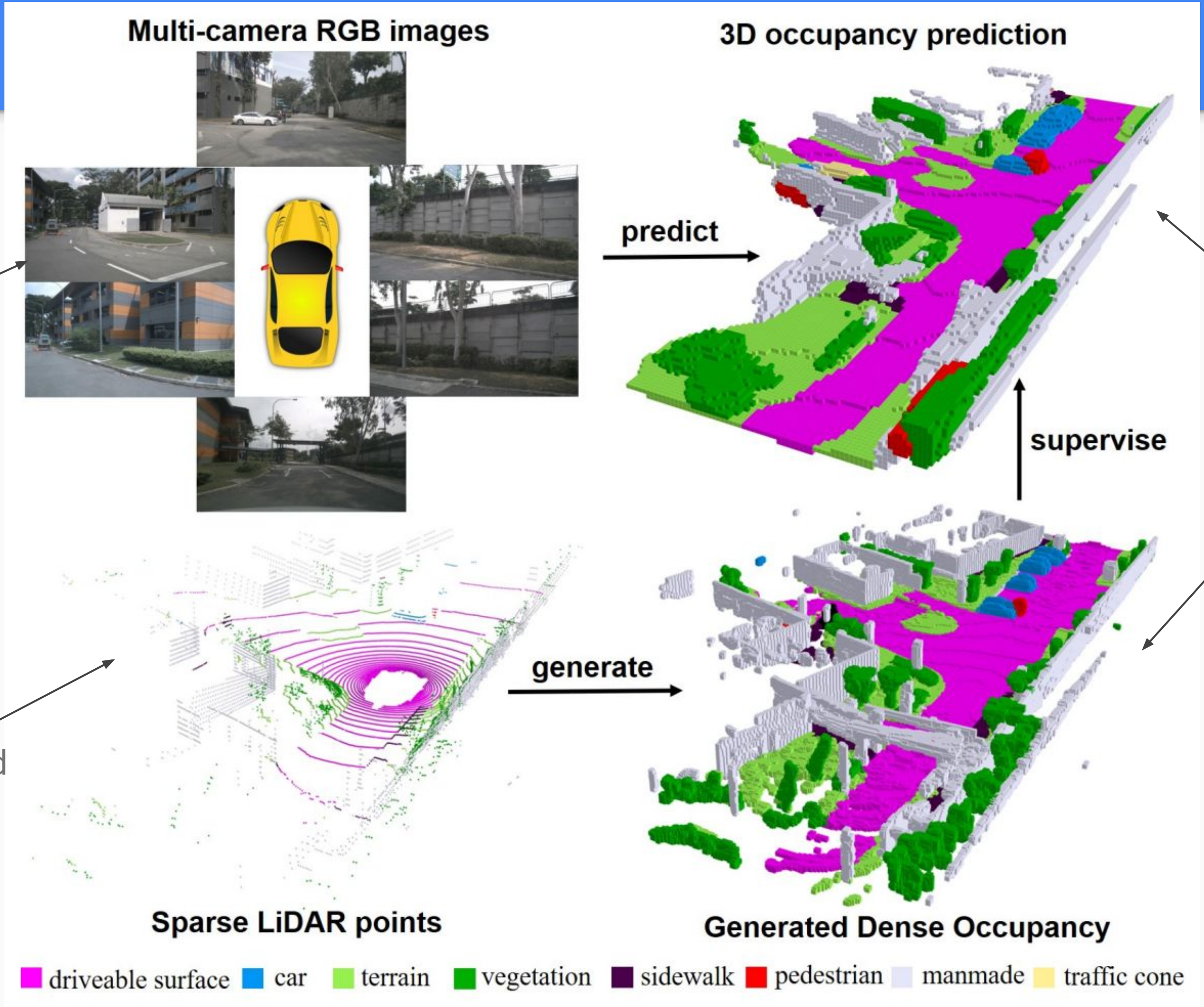
3D scene understanding in autonomous driving

Problem

Current **3D object detection** is not accurate in predicting diverse structures from camera images alone

Objective

To introduce the **SurroundOcc** method for predicting **3D occupancy** using multi-camera image processing



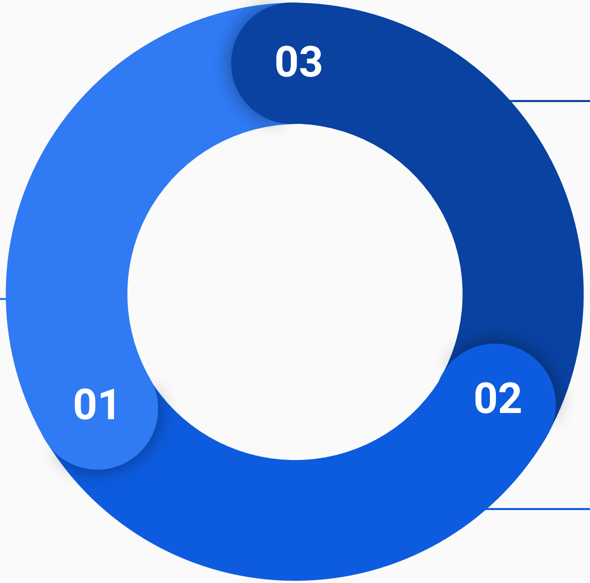
Multiple cameras

SurroundOcc interpretation of the data

Supervised by LiDAR points

Reconstructed and given semantic labeling

Related Work



Voxel-based Scene Representation

3D Scene Reconstruction

Vision-based 3D Perception

Related Work

Voxel-based Scene Representation

01

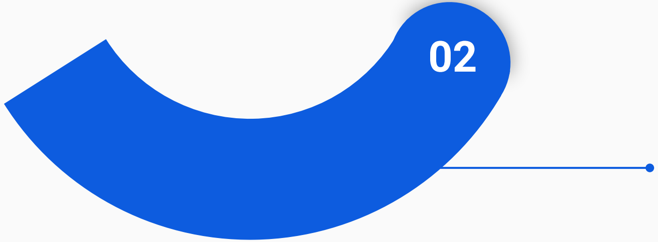
- Transforms 3D spaces into a spatial grid of voxels

- **MonoScene** pioneered outdoor scene reconstruction using RGB inputs
- **TPVFormer** extended this approach to multi-camera 3D semantic occupancy prediction

Related Work

- Direct reconstruction from RGB to 3D geometry using **SurfaceNet** and **Atlas** (MVS) methods
- Unfortunately, designed for indoor environments

- **NeuralRecon** and **TransformerFusion** fuse features from different angles and views for more precise 3D reconstructions

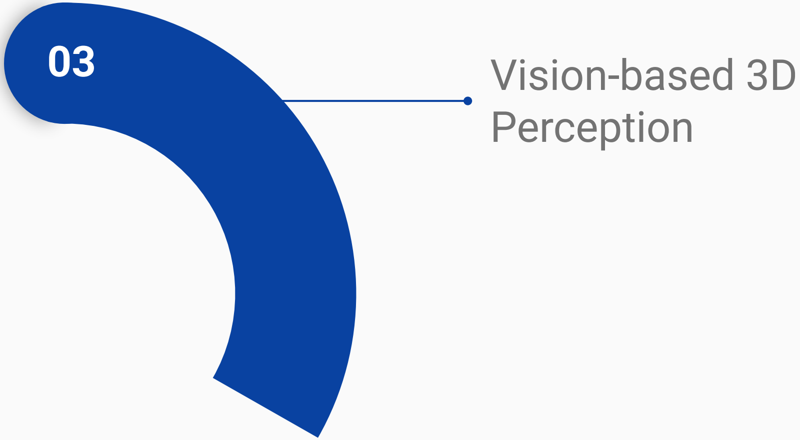


3D Scene Reconstruction

Related Work

- Vision-based methods to create 3D structures from image via monocular depth prediction and structure from motion (SfM)
- Misses fine details since interpreting from 2D images has its limits

- **BEV** was enhanced to **BEVFormer** using this method



Related Work

03

Vision-based 3D Perception

Voxel-based Scene Representation

01

02

3D Scene Reconstruction

Approach

Approach

Step 1

Feature Extraction

Multi-scale feature maps from each image using a 2D backbone network

Step 2

Spatial 2D-3D Attention

Lifting multi-camera image information to 3D volume features

Step 3

3D Convolutions

Upsampling low-resolution volume features to obtain fine-grained 3D representations

Approach

Step 1

Feature Extraction

Multi-scale feature maps from each image using a 2D backbone network

Step 2

Spatial 2D-3D Attention

Lifting multi-camera image information to 3D volume features

Step 3

3D Convolutions

Upsampling low-resolution volume features to obtain fine-grained 3D representations

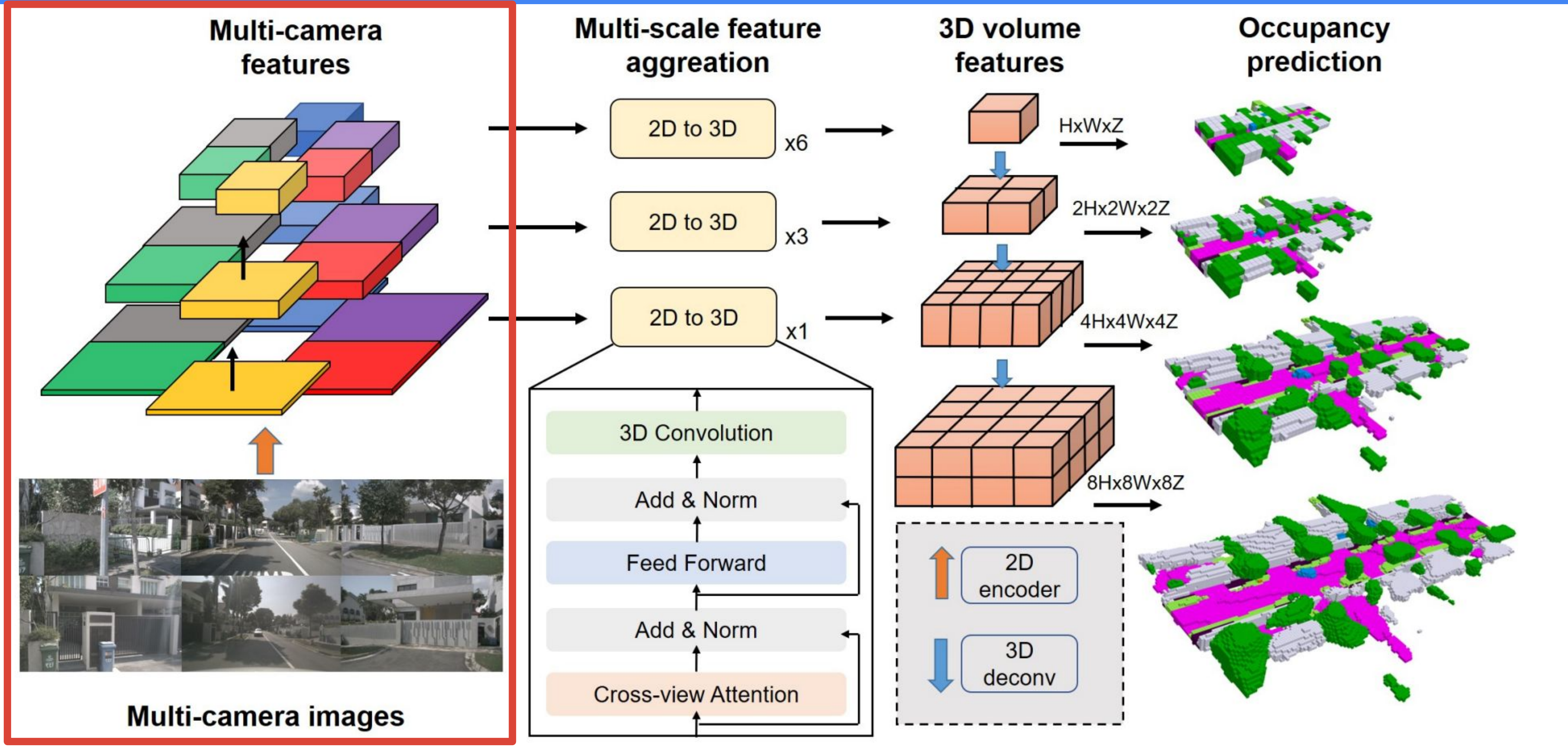


Figure 2. The pipeline of the proposed method. First, we use a backbone to extract multi-scale features of multi-camera images. Then we adopt 2D-3D spatial attention to fuse multi-camera information and construct 3D volume features in a multi-scale fashion. Finally, the 3D deconvolution layer is used to upsample 3D volumes and occupancy prediction is supervised in each level.

Approach

Step 1

Feature Extraction

Multi-scale feature maps from each image using a 2D backbone network

Step 2

Spatial 2D-3D Attention

Lifting multi-camera image information to 3D volume features

Step 3

3D Convolutions

Upsampling low-resolution volume features to obtain fine-grained 3D representations

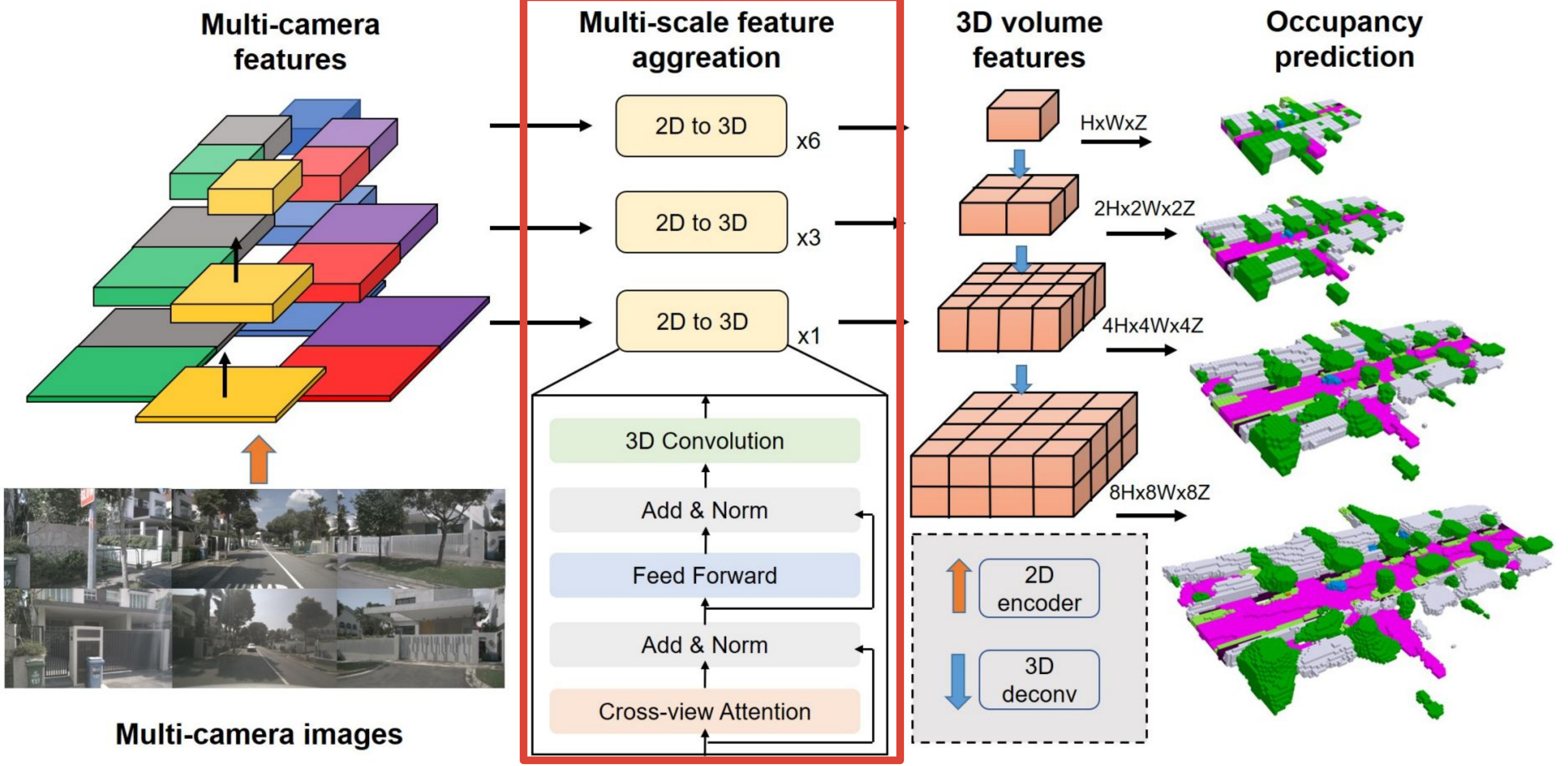


Figure 2. The pipeline of the proposed method. First, we use a backbone to extract multi-scale features of multi-camera images. Then we adopt 2D-3D spatial attention to fuse multi-camera information and construct 3D volume features in a multi-scale fashion. Finally, the 3D deconvolution layer is used to upsample 3D volumes and occupancy prediction is supervised in each level.

Approach

Step 1

Feature Extraction

Multi-scale feature maps from each image using a 2D backbone network

Step 2

Spatial 2D-3D Attention

Lifting multi-camera image information to 3D volume features

Step 3

3D Convolutions

Upsampling low-resolution volume features to obtain fine-grained 3D representations

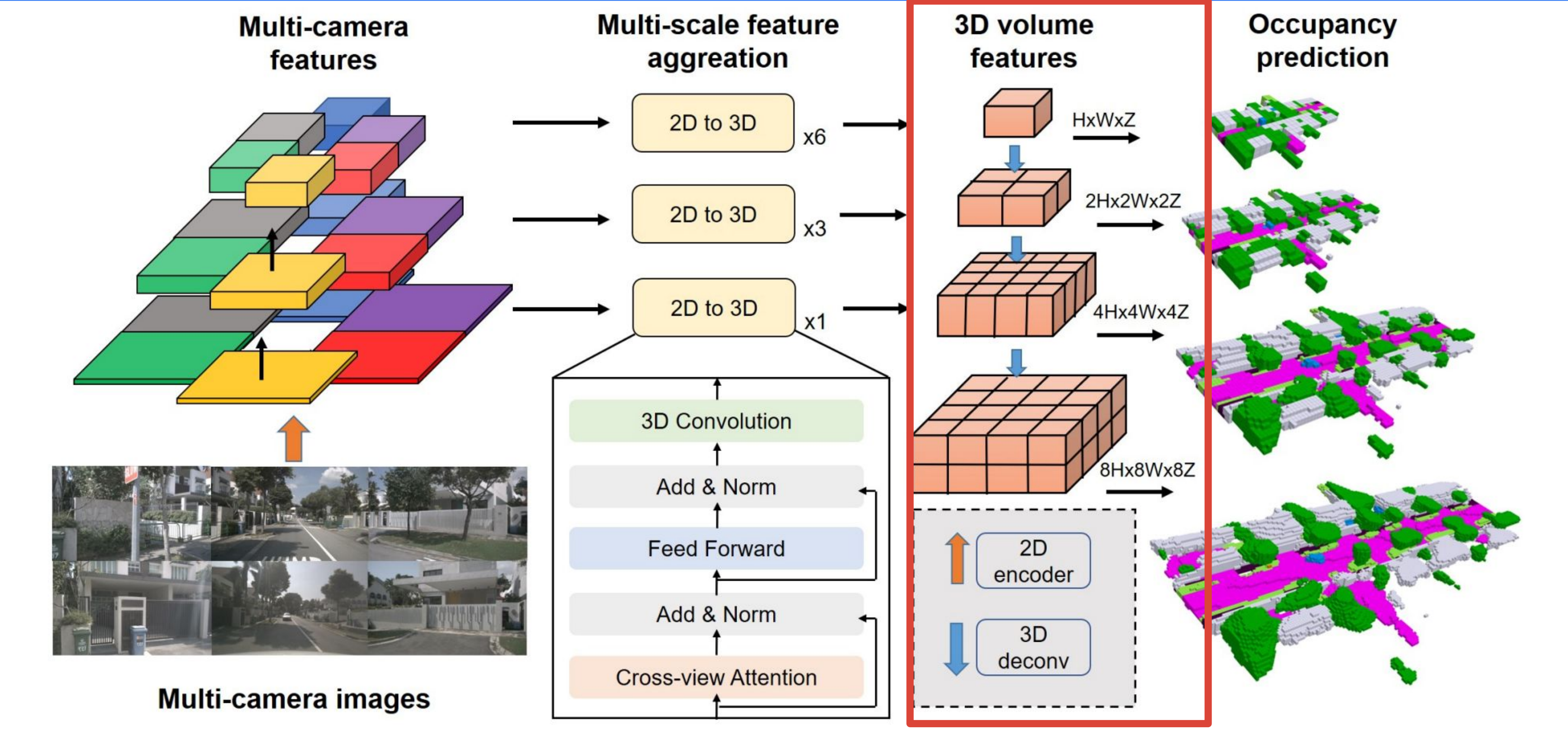


Figure 2. The pipeline of the proposed method. First, we use a backbone to extract multi-scale features of multi-camera images. Then we adopt 2D-3D spatial attention to fuse multi-camera information and construct 3D volume features in a multi-scale fashion. Finally, the 3D deconvolution layer is used to upsample 3D volumes and occupancy prediction is supervised in each level.

Approach Formulas

- Problem Formulation

$$V = G(I^1, I^2, \cdots I^N) \tag{1}$$

- 2D-3D Spatial Attention

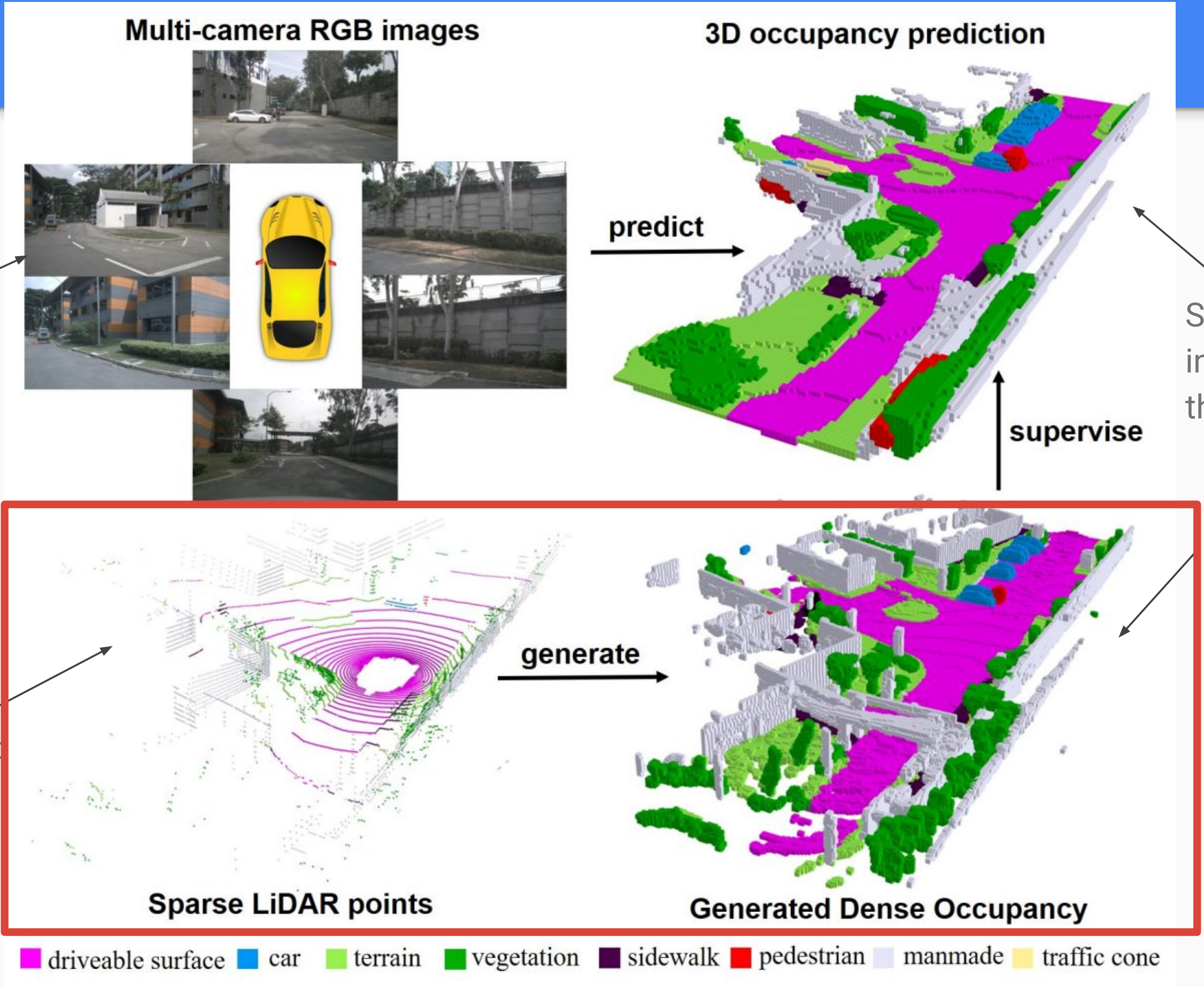
$$\text{DeformAttn}(q,p,x) = \sum_{i=1}^{N_{\text{head}}} \mathcal{W}_i \sum_{j=1}^{N_{\text{key}}} \mathcal{A}_{ij} \cdot \mathcal{W}'_i x(p + \Delta p_{ij})$$

$$F^p = \frac{1}{|\mathcal{V}_{\text{hit}}|} \sum_{i \in \mathcal{V}_{\text{hit}}} \text{DeformAttn}(Q^p, \mathcal{P}(q^p, i), X_i) \tag{2}$$

- Multi-scale Occupancy Prediction

$$Y_j = F_j + \text{Deconv}(Y_{j-1}) \tag{3}$$

Dense Occupancy
Ground Truth Generation



Multiple cameras

SurroundOcc interpretation of the data

Supervised by LiDAR points

Reconstructed and given semantic labeling

Dense Occupancy Ground Truth Generation

Data Fusion

Multi-frame LiDAR scans of dynamic and static scenes

+

Reconstruction

Filling holes and voxelizing the mesh to generate dense occupancy labels

→

Result

Dense Occupancy Ground Truth Generation for supervising models

Dense Occupancy Ground Truth Generation

Data Fusion

Multi-frame LiDAR scans of dynamic and static scenes

+

Reconstruction

Filling holes and voxelizing the mesh to generate dense occupancy labels

→

Result

Dense Occupancy Ground Truth Generation for supervising models

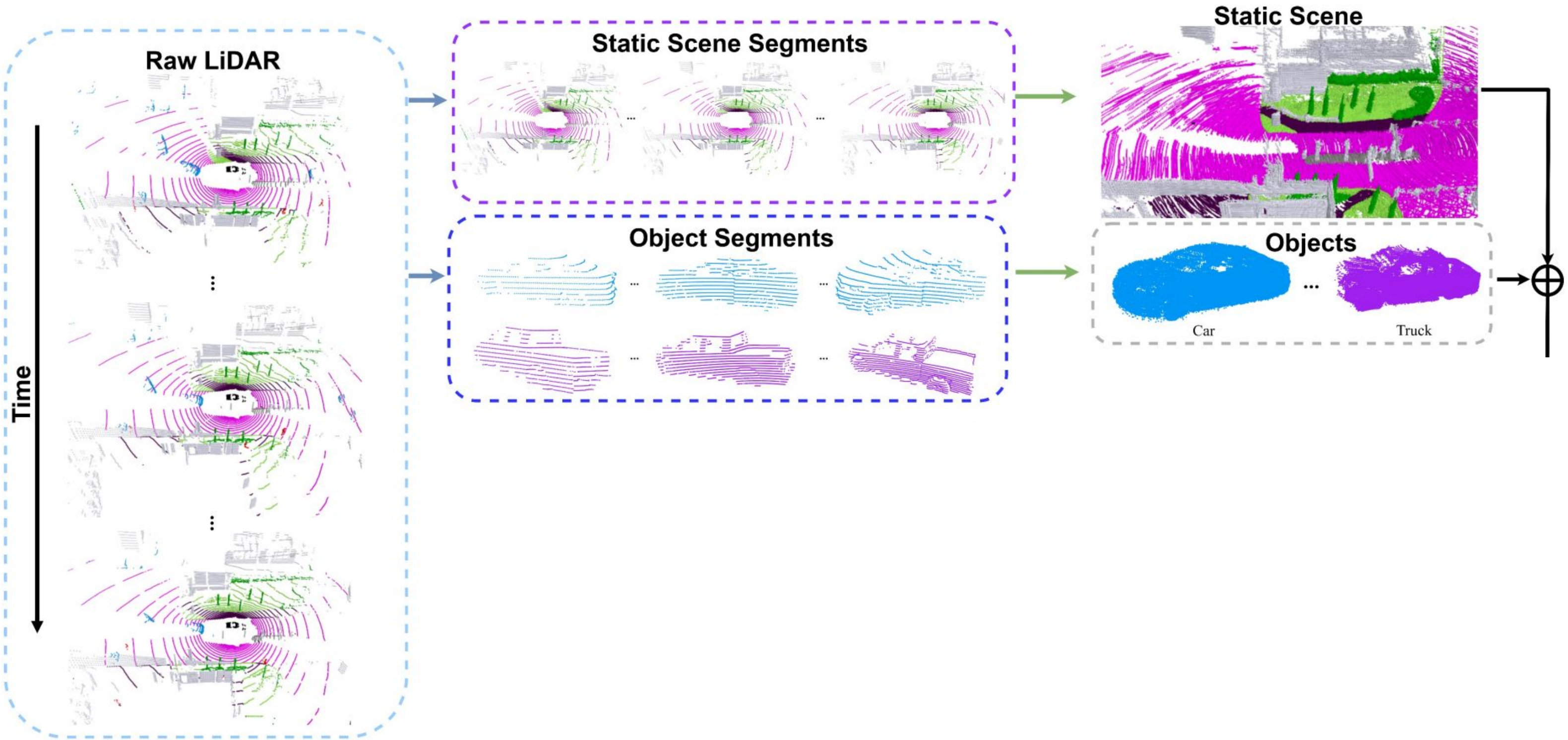


Figure 4. Dense occupancy ground truth generation. We first traverse all frames to stitch the multi-frame LiDAR points of dynamic objects and static scenes separately, and then merge them into a complete scene. Subsequently, we employ Poisson Reconstruction to densify the points and voxelize the resulting mesh to obtain a dense 3D occupancy. Finally, we use the Nearest Neighbor (NN) algorithm to assign semantic labels to dense voxels.

Dense Occupancy Ground Truth Generation

Data Fusion

Multi-frame LiDAR scans of dynamic and static scenes

+

Reconstruction

Filling holes and voxelizing the mesh to generate dense occupancy labels

→

Result

Dense Occupancy Ground Truth Generation for supervising models

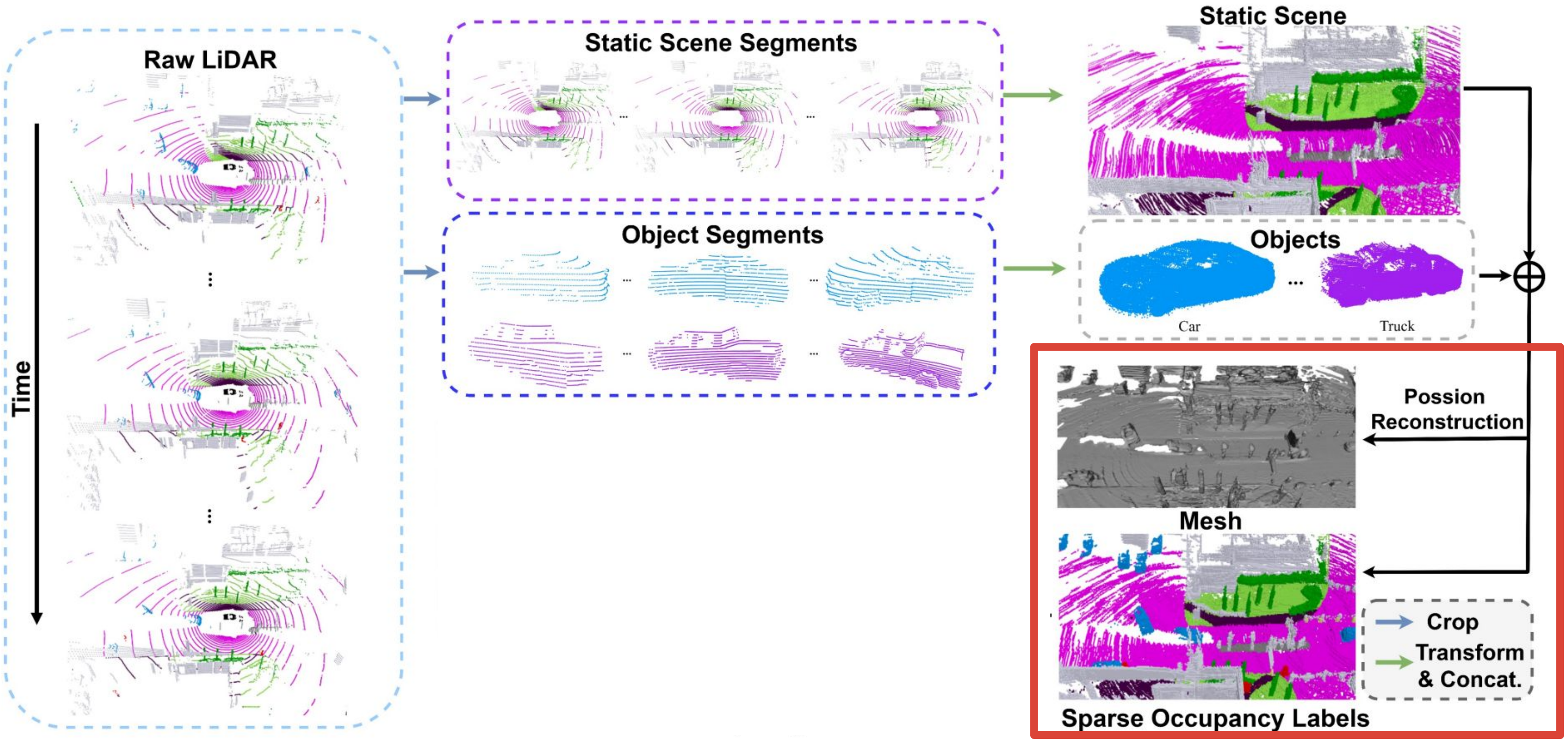


Figure 4. Dense occupancy ground truth generation. We first traverse all frames to stitch the multi-frame LiDAR points of dynamic objects and static scenes separately, and then merge them into a complete scene. Subsequently, we employ Poisson Reconstruction to densify the points and voxelize the resulting mesh to obtain a dense 3D occupancy. Finally, we use the Nearest Neighbor (NN) algorithm to assign semantic labels to dense voxels.

Dense Occupancy Ground Truth Generation

Data Fusion

Multi-frame LiDAR scans of dynamic and static scenes

+

Reconstruction

Filling holes and voxelizing the mesh to generate dense occupancy labels

→

Result

Dense Occupancy Ground Truth Generation for supervising models

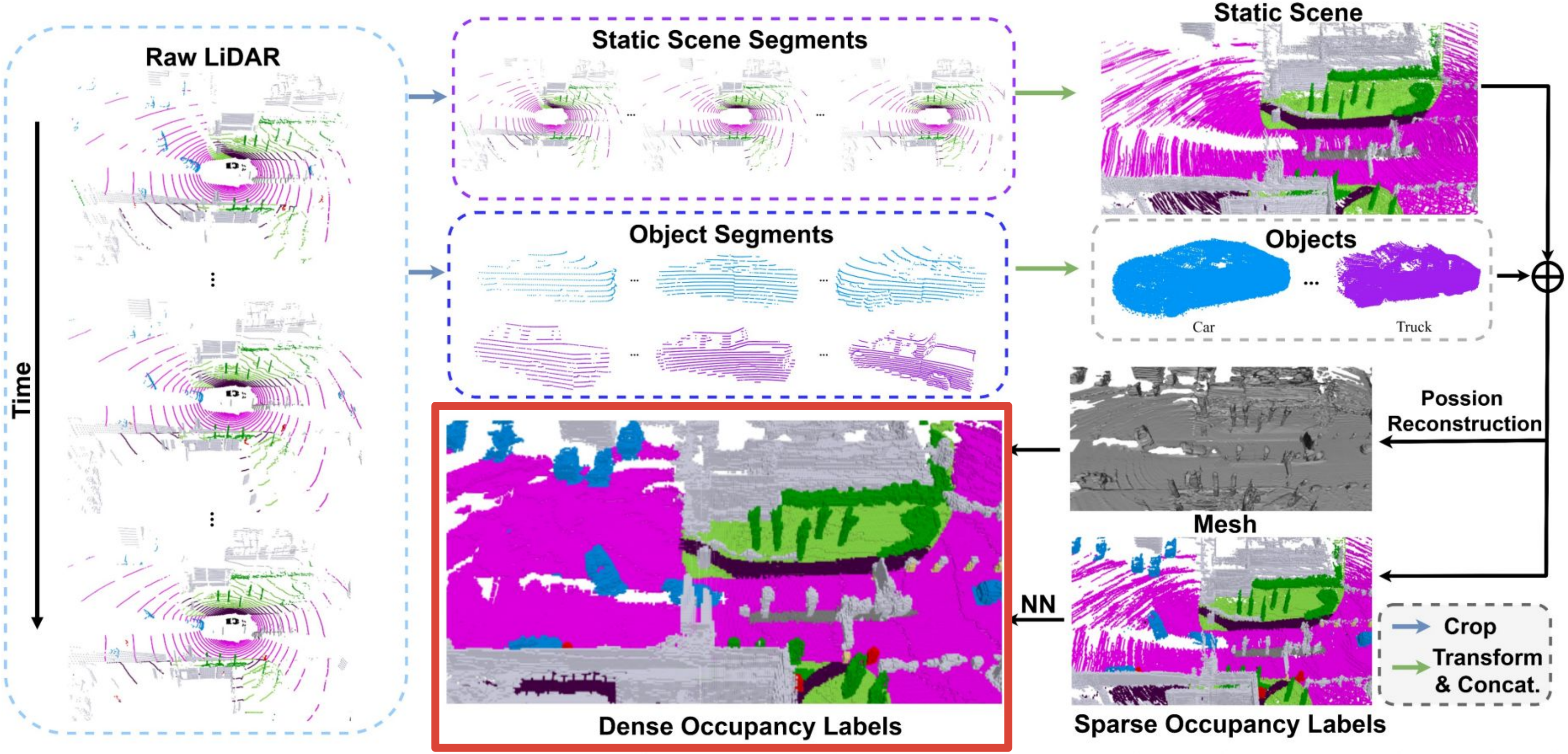
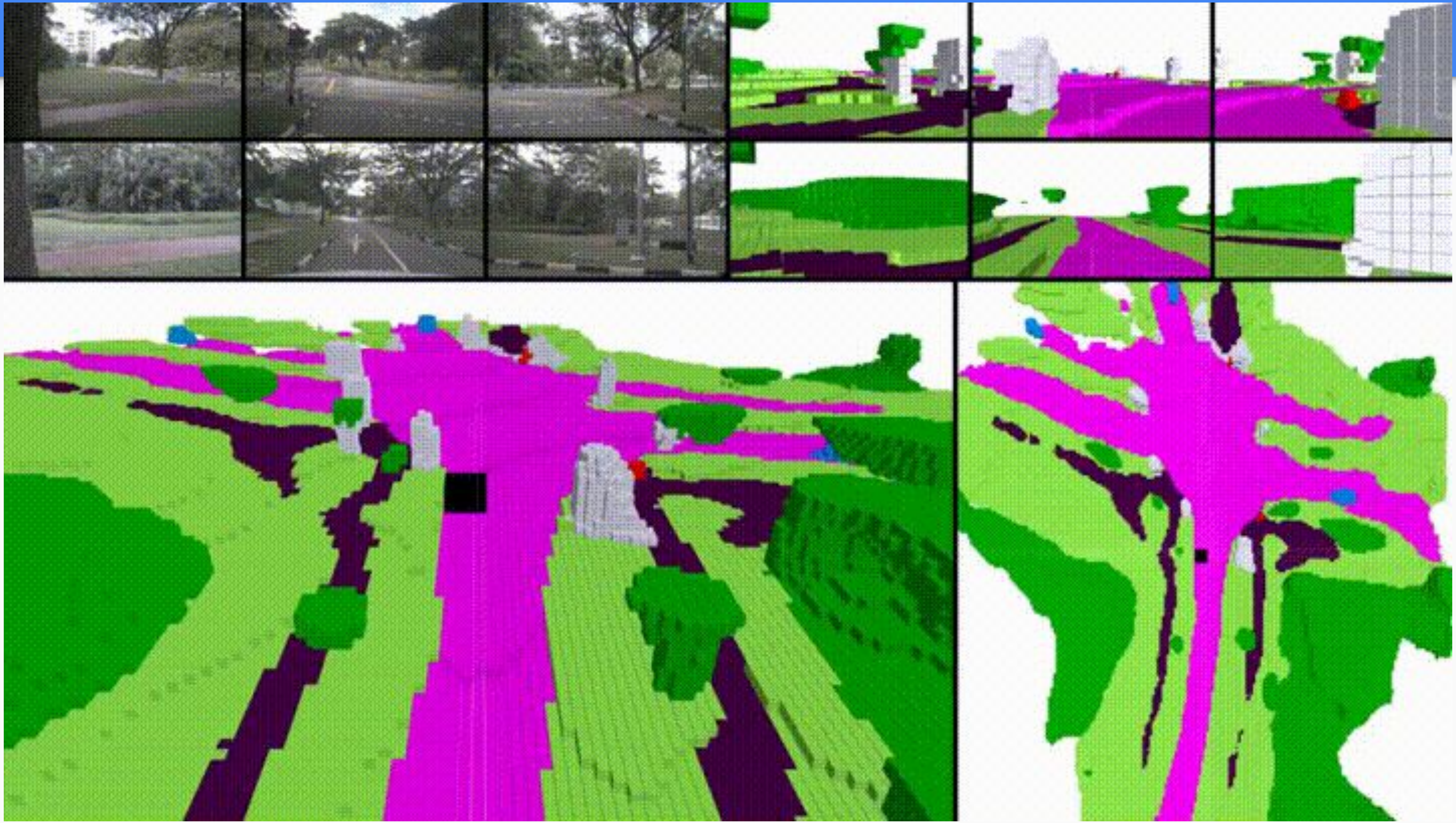
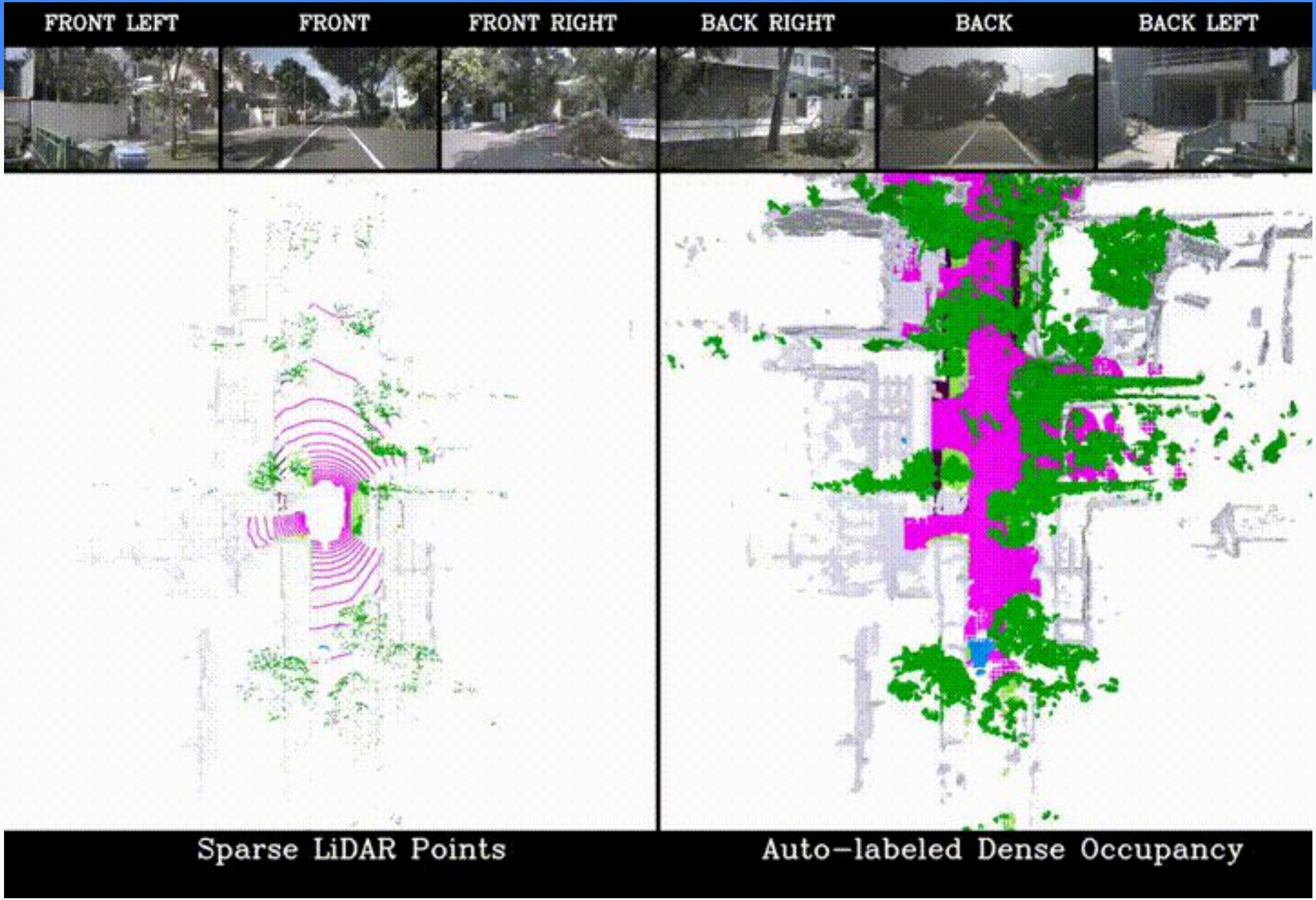


Figure 4. Dense occupancy ground truth generation. We first traverse all frames to stitch the multi-frame LiDAR points of dynamic objects and static scenes separately, and then merge them into a complete scene. Subsequently, we employ Poisson Reconstruction to densify the points and voxelize the resulting mesh to obtain a dense 3D occupancy. Finally, we use the Nearest Neighbor (NN) algorithm to assign semantic labels to dense voxels.

Experiments




FRONT LEFT
FRONT
FRONT RIGHT
BACK RIGHT
BACK
BACK LEFT
Sparse LiDAR Points
Auto-labeled Dense Occupancy

Experiments

Datasets

- nuScenes
  - Training and validating the model
- SemanticKITTI
  - Provided annotated LiDAR data for assessing the model's performance in monocular semantic scene completion tasks
- 8x RTX-3090 GPUs

Results

SurroundOcc achieved the best metrics in IoU and mIoU. This indicates superior **occupancy prediction accuracy**.

Experiments

"To further demonstrate the superiority of our method, we also conduct monocular 3D semantic scene completion on SemanticKITTI dataset. **Although our method is not designed for monocular perception** *and* cross-view attention will be ineffective for the monocular setting, our method **still achieves state-of-the-art performance** on this benchmark."

Experiments

Method	SC IoU	SSC mIoU	barrier	bicycle	bus	car	const. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	drive. suf.	other flat	sidewalk	terrain	manmade	vegetation
MonoScene [8]	23.96	7.31	4.03	0.35	8.00	8.04	2.90	0.28	1.16	0.67	4.01	4.35	27.72	5.20	15.13	11.29	9.03	14.86
Atlas [37]	28.66	15.00	10.64	5.68	19.66	24.94	8.90	8.84	6.47	3.28	10.42	16.21	34.86	15.46	21.89	20.95	11.21	20.54
BEVFormer [29]	30.50	16.75	14.22	6.58	23.46	28.28	8.66	10.77	6.64	4.05	11.20	17.78	37.28	18.00	22.88	22.17	13.80	22.21
TPVFormer [22]	11.51	11.66	16.14	7.17	22.63	17.13	8.83	11.39	10.46	8.23	9.43	17.02	8.07	13.64	13.85	10.34	4.90	7.37
TPVFormer*	30.86	17.10	15.96	5.31	23.86	27.32	9.79	8.74	7.09	5.20	10.97	19.22	**38.87**	21.25	24.26	**23.15**	11.73	20.81
SurroundOcc	**31.49**	**20.30**	**20.59**	**11.68**	**28.06**	**30.86**	**10.70**	**15.14**	**14.09**	**12.06**	**14.38**	**22.26**	37.29	**23.70**	**24.49**	22.77	**14.89**	**21.86**

Table 1. **3D semantic occupancy prediction results on nuScenes validation set.** Except TPVFormer [22], all methods are trained with dense occupancy labels. To fairly compare, we further use dense ground truth to train the TPVFormer, which is denoted as TPVFormer*.

Method	SC IoU	SSC mIoU	road (15.30%)	sidewalk (11.13%)	parking (1.12%)	other-grnd (0.56%)	building (14.1%)	car (3.92%)	truck (0.16%)	bicycle (0.03%)	motorcycle (0.03%)	other-veh. (0.20%)	vegetation (39.3%)	trunk (0.51%)	terrain (9.17%)	person (0.07%)	bicyclist (0.07%)	motorcyclist. (0.05%)	fence (3.90%)	pole (0.29%)	traf.-sign (0.08%)
LMSCNet [46]	31.38	7.07	46.70	19.50	13.50	3.10	10.30	14.30	0.30	0.00	0.00	0.00	10.80	0.00	10.40	0.00	0.00	0.00	5.40	0.00	0.00
3DSketch [11]	26.85	6.23	37.70	19.80	0.00	0.00	12.10	17.10	0.00	0.00	0.00	0.00	12.10	0.00	16.10	0.00	0.00	0.00	3.40	0.00	0.00
AICNet [27]	23.93	7.09	39.30	18.30	19.80	1.60	9.60	15.30	0.70	0.00	0.00	0.00	9.60	1.90	13.50	0.00	0.00	0.00	5.00	0.10	0.00
JS3C-Net [59]	34.00	8.97	47.30	21.70	19.90	2.80	12.70	20.10	0.80	0.00	0.00	4.10	14.20	3.10	12.40	0.00	0.20	0.20	8.70	1.90	0.30
MonoScene [8]	34.16	11.08	54.70	27.10	24.80	5.70	14.40	18.80	3.30	0.50	0.70	**4.40**	**14.90**	2.40	19.50	1.00	1.40	**0.40**	11.10	3.30	2.10
TPVFormer [22]	34.25	11.26	55.10	27.20	27.40	6.50	14.80	19.20	**3.70**	1.00	0.50	2.30	13.90	2.60	**20.40**	1.10	**2.40**	0.30	11.00	2.90	1.50
SurroundOcc	**34.72**	**11.86**	**56.90**	**28.30**	**30.20**	**6.80**	**15.20**	**20.60**	1.40	**1.60**	**1.20**	**4.40**	**14.90**	**3.40**	19.30	**1.40**	2.00	0.10	**11.30**	**3.90**	**2.40**

Table 3. **Monocular Semantic scene completion results on SemanticKITTI test set.** For fair comparison, we use the performances of RGB-inferred versions of the first four methods, which are reported in MonoScene [8]. Although our method is not designed for monocular perception, we still outperform other methods for a large margin.

Experiments

Table	Experiment
4	3D Scene Reconstruction Results
5	2D-3D Spatial Attention
6	Multi-scale Occupancy Prediction
7	Dense Occupancy Supervision

Method	Acc ↓	Comp ↓	Prec ↑	Recall ↑	**CD** ↓	**F-score** ↑
SurroundDepth [57]	1.747	1.384	0.261	0.353	3.130	0.293
AdaBins [4]	1.989	1.287	0.233	0.347	3.275	0.271
NeWCRFs [64]	2.163	1.233	0.214	0.348	3.396	0.257
Atlas [37]	**0.679**	1.685	0.407	0.546	2.365	0.458
TransformerFusion [6]	0.771	1.434	0.375	0.591	2.205	0.453
SurroundOcc	0.724	**1.226**	**0.414**	**0.602**	**1.950**	**0.483**

Table 4. **3D scene reconstruction results on nuScenes validation set**. F-score and CD are the main metrics.

Method	SC IoU	SSC mIoU
w/o spatial attention	29.78	17.34
BEV-based attention	30.45	18.94
Ours	**31.49**	**20.30**

Table 5. The ablation study of 2D-3D spatial attention. "w/o spatial attention" indicates that we average all multi-camera features in a grid.

Method	SC IoU	SSC mIoU
w/o multi-scale structure	30.41	18.22
w/o multi-scale supervision	31.16	19.73
Ours	**31.49**	**20.30**

Table 6. The ablation study of multi-scale occupancy prediction. "w/o multi-scale structure" means that we do not add multi-scale skip connection.

Supervision	SC IoU	SSC mIoU
sparse LiDAR points	11.96	12.17
sparse occupancy labels	30.58	18.83
dense occupancy labels	**31.49**	**20.30**

Table 7. The ablation study of dense occupancy supervision. The model trained with our dense occupancy ground truth is much better than that trained with sparse LiDAR points.

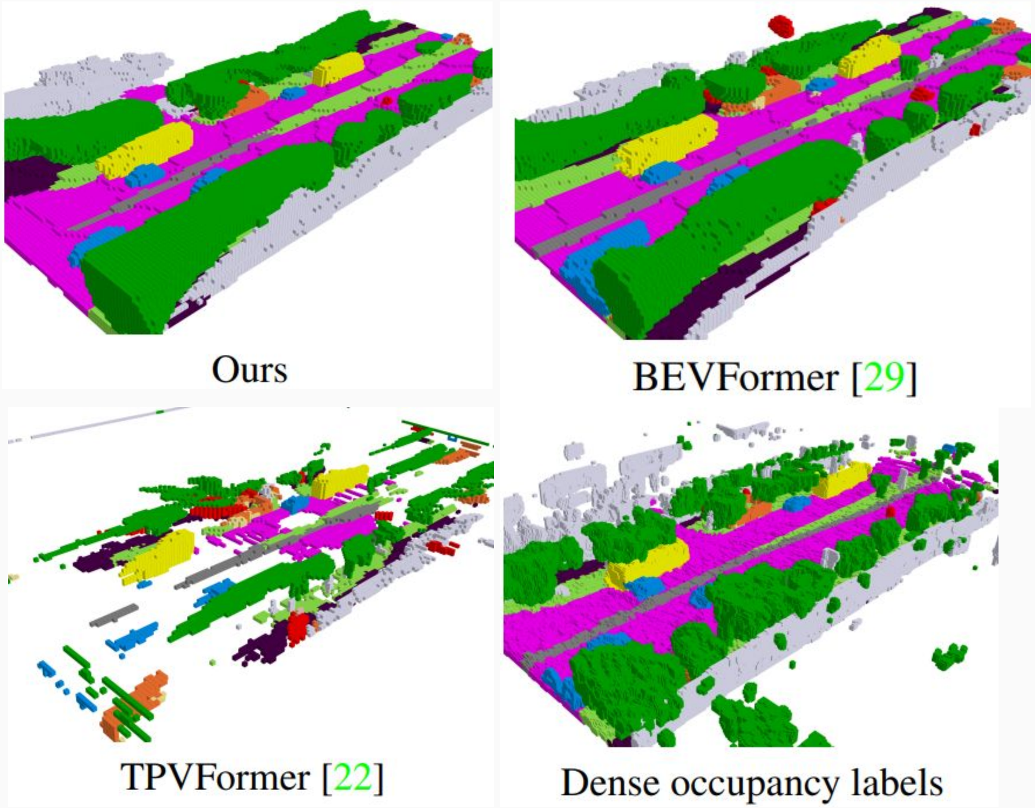
Experiments

Method	Latency (s)	Memory (G)
SurroundDepth [57]	0.73	12.4
NeWCRFs [64]	1.07	14.5
Adabins [4]	0.75	15.5
BEVFormer [29]	**0.31**	**4.5**
TPVFormer [22]	0.32	5.1
MonoScene [8]	0.87	20.3
Ours	0.34	5.9

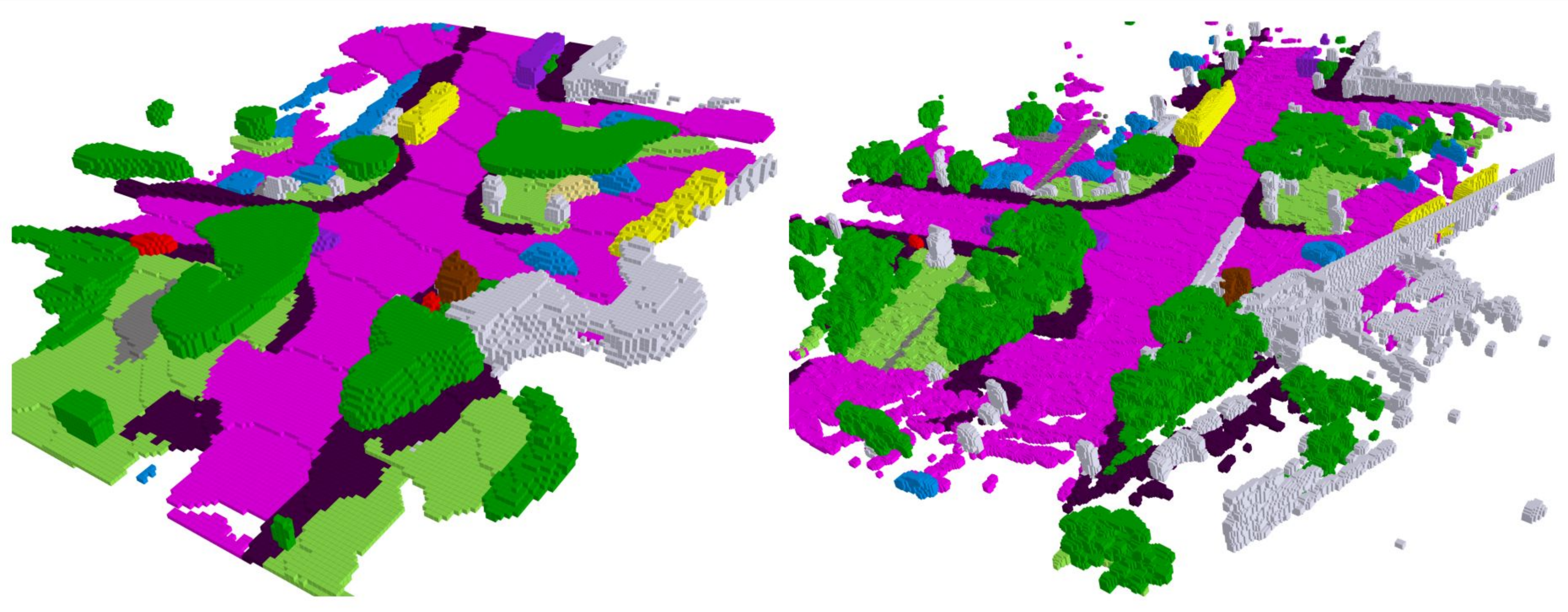
Table 8. The model efficiency of different methods. The experiments are conducted on one RTX 3090 with six multi-camera images, whose resolutions are 1600x900.

"Compared with BEVFormer, our method **slightly** increases inference time and memory and we think the increased burden is acceptable"

$$
\begin{aligned}
\mathrm{IoU} &= \frac{TP}{TP + FP + FN} \\
\mathrm{mIoU} &= \frac{1}{C}\sum_{i=1}^{C}\frac{TP_i}{TP_i + FP_i + FN_i}
\end{aligned} \tag{4}
$$



Experiments



Conclusion

- SurroundOcc **advances multi-camera 3D scene reconstruction** with high precision in occupancy prediction

- Future work aims to extend **single frame** to **multi frame** occupancy for occupancy flow features

Questions?