SurfOcc: Surface-based Feature Lifting for Vision-centric 3D Occupancy Prediction

Tonghui Ye[1,2], Zhi Gao[1,2]✉, Zhipeng Lin[3], Xinyi Liu[1], and Ronghe Jin[1]

[1] School of Remote Sensing and Information Engineering,
Wuhan University, Wuhan, China
[2] Hubei Luojia Laboratory, Wuhan 430070, China
[3] Department of Mechanical and Automation Engineering,
The Chinese University of Hong Kong, Hong Kong SAR 999077, China
{ythyth, gaozhinus, liuxy0319, huanhexiao}@whu.edu.cn,
zplin@link.cuhk.edu.hk

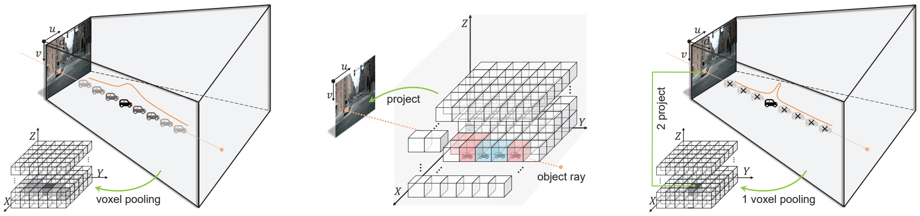
Abstract. 3D occupancy prediction has been an emerging trend in 3D perception for its superiority in preserving exquisite geometric and semantic details. However, existing vision-based approaches either leave features unrefined or neglect depth ambiguity due to defective 2D-to-3D feature lifting modules, leading to imprecise prediction results. In this paper, we introduce SurfOcc, a vision-centric 3D occupancy prediction framework which addresses these limitations fundamentally. SurfOcc decouples the learning process of observed surfaces and occluded regions while seamlessly integrating them into an end-to-end architecture. Specifically, we first propose surface-based feature lifting to precisely locate observed surfaces and enhance the selected surface voxels via cross-attention during feature lifting. Then we design a feature diffuser which incorporates both local and global features to diffuse the reliable surface features to occluded regions. Experiments show that SurfOcc achieves state-of-the-art performance with 13.75 mIoU on SemanticKITTI and 42.38 mIoU on Occ3D-nuScenes, which also demonstrates the potential of SurfOcc in handling occlusion situations. Code is available at https://github.com/sullicsullic/SurfOcc.

Keywords: Autonomous Driving · 3D Computer Vision · 3D Occupancy Prediction · Feature Lifting

1 Introduction

An essential component of automation applications like robotics and autonomous driving is the ability to perceive the environment. Even while 3D object detection[34,27,32,49] has advanced significantly, there is a growing interest in 3D occupancy prediction[3,24,48] because it can preserve more details and even describe irregular items. With superior geometry demonstration[42] and semantic preservation[21,22] capabilities, voxel-based representation is intrinsically well-suited for 3D feature learning. This makes it straightforward for 3D occupancy

✉ Corresponding author.



(a) Lift-Splat-based (b) Attention-based (c) Surface-based

Fig. 1: **Comparison of feature lifting methods. (a)** Lift-Splat-based methods can only represent observed surface in each optical ray but can't depict the occluded regions. **(b)** Attention-based methods lift same feature to voxels along an object ray. Due to their alignment with the car's optical ray, the red voxels receive the car's feature even though only the blue voxels depict the car's actual location. **(c)** Only the surface voxels (positions with the highest depth confidence) are enhanced as observed surfaces. Through cross-attention, surface voxels are projected onto 2D image feature maps to obtain updates.

network to depict both foreground and background objects with different shapes, using different geometric structures of 3D cubes. With the proliferation of datasets[41,45,42,23] and solutions[24,17,55,52,18], 3D occupancy prediction is paving the way for novel developments in 3D perception.

Relying on explicit depth measurements, LiDAR solutions[38,8,50] have been leading in performance for a long time. Some works like 3DPPE[39] even transform images into 3D points for detection. However, LiDAR sensors suffer from high cost and sparse scanned points, which motivates the study of vision-centric approaches for low cost, rich visual cues and high generality. As camera images store scene information in 2D planes, feature lifting, the process of constructing 3D features from 2D inputs, is crucial for generating accurate 3D outputs.

Up to now, there have been primarily two solutions for feature lifting in vision-centric 3D perception (see figs. 1a and 1b). Methods[16,26,55] based on Lift-Splat (see fig. 1a) employ a straightforward lift and splat approach[35,36] that is ubiquitous in 3D object detection[32,28,26,16], adhering to a 2D-to-3D paradigm. By taking the outer product of estimated depths and 2D features, they first lift 2D image features into 3D frustums. Next they sample the 3D voxel features from the 3D frustums using a pooling mechanism[35,16,15]. oh[39] Though having depth priors, this type of solution only represents observed surface in each optical ray, leaving occluded regions unconcerned.

Instead of Lift-Splat, some other works[46,27,31,48,47] utilize attention mechanism[43,5,56] for feature lifting, following a 3D-to-2D paradigm. To begin with, key-value pairs are taken from image features, and a predefined number of embeddings are initialized to serve as 3D queries in the ego or LiDAR coordinate system. Then the 3D queries are projected to 2D image feature maps to get the corresponding pixels according to camera parameters. Finally, the 3D features are lifted through cross-attention between 3D queries and 2D image features.

Though dense 3D features are constructed, the lack of depth information leads to feature ambiguity. Specifically, same feature is sampled by voxels along a ray (see fig. 1b), hindering the model from distinguishing between instances and air.

To address the aforementioned limitations, we propose an end-to-end framework for 3D occupancy prediction called SurfOcc. Since sensors can only observe the visible surfaces of objects and cannot access information about internal parts concealed by self-occlusion[23], our geometrically-motivated approach is designed to propagate features from observed surfaces to occluded regions. As shown in fig. 2, rather than predicting the entire scene synchronously, we conceptually divide the entire prediction process into two phases and integrate them in an end-to-end manner. Specifically, during feature lifting phase, to mitigate depth ambiguity and refine lifted features, we propose surface-based feature lifting (see fig. 1c) to select and enhance surface voxels (positions with the highest depth confidence) as observed surfaces, which are the few reliable informational units in 3D features. During feature diffusion phase, we transform voxel features into pseudo points and design a feature diffuser to diffuse the reliable features to occluded regions, utilizing local and global features. Our experiments demonstrate that our two-phase prediction scheme is particularly effective in handling complex scenes, especially those involving occlusion. Our contributions are summarized as follows:

- We propose SurfOcc, an end-to-end 3D occupancy prediction framework which predicts the observed surfaces in feature lifting phase and associates the occluded regions in feature diffusion phase. The specially designed two-phase scheme ensures the model to accurately acquire information from observable surfaces and use it as a basis to make predictions at the scene level.
- We propose a novel feature lifting method, termed surface-based feature lifting, which accurately lifts image features into 3D space by locating and enhancing surface voxels. Grounded in geometry, our method resolves the long-standing issues of feature unrefinement and ambiguity inherent in existing feature lifting methods.
- We evaluate the proposed SurfOcc on both monocular and surround-view settings. SurfOcc surpasses existing approaches and achieves 13.75 mIoU on SemanticKITTI[1] and 42.38 mIoU on Occ3D-nuScenes[41]. Experimental results demonstrate the potential of SurfOcc in handling occlusion situations and advancing scene understanding.

2 Related Works

2.1 3D Occupancy Prediction

3D occupancy prediction generates 3D volumetric semantics to depict the detailed occupancy states and semantics of a scene. SSCNet[40] is the first to jointly inferences both geometry and semantics leveraging Truncated Signed Distance Function (TSDF). The follow-ups usually enhance the geometrical information with explicit priors like depth[30,10,20,8,19], occupancy grids[38,50] or point cloud[37]. Recently, the pioneer work MonoScene[3] proposes the first vision-centric method using a 3D U-Net. TPVFormer[17] promotes 3D occupancy prediction to multi-view case with tri-perspective view representation. SurroundOcc[48] generates the 3D feature volume by cross-attention, which introduces depth ambiguity. Employing masked-attention[7], OccFormer[55] predicts the final occupancy in a dual and multi-scale manner. VoxFormer[24] tackles the lack of depth by generating voxel proposals at the first stage, but the networks in two stages are trained separately, which is complex and inelegant. SceneRF[4] proposes PrSamp to implicitly learn to correlate high mixture values with surface locations. Differently, we devise an end-to-end pipeline to seamlessly integrate the phase which predicts the observed surfaces with the phase which diffuses the features to occluded regions.

2.2 Feature Lifting Methods

Lift-Splat-based Feature Lifting. These methods[36,16,26,28,32] usually follow the mechanism pioneered by Lift-Splat-Shoot[35] and improved it from different perspectives. CaDDN[36] puts forward a linear-increasing discretization (LID) to provide balanced depth estimations. BEVdet[16] and BEVFusion[32] improve the computational efficiency by optimizing the underlying computational logic. BEVStereo[25] tries to enhance performance by improving depth quality using temporal stereo depth. However, depth distribution only characterizes the geometry of observed surfaces but fails to capture the overall geometric structure. In this work, we overcome this limitation by diffusing surface features to occluded regions to help our model understand the overall geometry.

Attention-based Feature Lifting. These methods[27,51,46,31,48,47] update 3D features leveraging cross-attention between 3D queries and 2D image features, but this paradigm usually introduces depth ambiguity. BEVFormer[27] is the first to lift 3D feature through attention at both spatial and temporal space, and an updated version[51] improves the performance via adding perspective information. SurroundOcc[48] introduces this paradigm into 3D occupancy prediction in a multi-scale manner, but the depth ambiguity is not been addressed. DA-BEV[54] proposes a depth-aware cross-attention to solve depth ambiguity and OPEN[13] advocates for focusing on the center of objects. In this work, we argue that it is imperative for the model to choose the observed surfaces (surface voxels) at an early stage, which is of great help in mitigating depth ambiguity.

3 Methodology

In this section, we propose SurfOcc, which predicts the observed surfaces and associates the occluded regions separately, rather than predicts the whole scene directly. We first introduce the overall architecture in section 3.1, after describing the problem formulation. In section 3.2 we present surface-based feature lifting

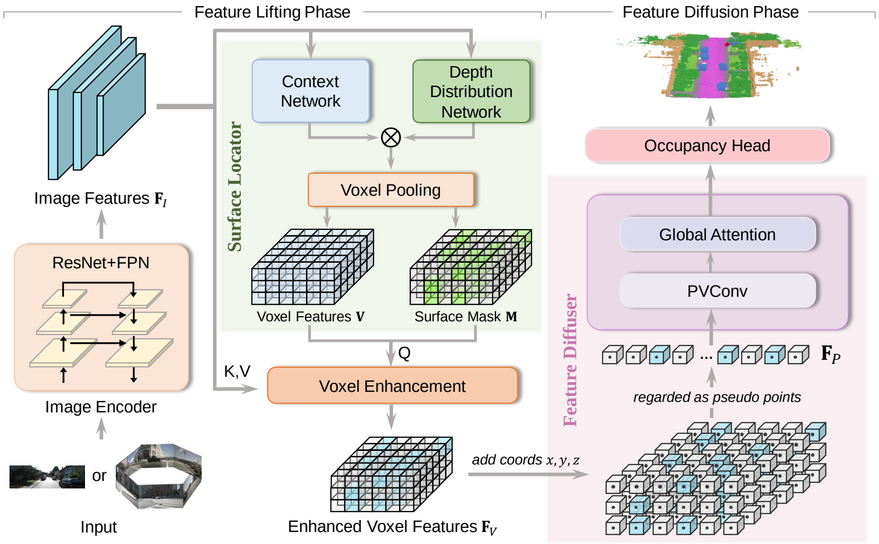


Fig. 2: **Overall architecture of SurfOcc.** There are two phases in the entire end-to-end pipeline. The feature lifting phase consists of three parts: the image encoder extracts image features from input image; the surface locator generates coarse voxel features and surface mask from image features; the voxel enhancement module takes the key-value pairs from image features and updates coarse voxel features. Enhanced voxel features are taken as pseudo points in feature diffusion phase, then both local features and global features are captured.

for constructing accurate 3D feature. Finally, we propose feature diffuser which propagates features to occluded regions in section 3.3.

Problem Formulation. 3D occupancy prediction aims to predict a dense semantic scene within a certain volume. Specifically, this task takes a monocular image $\mathbf{I} \in \mathbb{R}^{H^I \times W^I \times 3}$ or N surround-view images $\{\mathbf{I}_i \in \mathbb{R}^{H^I \times W^I \times 3}\}$ as input, where $i = 1, \ldots, N$. The output should be a labelled volume $\mathbf{V}_O \in C^{X^V \times Y^V \times Z^V}$, where X^V, Y^V, Z^V denote the length, width and height of the volume respectively. The labels are divided into $M + 1$ categories $C = \{c_0, c_1, \ldots, c_M\}$, with c_0 denoting the free voxel (air) and $\{c_1, \ldots, c_M\}$ denoting other semantic categories. Here M denotes the number of interested categories. Each voxel in $\mathbf{V}_O$ is occupied by a category in C.

3.1 Overall Architecture

As illustrated in fig. 2, SurfOcc has two main phases, namely feature lifting phase and feature diffusion phase. SurfOcc first locates observed surfaces as surface voxels in feature lifting phase, then diffuse these feature to occluded regions in feature diffusion phase. We argue that the feature lifting phase provides a robust foundation for subsequent feature diffusion, and the sparse surface features lifted during this phase play a vital role in the overall task.

In feature lifting phase, we propose surface-based feature lifting (section 3.2). During feature lifting, extracted multi-scale 2D features go through surface locator and then do voxel enhancement. Surface locator simultaneously constructs coarse voxel features $\mathbf{V}$ and predicts a binary surface mask $\mathbf{M}$ in which the value of 1 represents the postion of surface voxel. Then surface voxels are enhanced via deformable cross-attention with image features $\mathbf{F}_I$. The surface voxels contain reliable information which is lifted from 2D features guided by depth, and they represent the observed surfaces in the scene.

In feature diffusion phase, we introduce a feature diffuser (section 3.3) to propagate correct features to occluded regions. The diffuser refines local context on a per-voxel basis and aggregates global features to infer occluded regions. Finally a light-weighted occupancy head is used to upsample and make final predictions. To save computations, the volume spatial resolution $X \times Y \times Z$ of $\mathbf{V}, \mathbf{M}$ and $\mathbf{F}_V$ is lower than output resolution $X^V \times Y^V \times Z^V$.

3.2 Surface-based Feature Lifting

Surface-based feature lifting aims to lift 2D features to the correct positions of the 3D volume and mitigate feature ambiguity from the root. In brief, surface locator provides the locations of surface voxels and then these surface voxels are enhanced by cross-attention.

Surface Locator. Surface locator follows the paradigm of Lift-Splat-Shoot[35]. Based on this so-called forward projection paradigm, we made some adaptive changes. The structure of surface locator is shown in fig. 3. Image feature $\mathbf{F}_I$ is first processed by a context network and depth distribution network, which are adopted from CaDDN[36]. Typically, context network generates refined image features $\mathbf{F}_{con} \in \mathbb{R}^{H_F \times W_F \times C}$ and depth distribution network generates coarse depth distribution $\mathbf{D}_{dist} \in \mathbb{R}^{H_F \times W_F \times D}$, where H_F and W_F denote height and width of feature map respectively, C and D denote context channels and number of depth bins respectively. Then a frustum feature $\mathbf{G}_V$ is generated by taking the outer product of $\mathbf{D}_{dist}$ and $\mathbf{F}_{con}$, denoted as:

$$\mathbf{G}_V(u, v) = \mathbf{D}_{dist}(u, v) \otimes \mathbf{F}_{con}(u, v) \tag{1}$$

The outer product in eq. (1) is computed for each pixel, where (u, v) are the feature pixel location. Finally $\mathbf{G}_V$ is transformed to coarse voxel features $\mathbf{V} \in \mathbb{R}^{X \times Y \times Z \times C}$ leveraging voxel pooling, which is based on BEVDet[16,15].

During voxel pooling, every grid in frustum $\mathbf{G}_V$ is assigned to its nearest voxel in $\mathbf{V}$ and sum pooling is then performed. Based on this mechanism, voxel pooling is suitable for selecting surface voxels, which represent observed surfaces.

For each pixel in $\mathbf{D}_{dist}$, we set the value in the bin with the highest confidence among D bins to 1. Values in the other $D-1$ bins are set to zero. In this manner, a one-hot depth distribution $\mathbf{D}_{one-hot}$ is generated.

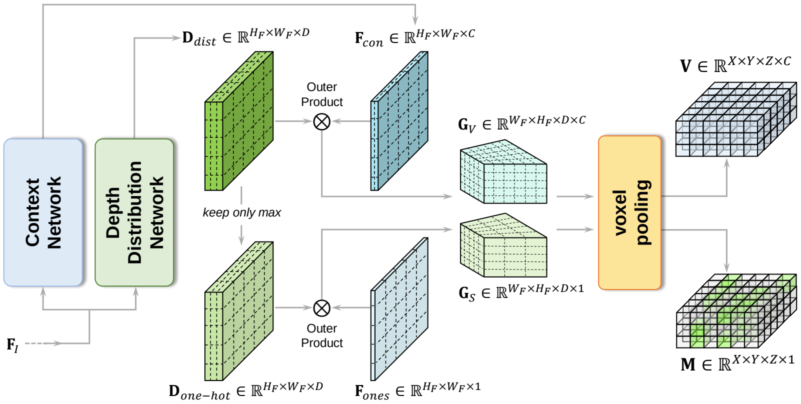

$\mathbf{D}_{dist} \in \mathbb{R}^{H_F \times W_F \times D}$
$\mathbf{F}_{con} \in \mathbb{R}^{H_F \times W_F \times C}$
Outer Product
$\mathbf{V} \in \mathbb{R}^{X \times Y \times Z \times C}$
$\mathbf{G}_V \in \mathbb{R}^{W_F \times H_F \times D \times C}$
Context Network
Depth Distribution Network
keep only max
voxel pooling
$\mathbf{G}_S \in \mathbb{R}^{W_F \times H_F \times D \times 1}$
Outer Product
$\mathbf{F}_I$
$\mathbf{M} \in \mathbb{R}^{X \times Y \times Z \times 1}$
$\mathbf{D}_{one-hot} \in \mathbb{R}^{H_F \times W_F \times D}$
$\mathbf{F}_{ones} \in \mathbb{R}^{H_F \times W_F \times 1}$

Fig. 3: **Surface locator.** The primary function of surface locator is to generate both coarse voxel features and surface mask. The following steps are included in their generation process: after obtaining the frustums through outer product using depth distribution and corresponding features, voxel pooling is performed to transform frustums into voxel representation.

Now that the goal is to select surface voxels, the only thing that matters is location information and the specific context information is unnecessary. We use an all-1 matrix $\mathbf{F}_{ones}$ to replace context feature, and the surface frustum $\mathbf{G}_S$ is computed as:

$$\mathbf{G}_S(u, v) = \mathbf{D}_{one-hot}(u, v) \otimes \mathbf{F}_{ones}(u, v) \tag{2}$$

Features in $\mathbf{G}_S$ contains the depth information derived from $\mathbf{D}_{one-hot}$, supervised by LiDAR provided by datasets. Via voxel pooling, a binary matrix $\mathbf{M} \in \mathbb{R}^{X \times Y \times Z \times 1}$ containing the locations of surface voxels (green voxels in $\mathbf{M}$ in the bottom right corner of fig. 3) is generated.

Voxel Enhancement. Following surface locator, we then attend to image features $\mathbf{F}_I$ with surface voxels to gain rich visual features of the 3D scene, leveraging attention mechanism[43]. To alleviate the huge computation cost, we utilize deformable attention[56]. In vanilla deformable attention, queries focus on local regions and sample N_p points around the reference point to update attention results, denoted as:

$$\text{DeformAttn}(\mathbf{q}, \mathbf{p}, \mathbf{F}) = \sum_{m=1}^{M} \mathbf{W}_m [\sum_{p=1}^{N_p} \mathbf{A}_{mp} \mathbf{W}'_m \mathbf{F} \left(\mathbf{p} + \delta \mathbf{p}_{mp}\right)] \tag{3}$$

where $\mathbf{q}$ denotes query, $\mathbf{p}$ denotes reference point, and $\mathbf{F}$ represents input features. m and p index the attention head and the sampled point. $\delta\mathbf{p}_{mp}$ and $\mathbf{A}_{mp}$ denote the sampling offset and normalized attention weight of the p^{th} sampling

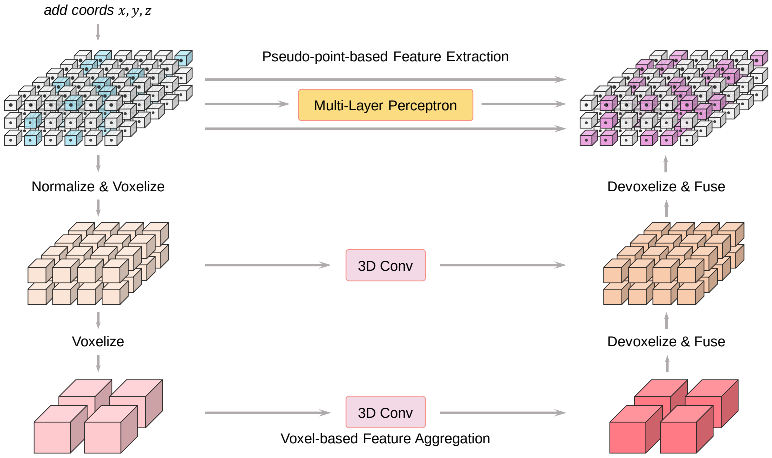


Fig. 4: **Feature diffuser.** Features are diffused to occluded regions via PVConv[33].

point in the m^{th} attention head, respectively. $\mathbf{F}\left(\mathbf{p}+\delta\mathbf{p}_{mp}\right)$ is the feature at location $\mathbf{p}+\delta\mathbf{p}_{mp}$ extracted by bilinear interpolation. $\mathbf{W}_m$ and $\mathbf{W}'_m$ are learnable weights.

To enhance surface voxels, we leverage deformable cross-attention. For each surface voxel as query $\mathbf{q}_a$, we first calculate its real-world coordinate $\mathbf{p}_a$ based on the volume resolution $X \times Y \times Z$ and the interested scene range. Then we project the 3D point coordinate to 2D image features $\mathbf{F}_I$ according to camera parameters. However, it is not certain whether the projected 2D point falls on an image due to the field of view. We only use the image the 2D point falls on, termed as $\mathcal{V}_{\text{hit}}$. Afterwards, the projected 2D point is used as the reference point of query $\mathbf{q}_a$, and we sample the features from $\mathcal{V}_{\text{hit}}$ around the reference point. Finally, the output feature is a weighted sum of sampled features according to deformable attention in eq. (3), denoted as:

$$\operatorname{DeformCrossAttn}\left(\mathbf{q}_a, \mathbf{F}_I\right)=\frac{1}{\left|\mathcal{V}_{\text{hit}}\right|} \sum_{i \in \mathcal{V}_{\text{hit}}} \operatorname{DeformAttn}\left(\mathbf{q}_a, \mathcal{P}(\mathbf{p}_a, i), \mathbf{F}_{I,i}\right) \quad (4)$$

where i indexes the images, and there is at most 1 image for monocular dataset. $\mathcal{P}$ is the projection function that projects location $\mathbf{p}_a$ to i^{th} image to obtain reference point.

At training stage, coarse features $\mathbf{G}_V$ will be more and more sparse as the iteration progresses. The reason of this is that under the supervision of depth information, depth distribution will be concentrated in one depth bin. That is to say, non-surface voxels will become increasingly obscure. To avoid this and obtain updated voxel features $\mathbf{F}_V$, after the enhancement, we replace the non-surface voxels with learnable parameters, maintaining the same number of channels.

3.3 Feature Diffuser

The structure of feature diffuser is shown in fig. 4, and this U-Net-like architecture is commonly used in voxel-based approaches for multi-scale information such as in HEDNet[53]. We first calculate the real-world coordinates for all voxels in $\mathbf{F}_V$ based on the volume resolution $X \times Y \times Z$ and the interested scene range, then we take the x, y, z coordinates as three channels to expand the voxel features. The expanded voxel features are regarded as pseudo points features $\mathbf{F}_P$ comprising three positional channels representing coordinates and additional feature channels, which will be termed pseudo points hereafter.

To learn both local features and global features from pseudo points, we utilize PVconv[33] to further process pseudo points. To begin with, all coordinates are normalized to $[0, 1]$ with the gravity center of pseudo points as origin. During normalization, features $\mathbf{F}_P$ keep unchanged. Afterwards, we merge the pseudo points to bigger voxels by aggregating all features whose coordinates fall into the bigger voxel grid. A stack of 3D convolutions are applied to aggregate local features in bigger voxels. After getting coarse-grained local features, we use an MLP for each individual pseudo point to get fine-grained features as global features. As we need to fuse the local features and global features, we devoxelize the bigger voxels back to the domain of pseudo points. To ensure that the features mapped to each pseudo point are distinct and do not share the same local feature, we leverage the trilinear interpolation for devoxelization. Finally, low-resolution local features and high-resolution global features are fused with another MLP.

3.4 Loss Functions

Following VoxFormer[24], we directly use the widely used loss function $\mathcal{L}_{\text{occ}} = \mathcal{L}_{\text{wce}} + \mathcal{L}_{\text{scal}}^{\text{sem}} + \mathcal{L}_{\text{scal}}^{\text{geo}}$ for the occupancy network to supervise the occupancy head, where $\mathcal{L}_{\text{wce}}$ is a weighted cross-entropy loss, $\mathcal{L}_{\text{scal}}^{\text{sem}}$ and $\mathcal{L}_{\text{scal}}^{\text{geo}}$ are off-the-shelf SSC losses derived from MonoScene[3]. To ensure the geometric positions of surface voxels are correct, the depth distribution $\mathbf{D}_{dist}$ is supervised by the projections of LiDAR points, with the binary cross-entropy loss $\mathcal{L}_{\text{depth}}$. The final training loss is a simple summation:

$$\mathcal{L} = \mathcal{L}_{\text{occ}} + \mathcal{L}_{\text{depth}} \tag{5}$$

4 Experiments

4.1 Datasets

SemanticKITTI [1] is a popular semantic scene understanding dataset based on KITTI Odometry Benchmark[11] including 22 outdoor driving scenarios. The dataset provides dense semantic annotations for each LiDAR sweep represented as 256 × 256 × 32 grids of 0.2 m voxels. The voxels are labelled with 20 classes (19 semantics and 1 free), and the 22 sequences are split into 10/1/11 for train/val/test. For SSC or 3D occupancy prediction, the dataset only focuses on scenes within 51.2 m to the front of the car, 25.6 m to the left and right, and 6.4 m in height.

Table 1: **3D occupancy prediction results on SemanticKITTI[1] validation set.** ∗ means the methods are adapted for the RGB inputs, which are implemented and reported in MonoScene[3]. The symbol ⋄ means the performance is achieved by our implementation using its official code. The top two performances are marked by red and green respectively.

Method	Backbone	mIoU	road (15.30%) ■	sidewalk (11.13%) ■	parking (1.12%) ■	other-ground (0.56%) ■	building (14.1%) ■	car (3.92%) ■	truck (0.16%) ■	bicycle (0.03%) ■	motorcycle (0.03%) ■	other-vehicle (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%) ■	traffic-sign (0.08%) ■
LMSCNet*[38]	-	6.70	40.68	18.22	4.38	0.00	10.31	18.33	0.00	0.00	0.00	0.00	13.66	0.02	20.54	0.00	0.00	0.00	1.21	0.00	0.00
3DSketch*[6]	-	7.50	41.32	21.63	0.00	0.00	14.81	18.59	0.00	0.00	0.00	0.00	19.09	0.00	26.40	0.00	0.00	0.00	0.73	0.00	0.00
AICNet*[19]	-	8.31	43.55	20.55	11.97	0.07	12.94	14.71	4.53	0.00	0.00	0.00	15.37	2.90	28.71	0.00	0.00	0.00	2.52	0.06	0.00
JS3C-Net*[50]	-	10.31	50.49	23.74	11.94	0.07	15.03	24.65	4.41	0.00	0.00	6.15	18.11	**4.33**	26.86	0.67	0.27	0.00	3.94	3.77	1.45
MonoScene[3]	-	11.50	57.47	**27.05**	15.72	**0.87**	14.24	23.55	7.83	0.20	0.77	3.59	18.12	2.57	**30.76**	1.79	1.03	0.00	**6.39**	4.11	2.48
TPVFormer[17]	R101-DCN	11.36	56.50	25.87	**20.60**	0.85	13.88	23.81	8.08	0.36	0.05	4.35	16.92	2.26	30.38	0.51	0.89	0.00	5.94	3.14	1.52
OccFormer⋄[55]	R50	**13.03**	**58.66**	26.34	19.1	0.77	14.41	25.19	**19.32**	**1.03**	**1.43**	**9.51**	**19.16**	3.25	30.64	**2.58**	**3.91**	0.00	5.60	4.02	**2.68**
VoxFormer[24]	R50	12.35	54.76	26.35	15.5	0.7	**17.65**	**25.79**	5.63	0.59	0.51	3.77	**24.39**	**5.08**	29.96	1.78	**3.23**	0.00	**7.64**	**7.11**	**4.18**
SurfOcc(ours)	R50	**13.75**	**59.43**	**28.59**	**19.78**	**2.15**	**16.51**	**26.16**	**22.77**	**0.85**	**0.88**	**9.64**	19.09	3.74	**33.73**	**1.82**	2.90	0.00	5.86	**4.75**	2.67

Table 2: **3D occupancy prediction results on Occ3D-nuScenes[41] validation set.** The symbol † indicates that the result is reported with utilization of camera mask during training. The top two performances are marked by red and green respectively.

Method	Backbone	Temporal	mIoU	others ■	barrier ■	bicycle ■	bus ■	car ■	const. veh. ■	motorcycle ■	pedestrian ■	traffic cone ■	trailer ■	truck ■	drive. suf. ■	other flat ■	sidewalk ■	terrain ■	manmade ■	vegetation ■
MonoScene[3]	R101-DCN	×	6.06	1.75	7.23	4.26	4.93	9.38	5.67	3.98	3.01	5.90	4.45	7.17	14.91	6.32	7.92	7.43	1.01	7.65
Lift-Splat-based feature lifting																				
BEVDet[16]	R101-DCN	√	19.38	4.39	30.31	0.23	32.26	34.47	12.97	10.34	10.36	6.26	8.93	23.65	52.27	24.61	26.06	22.31	15.04	15.10
OccFormer[55]	R101-DCN	×	21.93	5.94	30.29	12.32	34.40	39.17	14.44	16.45	17.22	9.27	13.90	26.36	50.99	30.96	34.66	22.73	6.76	6.97
BEVDet4D†[14]	Swin-B	√	42.02	**12.15**	49.63	25.10	**52.02**	54.46	**27.87**	27.99	**28.94**	27.23	**36.43**	42.44	82.31	43.29	**54.62**	**57.90**	**48.61**	**43.55**
Attention-based feature lifting																				
BEVFormer[27]	R101-DCN	√	26.88	5.85	37.83	17.87	40.44	42.43	7.36	23.88	21.81	20.98	22.38	30.70	55.35	28.36	36.0	28.06	20.04	17.69
TPVFormer[17]	R101-DCN	×	27.83	7.22	38.90	13.67	40.78	45.90	17.23	19.99	18.85	14.30	26.69	34.17	55.65	35.47	37.55	30.70	19.40	16.78
CTF-Occ[41]	R101-DCN	-	28.53	8.09	39.33	20.56	38.29	42.24	16.93	24.52	22.72	21.05	22.98	31.11	53.33	33.84	37.98	33.23	20.79	18.00
SurroundOcc†[48]	InternImage-B	×	40.70	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PanoOcc†[47]	R101-DCN	√	**42.13**	**11.67**	**50.48**	**29.64**	49.44	**55.52**	23.29	**33.26**	**30.55**	**30.99**	34.43	**42.57**	**83.31**	**44.23**	54.40	56.04	45.94	**40.40**
Surface-based feature lifting																				
SurfOcc†(ours)	R101-DCN	×	**42.38**	11.06	**49.95**	**27.71**	**51.12**	**55.96**	**28.54**	**30.07**	28.67	**27.44**	**35.28**	**44.42**	**82.99**	**46.35**	**56.09**	**59.70**	**47.26**	37.88

Occ3D-nuScenes is a newly proposed 3D occupancy prediction benchmark derived from the nuScenes dataset[2], which comprises 700 scenes for training and 150 scenes for validation. The dataset provides semantic annotations for each key frame represented as $200 \times 200 \times 16$ grids of 0.4 m voxels. The dataset covers a spatial range of -40 m to 40 m along the X and Y axis, and -1 m to 5.4 m along the Z axis. The semantic labels contain 17 classes (including "others"). To further enhance the 3D occupancy prediction benchmark, the dataset also provides visibility masks for both LiDAR and camera modality.

4.2 Implementation Details

Network Structures. For SemanticKITTI, we adopt ResNet50[12] to extract image features following VoxFormer[24]. For Occ3D-nuScenes, we adopt

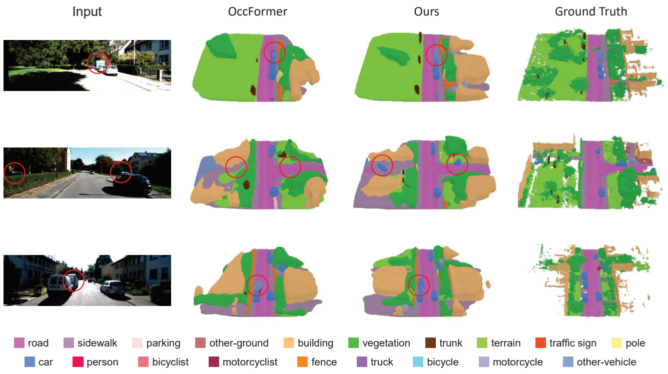


Fig. 5: **Qualitative results on SemanticKITTI validation set.** Our model exhibits superior performance when dealing with occlusion scenarios (red circles).

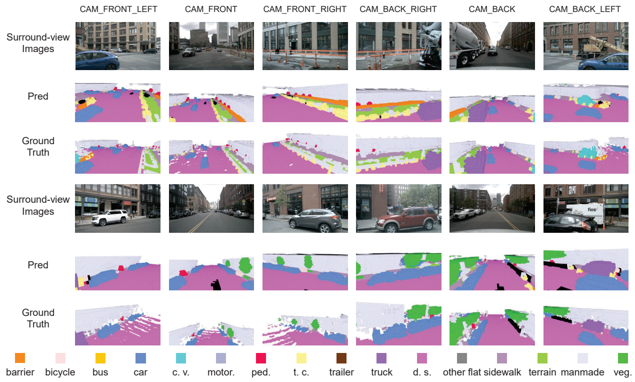


Fig. 6: **Qualitative results on Occ3D-nuScenes validation set.** We present the occupancy grids from different camera angles for better observation.

ResNet101-DCN[12,9] that initialized from FCOS3D[44] checkpoint. By default, we take the output multi-scale features from FPN[29] with size of $1/8$, $1/16$ and $1/32$. The feature dimension is set as $C = 128$. In feature lifting phase, the shape of voxel features is $128 \times 128 \times 16$ for SemanticKITTI and $100 \times 100 \times 8$ for Occ3D-nuScenes, with 128 channels. For both datasets, 3 deformable cross-attention layers and 8 sampling points are used around each reference points. In feature diffusion phase, the maximum resolution for PVConv[33] is 32 for SemanticKITTI and 50 for Occ3D-nuScenes. The occupancy head only contains a shallow ResNet3D, which upscales the voxel features to the same shape as the ground truth for full-scale evaluation.

Training Setup. We crop RGB images of cam2 (left camera) in SemanticKITTI to size 1220×370 and images from all 6 perspectives in Occ3D-nuScenes to

Table 3: **Ablation study for architecture.**

	Surface-based Feature Lifting		Feature Diffuser	mIoU↑
	Surface Locator	Voxel Enhancement		
1)	√	√	√	**13.75**
2)	√	√		12.69(-1.06)
3)	√		√	12.07(-1.68)
4)		√	√	11.90(-1.85)

size 704 × 256. For data augmentation, we apply random scaling, flipping, and rotation following BEVDet[16]. We train the model end-to-end for 30 epoches on SemanticKITTI and 24 epoches on Occ3D-nuScenes. During training, the AdamW optimizer with an initial learning rate of 1e-4 and a weight decay of 0.01 is used. All experiments are conducted with a batch size of 4 on 4 NVIDIA A100 GPUs with 40G memory.

4.3 Metrics

For both datasets, we report the mean Intersection over Union (mIoU) for 3D occupancy prediction. For Occ3D-nuScenes, the benchmark calculates the mIoU for 17 semantic categories in the visible region of the camera.

4.4 3D Occupancy Prediction Results

As is shown in table 1, we first conduct monocular 3D occupancy prediction on SemanticKITTI[1] and compare with existing state-of-the-art methods. SurfOcc outperforms all competitors and achieves state-of-the-art performance. We can observe that SurfOcc shows especially good performance in differentiating objects with significant volume and appreciable thickness (car, truck). We surpass the two-stage VoxFormer[24] with a margin of 1.4 mIoU through an end-to-end training approach, seamlessly integrating depth estimation into the process. Results have demonstrated the effectiveness of this enhancement, particularly in categories that rely on accurate depth measurements, such as various grounds. Qualitative results can be seen in fig. 5.

We also conduct experiments on Occ3D-nuScenes[41], as is shown in table 2. Despite the omission of temporal information in our approach, our work still outperforms recent state-of-the-art methods, including Lift-Splat-based methods and attention-based methods. The results demonstrate SurfOcc's remarkable ability to identify objects with significant volume and prominent thickness, as well as those objects with clear depth information. Qualitative results can be seen in fig. 6.

4.5 Ablation Studies

Ablation on the Architecture. We conduct architecture ablation as shown in table 3. Line **1)** shows the performance of complete SurfOcc model. **2)** When the

Table 4: **Ablation study for feature scales.**

	Feature Scale				mIoU↑
	1/4	1/8	1/16	1/32	
1)				√	12.26
2)			√	√	12.84
3)		√	√	√	13.75
4)	√	√	√	√	13.87

Table 5: **Ablation study for feature diffuser.** Resolution means the max-resolution of voxelization when applying PVConv. Memory means the consumption of this module.

	Resolution	mIoU↑	Memory(G)
1)	16	12.88	7.9
2)	32	13.75(+0.87)	8.2(+0.3)
3)	64	13.96(+1.08)	9.6(+1.7)

Table 6: **Model size comparison.**

Method	Backbone	Dataset	mIoU↑	Params↓
OccFormer[55]	R50	SemanticKITTI	13.03	200M
SurfOcc(ours)	R50	SemanticKITTI	13.75	94M
SurfOcc(ours)	R101-DCN	Occ3D-nuScenes	42.38	110M
PanoOcc[47]	R101-DCN	Occ3D-nuScenes	42.13	115M

feature diffuser is removed, mIoU drops to 12.69. As the diffuser helps the model understand the environment, particularly occluded regions, inferring without it leads to a lack of information. **3)-4)** Removing either module in surface-based feature lifting results in a varying degree of performance degradation.

Ablation on Surface-based Feature Lifting. Line **3)-4)** in table 3 show how surface-based feature lifting affects the performance of the model and show the comparison of different feature lifting methods. We first remove voxel enhancement during feature lifting to imitate Lift-Splat-based feature lifting method. Feature volume is generated with no surface voxels and without refinement, resulting in mIoU drops to 12.07. We then remove surface locator and generate feature volume via cross-attention between predefined embeddings and image features to imitate attention-based feature lifting method. The mIoU drops even more severely to 11.90. We tried to use a depth network instead of the depth distribution network, but it drastically reduced the accuracy and made the fitting process very slow. These experiments confirm that surface-based feature lifting is effective and superior to other feature lifting methods.

We also conduct abaltion study for feature scales of cross-attention in table 4. Feature scale is relative to the input image size. The performance improves as the number of scales increases but the gains become marginal when four scales are used. Given the model's size, we opt for a three-scale approach, which strikes a balance between performance and efficiency.

Ablation on Feature Diffuser. The primary influencing factor is the maximum resolution of voxelization in PVConv[33]. As is shown in table 5, higher resolution leads to better model performance. When the resolution is too low, the model exhibits poorer performance due to the scarcity and coarseness of local information. While increasing the resolution from 32 to 64 will enhance performance, it concurrently incurs a computational cost that does not align proportionally with the observed improvement in the experiment. In our final implementation, we set the resolution to 32, yielding excellent performance and avoiding excessive computational costs.

4.6 Model Size

Our approach is geometrically intuitive and optimized for complexity, ensuring a lightweight design that maximizes performance while minimizing complexity. As shown in table 6, our model achieves better performance on both datasets while using fewer parameters.

5 Conclusion

In this paper, we present SurfOcc, a robust vision-centric 3D occupancy prediction framework. SurfOcc locates observed surfaces during feature lifting phase and propagates features to occluded regions during feature diffusion phase. To effectively refine lifted features and mitigate feature ambiguity, we propose surface-based feature lifting to extract and enhance voxel features. Additionally, we design a feature diffuser to support our model's inference of occluded parts, and the end-to-end two-phase scheme enhances the model's holistic understanding of 3D scenes. The comparison on SemanticKITTI and Occ3D-nuScenes demonstrates the superiority of SurfOcc in both monocular and multi-view settings.

Limitation. SurfOcc performs less effectively in identifying small objects, which we attribute to the adoption of a smaller feature scale during the training process for efficiency. In future work, we will attempt to address this limitation by incorporating larger-scale and sparse representation.

Acknowledgments. This work was supported by the Civilian Space Project under Grant D040103 and Grant D010206.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., Gall, J.: Semantickitti: A dataset for semantic scene understanding of lidar sequences. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9297–9307 (2019)

2. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020)
3. Cao, A.Q., de Charette, R.: Monoscene: Monocular 3d semantic scene completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3991–4001 (2022)
4. Cao, A.Q., de Charette, R.: Scenerf: Self-supervised monocular 3d scene reconstruction with radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9387–9398 (2023)
5. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
6. Chen, X., Lin, K.Y., Qian, C., Zeng, G., Li, H.: 3d sketch-aware semantic scene completion via semi-supervised structure prior. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4193–4202 (2020)
7. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1290–1299 (2022)
8. Cheng, R., Agia, C., Ren, Y., Li, X., Bingbing, L.: S3cnet: A sparse semantic scene completion network for lidar point clouds. In: Conference on Robot Learning. pp. 2148–2161. PMLR (2021)
9. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 764–773 (2017)
10. Garbade, M., Chen, Y.T., Sawatzky, J., Gall, J.: Two stream 3d semantic scene completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)
11. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE conference on computer vision and pattern recognition. pp. 3354–3361. IEEE (2012)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
13. Hou, J., Wang, T., Ye, X., Liu, Z., Gong, S., Tan, X., Ding, E., Wang, J., Bai, X.: Open: Object-wise position embedding for multi-view 3d object detection. arXiv preprint arXiv:2407.10753 (2024)
14. Huang, J., Huang, G.: Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. arXiv preprint arXiv:2203.17054 (2022)
15. Huang, J., Huang, G.: Bevpoolv2: A cutting-edge implementation of bevdet toward deployment. arXiv preprint arXiv:2211.17111 (2022)
16. Huang, J., Huang, G., Zhu, Z., Ye, Y., Du, D.: Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. arXiv preprint arXiv:2112.11790 (2021)
17. Huang, Y., Zheng, W., Zhang, Y., Zhou, J., Lu, J.: Tri-perspective view for vision-based 3d semantic occupancy prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9223–9232 (2023)
18. Lee, J., Im, W., Lee, S., Yoon, S.E.: Diffusion probabilistic models for scene-scale 3d categorical data. arXiv preprint arXiv:2301.00527 (2023)

19. Li, J., Han, K., Wang, P., Liu, Y., Yuan, X.: Anisotropic convolutional networks for 3d semantic scene completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3351–3359 (2020)
20. Li, J., Liu, Y., Yuan, X., Zhao, C., Siegwart, R., Reid, I., Cadena, C.: Depth based semantic scene completion with position importance aware loss. IEEE Robotics and Automation Letters **5**(1), 219–226 (2019)
21. Li, Y., Chen, Y., Qi, X., Li, Z., Sun, J., Jia, J.: Unifying voxel-based representation with transformer for 3d object detection. Advances in Neural Information Processing Systems **35**, 18442–18455 (2022)
22. Li, Y., Qi, X., Chen, Y., Wang, L., Li, Z., Sun, J., Jia, J.: Voxel field fusion for 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1120–1129 (2022)
23. Li, Y., Li, S., Liu, X., Gong, M., Li, K., Chen, N., Wang, Z., Li, Z., Jiang, T., Yu, F., et al.: Sscbench: A large-scale 3d semantic scene completion benchmark for autonomous driving. arXiv preprint arXiv:2306.09001 (2023)
24. Li, Y., Yu, Z., Choy, C., Xiao, C., Alvarez, J.M., Fidler, S., Feng, C., Anandkumar, A.: Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9087–9098 (2023)
25. Li, Y., Bao, H., Ge, Z., Yang, J., Sun, J., Li, Z.: Bevstereo: Enhancing depth estimation in multi-view 3d object detection with temporal stereo. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 1486–1494 (2023)
26. Li, Y., Ge, Z., Yu, G., Yang, J., Wang, Z., Shi, Y., Sun, J., Li, Z.: Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 1477–1485 (2023)
27. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Qiao, Y., Dai, J.: Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In: European conference on computer vision. pp. 1–18. Springer (2022)
28. Liang, T., Xie, H., Yu, K., Xia, Z., Lin, Z., Wang, Y., Tang, T., Wang, B., Tang, Z.: Bevfusion: A simple and robust lidar-camera fusion framework. Advances in Neural Information Processing Systems **35**, 10421–10434 (2022)
29. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
30. Liu, S., Hu, Y., Zeng, Y., Tang, Q., Jin, B., Han, Y., Li, X.: See and think: Disentangling semantic scene completion. Advances in Neural Information Processing Systems **31** (2018)
31. Liu, Y., Wang, T., Zhang, X., Sun, J.: Petr: Position embedding transformation for multi-view 3d object detection. In: European Conference on Computer Vision. pp. 531–548. Springer (2022)
32. Liu, Z., Tang, H., Amini, A., Yang, X., Mao, H., Rus, D.L., Han, S.: Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In: 2023 IEEE international conference on robotics and automation (ICRA). pp. 2774–2781. IEEE (2023)
33. Liu, Z., Tang, H., Lin, Y., Han, S.: Point-voxel cnn for efficient 3d deep learning. In: Conference on Neural Information Processing Systems (NeurIPS) (2019)
34. Mao, J., Xue, Y., Niu, M., et al.: Voxel transformer for 3d object detection. ICCV (2021)

35. Philion, J., Fidler, S.: Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16. pp. 194–210. Springer (2020)
36. Reading, C., Harakeh, A., Chae, J., Waslander, S.L.: Categorical depth distribution network for monocular 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8555–8564 (2021)
37. Rist, C.B., Emmerichs, D., Enzweiler, M., Gavrila, D.M.: Semantic scene completion using local deep implicit functions on lidar data. IEEE transactions on pattern analysis and machine intelligence **44**(10), 7205–7218 (2021)
38. Roldao, L., de Charette, R., Verroust-Blondet, A.: Lmscnet: Lightweight multiscale 3d semantic completion. In: 2020 International Conference on 3D Vision (3DV). pp. 111–119. IEEE (2020)
39. Shu, C., Deng, J., Yu, F., Liu, Y.: 3dppe: 3d point positional encoding for multi-camera 3d object detection transformers. arXiv preprint arXiv:2211.14710 (2023)
40. Song, S., Yu, F., Zeng, A., Chang, A.X., Savva, M., Funkhouser, T.: Semantic scene completion from a single depth image. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1746–1754 (2017)
41. Tian, X., Jiang, T., Yun, L., Wang, Y., Wang, Y., Zhao, H.: Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. arXiv preprint arXiv:2304.14365 (2023)
42. Tong, W., Sima, C., Wang, T., Chen, L., Wu, S., Deng, H., Gu, Y., Lu, L., Luo, P., Lin, D., et al.: Scene as occupancy. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8406–8415 (2023)
43. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
44. Wang, T., Zhu, X., Pang, J., Lin, D.: Fcos3d: Fully convolutional one-stage monocular 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 913–922 (2021)
45. Wang, X., Zhu, Z., Xu, W., Zhang, Y., Wei, Y., Chi, X., Ye, Y., Du, D., Lu, J., Wang, X.: Openoccupancy: A large scale benchmark for surrounding semantic occupancy perception. arXiv preprint arXiv:2303.03991 (2023)
46. Wang, Y., Guizilini, V.C., Zhang, T., Wang, Y., Zhao, H., Solomon, J.: Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In: Conference on Robot Learning. pp. 180–191. PMLR (2022)
47. Wang, Y., Chen, Y., Liao, X., Fan, L., Zhang, Z.: Panoocc: Unified occupancy representation for camera-based 3d panoptic segmentation. arXiv preprint arXiv:2306.10013 (2023)
48. Wei, Y., Zhao, L., Zheng, W., Zhu, Z., Zhou, J., Lu, J.: Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 21729–21740 (2023)
49. Yan, J., Liu, Y., Sun, J., Jia, F., Li, S., Wang, T., Zhang, X.: Cross modal transformer: Towards fast and robust 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 18268–18278 (2023)
50. Yan, X., Gao, J., Li, J., Zhang, R., Li, Z., Huang, R., Cui, S.: Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 3101–3109 (2021)

51. Yang, C., Chen, Y., Tian, H., Tao, C., Zhu, X., Zhang, Z., Huang, G., Li, H., Qiao, Y., Lu, L., et al.: Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17830–17839 (2023)
52. Yao, J., Li, C., Sun, K., Cai, Y., Li, H., Ouyang, W., Li, H.: Ndc-scene: Boost monocular 3d semantic scene completion in normalized device coordinates space. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9455–9465 (2023)
53. Zhang, G., Junnan, C., Gao, G., Li, J., Hu, X.: Hednet: A hierarchical encoder-decoder network for 3d object detection in point clouds. Advances in Neural Information Processing Systems **36** (2024)
54. Zhang, H., Li, H., Liao, X., Li, F., Liu, S., Ni, L.M., Zhang, L.: Da-bev: Depth aware bev transformer for 3d object detection. arXiv preprint arXiv:2302.13002 (2023)
55. Zhang, Y., Zhu, Z., Du, D.: Occformer: Dual-path transformer for vision-based 3d semantic occupancy prediction. arXiv preprint arXiv:2304.05316 (2023)
56. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)