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

Yi Wei¹,²*, Linqing Zhao³*, Wenzhao Zheng¹,², Zheng Zhu⁴, Jie Zhou¹,², Jiwen Lu¹,²†

¹Beijing National Research Center for Information Science and Technology, China
²Department of Automation, Tsinghua University, China
³School of Electrical and Information Engineering, Tianjin University, China
⁴PhiGent Robotics

{y-wei19,zhengwz18}@mails.tsinghua.edu.cn; linqingzhao@tju.edu.cn;
zhengzhu@ieee.org; {jzhou, lujiwen}@tsinghua.edu.cn

Abstract

3D scene understanding plays a vital role in vision-based autonomous driving. While most existing methods focus on 3D object detection, they have difficulty describing real-world objects of arbitrary shapes and infinite classes. Towards a more comprehensive perception of a 3D scene, in this paper, we propose a SurroundOcc method to predict the 3D occupancy with multi-camera images. We first extract multi-scale features for each image and adopt spatial 2D-3D attention to lift them to the 3D volume space. Then we apply 3D convolutions to progressively upsample the volume features and impose supervision on multiple levels. To obtain dense occupancy prediction, we design a pipeline to generate dense occupancy ground truth without expansive occupancy annotations. Specifically, we fuse multi-frame LiDAR scans of dynamic objects and static scenes separately. Then we adopt Poisson Reconstruction to fill the holes and voxelize the mesh to get dense occupancy labels. Extensive experiments on nuScenes and SemanticKITTI datasets demonstrate the superiority of our method. Code and dataset are available at https://github.com/weiyithu/SurroundOcc.

1. Introduction

Understanding the 3D geometry of the surrounding scene serves as the basic procedure in an autonomous driving system. While LiDAR is a direct solution to capture this geometric information, it suffers from high-cost sensors and sparse scanned points, limiting its further application. Recently, vision-centric autonomous driving has attracted extensive attention as a promising direction. Taking multi-camera images as inputs, it has demonstrated competitive performance in various 3D perception tasks including depth estimation [17, 58], 3D object detection [30, 27, 34, 20], and semantic map construction [66, 19, 1].

Although multi-camera 3D object detection plays an important role [30, 27, 34, 20] in vision-based 3D perception, it is easy to suffer from the long-tail problem and difficult to recognize all classes of objects in the real world. Complementary to 3D detection, reconstructing surrounding 3D scenes can better help the downstream perception tasks. Recent works [17, 58] incorporate information from multiple views and predict surrounding depth maps. However, depth maps only predict the nearest occupied point in each optical ray and are unable to recover the occluded parts of the 3D

*Equal contribution.
†Corresponding author.
scene. Different from depth-based methods, another trend [7, 21] is to directly predict the 3D occupancy of the scene. It describes a 3D scene by assigning an occupied probability to each voxel in the 3D space. We advocate 3D occupancy to be a good 3D representation for multi-camera scene reconstruction, which naturally guarantees the multi-camera geometry consistency and is able to recover occluded parts. Also, it is flexible to extend to other 3D downstream tasks such as 3D semantic segmentation [69, 73, 15]. As one of the pioneering works, MonoScene [7] infers the dense 3D voxelized semantic scene with monocular images. However, simply fusing multi-camera results with cross-camera post-processing will lead to low performance [30]. TPVFormer [21] uses sparse LiDAR points as supervision, which results in sparse occupancy prediction.

To address this, we propose a SurroundOcc method, which aims to predict dense and accurate 3D occupancy with multi-camera images input. We first use a 2D backbone network to extract multi-scale feature maps from each image. Then we perform 2D-3D spatial attention to lift multi-camera image information to 3D volume features instead of BEV features. A 3D convolution network is then employed to progressively upsample the low-resolution volume features and fuse them with high-resolution ones to obtain fine-grained 3D representations. At each level, we use a decayed weighted loss to supervise the network. To get dense predictions, we need dense occupancy labels. However, the mainstream multi-camera dataset nuScenes [6] only provides sparse LiDAR points. To avoid expensive occupancy annotations, we devise a pipeline to generate dense occupancy ground truth with the existing 3D detection and 3D semantic segmentation labels. Specifically, we first combine multi-frame points of dynamic objects and static scenes respectively. Then we leverage Poisson Reconstruction [23] algorithm to further fill the holes. Finally, NN and voxelization are used to obtain dense 3D occupancy labels.

With the dense occupancy ground truth, we train the model and compare it with other state-of-the-art methods on nuScenes [6] dataset. Both the quantitative results and visualizations demonstrate the effectiveness of our method. Moreover, we further conduct experiments on SemanticKITTI dataset [2]. Although our method is not designed for the monocular setting, it achieves state-of-the-art performance on the monocular 3D semantic scene completion benchmark.

2. Related Work

**Voxel-based Scene Representation:** How to effectively represent a 3D scene lies at the core of autonomous driving perception. Voxel-based scene representation voxelizes the 3D space into discretized voxels and describes each voxel by a vector feature. It has empowered the success of numerous methods on the lidar segmentation [33, 52, 11, 62, 61] and 3D scene completion [7, 46, 9, 25, 59, 29, 28] tasks. For the 3D occupancy prediction task, we also advocate the voxel representation as it is more suitable to model the occupancy field of a 3D scene. MonoScene [7] is the first work to reconstruct outdoor scenes using only RGB inputs. TPVFormer [21] further generalizes it to multi-camera 3D semantic occupancy prediction. However, its lack of dense supervision results in sparse occupancy prediction. Differently, we devise a pipeline to generate dense occupancy ground truth for training and our occupancy prediction is much denser.

**3D Scene Reconstruction:** 3D reconstruction [42, 49, 48, 41, 71, 16, 4, 38, 50, 5] is a traditional but important topic in computer vision. One way is through depth estimation which predicts a depth value for each pixel in the image. While early methods require full depth annotations to supervise the depth estimation model [14, 24, 67], later research focuses on self-supervised depth estimation as it does not require intensive human annotations [71, 16, 4, 68, 63, 57, 70, 44, 8]. Recently, SurroundDepth [58] further incorporates the interactions between surrounding views to capture more spatial correlations. Different from depth estimation, 3D scene reconstruction methods [22, 38, 50, 5, 7] directly reconstruct a comprehensive and accurate 3D geometry of a scene. SurfaceNet [22] employs a 3D convolutional network to transform RGB colors to 3D surface occupancy from two images. Atlas [38] further extends it to the multi-view setting and utilized learned features to predict occupancy. NeuralRecon [50] and TransformerFusion [5] fuse the learned image features from different views in an online manner for more accurate 3D reconstructions. However, most of these 3D scene reconstruction methods are designed for indoor scenes, which are different from the multi-camera setting in the outdoor environment.

**Vision-based 3D Perception:** The lack of direct geometric input demands vision-based 3D surround perception methods to infer the 3D scene geometry based on semantic cues. Depth-based methods explicitly predict depth maps for image inputs to extract 3D geometric information of the scene before perception [55, 37, 45, 40, 27, 36, 31, 43, 20, 66]. The widely adopted pipeline is to predict categorical depth distributions and leverage them to project semantic features into 3D space [43]. Other methods implicitly learn 3D features without producing explicit depth maps [54, 30, 56, 35, 53, 64]. For example, BEVFormer [30] adopts cross attention to progressively refine BEV grid features from 2D image features. While most existing works employ BEV representation to describe a scene, we propose to reconstruct a 3D scene using volumetric occupancy representation, which provides the more fine-grained and comprehensive modeling of the scene.
3. Approach

3.1. Problem Formulation

In this work, we aim to predict the 3D occupancy of surrounding scenes with multi-camera images $I = \{I^1, I^2, \ldots I^N\}$. Formally, the 3D occupancy prediction is represented as:

$$ V = G(I^1, I^2, \ldots I^N) $$

where $G$ is a neural network and $V \in \mathbb{R}^{H \times W \times Z}$ is the 3D occupancy. The value of $V$ is between 0 and 1, representing the occupied probability of the grids. Lifting $V$ to an $(L, H, W, Z)$ tensor, we can obtain the 3D semantic occupancy, where $L$ is the class number and class 0 means non-occupied grids.

3D occupancy is a good representation for multi-camera 3D scene reconstruction. First, since 3D occupancy is predicted in 3D space, it theoretically satisfies the multi-camera consistency. Second, it is possible for networks to predict occluded areas according to the surrounding semantic information, which is unavailable in depth estimation. Third, 3D occupancy is easy to extend to other downstream tasks, such as 3D semantic segmentation and scene flow estimation.

3.2. Overview

Figure 2 shows the pipeline of our method. Given a set of surrounding multi-camera images, we first use a backbone network (e.g. ResNet-101 [18]) to extract $N$ cameras’ and $M$ levels’ multi-scale features $X = \{\{X_i^j\}_{i=1}^N\}_{j=1}^M$. For each level, we use a transformer to fuse multi-camera features with spatial cross attention. The output of the 2D-3D spatial attention layer is a 3D volume feature instead of the BEV feature. Then the 3D convolution network is utilized to upsample and combine multi-scale volume features. The occupancy prediction in each level is supervised by the generated dense occupancy ground truth with a decayed loss weight.

3.3. 2D-3D Spatial Attention

Many 3D scene reconstruction methods [7, 38] integrate multi-view 2D features into 3D space by reprojecting 2D features back to the 3D volumes with known poses. The grid feature is calculated by simply averaging all 2D features in this grid. However, this kind of method assumes that different views contribute equally to the 3D volumes.
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.

which does not always hold true, especially when some views are occluded or blurred.

To tackle the issue, we leverage cross-view attention to fuse multi-camera features. We project 3D reference points to 2D views and use deformable attention [72, 30] to query points and aggregate information. As shown in Figure 3, instead of 2D BEV queries, we build 3D volume queries to further reserve 3D space information. Specifically, 3D volume queries are defined as $Q \in \mathbb{R}^{C \times H \times W \times Z}$. For each query, we project its corresponding 3D point to 2D views according to the given intrinsic and extrinsic. We only use the views that the 3D reference point hits. Then we sample 2D features around these projected 2D positions. The output $F \in \mathbb{R}^{C \times H \times W \times Z}$ of the cross-view attention layer is a weighted sum of sampled features according to the deformable attention mechanism:

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

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

where $F^p$ and $Q^p$ indicate the $p$th element of the output features and 3D volume queries. $q^p$ is the corresponding 3D positions of queries and $P$ is the 2D to 3D project function. $V_{\text{hit}}$ represents the hit views of 3D query points. $W_i$ and $W'_i$ are the learnable weights and $A_{ij} \in [0, 1]$ is the attention weight calculated by the dot product of query and key. $x(p + \Delta p_{ij})$ is the 2D feature at location $p + \Delta p_{ij}$. Instead of performing expensive 3D self-attention, we use the 3D convolution to interact features between neighboring 3D voxels.

### 3.4. Multi-scale Occupancy Prediction

We further extend the 2D-3D spatial attention to a multi-scale fashion. Different from 3D detection task, 3D scene reconstruction needs more low-level features to help the network learn fine-grained details. To tackle the issue, we design a 2D-3D U-Net architecture. Specifically, given multi-scale 2D features $\{X_j\}_{j=1}^M$, we adopt different number of 2D-3D spatial attention layers to extract multi-scale 3D volume features $\{F_j \in \mathbb{R}^{C_j \times H_j \times W_j \times Z_j}\}_{j=1}^M$.

Then we upsample $j - 1$th level 3D volume features $Y_{j-1}$ with 3D deconvolution layer and fuse it with $F_j$:

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

For each level, the network outputs an occupancy prediction result with different resolution $V_j \in \mathbb{R}^{C_j \times H_j \times W_j \times Z_j}$. To get powerful both high-level and low-level 3D features, the network is supervised at each scale. Specifically, we use the cross-entropy loss and scene-class affinity loss introduced in [7] as the supervision signals. For 3D semantic occupancy prediction, we adopt a multi-class cross-entropy loss and for 3D scene reconstruction we change it to a two-class formulation. Since the high-resolution prediction is more important, we use a decayed loss weight $\alpha_j = \frac{1}{2^j}$ for $j$th level supervision.
4. Dense Occupancy Ground Truth

In our experiments, we find that the network supervised by sparse LiDAR points is unable to predict dense enough occupancy. Thus, it is necessary to generate dense occupancy labels. However, as mentioned in SemanticKITTI [2], it is complex and needs huge human efforts to annotate the dense occupancy of a 3D scene which has millions of voxels. To this end, we design a pipeline to generate dense occupancy ground truth leveraging existing 3D detection and 3D semantic segmentation labels without extra human annotations, which is shown in Figure 4. An intuitive way is to directly transform a multi-frame LiDAR point sequence into a unified coordinate system, and then voxelize the concatenated dense points into voxel grids. However, such a straightforward solution is only applicable to completely static scenes and ignores moving objects. Moreover, multi-frame point clouds are not dense enough and there still exist many holes, which will result in wrong occupancy labels. To address these issues, we propose to stitch multi-frame LiDAR points of dynamic objects and static scenes separately. In addition, we adopt Poisson Reconstruction [23] to fill up the holes and voxelize the obtained mesh to get dense volumetric occupancy. Since the LiDAR scans surface points, our method also generates surface occupancy.

4.1. Multi-frame Point Cloud Stitching

We propose a two-stream pipeline to stitch static scenes and objects separately and merge them into a complete scene before voxelization. Specifically, for each frame, we first cut out movable objects from the LiDAR points according to 3D bounding box labels, so that we can obtain the 3D points of a static scene and movable objects. After traversing all frames in the scene, we integrate the collected static scene segments and object segments into a set respectively. To combine the multi-frame segments, we then transform their coordinates into the world coordinate system via the known calibrated matrices and ego-poses. We denote the transformed static scene segments and object segments as \( P_{ss} = \{ P_{ss}^1, P_{ss}^2, \cdots, P_{ss}^n \} \) and \( P_{os} = \{ P_{os}^1, P_{os}^2, \cdots, P_{os}^m \} \), where \( n \) and \( m \) are the numbers of frames and objects in the sequence, respectively. Note that the same objects in different frames can be recognized according to the bounding box index. Therefore, we can represent the whole static scene as \( P_s = \{ P_{ss}^1, P_{ss}^2, \cdots, P_{ss}^n \} \) while the objects as \( P_o = \{ P_{os}^1, P_{os}^2, \cdots, P_{os}^m \} \), respectively, where \([\cdot]\) is the concatenation operator. Finally, according to the objects’ locations and ego-pose of the current frame, the 3D points of this frame can be obtained by merging static scene and objects: \( P = [T_s(P_s), T_o(P_o)] \), where \( T_s \) and \( T_o \) are the transformations of static scenes and objects from world coordinate system to the current frame coordinate system. In this way, the occupancy labels of the current frame leverage the LiDAR points of all frames in the sequence.

4.2. Densifying with Poisson Reconstruction

While the density of \( P \) is much larger than a single-frame LiDAR, there still exists many interspaces and the points are not evenly distributed, which is caused by the limited LiDAR beams. To address this, we first compute the normal vectors according to the spatial distribution in local neighborhoods. Then we reconstruct \( P \) to a triangular mesh \( \mathcal{M} \) via Poisson Surface Reconstruction [23], whose input is point cloud with normal vectors, and the output is a triangular mesh (see Figure 4). The obtained mesh \( \mathcal{M} = \{ V, E \} \) fills up the holes of point clouds with evenly distributed vertices \( V \), so that we can further convert the mesh into dense voxel \( V_d \).

4.3. Semantic Labeling with NN Algorithm

Having obtained the occupancy of dense voxels \( V_d \), we aim to assign semantic labels to each voxel, since position reconstruction can only be applied to 3D space, not semantic space. To this end, we propose to leverage Nearest Neighbors (NN) algorithm to search the nearest semantic label for each voxel. Specifically, we first voxelize \( P \) with semantics into voxels \( V_s \), which are sparser than \( V_d \) due to limited LiDAR beams. Then for each occupied voxel in \( V_d \), we use NN to search the nearest voxel in \( V_s \) and assign the semantic label to it. Note that this process can be accelerated by parallel computing on the GPU. Thus, all occupied voxels in \( V_d \) can obtain their semantic labels from \( V_s \).

Figure 5 shows a detailed visual comparison between single-frame LiDAR points, sparse occupancy labels and dense occupancy labels. We observe that our dense voxels can provide much more realistic occupancy labels with
clear semantic boundaries.

We think it is not trivial to propagate the sparse semantic label to a dense one since it is an ill-posed problem. The proposed multi-frame point cloud stitching can aggregate multi-frame semantic information and provide dense enough reference points for NN. However, we find that NN is sensitive to the annotation noise in original LiDAR semantic labels, and we will try to solve it as the future work.

5. Experiments

5.1. Experimental Setup

Dataset: We conduct multi-camera experiments on nuScenes dataset [6], which is a large-scale autonomous driving dataset. Since the 3D semantic and 3D detection labels are unavailable in test set and we cannot generate dense occupancy labels, we use the training set to train the model and validation set for evaluation. The occupancy prediction range is set as \([-50_{m}, 50_{m}]\) for \(X, Y\) axis and \([-5_{m}, 3_{m}]\) for \(Z\) axis. The final output occupancy has the shape 200x200x16 with 0.5m voxel size. The input image resolution is 1600x900.

To further demonstrate the effectiveness of our method, we conduct monocular semantic scene completion experiment on SemanticKITTI dataset [2]. SemanticKITTI has annotated outdoor LiDAR scans with 21 semantic labels. The ground truth is voxelized as 256x256x32 grid with 0.2m voxel size. We evaluate our model on the test set.

Implementation Details: The whole network architecture obtains \(M = 4\) levels and we do not add the skip connection in level 0. For nuScenes dataset, we adopt ResNet101 DCN [18, 13] with the initial weight from FCOS3D [54] as the backbone to extract image features. The features of stage 1,2,3 are fed to FPN [32] and used as multi-scale image features. The number of 2D-3D spatial attention layers are set as 1, 3, 6 for three levels. For SemanticKITTI dataset, following MonoScene [7], we use a pretrained EfficientNetB7 [51] as the backbone to generate multi-scale image features. We also adopt FPN to further fuse the features of different levels. We set the number of 2D-3D spatial attention layers as 1, 3, 8. All experiments are conducted on 8 RTX 3090s. For Possion Reconstruction, we accumulate both key-frame and non key-frame data. In details, we use LiDAR frames at 20Hz, resulting in 400 frames and around 13 million points for a 20 second sequence.

Evaluation Metrics: For 3D semantic occupancy prediction, we use the intersection over union (IoU) of occupied voxels, ignoring their semantic class as the evaluation metric of the scene completion (SC) task and the mIoU of all semantic classes for the SSC task.

\[
\text{IoU} = \frac{TP}{TP + FP + FN} \quad \text{mIoU} = \frac{1}{C} \sum_{i=1}^{C} \frac{TP_i}{TP_i + FP_i + FN_i}
\]

where \(TP, FP, FN\) indicate the number of true positive, false positive, and false negative predictions. \(C\) is the class number.

For 3D scene reconstruction, we first convert occupancy prediction to point clouds. Following [38, 5], we use 3D metrics for evaluation, which is shown in Table 2. Chamfer distance (CD) and F-score are the main metrics since they consider both precision and recall. Please refer to the supplementary for more evaluation metric details. For all evaluation, we adopt dense occupancy as ground truth since sparse LiDAR points cannot fully evaluate the quality of occupancy reconstruction.

<table>
<thead>
<tr>
<th>Method</th>
<th>SC IoU</th>
<th>SSC mIoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>MonoScene [7]</td>
<td>23.96</td>
<td>7.31</td>
</tr>
<tr>
<td>Atlas [38]</td>
<td>28.66</td>
<td>15.00</td>
</tr>
<tr>
<td>BEVFormer [30]</td>
<td>30.50</td>
<td>16.75</td>
</tr>
<tr>
<td>TPVFormer [21]</td>
<td>11.51</td>
<td>11.66</td>
</tr>
<tr>
<td>TPVFormer*</td>
<td>30.86</td>
<td>17.10</td>
</tr>
</tbody>
</table>

Table 2. Evaluation metrics for 3D scene reconstruction. 

\[
\begin{array}{|c|c|c|c|c|c|c|c|c|c|}
\hline
\text{Method} & \text{barrier} & \text{bicycle} & \text{bus} & \text{car} & \text{const. veh} & \text{monkey.cyc} & \text{Pedestrian} & \text{traffic cone} & \text{trailer} & \text{train} & \text{drive. suf.} & \text{other flat} & \text{sidewalk} & \text{terrain} & \text{manmade} & \text{vegetation} \\
\hline
\hline
\hline
\end{array}
\]
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 [7]. Although our method is not designed for monocular perception, we still outperform other methods for a large margin.

![Image of challenging scenes. Although the quality of RGB images degrades in rainy days and nights, our method can still predict detailed occupancy. Better viewed when zoomed in.](image)

Figure 6. An example of challenging scenes. Although the quality of RGB images degrades in rainy days and nights, our method can still predict detailed occupancy. Better viewed when zoomed in.

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

5.2. 3D Semantic Occupancy Prediction

We first conduct multi-camera 3D semantic occupancy prediction on nuScenes [6] dataset and compare with several state-of-the-art methods [7, 30, 21, 38]. For MonoScene [7], we concatenate the multi-camera occupancy predictions and voxelize them with the 0.5m voxel size, which is same as our setting. For BEVFFormer [30], we add a 3D segmentation head to predict semantic occupancy.

As shown in Table 1, our method achieves state-of-the-art performance. We also show some qualitative results in Figure 6 and Figure 7. See supplementary material for more video demos and qualitative comparisons.

Especially in Figure 6, we show rainy day and night visualization. Although the quality of RGB images degrades in these two challenging scenes, our method can still predict fine details. The LiDAR sensor suffers from rainy days and easily misses points. With multi-frame aggregation and Possion Reconstruction, we dramatically densify the labels and provide strong supervision, which is crucial for the challenging scenarios. Moreover, we conduct color jitter augmentation, which increases the robustness of brightness change. We also note that some parts cannot be observed by LiDAR sensor, such as the back side of the car. Surprisingly, our model can predict the complete shape according to the surrounding information.

To further demonstrate the superiority of our method, we also conduct monocular 3D semantic scene completion on SemanticKITTI dataset [2]. Table 3 shows the results. 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.
<table>
<thead>
<tr>
<th>Method</th>
<th>SC IoU</th>
<th>SSC mIoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o spatial attention</td>
<td>29.78</td>
<td>17.34</td>
</tr>
<tr>
<td>BEV-based attention</td>
<td>30.45</td>
<td>18.94</td>
</tr>
<tr>
<td>Ours</td>
<td>31.49</td>
<td>20.30</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>SC IoU</th>
<th>SSC mIoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o multi-scale structure</td>
<td>30.41</td>
<td>18.22</td>
</tr>
<tr>
<td>w/o multi-scale supervision</td>
<td>31.16</td>
<td>19.73</td>
</tr>
<tr>
<td>Ours</td>
<td>31.49</td>
<td>20.30</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Supervision</th>
<th>SC IoU</th>
<th>SSC mIoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>sparse LiDAR points</td>
<td>11.96</td>
<td>12.17</td>
</tr>
<tr>
<td>sparse occupancy labels</td>
<td>30.58</td>
<td>18.83</td>
</tr>
<tr>
<td>dense occupancy labels</td>
<td>31.49</td>
<td>20.30</td>
</tr>
</tbody>
</table>

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.

5.3. 3D Scene Reconstruction

Another important application of 3D occupancy prediction is 3D scene reconstruction. Due to this reason, we further evaluate 3D reconstruction performance without using multi-class semantic labels. We do the comparison with state-of-the-art multi-camera depth estimation methods (SurroundDepth [58]), monocular depth estimation methods (AdaBins [3] and NeWCRFs [65]) and 3D reconstruction method (Atlas [38] and TransformerFusion [5]). For the self-supervised methods SurroundDepth [58], we use depth ground truth to supervise them. To evaluate depth estimation methods in 3D space, following [38, 5], we run TSDF fusion [12, 39] to fuse multi-camera depth as point clouds. Note that we fuse the multi-camera depth maps of the same timestamp but not the multi-frame depths. Thus, there is no need to specially deal with movable objects. As shown in Table 4, SurroundOcc achieves state-of-the-art performance on most metrics, and outperforms other methods by a large margin, which verifies the effectiveness of the proposed method.

5.4. Ablation Study

2D-3D Spatial Attention: Table 5 shows the ablation results for 2D-3D spatial attention. Without spatial attention, we directly average all multi-camera features in a grid. However, we find this straightforward fusion method performs worse than spatial attention. The potential reason is that the contribution of each view is different for a 3D grid. Moreover, the ablation study shows that 3D-based cross-view attention is more effective than BEV-based cross-view attention since it can preserve 3D space information.

Multi-scale Occupancy Prediction: We conduct an ablation study on multi-scale structure and multi-scale supervision in Table 6. The experimental results show that these two multi-scale designs can boost the performance. The multi-scale skip connection can help the network learn low-level fine-grained features, which is important for detailed high-resolution occupancy prediction. Moreover, the 3D volume features in all levels will be enhanced by multi-scale supervision.

Dense Occupancy Supervision: The results in Table 7 demonstrate the importance of using dense occupancy as ground truth. Compared with sparse LiDAR points, the sparse occupancy labels fuse multi-frame points and can provide more powerful supervision. The pose reconstruction and NN algorithm can fill the holes and further densify the occupancy labels, which will boost model’s performance. Figure 7 can better illustrate the effectiveness of dense supervision. We can see that the model trained with our occupancy labels can predict much denser occupancy than that trained with LiDAR points.

5.5. Model Efficiency

We compare the inference time and inference memory of different methods in Table 8. The experiments are conducted on one RTX 3090 with six multi-camera images, whose resolutions are 1600x900. We find that our method can achieve both high performance and efficiency. Compared with BEVFormer, our method slightly increases inference time and memory and we think the increased burden is acceptable.

6. Limitations and Future Work

Currently, we only explore single-frame occupancy prediction. However, for the downstream modules, e.g. motion prediction and planning, occupancy flow is more important. As the future work, we will design a framework to build...
occupancy flow dataset and utilize multi-frame surrounding images as the inputs. Moreover, LiDAR data is not always available. Self-supervised occupancy prediction with only RGB data is a valuable but challenging direction.

7. Conclusion

In this paper, we propose SurroundOcc for multi-camera 3D occupancy prediction. We utilize 2D-3D spatial attention to integrate 2D features to the 3D volume in a multi-scale fashion, which is further upsampled and fused by the 3D deconvolution layer. Moreover, we devise a pipeline to generate dense occupancy ground truth. We stitch multi-frame LiDAR points of dynamic objects and static scenes separately and utilize Poisson Reconstruction to fill the holes. The comparison on nuScenes and SemanticKITTI datasets demonstrates the superiority of our method.

Acknowledgement

This work was supported in part by the National Key Research and Development Program of China under Grant 2022ZD0160102 and in part by the National Natural Science Foundation of China under Grant 62125603.
References


[31] Tingting Liang, Hongwei Xie, Kaicheng Yu, Zhongyu Xia, Zhiwei Lin, Yongtao Wang, Tao Tang, Bing Wang, and Zhi


[61] Dongqiangzi Ye, Zixiang Zhou, Weijia Chen, Yufei Xie, Yu Wang, Panqu Wang, and Hassan Foroosh. Lidarmultinet:
Towards a unified multi-task network for lidar perception.  


