2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

OCC-VO: Dense Mapping via 3D Occupancy-Based Visual Odometry for Autonomous Driving

Heng Li, Yifan Duan, Xinran Zhang, Haiyi Liu, Jianmin Ji and Yanyong Zhang*

***Abstract*— Visual Odometry (VO) plays a pivotal role in autonomous systems, with a principal challenge being the lack of depth information in camera images. This paper introduces OCC-VO, a novel framework that capitalizes on recent advances in deep learning to transform 2D camera images into 3D semantic occupancy, thereby circumventing the traditional need for concurrent estimation of ego poses and landmark locations. Within this framework, we utilize the TPV-Former to convert surround view cameras' images into 3D semantic occupancy. Addressing the challenges presented by this transformation, we have specifically tailored a pose estimation and mapping algorithm that incorporates Semantic Label Filter, Dynamic Object Filter, and finally, utilizes Voxel PFilter for maintaining a consistent global semantic map. Evaluations on the Occ3D-nuScenes not only showcase a 20.6% improvement in Success Ratio and a 29.6% enhancement in trajectory accuracy against ORB-SLAM3, but also emphasize our ability to construct a comprehensive map. Our implementation is open-sourced and available at: https://github.com/USTCLH/OCC-VO.**

I. INTRODUCTION

Visual odometry (VO) and Visual Simultaneous Localization And Mapping (SLAM) represent a fundamental technology in robotics and autonomous systems [1]. Given that camera images do not have depth information, the fundamental task of VO and Visual SLAM is thus to concurrently estimate the ego poses and the landmark locations, with this process commonly referred to as Bundle Adjustment (BA) [2]. BA is a complex optimization problem that can easily fail to achieve satisfactory results when faced with challenging circumstances, e.g., poor quality data, degenerate motions, and inadequate initial values for optimization [3]. Additionally, to meet the real-time computation requirement, the scale of the BA problem is typically controlled, with BA-based visual SLAM algorithms such as [4] that yield sparse maps without geometric details or semantic information.

Meanwhile, in the field of deep learning, significant progress has recently been made in the task of 3D semantic occupancy prediction [5]. This task allows for the transformation of 2D image frames into 3D semantic occupancy, thus rectifying the shortcoming of lacking depth information in images. By incorporating this task, we are able to simplify the core BA problem in traditional Visual SLAM, eliminating the need for simultaneous estimation of landmark locations. In other words, by treating the 3D semantic occupancy as a point cloud, the BA problem is thus transformed into a point cloud registration problem.

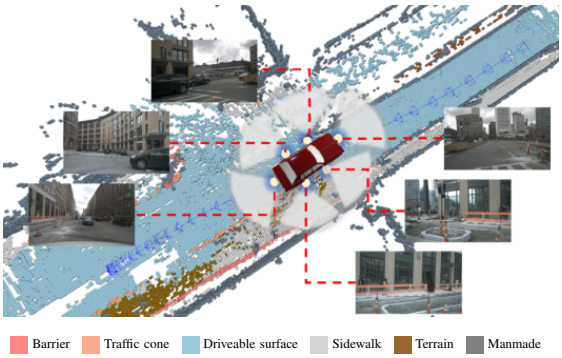


Fig. 1. Our approach transforms surround view cameras' image sequence into trajectories and global semantic maps. Such transformation can enhance scene understanding for downstream tasks in challenging environments.

However, the challenges arise because the 3D semantic occupancy differs from the original capture of the scene structure, e.g., Lidar scans. As a consequence, using such data to perform point cloud registration brings several issues. Firstly, the coarse resolution of the 3D semantic occupancy introduces uncertainty in the estimation of landmark positions, subsequently affecting the accuracy of registration. Secondly, due to the imperfect neural network models, the landmarks may be inaccurately constructed or even partially missed. Lastly, it is important to differentiate between stationary environments and dynamic objects, especially in applications like autonomous driving, as their matches can result in less accurate pose estimation.

In this work, we design and develop OCC-VO, a framework that takes surround view cameras' images as input and outputs a dense semantic map, which facilitates enhanced scene understanding for downstream tasks such as perception and navigation. Within this framework, we employ the open-source 3D semantic occupancy prediction module known as TPV-Former [6], to convert surround view camera images into 3D semantic occupancy.

To cope with the issues in registration mentioned above, we devise a pose estimation and mapping algorithm tailored for 3D semantic occupancy. Specifically, we start with the well-known GICP algorithm [7] commonly used in Lidar-based SLAM as our baseline. This algorithm relies on feature matching and iterative optimization for the alignment of point clouds. Given the unique characteristics of the 3D semantic occupancy transformed from images, we introduce semantic constraints into our registration process, similar to

The work is partially supported by the National Natural Science Foundation of China (No.62332016), Anhui Province Development and Reform Commission 2021 New Energy and Intelligent Connected Vehicle Innovation Project.

* The corresponding author.
School of Computer Science and Technology, University of Science and Technology of China, Hefei, 230026, China {li_heng, dyf0202, zxrr, lhyakn}@mail.ustc.edu.cn, {jianmin, yanyongz}@ustc.edu.cn.

ColorICP [8]. These semantic constraints prove to be highly effective in scenarios where geometric structures may be similar but with different semantics, such as road surfaces in autonomous driving contexts. Additionally, we implement a dynamic object filter to improve both map accuracy and pose estimation precision. Finally, during the mapping phase, we leverage the idea in PFilter [9] to eliminate unreliable points, building a more robust global semantic map. The end result is a fine-tuned pose estimation and a highly accurate map as depicted in Fig. 1.

The evaluation of our approach is conducted using the Occ3D-nuScenes dataset [10], which is an extension of the nuScenes dataset [11] augmented with voxel labels. We specifically focus on the training and validation sets captured by 6 surround view cameras in 2Hz. The diverse scenarios in the dataset, spanning different countries, lighting conditions, weather conditions, and environments, enable a thorough assessment of OCC-VO. Our method demonstrates superior performance in terms of accuracy and robustness in autonomous driving scenarios, compared to traditional Visual SLAM algorithms. In particular, when tested against ORB-SLAM3 [12], our method demonstrates a Success Ratio that is improved by 20.6% and shows a considerable gain in trajectory accuracy, reducing the absolute pose error by 29.6%. Additionally, we assessed the completeness and precision of the map, proving its potential support for downstream tasks. However, we would like to point out that while our OCC-VO offers these advantages, it does require the presence of surround view cameras and incur more computation costs and longer latency. Thus in certain scenarios, it may not be a better alternative compared to other light-weight solutions.

The main contributions of this paper are as follows:

- To the best of our knowledge, we design and develop OCC-VO which is the first to integrate 3D semantic occupancy with VO. This combination rectifies the shortcoming of lacking depth information in images, enabling us to create dense and comprehensive maps, which facilitate enhanced scene understanding for downstream tasks in challenging environments.
- We have specifically designed a pose estimation and mapping module for the proposed framework, which addresses the inherent limitations of 3D semantic occupancy such as inference errors and uncertainty due to the coarse resolution. Under this framework, this allows us to achieve accurate and robust pose estimation as well as dense mapping.
- Through trajectory evaluation, ablation study and map evaluation on Occ3d-nuScenes, our method demonstrates its superiority over traditional Visual SLAM algorithms, proving its robustness and accuracy in complex environments, even with low-frequency input.

II. Related Work

A. 3D Semantic Occupancy Prediction

Recently, we have witnessed the rapid development of 3D Object Detection based on Bird-Eye-View representations in autonomous driving [13]. However, relying solely on 3D object detection is insufficient for reconstructing realistic scenes. Unlike 3D Object Detection, 3D semantic occupancy prediction aims to reconstruct realistic scenes with rich semantic information. Incorporating semantic details offers more promising information for downstream tasks such as planning and SLAM.

Due to the similarities between the 3D semantic occupancy prediction task and the 3D Semantic Scene Completion (SSC) task, we will also incorporate the relevant SSC-related work into our research. Most of the related work relies on utilizing rich geometric information, such as Lidar point clouds [14], [15] and RGB-D images [16]–[19], which provides valuable depth and spatial information. In contrast, some recent works aim to reduce the reliance on geometric and depth information. MonoScene [20] utilizes only a single RGB image to predict the semantic scene in indoor and outdoor environments. OCCDepth [21] incorporates implicit depth cues from stereo images to aid in reconstructing 3D geometric structures. Furthermore, TPV-Former [6] introduces a novel approach that leverages surround view cameras' images to construct tri-perspective representations, enabling the prediction of 3D semantic scenes.

B. Visual SLAM

Visual SLAM is a branch of SLAM that primarily relies on visual information, such as images or videos, to estimate the ego pose and construct a map of the environment. It has gained significant attention due to the widespread availability of cameras in modern robotic platforms.

Traditional visual SLAM typically employ feature-based techniques. An example is ORB-SLAM [4], which detects and tracks features in images to estimate camera motion and reconstruct a sparse 3D map. These methods often rely on the extraction and matching of distinctive visual features across multiple frames. On the other hand, LSD-SLAM [22] is a direct method that estimates camera motion and reconstructs a semi-dense 3D map by directly minimizing the photometric error, exploiting all available pixel information.

In recent years, learning-based Visual SLAM has become extremely popular, such as CNN-SLAM [23] DROID-SLAM [24]. Besides, the impact of implicit 3D representation on visual SLAM has been significant, as demonstrated by the iMAP [25] and the NICE-SLAM [26], which use RGB-D camera to SLAM.

Recent years have also seen that Visual SLAM can benefit from occupancy prediction. By incorporating the 2D predicted semantic occupancy into the mapping and localization process, BEV-SLAM [27] enhances the robustness and accuracy of the SLAM system. Our work, OCC-VO, takes this a step further by incorporating the predicted 3D semantic occupancy into the mapping and localization process.

III. Preliminaries

Before elaborating on the core components of our OCC-VO, it is crucial to outline some foundational algorithms. This section offers a brief overview of TPV-Former (Sec. III-A) and GICP (Sec. III-B), both of which have a significant impact on the proposed approach.

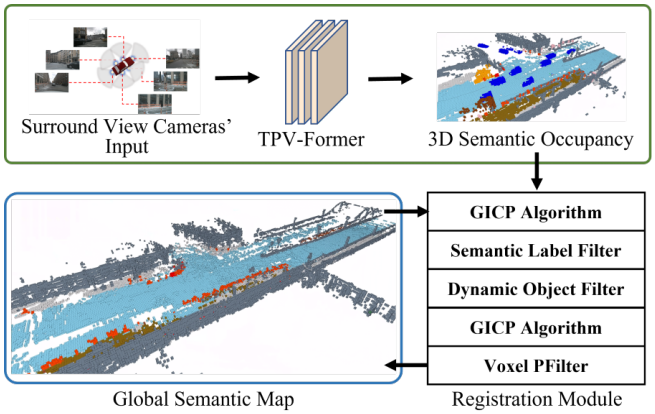


Fig. 2. Pipeline of our proposed OCC-VO.

A. 3D Semantic Occupancy Prediction

We use TPV-Former [6] to obtain the 3D semantic occupancy. TPV-Former introduces a novel approach for characterizing 3D scenes utilizing a tri-perspective view (TPV) representation. To assemble TPV features, the process begins by feeding surround view cameras' images into an image backbone network for feature extraction. Following this, TPV-Former utilizes a cross-attention mechanism to assimilate image features into TPV features and a hybrid-attention mechanism to foster interactivity among three planes by sampling the corresponding positions in the three planes. Finally, dense voxel features are obtained by broadcasting each plane along the corresponding orthogonal direction and summing these features, and a lightweight decoder takes the voxel features to predict whether each voxel is occupied and the semantic label of the occupied voxel.

B. GICP Algorithm

The GICP algorithm [7] is an enhanced version of the ICP algorithm [28]. This enhancement incorporates a Gaussian probability model into the ICP's cost function. The covariance matrix in this model performs a role analogous to weighting, which aids in mitigating the influence of outliers during the computation process. GICP retains the other segments of the algorithm unchanged, which assists in decreasing complexity and maintaining computational speed.

In typical implementations, we obtain the point clouds matching results by utilizing a KD-tree to find the nearest neighbor points, denoted as $A = \{a_i\}_{i=1,2,...,N}$ and $B = \{b_i\}_{i=1,2,...,N}$, where a_i and b_i are corresponding points. At this point, we continue to assume that each point in both sets follows a Gaussian distribution, $a_i \sim \mathcal{N}(\hat{a_i}, C_i^A)$, $b_i \sim \mathcal{N}(\hat{b_i}, C_i^B)$.

Assuming that $\mathbf{T}$ is the transformation, we define the residual $d_i^{(\mathbf{T})} = b_i - \mathbf{T}a_i$ and define $F_i(\mathbf{T}) = {d_i^{(\mathbf{T})}}^T (C_i^B + \mathbf{T}C_i^A\mathbf{T}^T)^{-1} d_i^{(\mathbf{T})}$. As the GICP algorithm proposes, we perform the following optimization procedure:

$$\mathbf{T}^\star = \underset{\mathbf{T}}{\operatorname{argmin}} \sum_i F_i(\mathbf{T}), \tag{1}$$

where $\mathbf{T}^\star$ represents the optimal transformation.

IV. Methodology

A. Overview

As shown in Fig. 2, the 6 images captured by the surround view cameras at the same time are converted into 3D semantic occupancy by TPV-Former mentioned in Sec. III-A at the beginning. The resulting 3D semantic occupancy is treated as point cloud, and the pose of each frame is estimated through registration between it and the global semantic map. In specific, we employed the GICP algorithm (Sec. III-B) twice. At the beginning, the coarse correspondence between points is established. Following this, we engage the Semantic Label Filter (Sec. IV-B) and Dynamic Object Filter (Sec. IV-C) to discard erroneous matches, thus refining the accuracy of the second GICP application. Once a precise pose is determined, we leverage the Voxel PFilter (Sec. IV-D) to merge the frame of data into the global semantic map, rectifying errors in TPV-Former's inference of the map for global consistency.

B. Semantic Label Filter

In point cloud registration, a common cause of failure is the instability encountered on smooth planes, a common structure, e.g., the road, in autonomous driving scenarios. Specifically, the alignment of point clouds tends to falter due to inadequate geometric constraints, causing a "slip" in the alignment [8]. This problem is hard to solve without introducing other sensors in point cloud-based SLAM systems [29]. Fortunately, the point cloud from TPV-Former contains semantic labels besides geometric information.

Hence, we propose the use of a Semantic Label Filter by introducing semantic constraints to tackle this instability mentioned above. The Semantic Label Filter functions by eliminating pairs of points with mismatching semantic labels during the optimization process. This method effectively prevents erroneous point matches between various objects or surfaces from impacting the optimization solution.

To simplify the following formula, we continue our discussion by adopting Iverson bracket [30] defined as follows:

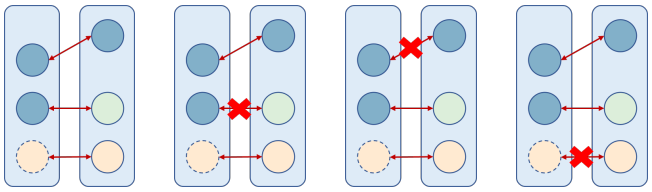
$$[P] = \begin{cases} 1, & \text{if } P \text{ is True,} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

In this way, we employ $P_S(a_i, b_i)$ as the mathematical representation of the Semantic Label Filter. Within this function, a_i is derived from the input 3D semantic occupancy, while b_i is the corresponding point for a_i in the global semantic map found through the first GICP. $P_S(a_i, b_i)$ is defined as $P_S(a_i, b_i) = (SL_{a_i}$ is equals to $SL_{b_i})$, where SL represents the semantic label of a_i and b_i. Then we incorporate it into the cost function of GICP algorithm (Eq. 1), specifically as follows:

$$\mathbf{T}^\star = \underset{\mathbf{T}}{\operatorname{argmin}} \sum_i F_i(\mathbf{T})[P_S(a_i, b_i)]. \tag{3}$$

C. Dynamic Object Filter

In autonomous driving scenarios, dynamic vehicles and pedestrians introduce significant disturbances to registration accuracy. One straightforward approach is using semantic labels to eliminate the occupancy of potential moving objects,



(a) Nearest points match (b) Semantic Label Filter (c) Dynamic Object Filter (d) Voxel PFilter

Fig. 3. Three filters we propose. The rectangular boxes represent 3D semantic occupancy and the global semantic map separately, with each circle representing a specific point. The color of the circle represents semantic labels, and the dashed circle border indicates that the point is transient with low p-Index defined in Sec. IV-D. (a) shows three poor point-pair matches; (b) removes the match with different labels; (c) eliminates the one from dynamic objects; and (d) filters out the one containing a low p-Index value.

such as various types of vehicles and humans. We refer to this method as Label-based Object Filter.

Yet, indiscriminately eliminating these potential dynamic objects can't always favorably contribute to pose estimation. Specifically, stationary objects often provide effective constraints for registration and their removal may lead to scene degradation conversely. This issue becomes more severe when these objects are large vehicles such as engineering vehicles or buses, as they occupy a significant portion of the field of view of cameras.

With the problem clearly outlined, we design the Dynamic Object Filter to enhance the performance of OCC-VO when dealing with highly dynamic scenarios. Specifically, objects with potential for motion are separated from both the 3D semantic occupancy and the global semantic map using semantic labels. Each object is then subjected to point cloud clustering, with these clusters subsequently treated as unified objects. Utilizing the transformation results from the first GICP shown in Fig. 2, we compare the position of each object. This allows us to identify relative displacements and decide if an object is dynamic and should be removed. Employing the static part of each input, a refined registration is conducted, leading to enhanced precision in pose estimation.

Using the aforementioned algorithm, we have acquired the point set DA corresponding to dynamic objects in the 3D semantic occupancy, as well as the point set DB associated with dynamic objects in the global semantic map. Thus the Dynamic Object Filter can be defined as $P_D(a_i, b_i) = (a_i \notin DA \text{ and } b_i \notin DB)$. Similar to Sec. IV-B, the cost function, i.e., Eq. 3, can be extended as:

$$\mathbf{T}^\star = \underset{\mathbf{T}}{\operatorname{argmin}} \sum_i F_i(\mathbf{T})[P_S(a_i, b_i)][P_D(a_i, b_i)]. \tag{4}$$

D. Voxel PFilter

Considering that for the same object or surface, the 3D semantic occupancy predicted between adjacent frames might have inconsistent grid representations, which is quite common in this field of work. As a result, we suggest the incorporation of Voxel PFilter in the registration process to merge the more reliable points in the 3D semantic occupancy into the global semantic map. This modification aims to maintain the global consistency of the map and correct noise induced by network inference.

Voxel PFilter is an improved version of PFilter [9], a feature selection algorithm for Lidar-based SLAM. To adapt to the problems of this work, we redefine the metric, i.e., p-Index, proposed in PFilter. In our implementation, the p-Index is a metric designed to determine the persistence level of a voxel. For clarify, the meaning of the voxel being occupied in this section is that there is a point in the voxel, which is the data format we actually deal with in OCC-VO. During the mapping process, two attributes are recorded for each occupied voxel: the time when the voxel is first predicted as occupied, denoted as t_0, and the number of times it has been marked as occupied, denoted as f. The p-Index of voxel v at time t is simply defined as $\frac{f}{t-t_0}$.

The voxel with a high p-Index value is consistently predicted to be occupied, indicating that it has a high probability of actually being occupied. This is what we call the persistent voxel. In contrast, the transient voxel, often caused by the network's erroneous prediction, exhibits lower values.

Similar to Sec. IV-C, we define Voxel PFilter as $P_V(b_i) = (\text{p-Index}(v) > 0.5 | b_i \text{ in } v)$, where 0.5 is derived from experience. Now substituting Voxel PFilter into Eq. 4 yields:

$$\mathbf{T}^\star = \underset{\mathbf{T}}{\operatorname{argmin}} \sum_i F_i(\mathbf{T})[P_S(a_i, b_i)][P_D(a_i, b_i)][P_V(b_i)]. \tag{5}$$

Furthermore, a downsampling procedure is applied to the persistent points. This involves conducting a weighted average of three-dimensional coordinates within each grid based on the p-Index. Only the points newly generated through this downsampling are retained, with a fresh p-Index calculation performed for them.

V. Experiments

We first introduce the setup of our experiment in Sec. V-A. Then we evaluate OCC-VO using the Occ3D-nuScenes datasets, contrasting its performance with the traditional method, i.e., ORB-SLAM3 [12] and learning-based method, i.e., DROID-SLAM [24] in Sec. V-B. Ablation studies are performed to underscore the potency of our methodology in Sec. V-C. Further, we illustrate that OCC-VO can adeptly construct a comprehensive and accurate 3D semantic map in autonomous driving scenarios in Sec. V-D. Lastly, we conduct a execution time analysis, demonstrating the real-time capabilities of our algorithm in Sec. V-E.

A. Experimental Setup

In our experiments, we evaluate OCC-VO using the validation sets of the Occ3D-nuScenes dataset [10]. The Occ3D-nuScenes dataset serves as a comprehensive benchmark for 3D semantic occupancy prediction. Building upon the nuScenes dataset, it incorporates voxel labels and is designed specifically for autonomous driving applications. The dataset offers a varied and realistic collection of sensor data from multiple urban settings, positioning it as an ideal benchmark to gauge the performance and robustness of our algorithm across diverse conditions. Distinctive in its breadth and detail, the dataset covers over 1000 scenes, each lasting about 20 seconds, which spread across different countries, lighting settings, weather variations, and environments.

TABLE I

SUCCESS RATIO AND RMSE OF APE [M] OF VARIOUS METHODS.

Method	Frame Rate	Input	Success Ratio	RMSE[m]
ORB-SLAM3 [12]	2Hz	Front camera	0.000	-
ORB-SLAM3 [12]	12Hz	Front camera	0.787	0.199
DROID-SLAM [24]	2Hz	Front camera	0.753	0.282
OCC-VO (ours)	2Hz	Surround cameras	**0.993**	**0.140**
OCC-VO (ours)	2Hz	3D semantic occupancy ground truth*	1.000	0.122

The best results are highlighted in the bold face.

TABLE II

SUCCESS RATIO AND RMSE OF APE [M] OF OCC-VO FOR ABLATION STUDY.

Semantic Label Filter	Dynamic Object Filter	Label-based Object Filter	Voxel PFilter	Success Ratio	RMSE[m]
				0.973	0.220
✓				0.973	0.206
✓		✓		0.973	0.218
✓	✓			0.980	0.173
✓	✓		✓	**0.993**	**0.140**

The best results are highlighted in the bold face.

On the training sets, we train the TPV-Former on the 1600x900 image sequences captured at 2Hz by six surround view cameras. These sequences, spanning a total of 700 scenes, are annotated with voxel labels serving as the ground truth. Referring to TPV-Former's implementation, we set the occupancy prediction range as $[-40m, 40m]$ for the X and Y axes, and $[-1.0m, 5.4m]$ for the Z axis. The final output 3D semantic occupancy is presented in a 200x200x16 shape with a voxel size of 0.4m.

In the subsequent sections, we focus on experiments conducted on 150 validation sequences. All experiments employ 2Hz image sequences, as provided by Occ3D-nuScenes, with a Dynamic Object Filter displacement threshold set at 2 meters. In addition, ORB-SLAM3 operates at 12Hz because it can't work with 2Hz input.

B. Trajectory Evaluation

We employ the Root Mean Square Error (RMSE) of Absolute Pose Error (APE) [31] of the predicted trajectories as the primary evaluation metric. Furthermore, due to the complexity of autonomous driving scenarios, these algorithms might yield results that deviate significantly from the correct trajectory in some cases, such as numerous dynamic objects and poor light conditions. Considering that the average APE for most samples is around 0.2 meters, any instance with an APE greater than 5 meters can be deemed a failure. Thus, we have introduced a success ratio metric, which represents the proportion of results with an APE of less than 5 meters. This is employed to filter out the inferior results and subsequently compute the aforementioned RMSE.

Table I present our experimental results. It's evident that OCC-VO achieves a higher success ratio and a reduced APE. Specifically, OCC-VO boasts a success ratio of 99.3% and an RMSE of APE at 0.140 meters. This represents a 20.6% increase in success ratio and a 29.6% improvement in trajectory accuracy compared to ORB-SLAM3, even when fed with a lower frequency input. Based on our analysis of experimental samples, the boost in success ratio is primarily observed in scenarios like rapid turns and poor lighting conditions, while the decrease in APE is attributed to robustness in complex environment such as numerous dynamic objects and extensive occlusions. In addition, we use the 3D semantic occupancy ground truth provided by Occ3D-nuScenes as the input of registration module, getting a success ratio of 100% and an RMSE of APE at 0.122 meters, which shows the potential of OCC-VO, especially when applied with more accurate predict networks.

C. Ablation Study

We next present the results of the ablation study. As shown in Table II, the experiment with different filter combinations exhibits varying levels of success ratio and accuracy. When no filters were employed, we observed the baseline performance with a success ratio of 97.3% and an RMSE of APE at 0.220 meters. As we incrementally introduced the three filters, there was a consistent uptrend in the algorithm's performance: the success ratio advanced from 97.3% to 98.0% and finally to 99.3%, while the RMSE of APE progressed from 0.220 meters to 0.206 meters, then to 0.173 meters and settled at 0.140 meters with the full filter set. Notably, when comparing the Dynamic Object Filter with the Label-based Object Filter, the latter did not enhance the success ratio and even resulted in an RMSE of APE decrease to 0.218 meters, underscoring the superior efficacy of the Dynamic Object Filter's design.

D. Map Evaluation

To highlight the powerful capabilities of OCC-VO in dense outdoor mapping, we perform map evaluations. In this experiment, we generate map ground truth using the 3D semantic occupancy ground truth and the pose ground truth. The algorithm's output is assessed using accuracy, accuracy ratio and completion ratio. Accuracy quantifies the RMSE of the distance between sampled points from the reconstructed map and the nearest map ground truth point. Precision and completion ratio separately measure the proportion of points in the output reconstructed properly and the proportion of points in the map ground truth that are reconstructed. In our calculation, a map point is deemed "reconstructed" if its distance to the nearest reconstructed point is less than 0.4 meters, since the voxel size is 0.4 meters.

In our experiment, we find that even outdoor-capable visual SLAM algorithms, like DROID-SLAM [24], fail to produce accurate and complete maps in such autonomous

TABLE III

PARTIAL VISUALIZATION RESULTS ON OCC3D-NUSCENES VALIDATION SEQUENCES

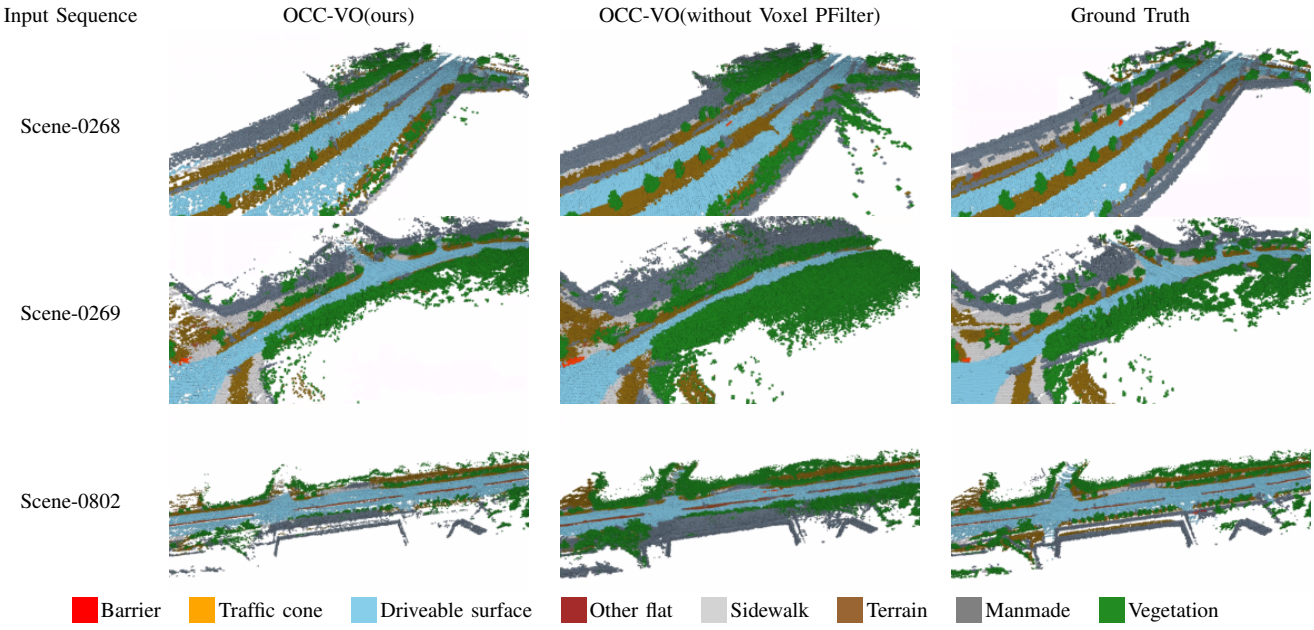


TABLE IV

THE ACCURACY [M], PRECISION [<0.4M] AND COMPLETION RATIO [<0.4M] OF OCC-VO

Method	Acc.[m]	Precision	Comp. Ratio
OCC-VO(ours)	**0.111**	**0.724**	0.725
OCC-VO(w/o Voxel PFilter)	0.125	0.572	**0.875**

driving scenes. Thus we only conducted experiments under the conditions of OCC-VO with and without Voxel PFilter. As shown in Table IV, when using Voxel PFilter, the algorithm exhibited higher accuracy and precision, but due to the filtering of transient points, completeness slightly decreased, with respective values of 0.111 meters, 72.4% and 72.5%. Without Voxel PFilter, the algorithm get an 87.5% completeness, but decrease in accuracy and precision, with values of 0.125 meters and 57.2% respectively. The results show that the OCC-VO maintains its ability to perform accurate and dense mapping in complex autonomous driving scenarios, thereby demonstrating the capability to assist with downstream tasks such as navigation. As illustrated in Table III, we present the qualitative visual results of our algorithm. These visual representations show that many static semantic details, such as vegetation, driveable roads, and barrier, are accurately reconstructed by our algorithm.

E. Execution Time Analysis

Owing to the high video memory demands of the 3D semantic occupancy prediction network, we conducted training and testing on a server equipped with 4 Intel Xeon Gold 6230R CPUs @ 2.10GHz, 8 NVIDIA A100 GPUs and 754 GB RAM. For the registration module, testing was executed directly on a Intel i9-13900K CPU @ 3.00GHz and 128 GB RAM personal computer. Given that the 3D semantic occupancy prediction network primarily relies on GPU performance, while the registration module leans more towards CPU capabilities, it is reasonable to evaluate and analyze their execution time on separate hardware platforms.

TABLE V

AVERAGE EXECUTION TIME [MS/FRAME] OF OCC-VO

Module		Time[ms/frame]	
3D semantic occupancy prediction (A100)		267	
Registration(i9-13900K)	GICP Algorithm	69	371
	Dynamic Object Filter	4	
	GICP Algorithm	70	
	Voxel PFilter	166	
	Other	62	

A detailed execution time analysis is shown in Table V. The 3D semantic occupancy prediction network consumes 5401MB of GPU memory and requires 267 ms per inference. The registration module takes another 371 ms for each computation, pointing to possible limitations of the current Python-based implementation. Notably, both modules can operate in a pipelined manner. Thus, even with a 2Hz input, they ensure seamless processing without causing a backlog in the queue. Within the scope of our study, our attention has been on the integration and assessment of these modules in the OCC-VO system, highlighting that both the trajectory accuracy and map completeness are commendable. As future endeavors, for faster execution speed, we can refining the network's efficiency and migrating the registration to C++.

VI. CONCLUSION

In our work, we introduce OCC-VO, a novel VO framework leveraging 3D semantic occupancy, making it distinct from traditional Visual SLAM. By using the designed filters, this innovation not only facilitates the generation of denser maps but also produces more accurate trajectories in autonomous driving scenarios. Our experiments on the Occ3D-nuScenes dataset demonstrate OCC-VO's superior performance in terms of accuracy and robustness in autonomous driving scenarios. In future work, we intend to integrate modules such as loop closure detection into OCC-VO, advancing it towards a SLAM system.

REFERENCES

[1] W. Chen, G. Shang, A. Ji, C. Zhou, X. Wang, C. Xu, Z. Li, and K. Hu, "An overview on visual slam: From tradition to semantic," *Remote Sensing*, vol. 14, no. 13, p. 3010, 2022.

[2] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings*. Springer, 2000, pp. 298–372.

[3] A. Macario Barros, M. Michel, Y. Moline, G. Corre, and F. Carrel, "A comprehensive survey of visual slam algorithms," *Robotics*, vol. 11, no. 1, p. 24, 2022.

[4] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics (T-RO)*, vol. 31, no. 5, pp. 1147–1163, 2015.

[5] L. Roldao, R. De Charette, and A. Verroust-Blondet, "3d semantic scene completion: A survey," *International Journal of Computer Vision (IJCV)*, vol. 130, no. 8, pp. 1978–2005, 2022.

[6] Y. Huang, W. Zheng, Y. Zhang, J. Zhou, and J. Lu, "Tri-perspective view for vision-based 3d semantic occupancy prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 9223–9232.

[7] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp." in *Robotics: science and systems*, vol. 2, no. 4. Seattle, WA, 2009, p. 435.

[8] J. Park, Q.-Y. Zhou, and V. Koltun, "Colored point cloud registration revisited," in *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2017, pp. 143–152.

[9] Y. Duan, J. Peng, Y. Zhang, J. Ji, and Y. Zhang, "Pfilter: Building persistent maps through feature filtering for fast and accurate lidar-based slam," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 11 087–11 093.

[10] 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," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, 2024.

[11] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2020, pp. 11 621–11 631.

[12] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics (T-RO)*, vol. 37, no. 6, pp. 1874–1890, 2021.

[13] R. Qian, X. Lai, and X. Li, "3d object detection for autonomous driving: A survey," *Pattern Recognition*, vol. 130, p. 108796, 2022.

[14] C. B. Rist, D. Emmerichs, M. Enzweiler, and D. M. Gavrila, "Semantic scene completion using local deep implicit functions on lidar data," *IEEE transactions on pattern analysis and machine intelligence (T-PAMI)*, vol. 44, no. 10, pp. 7205–7218, 2021.

[15] M. Zhong and G. Zeng, "Semantic point completion network for 3d semantic scene completion," in *European Conference on Artificial Intelligence (ECAI)*. IOS Press, 2020, pp. 2824–2831.

[16] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2017, pp. 1746–1754.

[17] M. Garbade, Y.-T. Chen, J. Sawatzky, and J. Gall, "Two stream 3d semantic scene completion," in *Proceedings of the IEEE conference on computer vision and pattern recognition Workshops (CVPRW)*, 2019, pp. 0–0.

[18] S. Li, C. Zou, Y. Li, X. Zhao, and Y. Gao, "Attention-based multi-modal fusion network for semantic scene completion," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, vol. 34, no. 07, 2020, pp. 11 402–11 409.

[19] S.-C. Wu, K. Tateno, N. Navab, and F. Tombari, "Scfusion: Real-time incremental scene reconstruction with semantic completion," in *2020 International Conference on 3D Vision (3DV)*. IEEE, 2020, pp. 801–810.

[20] A.-Q. Cao and R. de Charette, "Monoscene: Monocular 3d semantic scene completion," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2022, pp. 3991–4001.

[21] R. Miao, W. Liu, M. Chen, Z. Gong, W. Xu, C. Hu, and S. Zhou, "Occdepth: A depth-aware method for 3d semantic scene completion," *arXiv preprint arXiv:2302.13540*, 2023.

[22] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 834–849.

[23] K. Tateno, F. Tombari, I. Laina, and N. Navab, "Cnn-slam: Real-time dense monocular slam with learned depth prediction," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2017, pp. 6243–6252.

[24] Z. Teed and J. Deng, "Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras," *Advances in neural information processing systems (NeurIPS)*, 2021.

[25] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, "Nice-slam: Neural implicit scalable encoding for slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 12 786–12 796.

[26] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "imap: Implicit mapping and positioning in real-time," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, vol. 34, 2021, pp. 16 558–16 569.

[27] J. Ross, O. Mendez, A. Saha, M. Johnson, and R. Bowden, "Bev-slam: Building a globally-consistent world map using monocular vision," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 3830–3836.

[28] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. Spie, 1992, pp. 586–606.

[29] J. Zhang, M. Kaess, and S. Singh, "On degeneracy of optimization-based state estimation problems," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 809–816.

[30] K. E. Iverson, "A programming language," in *Proceedings of the May 1-3, 1962, spring joint computer conference*, 1962, pp. 345–351.

[31] M. Grupp, "evo: Python package for the evaluation of odometry and slam." https://github.com/MichaelGrupp/evo, 2017.