SfmOcc: Vision-Based 3D Semantic Occupancy Prediction in Urban Environments

Rodrigo Marcuzzi Lucas Nunes Elias Marks Louis Wiesmann
Thomas Läbe Jens Behley Cyrill Stachniss

***Abstract*—Semantic scene understanding is crucial for autonomous systems and 3D semantic occupancy prediction is a key task since it provides geometric and possibly semantic information of the vehicle's surroundings. Most existing vision-based approaches to occupancy estimation rely on 3D voxel labels or segmented LiDAR point clouds for supervision. This limits their application to the availability of a 3D LiDAR sensor or the costly labeling of the voxels. While other approaches rely only on images for training, they usually supervise only with a few consecutive images and optimize for proxy tasks like volume reconstruction or depth prediction. In this paper, we propose a novel method for semantic occupancy prediction using only vision data also for supervision. We leverage all the available training images of a sequence and use bundle adjustment to align the images and estimate camera poses from which we then obtain depth images. We compute semantic maps from a pre-trained open-vocabulary image model and generate occupancy pseudo labels to explicitly optimize for the 3D semantic occupancy prediction task. Without any manual or LiDAR-based labels, our approach predicts full 3D occupancy voxel grids and achieves state-of-the-art results for 3D occupancy prediction among methods trained without labels.**

***Index Terms*—Semantic Scene Understanding, Deep Learning Methods**

I. Introduction

In the context of outdoor navigation, scene understanding plays a crucial role to enable safe navigation in complex environments. 3D perception tasks often rely on costly 3D sensors like LiDAR, whereas, vision-centric scene understanding seeks to provide meaningful information about the surrounding scene using only RGB cameras.

Among the different vision-centric tasks, 3D semantic occupancy prediction aims to represent the 3D geometric structure of the surrounding scene from a setup of surrounding cameras, this means, extracting 3D information from images. Other tasks for geometric and semantic scene understanding from images include 3D object detection and depth estimation. Compared to 3D object detection [19], [20], [29], occupancy prediction [16], [17], [27], [34] provides a more fine-grained representation and in contrast to monocular depth estimation [12], [38], it seeks to also reconstruct the environment in occluded areas.

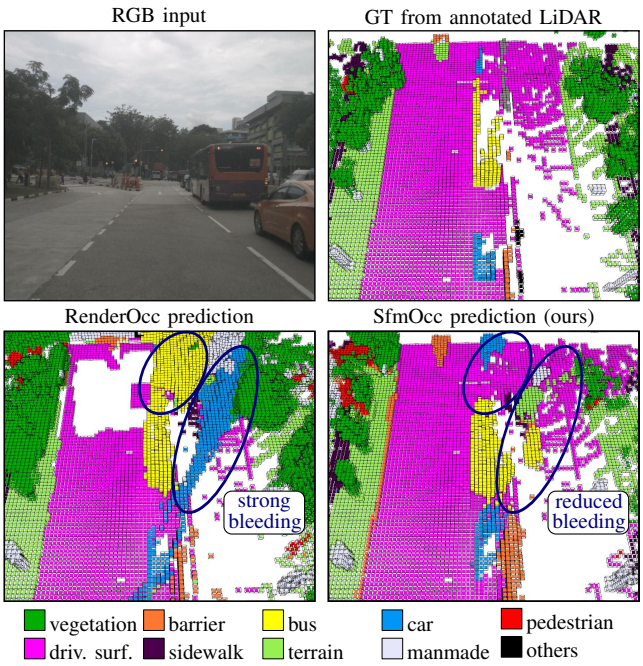


Fig. 1: Approaches that do not rely on ground-truth occupancy labels for training like RenderOcc [27] usually perform implicit supervision and have bleeding effects in their predictions as highlighted by circles in the image. Using only RGB images for training, we explicitly supervise for occupancy, which allows us to predict occupancy and reduce the bleeding.

State-of-the-art approaches for 3D semantic occupancy prediction [17], [34], [42] rely on 3D voxel labels as an explicit supervision signal to learn the full occupancy of the surrounding scene. Due to the costly annotation process, some recent approaches project segmented LiDAR scans [3], [27] into the images and use these for supervision. Given the sparsity of a 3D LiDAR compared to camera images, they use multiple past and future scans to obtain denser supervision signals but rely on the availability of a LiDAR sensor. In contrast, recent research [16], [41] investigates using only images for training. These approaches leverage multiple temporally close RGB images jointly for supervision and consider multi-view consistency to train a depth estimator for the surrounding views but do not exploit all available images taken of the whole scene. Some methods that use LiDAR scans [3], [27] or RGB images [16], [41] provide implicit supervision to learn

Manuscript received: Dec 13, 2024; Revised: Feb 24, 2025; Accepted: Mar 27, 2025. This paper was recommended for publication by Editor Markus Vincze upon evaluation of the Associate Editor and Reviewers' comments.

All authors are with the Center for Robotics, University of Bonn, Germany. Cyrill Stachniss is additionally with the Department of Engineering Science at the University of Oxford, UK, and with the Lamarr Institute for Machine Learning and Artificial Intelligence, Germany.

This work has partially been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy, EXC-2070 – 390732324 – PhenoRob, and by the German Federal Ministry of Education and Research (BMBF) in the project "Robotics Institute Germany", grant No. 16ME0999.

Digital Object Identifier (DOI): see top of this page.

the occupancy by optimizing for a proxy task like volume rendering [26]. Due to their training scheme, the predictions of these methods usually present bleeding effects at the object boundaries as shown in Fig. 1.

In this paper, we tackle the task of 3D semantic occupancy prediction using only RGB images for training. We propose to explicitly supervise for 3D semantic occupancy prediction while relying only on image information to better predict the full geometry and semantics of the scene, also behind occluded voxels as shown in Fig. 1.

The main contribution of this paper is a method to generate occupancy pseudo labels relying solely on images that we can use to supervise networks for 3D semantic occupancy and predict the full occupancy of the surrounding area. We leverage all available images of the training set and use bundle adjustment to align the images of each scene and compute the camera poses. We exploit this information to generate depth images to replace the LiDAR supervision. We furthermore generate triangle meshes and use them to filter out depth values belonging to occluded objects. We leverage a foundation model to semantically segment the RGB images and combine them with the depth images to generate sparse occupancy pseudo labels as explicit supervision. We achieve state-of-the-art performance while using only unlabeled RGB images for training.

Our experiments suggest that without any labels, (i) our approach achieves state-of-the-art performance on 3D semantic occupancy prediction among methods using only images for training; and (ii) our depth filtering using a triangular mesh improves the performance of 3D occupancy prediction. The paper and our experimental evaluation back up these claims.

The implementation and generated pseudo labels used in our approach are available at https://github.com/PRBonn/SfmOcc.

II. Related Work

In this section, we examine works belonging to the main tasks related to 3D semantic occupancy prediction. We highlight recent research directions and common approaches together with the current challenges.

Depth Estimation is a key task in predicting 3D geometry given images as input. Early works on depth estimation [22], [43] rely on depth annotations for supervision. However, more recent methods [24], [28], [33], [36], [39] use a self-supervised approach to predict depth maps and ego-motion and supervise through photometric constraints [11] between adjacent frames. To adapt to multi-camera setups normally used for autonomous vehicles, some approaches [12], [38] tackle depth estimation using the multi-view information. In particular, SurroundDepth [38] uses structure-from-motion (sfm) in the overlap between cameras to obtain scale-aware pseudo depths to pre-train the models and increase performance. This highlights how even sparse depth supervision obtained from structure-from-motion can help in obtaining real-world depth predictions.

Instead of using only the small overlap between cameras, we propose to use all available images captured by the multi-camera setup for each scene. We perform bundle adjustment to obtain camera poses and sparse scale-aware depth images for the whole training set.

3D Semantic Occupancy Prediction, also known as semantic scene completion, aims to provide a complete understanding of the scene by estimating the 3D geometry and semantics within a predefined voxel grid. SemanticKITTI [1] introduced a benchmark for 3D semantic scene completion based on a LiDAR and stereo camera setting. While some works focus on reconstructing the surroundings using LiDAR data [7], [18], [44] others use monocular or stereo cameras [6], [17]. Occ3D-nuScenes [34] provides 3D semantic occupancy ground-truth for the nuScenes dataset [5], where the input is the six images of the surrounding camera setup. Various methods [10], [17], [37] build on top of 3D object detection architectures [15], [20] to extract 2D features and lift them into a common 3D space. Some of the most recent approaches [23], [40] focus on efficient supervision while others design new architectures to improve the performance [21], [42]. Due to the challenging nature of the 3D occupancy task, these methods rely on costly 3D ground-truth and laborious annotations. Instead of using 3D voxel labels, some approaches [3], [9], [27] train with 2D depth and semantic labels obtained by projecting the segmented LiDAR scan into the images, which requires annotation of the LiDAR data.

To avoid relying on expensive 3D sensors and labeling of 3D data, some approaches [16], [41], [42] use only unlabeled camera images and implicitly learn occupancy by supervising with a few subsequent images and optimizing for a proxy task like volume rendering [26]. These methods [9], [16], [41] use foundation models [32] to obtain semantic labels for each image during training.

We follow this line of work and only use images as supervision. However, instead of using a few sequential images for supervision, we propose to leverage all available images of the scene and bundle adjustment to generate sparse occupancy pseudo labels and explicitly supervise the network for occupancy prediction. We furthermore exploit a foundation model to generate semantics for each image, which are added to our pseudo labels to provide geometric and semantic supervision.

III. Our Approach to 3D Semantic Occupancy Prediction

A. Task Definition

Given a set of RGB images $\mathcal{I} = \{I_{1,t}, \dots, I_{N,t}\}$ from a setup with N surrounding cameras recorded at timestep t, the objective is to predict the geometry and semantics of the surroundings as a dense 3D voxel grid $\tilde{\boldsymbol{O}} \in \mathbb{R}^{H \times W \times D \times B}$ where H, W, D is the volume resolution and B is the number of semantic classes (including the "empty" class).

We show an overview of our approach, called SfmOcc, in Fig. 2. We first extract 3D voxel features $\boldsymbol{F} \in \mathbb{R}^{H \times W \times D \times C}$ from the input images $\mathcal{I}$ using a network G, where C is the feature dimension. The network G usually extracts 2D image features and projects them to 3D by either first predicting depth [27] or via attention [17]. From the voxel features $\boldsymbol{F}$, we obtain semantic logits and occupancy probabilities using two MLP heads ϕ_s and ϕ_o and supervise the model using

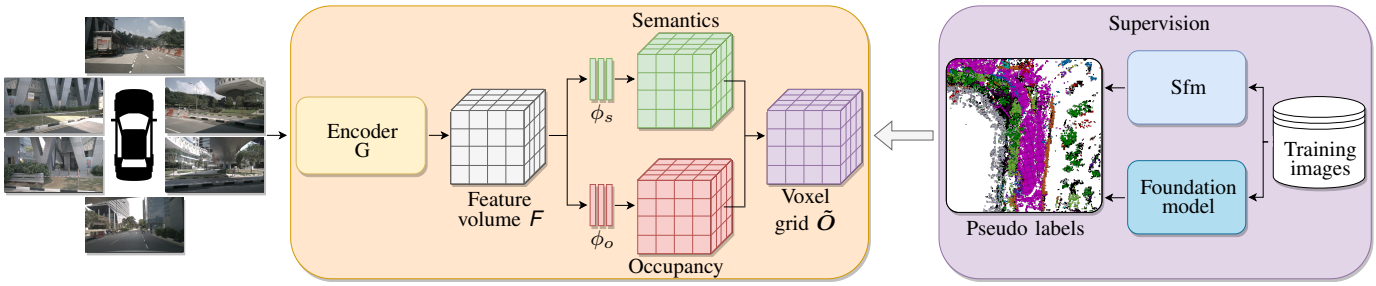
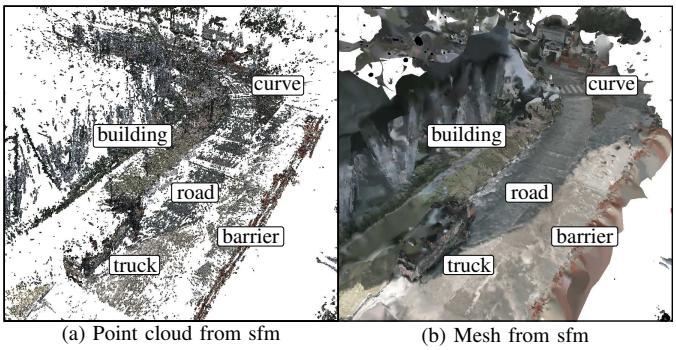


Fig. 2: Overview of our approach called SfmOcc. Given a set of RGB images from a multi-camera setup, we extract a 3D volume feature F and use MLP heads ϕ_s and ϕ_o to predict semantic logits and occupancy probabilities for each voxel, which we fuse to obtain the final voxel grid $\tilde{O}$. For supervision, we leverage all available training images and use structure-from-motion (sfm) and volume reconstruction to generate sparse occupancy pseudo labels. We use a foundation model to obtain semantic maps for the images and set the semantic class of each voxel. This way, we explicitly supervise our approach for semantic occupancy prediction while relying only on camera data.



(a) Point cloud from sfm (b) Mesh from sfm

Fig. 3: Point cloud (a) and triangle mesh (b) obtained from structure-from-motion for one scene. The scene depicts a parked truck on the left side of the road, construction barriers on the right, a building on the left, and a curvature at the end of the road.

sparse occupancy pseudo labels. We obtain the final semantic occupancy predictions by assigning the predicted semantic class to the voxels predicted as occupied. Note that our method is not bounded to a particular network G to perform occupancy prediction and could be used to train different models by replacing the supervision.

To obtain pseudo labels for each scene, we use all images recorded by the camera setup at all timesteps $t \in \{t_1, \dots, t_M\}$. We use bundle adjustment to align all $N \cdot M$ images in the scene, obtain camera poses, and then generate depth images as explained in the next section. We further use these depth images to perform volume reconstruction and obtain occupancy pseudo labels. We use a foundation model [32] to obtain 2D semantic maps for each image and generate semantic occupancy pseudo labels.

B. Depth Image Generation

We leverage all the available image information to obtain depth images that we can further use to generate occupancy pseudo labels. For each recorded scene, we use bundle adjustment [35] to align the RGB images $\{\{I_{1,1}, \dots, I_{N,1}\}, \dots, \{I_{1,M}, \dots, I_{N,M}\}\}$ captured by the N cameras at the M timesteps in the scene.

Given that the N cameras are rigidly attached, we optimize the poses $\mathsf{T}_{i,t} \in \mathbb{R}^{4\times4}$ of a single camera i for each timestep t and, for each other camera j, the transformation ${}^{j}\mathsf{T}_i \in \mathbb{R}^{4\times4}$ to camera i for the whole sequence. Furthermore, we fix the intrinsics K_i for each camera and do not optimize them, i.e., assuming a fixed, not-changing camera calibration. These two modifications allow us to optimize a smaller number of parameters and better constrain the adjustment, which in turn leads to more reliable results and is more robust to outliers from dynamic objects. This way, we obtain the camera poses for each camera, i.e., $\mathsf{T}_{j,t} = {}^{j}\mathsf{T}_i \mathsf{T}_{i,t}$ at each given timestep t in the scene. To simplify notation, when referring to a camera i at a timestep t, we drop the subscript t.

Given the optimized camera poses T_i, we use multi-view stereo reconstruction [8], [30] to generate a sparse point cloud $\mathcal{P}$ for the given scene as shown in Fig. 3 (a). We use the camera intrinsics K_i to project the point cloud $\mathcal{P}$ into each camera i and obtain a depth image $D_i^{\mathcal{P}} = \pi(\mathcal{P}, \mathsf{T}_i, \mathsf{K}_i)$, where π represents the projection of each point in the point cloud to the depth image, as shown in Fig. 4.

Each depth image $D_i^{\mathcal{P}}$ provides depth for the reconstructed regions. Due to the sparsity of the point cloud $\mathcal{P}$, the obtained depth images $D_i^{\mathcal{P}}$ often have pixels with wrong depth belonging to occluded objects or areas. This can be seen in the point cloud obtained when unprojecting a depth image $D_i^{\mathcal{P}}$ to 3D, as shown in Fig. 4.

C. Depth Filtering

As our point cloud $\mathcal{P}$ is sparse, points belonging to occluded objects are projected into the depth image $D_i^{\mathcal{P}}$, generating wrong depth values. If we obtain a closed surface without holes, we can avoid the projection of occluded points into the depth image. We can obtain such a surface by generating [13], [31] a triangle mesh $\mathcal{T}$ of the given scene using the poses from bundle adjustment to align the predicted depths from the multi-view stereo reconstruction. We show such mesh in Fig. 3 (b). Projecting the mesh we obtain the depth image $D_i^{\mathcal{T}} = \pi(\mathcal{T}, \mathsf{T}_i, \mathsf{K}_i)$ as shown in Fig. 4. We can observe that the mesh generation introduces blob-like artifacts and problems in the boundaries, which prevents us from directly using the depth image $D_i^{\mathcal{T}}$ for supervision. However, we can use this depth image $D_i^{\mathcal{T}}$ to filter out pixels with wrong depth in $D_i^{\mathcal{P}}$ belonging to occluded areas.

We compute the residual between both depth images $\hat{D}_i = |D_i^{\mathcal{P}} - D_i^{\mathcal{T}}|$ and filter out pixels with residual larger

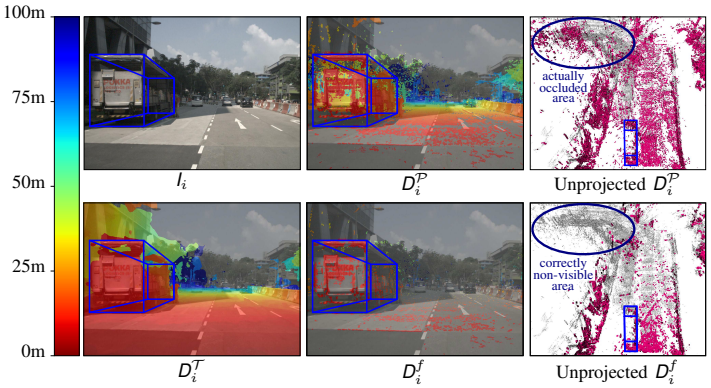


Fig. 4: Filtering of depth images. We project the point cloud $\mathcal{P}$ into the image I_i and obtain a depth image $D_i^{\mathcal{P}}$. Due to the sparsity of the input point cloud, pixels associated with occluded points have wrong depth values. This is better visible when unprojecting the depth image to 3D. We show the full sfm point cloud for reference. We project the triangle mesh $\mathcal{T}$ into the image I_i to obtain a depth image $D_i^{\mathcal{T}}$ and use it to filter out pixels with wrong depth. The result is D_i^f. We unproject it to 3D to better show the filtering. We draw a bounding box around the truck for better visualization.

than a threshold τ, which results in the filtered depth map D_i^f for image I_i:

$$D_i^f(u,v) = \begin{cases} D_i^{\mathcal{P}}(u,v) & \text{, if } \hat{D}_i(u,v) \leq \tau \\ 0 & \text{, otherwise} \end{cases}. \tag{1}$$

The depth image from the mesh $D_i^{\mathcal{T}}$ represents the closest surface to the camera and therefore the pixels (u, v) belonging to occluded objects have different depth values and a large residual in $\hat{D}_i$, which allows us to filter them out.

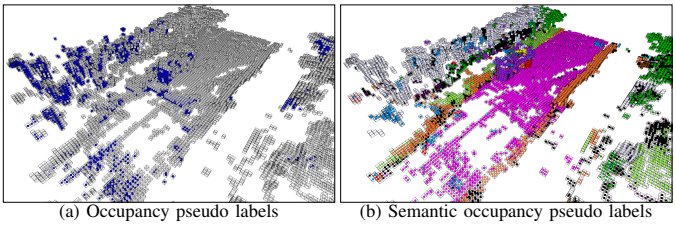
We show an example of the obtained depth image D_i^f in Fig. 4. Given the filtering, the result is less dense than $D_i^{\mathcal{P}}$ but pixels with wrong depth are discarded, as seen in Fig. 4, by unprojecting the depth images into 3D.

It is important to mention that during this process, also pixels with correct depth are removed, making the supervision even sparser. This filtering helps the network to better learn the depth and to generate better occupancy pseudo labels, as discussed in Sec. IV-F1.

D. Occupancy Pseudo Labels

The obtained depth images D_i^f can be used to replace the supervision in methods that use projected LiDAR for supervision [3], [27]. To explicitly supervise our method for occupancy, we can use the depth images along with the camera poses from bundle adjustment to generate sparse occupancy pseudo labels. Since we have depth images for the whole scene, we use the $N \cdot M$ depth images D_i^f to obtain a denser supervision, also behind occluded voxels in the current view.

We build a global voxel grid $\mathcal{V}_g \in \mathbb{R}^{H' \times W' \times D' \times B}$, where H', W', D' are the dimensions of the complete sequence and perform a similar operation to occupancy mapping [14]. For each ray of each image, we assume that the endpoint corresponds to a surface and that the line of sight between the camera center and endpoint is free space. To determine the voxels that need to be updated, we perform a ray-casting operation to determine voxels along the ray from the camera center to the endpoint. We use a 3D variant of the Bresenham's algorithm [4] to approximate the ray and step through the voxel grid from the camera center to the endpoint of the ray. We set all traversed voxels as "empty" and the voxel at the end of the ray as "occupied". We show examples of our generated pseudo labels from one or all timesteps in Fig. 5. With this approach, we combine all the different viewpoints of all cameras and obtain sparse occupancy pseudo labels for the whole scene, including free space and occluded voxels behind objects. This allows us to directly supervise occupancy probabilities instead of relying on proxy tasks like volume rendering [3], [16], [27].



(a) Occupancy pseudo labels (b) Semantic occupancy pseudo labels

Fig. 5: Generated occupancy pseudo labels. We show in (a) the occupancy labels generated from a single (blue) and from all depth images in the scene (gray) and in (b) the corresponding semantic classes for each occupied voxel.

E. Semantic Maps

To supervise for semantic occupancy, we extract semantic maps using a pre-trained open-vocabulary model [32]. This way, we obtain 2D semantic maps for each image without any 2D or 3D ground-truth label. To add semantic information in the pseudo label generation, instead of setting voxels as "occupied" at the end of a ray, we accumulate in a histogram the class of the ray, which we obtain from the corresponding semantic map. At the end of the generation, we perform majority voting for each voxel to obtain the final semantic class. Given the inconsistency of the semantics predicted by the pre-trained model for the different views, the obtained labels may not be consistent for the different objects or surfaces but still provide reasonable classes for the whole scene as shown in Fig. 5.

IV. Experimental Evaluation

The main focus of this work is a method to perform 3D semantic occupancy prediction, which is trained using only RGB images. We leverage all available training images to generate sparse pseudo labels and explicitly supervise for occupancy. We present our experiments to show the capabilities of our method, and support our key claims: (i) our approach achieves state-of-the-art performance on 3D semantic occupancy prediction among methods using only images for training, and (ii) our depth filtering using a triangular mesh improves the performance of 3D occupancy.

A. Implementation Details

We use BEVStereo [19] as the network G to obtain the 3D voxel features F. We predict 3D semantic occupancy for a voxel grid with resolution $H = 200$, $W = 200$,

$D = 16$ and voxel size $\{0.4 \times 0.4 \times 0.4\}\, m^3$. We keep the original architecture and only add the semantic and occupancy heads ϕ_s, ϕ_o, which both consist of two linear layers and softplus activation layers. We follow the training strategy of RenderOcc [27], resize the image to 512×1408 pix and use AdamW [25] optimizer to train for 12 epochs with learning rate of 10^{-4} and batch size 8. We supervise both the semantic logits and the occupancy probabilities using Cross Entropy loss between predictions and pseudo labels. All experiments are conducted on 8 NVIDIA A40 GPUs.

We filter static sequences where the car moved less than 1 m and discard complete scenes where less than 75% of the images were successfully aligned. We use the camera intrinsics K_i provided by the dataset and do not optimize them. To obtain semantic maps, we use Grounded SAM [32] and follow OccNerf [41] to prompt the model with multiple synonyms of each class name of the nuScenes dataset.

B. Experimental Setup

We evaluate our method on Occ3D-nuScenes [34] dataset, which provides 3D semantic occupancy ground-truth for the images of nuScenes [5]. nuScenes provides 1000 driving scenes with six surrounding-view cameras. The occupancy ground-truth covers a range of $[-40, 40]$ m in x and y direction and $[-1, 5.4]$ m in z direction with voxel size $\{0.4 \times 0.4 \times 0.4\}\, m^3$ meters and contains 17 semantic classes, the 16 classes of nuScenes plus an extra "empty" class. We use IoU to evaluate occupancy performance without considering semantics and mIoU to get the average across all non-empty semantic classes. Occ3D-nuScenes only provides labels for the training and validation set of nuScenes but not for the test set. To avoid running experiments in the same set of 150 scenes (validation) where we evaluate our performance, we separate 60 scenes from the training set and use this set as our validation set. This way, we run our experiments and evaluate our performance in different subsets of the data.

C. Pseudo Labels

We generate pseudo labels only for the training set. However, given that we use bundle adjustment, we do not consider scenes where the image alignment fails like scenes where the vehicle does not move, too dark scenes or scenes with too many dynamic objects. We obtain depth images and pseudo labels for 584 out of the 700 scenes in the training set and use 60 of the remaining scenes as validation set. The bundle adjustment and depth image generation take around 25 min per scene while the occupancy pseudo-label generation takes around 2 min per scene. We generate a total of $127,458$ depth images, which we use to generate $21,231$ voxel-level sparse pseudo labels covering the surroundings of the vehicle with the same voxel grid and range as Occ3D-nuScenes. The generated pseudo labels contain 18 semantic classes including the 17 of Occ3D-nuScenes, and an extra "uncertain" class, which represents occupied voxels with no class given by the pre-trained semantic model. We consider the voxels with this class for occupancy but not for semantic supervision.

Method	Mode	GT Sem.	IoU [%]	mIoU [%]
OccFormer [42]	3D	✓	-	21.9
TPVFormer [17]	3D	✓	-	27.8
CTF-Occ [34]	3D	✓	-	**28.5**
BEVStereoOcc	3D	✓	**51.6**	27.7
TPVFormer [17]	L	✓	17.2	13.6
RenderOcc [27]	L	✓	**45.9**	23.9
OccFlowNet [3]	L	✓	-	**26.1**
SimpleOcc [34]	C		-	7.1
SelfOcc [16]	C		45.0	9.3
OccNeRF [41]	C		45.0	9.5
GaussianOcc [9]	C		-	9.9
LangOcc [2]	C		51.8	11.8
SfmOcc (ours)	C		**57.7**	**17.7**

TABLE I: 3D semantic occupancy prediction performance on occ3D-nuScenes. In the column mode: 3D are methods trained with occupancy ground truth labels, L trained with LiDAR supervision, and C trained only with cameras. GT Sem. indicates the usage of ground-truth semantic labels for supervision. We denote our model trained with ground-truth labels as BEVStereoOcc.

Method	mIoU	bicycle	bus	car	cons. veh.	motorcycle	pedestrian	trailer	truck
SelfOcc [16]	10.5	0.1	6.6	13.2	0.0	0.4	2.4	0.0	7.7
GaussianOcc [9]	11.0	3.8	14.6	17.2	0.8	2.9	10.1	0.14	10.6
SfmOcc (ours)	**19.6**	**8.2**	**14.8**	**20.9**	**7.1**	**12.0**	**12.2**	**1.6**	**15.9**

TABLE II: 3D semantic occupancy prediction performance on scenes with many dynamic objects in Occ3D-nuScenes.

D. 3D Semantic Occupancy Prediction Performance

The first experiment evaluates the performance of our approach in Occ3D-nuScenes [34] dataset and shows that our approach achieves state-of-the-art performance among methods supervised using only camera data. In Tab. I, we compare against methods that use as supervision the ground-truth 3D labels (3D), LiDAR (L), and only cameras (C).

Our approach SfmOcc surpasses previous camera-only methods by at least 5.8 percent points both in terms of IoU and mIoU. We are able to predict better 3D geometry, shown by our higher IoU compared to the baselines and we even outperform methods trained with LiDAR supervision both in IoU and mIoU. We also train our same architecture with ground truth labels and denote it as BEVStereoOcc. Relying on images only, our approach SfmOcc is able to learn the free space but drops in terms of mIoU, probably because of wrong semantic classes in the generated pseudo labels.

E. Performance in Scenes with Dynamic Objects

We generate occupancy pseudo labels relying on bundle adjustment with the static scene assumption, where we do not reconstruct dynamic objects. In this experiment, we compare the performance of different image-based methods on scenes with many dynamic objects. We select 30 scenes with more dynamic objects in the validation set of Occ3D-nuScenes and evaluate image-based methods in Tab. II. Our approach outperforms other methods relying on images for training for all dynamic classes and in the average of all classes. Other methods that supervise with a few consecutive images also rely on multi-view geometry for supervision and therefore face the same problem.

#	Depth supervision	IoU [%]	mIoU [%]
A	LiDAR	53.8	16.4
B	Depth images $\boldsymbol{D}^{\mathcal{P}}$	40.5	12.3
C	Filtered depth images $\boldsymbol{D}^{f}$	**46.6**	**14.0**

TABLE III: Semantic occupancy prediction performance supervising only with depth images. We compare using depth images obtained by projecting LiDAR scans and by projecting the sfm point cloud $\mathcal{P}$ before $\boldsymbol{D}^{\mathcal{P}}$ and after $\boldsymbol{D}^{f}$ filtering.

#	Pseudo labels	IoU [%]	mIoU [%]
D	Depth images $\boldsymbol{D}^{\mathcal{P}}$	60.7	18.7
E	Filtered depth images $\boldsymbol{D}^{f}$	**68.3**	**20.5**

TABLE IV: Semantic occupancy prediction performance supervising with pseudo labels from unfiltered and filtered depth images.

#	Single image	All images	Volume rec.	IoU [%]	mIoU [%]
F	✓			38.4	13.9
G		✓		38.5	14.5
H	✓		✓	58.6	17.2
I		✓	✓	**68.3**	**20.5**

TABLE V: Influence in semantic occupancy prediction performance supervising with different sparse pseudo labels. Either using a single or multiple images to generate them and whether or not to use volume reconstruction to obtain supervision for the free space.

Pseudo labels	IoU [%]	mIoU [%]
Whole voxel grid	49.6	8.7
Visible area only	**50.4**	**15.7**

TABLE VI: Evaluation of our generated pseudo labels vs. the ground-truth voxel labels for the training set of Occ3D-nuScenes.

F. Ablation Studies

We show the influence of our design choices in the performance of the approach, namely the depth filtering and different ways of generating pseudo labels. We evaluate our model in the 60 scenes that we separated from the training set. We indicate each experimental setup with capital letters [A], [B], etc. in Tabs. III, IV and V.

1) Depth Filtering: In this experiment, we show the importance of the depth filtering to obtain depth supervision and generate the occupancy pseudo labels.

We first supervise only with depth images and the corresponding semantic maps. We follow RenderOcc [27] and randomly sample camera rays and use volume rendering to obtain their semantic and depth values. We show our results in Tab. III. In [A], we get depth images by projecting the LiDAR using the calibration between sensors. Due to the sparsity of the LiDAR, we follow a common practice [27] and use a window of 7 consecutive LiDAR scans for a denser supervision. We achieve 16.4% mIoU and 53.8% IoU. In [B], we use the depth images $\boldsymbol{D}^{p}$ obtained by projecting the point cloud $\mathcal{P}$ into the image plane. These depth images contain pixels with wrong depth due to the sparsity of the obtained point cloud, as explained in Sec. III-B and as shown in Fig. 4. Given the density of the supervision, we can train using the depth images for a single timestep, resulting in 12.3% mIoU and 40.5% IoU. In [C], we use the depth images $\boldsymbol{D}^{f}$ obtained after the depth filtering with the triangle mesh. We filter out pixels with wrong depth, as explained in Sec. III-C and shown in Fig. 4. This filtering allows us to achieve 14.0% mIoU and 46.7% IoU, improving 1.7 percent points and 6.1 percent points respectively and shows that we improve the performance.

For the next experiments we train using only the sparse occupancy pseudo labels generated with the depth images as explained in Sec. III-D and show the results in Tab. IV. In [D], we use the depth images $\boldsymbol{D}^{\mathcal{P}}$ obtained by projecting the point cloud $\mathcal{P}$ into the image plane, and in [E], we use the filtered depth images $\boldsymbol{D}^{f}$. Compared with Tab. III, training with pseudo labels instead of only using depth images improves the IoU by at least 14 percent points and the mIoU by around 4 percent points. As shown in Tab. IV, our depth filtering helps to remove invalid depth values due to occlusions and improves the performance from 60.7 to 68.3 in terms of IoU and from 18.7 to 20.5 in terms of mIoU. This highlights the importance of filtering wrong depth values before generating the occupancy pseudo labels.

2) Pseudo Label Generation: In this experiment, we show how the different ways of generating pseudo-labels presented in Sec. III-D influence the model performance.

In [F], we only use depth images for a given timestep unproject them into a point cloud, and voxelize them. This way, we get supervision only for the occupied voxels as shown in Fig. 5 (a) and only up to the first occupied voxel. The IoU reaches 38.44 due to the sparsity of the supervision and the fact that we can only supervise the occupied voxels.

In [G], we aggregate the point clouds from all the depth images for the scene and voxelize it. In this case, we obtain supervision for occupied voxels in the complete scene. This includes voxels occluded by objects, but which are visible from a different view, as shown in Fig. 5 (b). Here we only supervise occupied voxels and therefore the occupancy prediction performance is similar to [F], as shown by the IoU. However, since we have more voxel with semantic classes as supervision, the mIoU improves from 13.9 to 14.5.

In [H], we use the depth images from a single timestep to perform volume reconstruction and also set all the traversed voxels as "empty", as explained in Sec. III-D. Here, we only have values for the voxels up to the first object. While the mIoU improves around 3 percent points with respect to [G], the IoU improves by 20 percent points with respect to [F] and [G]. We argue that this is due to the supervision of empty voxels, which represent around 90% of the scene and help the model better understand the 3D geometry.

Finally, in [I], we perform volume reconstruction using all the depth images in the scene. This way, we obtain supervision for all voxels along the rays of each depth image in the scene. The IoU reaches 68.25 and mIoU 20.52 due to the supervision of both occupied and free voxels observed from the different points of view of the different cameras.

3) Evaluation of Pseudo Labels: To show the quality of our generated pseudo labels, we evaluate them against the ground-truth labels for the training set and show the results in Tab. VI.

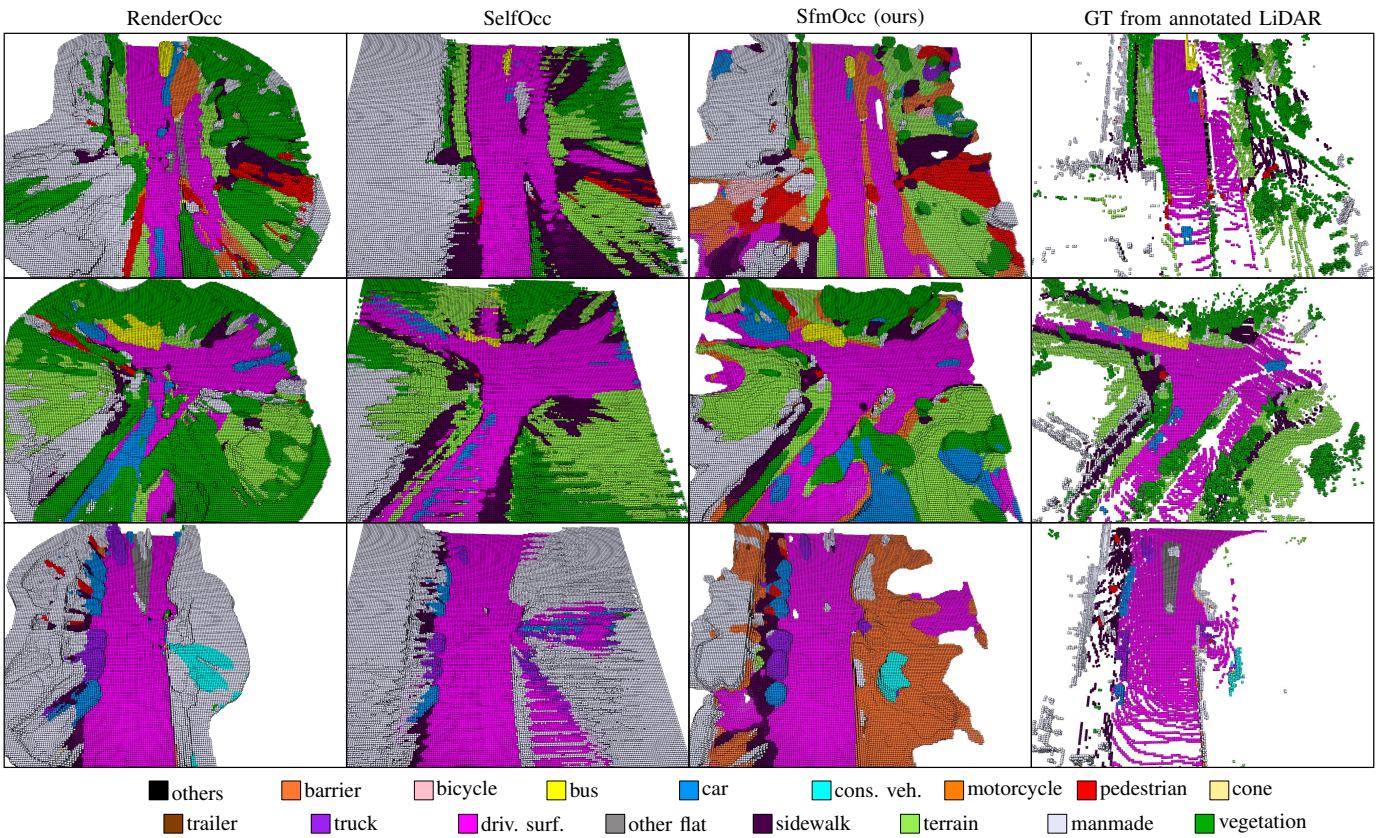


Fig. 6: Qualitative results of our approach SfmOcc. We compare against approaches that use depth images (RenderOcc) or multiple RGB images (SelfOcc) as supervision. They show a bleeding effect similar to monocular depth estimation predictions while our approach better learns the full shape of the objects, including the part that is not visible from the camera.

If we evaluate the whole voxel grid, our labels have an IoU of 49.6% and mIoU of 8.7% with respect to the ground-truth labels. During training, we do not consider the parts of the scene not seen by any camera for supervision, in which case our pseudo labels have an IoU of 50.4% and mIoU of 15.7%.

G. Qualitative Results

Finally, we compare qualitatively the semantic occupancy predictions of our approach SfmOcc with different state-of-the-art methods. Namely RenderOcc [27] trained using LiDAR and SelfOcc [16] supervised only with images [16]. We show predictions for each method in Fig. 6.

Training only with LiDAR depth, RenderOcc [27] shows bleeding at the boundaries of the objects, which is usually observed in methods that perform mono-depth estimation. This is because they only supervise each camera ray up to the first object and therefore are not able to learn the whole shape of the objects. This is similar to methods that supervise only with RGB images like SelfOcc [16], which provide, for each ray, the corresponding ray in other views and leverage multi-view consistency, and supervise using photometric constraints. On the other hand, our approach relies only on images for training but provides sparse supervision for multiple voxels in the scene, including occluded voxels. This allows the model to better learn the geometry of the scene and the shape of objects, not only the depth, and leads to more complete results. Furthermore, although the semantics of the pseudo labels are sometimes not consistent within an object, the model is able to learn a single class for the whole object. Because we do not use ground truth semantics, our method sometimes predicts wrong semantic classes.

In summary, our evaluation shows that our method achieves state-of-the-art performance on semantic occupancy prediction among methods trained only with camera data and that we are able to predict full occupancy instead of only performing depth estimation. However, due to the static surrounding assumption in our bundle adjustment system, we are currently not able to reconstruct dynamic objects in the pseudo label generation.

V. Conclusion

In this paper, we presented a novel approach to generate occupancy pseudo labels for 3D semantic occupancy prediction using only vision data. Our method allows us to supervise using only camera data and exploits the information from all the input images during training jointly. Without manual labeling and relying only on a foundation model for semantics, our approach achieves state-of-the-art performance among methods trained only with camera information and even competitive performance among methods trained with LiDAR. We implemented and evaluated our approach, provided comparisons to other existing techniques, and supported all claims made in this paper. The experiments suggest that we can use only images and structure-from-motion to generate supervision for occupancy estimation. Furthermore, we proposed a depth filtering method using a triangle mesh, which eliminates wrong depth values and improves the performance.

REFERENCES

[1] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2019.

[2] S. Boeder, F. Gigengack, and B. Risse. Langocc: Self-supervised open vocabulary occupancy estimation via volume rendering. *arXiv preprint*, arXiv:2407.17310, 2024.

[3] S. Boeder, F. Gigengack, and B. Risse. Occflownet: Towards self-supervised occupancy estimation via differentiable rendering and occupancy flow. *arXiv preprint*, arXiv:2402.12792, 2024.

[4] J.E. Bresenham. Algorithm for computer control of a digital plotter. In *Seminal graphics: pioneering efforts that shaped the field*, pages 1–6. 1998.

[5] H. Caesar, V. Bankiti, A. Lang, S. Vora, V. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuScenes: A Multimodal Dataset for Autonomous Driving. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[6] A.Q. Cao and R. De Charette. Monoscene: Monocular 3d semantic scene completion. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[7] R. Cheng, C. Agia, Y. Ren, X. Li, and L. Bingbing. S3cnet: A sparse semantic scene completion network for lidar point clouds. In *Proc. of the Conf. on Robot Learning (CoRL)*, 2021.

[8] S. Galliani, K. Lasinger, and K. Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2015.

[9] W. Gan, F. Liu, H. Xu, N. Mo, and N. Yokoya. Gaussianocc: Fully self-supervised and efficient 3d occupancy estimation with gaussian splatting. *arXiv preprint*, arXiv:2408.11447, 2024.

[10] W. Gan, N. Mo, H. Xu, and N. Yokoya. A simple attempt for 3d occupancy estimation in autonomous driving. *arXiv preprint*, arXiv:2303.10076, 2023.

[11] C. Godard, O. Mac Aodha, M. Firman, and G.J. Brostow. Digging into self-supervised monocular depth estimation. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2019.

[12] V. Guizilini, I. Vasiljevic, R. Ambrus, G. Shakhnarovich, and A. Gaidon. Full surround monodepth from multiple cameras. *IEEE Robotics and Automation Letters (RA-L)*, 7(2):5397–5404, 2022.

[13] V.H. Hiep, R. Keriven, P. Labatut, and J.P. Pons. Towards high-resolution large-scale multi-view stereo. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[14] A. Hornung, K. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees. *Autonomous Robots*, 34(3):189–206, 2013.

[15] J. Huang, G. Huang, Z. Zhu, Y. Ye, and D. Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint*, arXiv:2112.11790, 2021.

[16] Y. Huang, W. Zheng, B. Zhang, J. Zhou, and J. Lu. Selfocc: Self-supervised vision-based 3d occupancy prediction. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[17] Y. Huang, W. Zheng, Y. Zhang, J. Zhou, and J. Lu. Tri-perspective view for vision-based 3d semantic occupancy prediction. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[18] P. Li, R. Zhao, Y. Shi, H. Zhao, J. Yuan, G. Zhou, and Y. Zhang. LODE Locally Conditioned Eikonal Implicit Scene Completion from Sparse LiDAR. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2023.

[19] Y. Li, H. Bao, Z. Ge, J. Yang, J. Sun, and Z. Li. Bevstereo: Enhancing depth estimation in multi-view 3d object detection with temporal stereo. In *Proc. of the Conf. on Advancements of Artificial Intelligence (AAAI)*, 2023.

[20] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2022.

[21] Z. Li, Z. Yu, D. Austin, M. Fang, S. Lan, J. Kautz, and J.M. Alvarez. Fb-occ: 3d occupancy prediction based on forward-backward view transformation. *arXiv preprint*, arXiv:2307.01492, 2023.

[22] F. Liu, C. Shen, G. Lin, and I. Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(10):2024–2039, 2015.

[23] H. Liu, Y. Chen, H. Wang, Z. Yang, T. Li, J. Zeng, L. Chen, H. Li, and L. Wang. Fully sparse 3d occupancy prediction. *arXiv preprint*, arXiv:2312.17118, 2024.

[24] J. Liu, L. Kong, J. Yang, and W. Liu. Towards better data exploitation in self-supervised monocular depth estimation. *IEEE Robotics and Automation Letters (RA-L)*, 9(1):763–770, 2023.

[25] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *Proc. of the Intl. Conf. on Learning Representations (ICLR)*, 2019.

[26] B. Mildenhall, P. Srinivasan, M. Tancik, J. Barron, R. Ramamoorthi, and R. Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2020.

[27] M. Pan, J. Liu, R. Zhang, P. Huang, X. Li, H. Xie, B. Wang, L. Liu, and S. Zhang. Renderocc: Vision-centric 3d occupancy prediction with 2d rendering supervision. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2024.

[28] V. Patil, W. Van Gansbeke, D. Dai, and L. Van Gool. Don't forget the past: Recurrent depth estimation from monocular video. *IEEE Robotics and Automation Letters (RA-L)*, 5(4):6813–6820, 2020.

[29] J. Philion and S. Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2020.

[30] T. Pock, L. Zebedin, and H. Bischof. *TGV-fusion*. Springer, 2011.

[31] N. Poliarnyi. Out-of-core surface reconstruction via global tgv minimization. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2021.

[32] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint*, arXiv:2401.14159, 2024.

[33] L. Song, D. Shi, J. Xia, Q. Ouyang, Z. Qiao, S. Jin, and S. Yang. Spatial-aware dynamic lightweight self-supervised monocular depth estimation. *IEEE Robotics and Automation Letters (RA-L)*, 9(1):883–890, 2023.

[34] X. Tian, T. Jiang, L. Yun, Y. Mao, H. Yang, Y. Wang, Y. Wang, and H. Zhao. Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, 2024.

[35] B. Triggs, P.F. McLauchlan, R.I. Hartley, and A.W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Proc. of the Intl. Workshop on Vision Algorithms: Theory and Practice*, 1999.

[36] K. Wang, C. Liu, Z. Liu, F. Xiao, Y. An, X. Zhao, and S. Shen. Multi-view depth estimation by using adaptive point graph to fuse single-view depth probabilities. *IEEE Robotics and Automation Letters (RA-L)*, 9(7):6400–6407, 2024.

[37] Y. Wang, Y. Chen, X. Liao, L. Fan, and Z. Zhang. Panoocc: Unified occupancy representation for camera-based 3d panoptic segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[38] Y. Wei, L. Zhao, W. Zheng, Z. Zhu, Y. Rao, G. Huang, J. Lu, and J. Zhou. Surrounddepth: Entangling surrounding views for self-supervised multi-camera depth estimation. In *Proc. of the Conf. on Robot Learning (CoRL)*, 2023.

[39] X. Ye, X. Fan, M. Zhang, R. Xu, and W. Zhong. Unsupervised monocular depth estimation via recursive stereo distillation. *IEEE Trans. on Image Processing*, 30:4492–4504, 2021.

[40] Z. Yu, C. Shu, J. Deng, K. Lu, Z. Liu, J. Yu, D. Yang, H. Li, and Y. Chen. Flashocc: Fast and memory-efficient occupancy prediction via channel-to-height plugin. *arXiv preprint*, arXiv:2311.12058, 2023.

[41] C. Zhang, J. Yan, Y. Wei, J. Li, L. Liu, Y. Tang, Y. Duan, and J. Lu. Occnerf: Self-supervised multi-camera occupancy prediction with neural radiance fields. *arXiv preprint*, arXiv:2312.09243, 2023.

[42] Y. Zhang, Z. Zhu, and D. Du. Occformer: Dual-path transformer for vision-based 3d semantic occupancy prediction. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[43] Z. Zhang, C. Xu, J. Yang, J. Gao, and Z. Cui. Progressive hard-mining network for monocular depth estimation. *IEEE Trans. on Image Processing*, 27(8):3691–3702, 2018.

[44] H. Zou, X. Yang, T. Huang, C. Zhang, Y. Liu, w. li, F. Wen, and H. Zhang. Up-To-Down Network Fusing Multi-Scale Context for 3D Semantic Scene Completion. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2021.