

LiDAR based Traversable Regions Identification Method for Off-road UGV Driving

Yunxiao Shan^{1,3,4}, Yao Fu¹, Xiangchun Chen⁵, Hongquan Lin¹, Ziquan zhang¹, Jun Lin⁶, and Kai Huang²

Abstract—Traversable regions identification technology plays a crucial role in ensuring safe driving for unmanned ground vehicles in off-road environments. However, the unstructured terrain makes it challenging to identify traversable regions. To enhance the safety of off-road driving, a LiDAR-based traversable regions identification method is proposed in this paper. Firstly, a deep learning-based neural network is used to segment the traversable regions, obstacles, and vegetation. Next, an improved Gaussian Process(GP)-based modeling method is designed to model the traversable regions with a leading speed, and the obstacle point clouds are refined with a composite filter. Finally, field experiments have demonstrated that our proposed scheme outperforms existing state-of-the-art (SOTA) traditional and deep-learning-based methods in accurately identifying both road regions and obstacles, with precision improvements of up to 14% and recall improvements of up to 9%.

I. INTRODUCTION

In recent times, there has been a growing interest in the advancement of unmanned ground vehicles (UGVs) technology for off-road applications [1]. Unlike urban driving, off-road driving requires the ability to navigate through unstructured environments, and existing methods that were specifically designed for urban driving may not be effective in such environments. One of the main challenges of off-road driving is the lack of definition of traversability, as there are no salient curbs or buildings to guide the vehicle. Therefore, it is crucial to explore appropriate sensors and develop SOTA identification methods to identify traversable regions from the complicated background in off-road environments.

Both cameras and LiDARs are commonly used sensors for identifying traversable regions. Cameras can provide rich semantic information and identify traversable regions in most normal conditions, but their performance can be limited by diverse illumination and weather conditions. Additionally, cameras lack depth information, which can limit their performance in identifying traversable regions. LiDARs, on the other hand, are more robust to the environment and can provide both semantic and depth information by directly collecting 3D

¹The authors are with the School of Artificial Intelligence, Sun Yat-sen University, Zhuhai, Guangdong, P.R.China. & the Southern Marine Science and Engineering Guangdong Laboratory, Zhuhai, Guangdong, P.R.China.

²The authors are with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, Guangdong, P.R.China.

³the Shenzhen Institute, Sun Yat-sen University, Shenzhen, Guangdong, P.R.China.

⁴ Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou 510006, P.R.China.

⁵ PLA Army Academy of Artillery and Air Defense, China.

⁶ Sun Yat-sen University, Guangzhou, Guangdong, P.R.China.

The corresponding author is Kai Huang, huangk36@mail.sysu.edu.cn.

point clouds [2][3]. In off-road environments, LiDARs show excellent environmental adaptability [4].

Many methods were developed for the identification of traversable regions. Zhou *et al.* [5] proposed a fuzzy support vector machine for detecting the unstructured road with front-view monocular cameras. With cameras, Guin *et al.* [6] proposed a strategy to detect the traversable surface of marginally rural roads by deploying a deep neural network. Although the traversable region detection tasks can be well performed, failures are inevitable when cameras are completely exposed to changing lighting environment, which is common in the off-road environment. In light of the poor environmental robustness of cameras, Ahtiainen *et al.* [4] deployed LiDARs to identify traversable areas in vegetated regions. Gao *et al.* [7] proposed a method to extract off-road traversable regions using LiDAR data and a specific deep learning framework is designed to deal with the ambiguous area.

Although LiDARs have shown better identification ability of traversable regions for off-road applications with SOTA methods [8], applying them to identify traversable regions in off-road environments is not straightforward. In our applications, a UGV needs to drive along an unstructured road with vegetation aside and make appropriate decisions based on possible obstacles in the road. This requires simultaneous identification of traversable regions (road regions) and obstacles. Despite the presence of a vague road in our application, accurately identifying traversable regions is non-trivial. The penetrable vegetated boundaries are adverse to the reflection of LiDARs, and few salient features can be used for segmenting the road from the backgrounds. Additionally, the complex terrain and ground surface conditions make it very difficult to accurately model traversable regions while maintaining leading modeling speed. Furthermore, Detecting possible obstacles on the road is crucial for identifying obstacle-free areas and then ensuring driving safety. By accurately determining these obstacle-free (traversable) regions, the later-on decision-making and planning processes can be simplified.

To address the challenges of identifying traversable regions in off-road environments, a segmentation-modeling-refinement scheme is proposed. Our approach utilizes deep features to segment road regions, vegetation, and obstacles. Afterwards, an improved Gaussian Process method is used to model the road surface and exclude outlier point clouds. For segmented obstacle point clouds and excluded outliers, we developed a composite filter to filter out missegmentation and identify new obstacle point clouds from outliers. Field experiments demonstrate the superiority of our method.

Concretely, our contributions are listed as follows:

- An approach that combines deep learning and traditional methods is proposed for identify traversable regions in off-road environment, which has shown to be more effective than approaches that rely solely on either one of these techniques.
- A deep learning method is customized for segmenting the traversable regions, obstacles, and vegetation, an improved GP-based modeling method that incorporates a discretizing-based acceleration strategy is developed, and a composition filter is created to refine obstacle point clouds. The combination of these methods has enabled us to address the challenge of identifying traversable regions in complex environments, where traditional methods and deep learning alone have shown some limitations.

In the subsequent sections, we present a comprehensive review of the related work in Section II. Section III provides a detailed exposition of the methodology employed, while Section IV offers a presentation of the obtained results. The final Section V encapsulates the summary of our work, along with a discussion of future research directions.

II. RELATED WORK

According to the used sensors, the traversable regions identification methods can be categorized into camera based methods, LiDARs based methods and multi-sensor fusion methods.

1) *Camera based methods:* Camera-based traversable regions identification has traditionally relied on artificial features. Passani *et al.* [9] used a conditional random fields and uniformly re-weighted belief propagation based method to achieve pixel-wise segmentation for urban roads. Vitor *et al.* [10] introduced a novel method that created a classifier based on a set of probability distributions to improve performance in scenes with poor marked roads and challenging lighting conditions. Han *et al.* [11] proposed a patch-wise manifold ranking method and validated it on various unstructured road scenarios. Zhang *et al.* [12] introduced a system using a 2D grid map based on statistical features from depth camera-generated point clouds for mapping and path planning in intricate 3D terrain scenarios. While these traditional methods have demonstrated good performance in specific applications, they are often limited by their reliance on specific scenarios and may struggle to adapt to complex and dynamic environments. Recent deep learning-based methods are showing more potential for complex road scenarios. Oliveira *et al.* [13] proposed a deep segmentation network to balance efficiency and segmentation quality. Han *et al.* [14] proposed a fully convolutional neural network-based method that fused several image cues to improve road detection performance. Qiang *et al.* [13] proposed a unified neural network, DLT-Net, to achieve multiple tasks, including drivable areas, lane lines, and traffic objects. However, poor lighting conditions and fuzzy lane lines can still cause failures. While camera-based methods have achieved better performance, most existing methods are applied in urban structured environments, and few methods [15] exist for off-road applications.

2) *LiDARs based methods:* Recently, LiDAR-based methods have become widely used for traversable regions identification, particularly in challenging conditions for cameras. For urban applications, Xue *et al.* [16] proposed a LiDAR-based traversable regions identification method that utilizes Bayesian generalized kernel inference and bilateral filtering to estimate the attributes of unobserved cells. Tao *et al.* [17] used disparity to extract ground surfaces and obstacles from 3D point clouds and designed three kinds of features to detect curbs, ditches, and grasses. In rural and wild environments, most existing methods use environment-robust LiDARs. Ai *et al.* [18] proposed a LiDAR-based novel road boundary detection framework for surface mines, addressing challenges such as high generalization error and difficulty in boundary generation under unstructured environment. Qian *et al.* [19] proposed a ray-fitting technique for LiDAR-based ground segmentation, identifying traversable regions. Utilizing point cloud details and RANSAC-based polynomial fitting, it prevents over-segmentation, ensuring precise delineation and faster, highly accurate results compared to prior methods. Lee *et al.* [20] proposed a LiDAR-based traversability analysis method for urban environments, employing a self-training algorithm to enhance model performance without requiring manual labeling by utilizing unlabeled data. Ahtiainen *et al.* [4] used a support-vector machine classifier to identify traversable regions for outdoor environments and outperformed other purely geometrical analysis approaches. With deep learning methods, Gao *et al.* [7] proposed a robust off-road traversable regions extraction method with 3D LiDARs that achieves better performance and only needs to be trained by weakly supervised or semi-supervised methods. Xu *et al.* [21] addressed this issue by building two maps from original point clouds, an elevation map and a range map, and defining traversable regions by detecting obstacles and road areas. Their method consists of many artificial features-based filters, and many parameters are required to be regulated for the best performance.

3) *Multi-sensor fusion methods:* Several researchers also addressed to fuse multiple sensors to identify traversable regions. Gu *et al.* [22] proposed a road detection framework that combines a 3D LiDAR and a monocular camera. In this approach, the 3D point clouds obtained from LiDAR is projected onto the camera's image frame, allowing for the utilization of both range and color information. On the other hand, Su *et al.* [23] presented an illumination-invariant non-parametric model for urban road detection using a monocular camera and a single-line LiDAR sensor. This method differs from the previous fusion approaches mentioned. Unlike the above two fusion methods, Sock *et al.* [24] employed both cameras and LiDARs and assumes the data captured by these two sensors are independent and build separate traversability maps. Zhou *et al.* [25] proposed a novel LiDAR-vision-based method for terrain traversability mapping, including vision-based traversable area segmentation, LiDAR-based traversable regions extraction, and Bayesian fusion. Ni *et al.* [26] introduced a novel multi-phase fusion model for real-time robust 3D semantic segmentation employing LiDAR and cameras. The proposed method demonstrates promising detection results in both urban and off-road environments, effectively

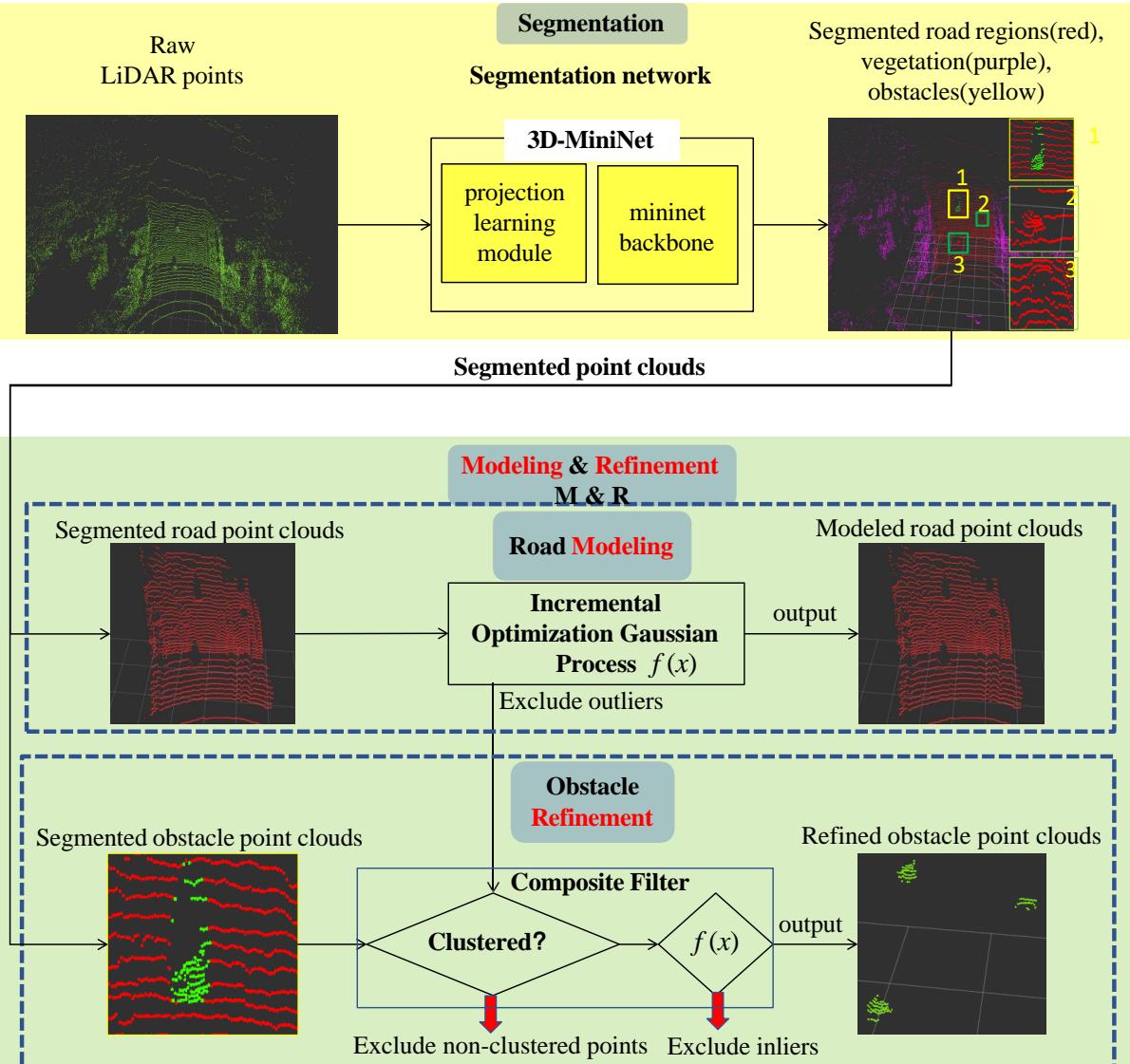


Fig. 1: The road point clouds are firstly segmented by a segmentation network and then the road regions are modeled and obstacles (yellow rectangle 1 denotes correctly segmented obstacles, green rectangle 2 and 3 denote obstacles that are not correctly segmented) will be refined.

assisting vehicles in safe navigation outdoors. Ruetz *et al.* [27] introduces a robust color fusion method that incorporates multi-return LiDAR measurements. This approach accurately estimates terrain traversability, particularly in challenging, densely vegetated environments. While fusion methods can take advantage of multiple sensors to complement each other, fusing two sets of heterogeneous data can be a challenging task. Additionally, we cannot ignore the potential increase in computational burden and the added complexity to the system.

Considering factors such as computational burden, system complexity, and adaptability to the environment, LiDARs are more suitable for identifying traversable regions in off-road environments. Although there are several state-of-the-art methods for identifying traversable regions, many of them do not account for potential obstacles. Additionally, some existing methods that can simultaneously identify traversable

regions and detect obstacles are not robust enough for various applications. Therefore, a simplified and robust method that can simultaneously model road regions and identify obstacles is still needed with the use of LiDARs.

III. APPROACH

Our proposed method consists of two parts, namely segmentation and modeling & refinement (M & R). Firstly, with the help of a neural network, the original point clouds are segmented to obtain road regions, vegetation, and obstacles. While deep features can accurately segment areas with prominent features, the possibility of mis-segmentation cannot be ruled out due to poor discriminative features of road regions from obstacles in road. To overcome this problem, an M & R process is employed to rectify the mis-segmentation. In detail, the segmented point clouds of the road regions are modeled

using an improved GP technique, and non-road point clouds are excluded as outliers. For the case of segmented point clouds of obstacles and excluded outliers, a composite filter is applied to correct segmentation errors and identify obstacle point clouds from outliers. The proposed scheme is illustrated in Fig. 1.

A. Segmentation

Although there are various state-of-the-art methods available, such as fully convolutional network [28] and convolutional recurrent network [29], they require heavy model parameters, making it difficult for real-time applications. 3D-MiniNet [30] is one of the most representative methods that provides accurate segmentation with a leading speed. The approach consists of a projection learning module and a FCN backbone. The original point clouds are transformed into depth maps through the depth-map projection, which is then input into the projection learning module to obtain a two-dimensional tensor representing the three-dimensional data. The tensor is then fed into FCN, which extracts low-level fine features using the depth map and incorporates both low-level and high-level features into the subsequent trunk. Finally, the three-dimensional point clouds with semantic labels are output. Further information about 3D-MiniNet can be found in [30].

Although the performance of 3D-MiniNet is validated in many classical scenarios, the issue of class-imbalance in our applications should also be addressed. In our off-road environment, over 57% point clouds belong to vegetation, while less than 1% point clouds belong to the obstacles in road regions. This class imbalance can significantly reduce the accuracy of segmentation. To account for this issue, we refer to SegNet [31] to smooth the resulting class weights.

B. Modeling & Refinement

To address the issue of mis-segmentation, an M & R strategy is deployed for the segmented road point clouds and obstacle point clouds, as shown in Fig. 1. Concretely, we firstly make use of an improved GP to model the road regions. This allows us to calssify the road point clouds as inliers, representing the road point clouds, and outliers, which indicated non-road point clouds. For the segmented obstacle point clouds and outliers, a composite filter is developed to filter out non-obstacle point clouds.

1) Road point clouds modeling: For the segmented road point clouds, an improved GP method is used for filtering out the non-road point clouds, as shown in Alg. 1, namely IOGP. Our IOGP mainly consists of two parts, acceleration module (line 1 to 3 Alg. 1) and incremental optimization module (line 4 to 26 Alg. 1).

Our first improvement is the acceleration strategy. As a non-parametric regression method, GP is widely used for modeling and prediction. A classical GP regression can be expressed as:

$$f(x) \sim GP(m(x), k(x, x')) \quad (1)$$

where $m(x)$ denotes mean functions and $k(x, x')$ is covariance functions.

Algorithm 1: Incremental Optimization GP (IOGP)

Input: Q_{seg_road} : segmented road point clouds
Output:
 $F(x) = (f_1(x), f_2(x), \dots, f_{sectorNum}(x))$: road model
 Q_{in} : road point clouds
 Q_{out} : non-road point clouds

```

1  $Q_{downSample} \leftarrow voxelFiltering(Q_{seg\_road});$ 
2  $sectors \leftarrow divide(Q_{downSample}, sectorNum);$ 
3 for  $sector \in sectors$  do
4    $Q_{seed} \leftarrow selectSeeds(Q_k);$ 
5    $\neg Q_{seed} \leftarrow Q_k - Q_{seed};$ 
6    $trainData \leftarrow Null;$ 
7   while true do
8      $trainData \leftarrow trainData \cup Q_{seed};$ 
9      $f_k(x) \leftarrow GPR(trainData);$ 
10     $flagExpand \leftarrow false;$ 
11    for  $p \in \neg Q_{seed}$  do
12       $flag = Eval(p, f_k(x));$ 
13      if  $flag$  then
14         $Q_{seed} \leftarrow p;$ 
15         $delete p \text{ in } \neg Q_{seed};$ 
16         $flagExpand \leftarrow true;$ 
17      end
18    end
19    if  $flagExpand$  is false then
20      | break;
21    end
22  end
23   $Q_{in} \leftarrow trainData;$ 
24   $Q_{out} \leftarrow \neg Q_{seed};$ 
25   $F(x) \leftarrow f_k(x);$ 
26 end
```

Nevertheless, The original GP method is time-consuming, especially when dealing with dense point clouds. As a result, we implemented an acceleration strategy to address this issue. First, we reduced the number of point clouds by using a Voxel Filter to downsample the point clouds (line 1 in Alg.1). This was necessary because the LiDAR with 40 beams we used in our application generated one hundred thousand points in one frame, making it difficult to process in real-time with the original GP method. Next, we used a discretization method to reduce the size of the covariance matrix. As the size of the covariance matrix in GP increases exponentially with the number of points, directly applying GP to the entire point cloud of road regions would take nearly 38 seconds to compute. To address this, we divided the entire region into many sub-regions, and each region was modeled by GP (line 2 in Alg. 1). While this discretization may lead to a loss of modeling accuracy as it ignores global association, it was necessary to compromise between accuracy and calculation speed for real-time application. We conducted experiments in Section IV that showed the trivial effects of discretization on identification accuracy. Our second improvement is aimed at improving the modeling accuracy. Inspired by [32], an incremental modeling process is deployed to model the point clouds of each sub-

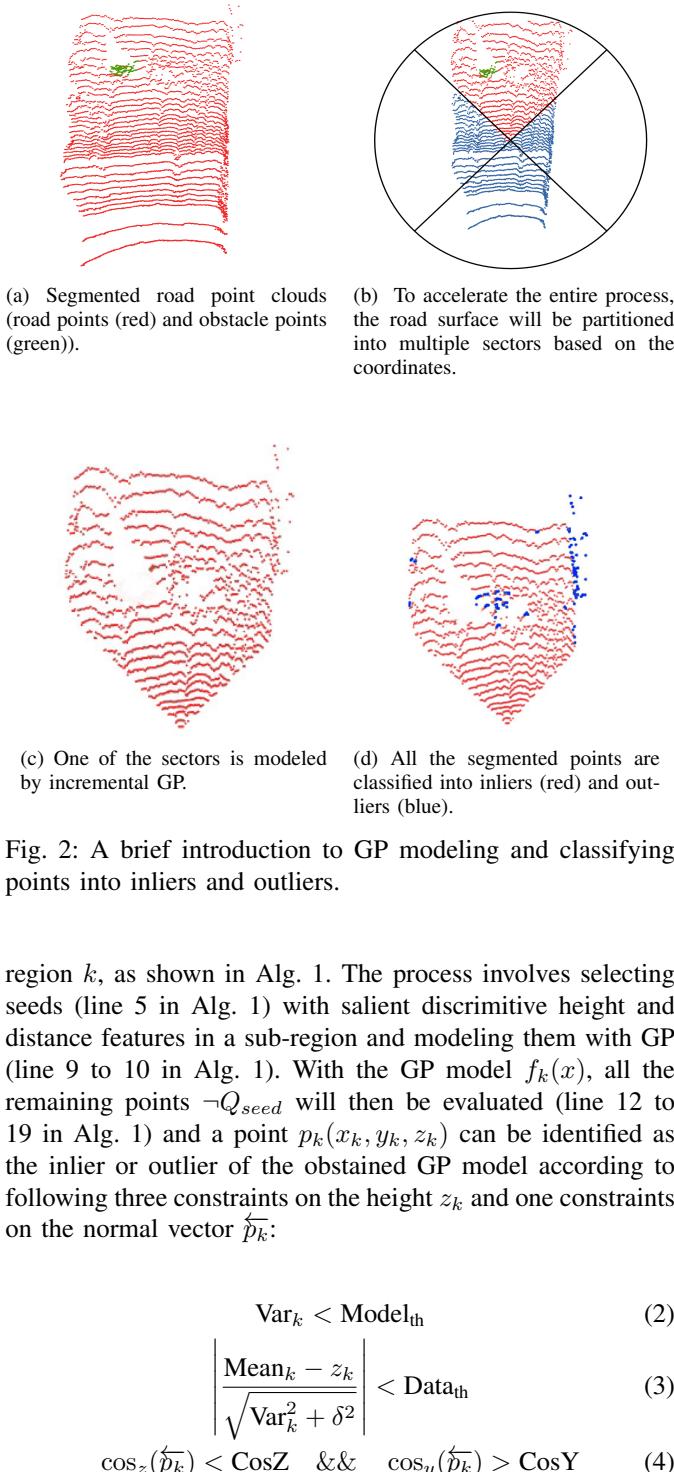


Fig. 2: A brief introduction to GP modeling and classifying points into inliers and outliers.

region k , as shown in Alg. 1. The process involves selecting seeds (line 5 in Alg. 1) with salient discriminative height and distance features in a sub-region and modeling them with GP (line 9 to 10 in Alg. 1). With the GP model $f_k(x)$, all the remaining points $\neg Q_{seed}$ will then be evaluated (line 12 to 19 in Alg. 1) and a point $p_k(x_k, y_k, z_k)$ can be identified as the inlier or outlier of the obstained GP model according to following three constraints on the height z_k and one constraints on the normal vector \hat{p}_k :

$$\text{Var}_k < \text{Model}_{th} \quad (2)$$

$$\left| \frac{\text{Mean}_k - z_k}{\sqrt{\text{Var}_k^2 + \delta^2}} \right| < \text{Data}_{th} \quad (3)$$

$$\cos_z(\hat{p}_k) < \text{CosZ} \quad \&\& \quad \cos_y(\hat{p}_k) > \text{CosY} \quad (4)$$

where Var_k denotes the variance of $f(x)$ on z_k , Model_{th} is a preset threshold, mean_k denotes mean value of $f(x)$ on z_k , δ is the noise and Dist_{th} is the distance threshold, \cos_z and \cos_y represent the cosine similarity between \hat{p}_k and z , and between \hat{p}_k and y , respectively.

Eq. (2) quantifies the level of uncertainty for each point, while Eq. (3) is similar to the Mahalanobis Distance [33] in that it measures the distance of each point from the GP distribution. The basic idea behind this process is that the inlier points of the model should be sufficiently close to the GP model (Eq. (3)). Additionally, incorporating these

points should further reduce the uncertainty of the model (Eq. (2)). Furthermore, the consistence of normal vectors is taken into consideration (Eq. (4)), which eliminates points that have a distribution similar to the GP distribution but do not belong to road regions. This incremental modeling process and point-wise optimization will continue until a certain iteration when no points are removed from $\neg Q_{seed}$. At this point, the algorithm terminates, and all remaining points in $\neg Q_{seed}$ are considered outliers of the model (line 19-21 Alg. 1), thereby significantly improving the accuracy of the segmentation. To provide additional clarification of this process, Fig. 2 explains the modeling process that was just described.

Algorithm 2: Composite Filter

Input:

$Q_{seg-obs}$: segmented obstacle point clouds,

Q_{out} : excluded outlier point clouds of road model

Output: Q_{obs} :refined obstacle point clouds

```

1  $Q_{clu-obs} \leftarrow \text{EuclidCluster}(Q_{seg-obs} + Q_{out})$ 
2 for  $q_{clu-obs} \in Q_{clu-obs}$  do
3    $flag\_success \leftarrow \text{Eval\_cluster}(q_{clu-obs});$ 
4   if  $\neg flag\_success$  then
5     | delete  $q_{clu-obs} \in Q_{clu-obs};$ 
6   end
7 end
8 for  $q \in Q_{clu-obs}$  do
9    $f_k(x) \leftarrow \text{LocateSector}(q);$ 
10   $flag = \text{Eval}(q, f_k(x));$ 
11  if  $flag$  then
12    | delete  $p \in Q_{clu-obs};$ 
13  end
14 end
15  $Q_{obs} \leftarrow Q_{clu-obs};$ 
```

2) *Obstacle point clouds refinement:* Compared to road regions, obstacles have poor feature salience and less reliable segmentation accuracy. Therefore, further refinement is required. To address this issue, a composite filter is designed to filter out non-obstacle point clouds, as shown in Alg. 2. The process starts by clustering the segmented point clouds and excluded outliers from the road modeling process using an Euclidean cluster. Points that cannot be clustered are filtered out (line 1 in Alg. 2). In light of the potential occurrence of multiple clusters in the clustering result with the predominance of road points, we established a set of criteria to eliminate these unremarkable clusters. Specifically, we first filtered out all clusters with z-axis height differences less than 0.1m. This step allowed us to effectively remove the majority of clusters composed solely of road points. However, in uneven terrains, some clusters consisting solely of road points may have a height difference exceeding 0.1m. Additionally, relying solely on the z-axis height difference for filtering might lead to the erroneous exclusion of some low obstacles. In such cases, clusters consisting only of road points tend to exhibit extended lengths in the x or y direction. Therefore, we can combine more criteria to filter the clusters composed solely of road points out: the first criterion is the height difference in the

z-axis within the range of 0.1 to 0.2 meters (the majority of road clusters exhibit a z-axis height difference of less than 0.2 meters in our application), and the second criterion is that there is a difference of more than 2 meters in the x or y direction. Furthermore, the selection of parameters in these two filtering criteria is determined based on the driving environment. Concretely, in relatively flat environments, the z-axis height difference threshold can be appropriately lowered, while in environments with uneven road surfaces, a higher threshold may be necessary. As for the extension length threshold in the x or y direction, it should be determined based on the width of the road surface. All other clusters are retained. The candidate obstacle point clouds are then rechecked by the established road model, and possible mis-segmentation road point clouds are filtered out. In detail, the point q of segmented obstacle point clouds is located into the corresponding sector and evaluated according to the constraints Eq. (2) - (4). If q does not meet these three constraints, it will be treated as an obstacle point (line 3 to 6 in Alg. 2); otherwise, it will be excluded from the candidate obstacle point clouds $Q_{clu-obs}$.

TABLE I: Our perception and computing equipment

Equipment	Parameters
CPU	Intel(R) Xeon(R) Gold 5118 CPU @ 2.30GHz
Memory	256GB
GPU	Nvidia Geforce RTX 3090
Operation system	Ubuntu 20.04
LiDAR	Hesai Pandar40M (mobile LiDAR)
Learning API	PyTorch 1.4

IV. EXPERIMENTS

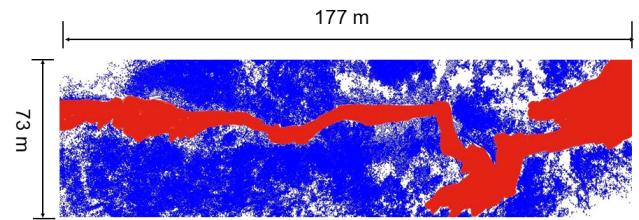
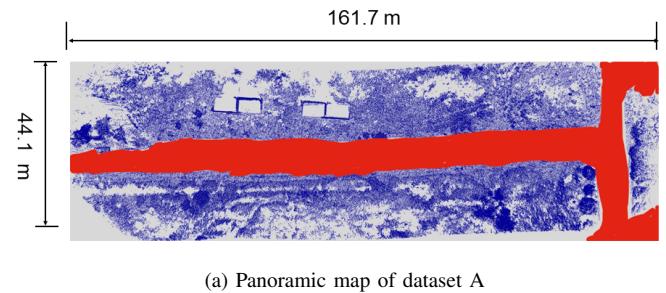
We conducted experiments using a field UGV, as depicted in Fig. 3. Our experimental environment is illustrated in Fig. 4, while Fig. 5 and Fig. 6 showcase the typical challenges in off-road environment, where mounds and pits are scattered and road is bumpy and winding. Additionally, Fig. 5 also highlights the image blurring problem caused by driving conditions and underexposure due to complex lighting. In our experiment, we first conducted a comparison test with several existing SOTA methods to demonstrate our competitive performance. Subsequently, we carried out an ablation study to investigate the modeling effects of IOGP and the refinement effects of the composite filter, respectively. Finally, the effects of several important parameters will be investigated.

A. Experiment setup

To equip our UGV with sensing capabilities, we have mounted sensing equipment that is capable of detecting obstacles and other hazards. To account for poor road conditions, we drive the UGV at a speed of 5km/h. The parameters of our sensing equipment are listed in Tab. I. We collected data for training and testing our scheme and other methods in scenarios depicted in Fig. 4, which covers an area of approximately 167.1m*44.1m and 173m*73m, respectively. The traversable regions in this area are surrounded by vegetation and obstacles can be found in the road regions, as shown in Fig. 5(b).



Fig. 3: In the front of our UGV, an ensemble of environmental perception system is installed (indicated by the yellow rectangle). Specifically, the perception system is composed of cameras, mobile LiDAR, and inertial navigation systems (IMU) in combination.



(b) Panoramic map of dataset B

Fig. 4: Our entire experimental scene was mapped using point clouds, with the traversable area shown in red and the vegetation in the environment shown in blue.

1) *Comparison counterparts:* To validate our method, we compared it with six existing methods, including 2 deep-learning based methods and 4 artificial-feature based methods. Tab. V shows key parameters used in these comparison counterparts. We briefly introduce these methods:

1. Occupation-grid based method (OG) [34]: This method rasterizes the environment into grids and considers two nearby grids with a vertical distance smaller than a critical distance as traversable regions.

2. Plane-fitting based method (PF) [35]: This method

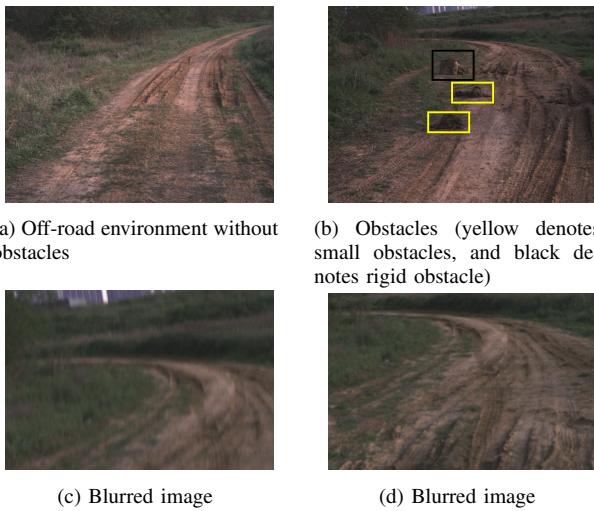


Fig. 5: The off-road environment. We show the blurred images caused by the bumpy road.



Fig. 6: Dataset B was collected in a more intricate environment. The road surface is more bumpy and winding, and the scene contains vehicle, truck and plants as obstacles, as well as pedestrians as dynamic obstacles.

iteratively selects prominent and highly confident seed points for the road surface model and obtains the road plane through plane fitting.

3. Geometric-angle based method (GA) [36]: This method converts point clouds into range images and then uses the geometric angle to identify the ground plane.

4. Elevation-difference based method (ED) [37]: This method extracts ground point clouds based on the maximum elevation difference h_{mn} . If h_{mn} is less than a pre-determined threshold h_{thr} , the cell $c(m, n)$ is considered a ground cell.

5. GndNet [38]: GndNet uses PointNet and Pillar Feature Encoding network to extract features and regress ground

height for each cell of the grid.

2) Metrics: For evaluating the identification performance of both road regions and obstacles, based on all the classes, we use the commonly-used precision (Eq. (5)), recall (Eq. (6)), $F1_{score}$ (Eq. (7)) and IoU (Eq. (8)). In the modeling comparison experiment, we also use these metrics. However, since there are only two classifications (inliers and outliers) in the modeling task, we use the accuracy of identifying inliers and outliers to measure the modeling methods. For this purpose, we treat road region point clouds as ground truth of inliers and obstacle point clouds as ground truth of outliers. This approach enables us to effectively measure the performance of the modeling methods in identifying road regions and obstacles.

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

$$F1score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (7)$$

$$IoU = \frac{TP}{TP + FP + FN} \quad (8)$$

B. Setup of Semantic Segmentation Network

1) Dataset: To evaluate the effectiveness of the proposed methodology, two datasets, namely Dataset A and Dataset B, were collected in two distinct, unstructured outdoor scenarios. Dataset A comprises 3923 frames of point cloud data encompassing outdoor road terrain and barriers such as trees and mounds, while Dataset B augments the obstacles with pedestrians as dynamic elements and consists of 1928 frames of point cloud data. During the training procedure, 2960 and 1350 frames were stochastically selected from Dataset A and Dataset B, respectively, as the training set, while the testing phase involved using the remaining 972 and 578 frames from Dataset A and Dataset B, respectively. The datasets were annotated with three kinds of labels, road regions, obstacles, and vegetation, with respective proportions of 41.81%, 0.89%, and 57.30%. All the deep learning-based networks in this paper were trained and tested on the same dataset. In order to better showcase the environment in which our field experiments were conducted, we utilized the SLAM algorithm [39] to map the entire scene. Fig. 7 illustrates an annotation error in different views. When labeling from the front view (Fig. 7(a)), all points with protruding features were incorrectly labeled as obstacles, whereas viewing the same data from the top view (Fig. 7(b)) revealed many road points labeled as obstacles, highlighting the importance of labeling from multiple viewpoints. Therefore, during the manual labeling of point clouds, to prevent potential labeling errors arising from occlusion of points in three-dimensional space, we also employed the obtained SLAM images for secondary confirmation, ensuring the labeling is as accurate as possible. Specifically, during labeling, we treated dynamic obstacles as simple static obstacles and annotated them individually in each frame, ensuring the accuracy of the annotations [3][40].

TABLE II: Comparison results of different modeling methods

Method	Precision	Inlier Recall	F1score	Precision	Outlier Recall	F1score	FPS
RANSAC	0.987	0.618	0.476	0.064	0.430	0.111	160
GP	0.964	0.778	0.694	0.192	0.082	0.115	1.9
GP+Incremental process	0.979	0.629	0.492	0.080	0.480	0.136	2.2
GP+Incremental process+Acceleration	0.975	0.626	0.489	0.073	0.463	0.127	178.6

TABLE III: Comparison results (Dataset A)

Method	Road Region				Obstacles				FPS
	Precision	Recall	F1score	IoU	Precision	Recall	F1score	IoU	
OG [34]	0.705	0.493	0.580	0.408	-	-	-	-	172.1
PF [35]	0.428	0.617	0.506	0.339	-	-	-	-	38.6
GA [36]	0.236	0.348	0.281	0.163	-	-	-	-	69.9
ED [37]	0.557	0.914	0.692	0.529	-	-	-	-	65.3
GndNet [38]	0.810	0.748	0.757	0.628	-	-	-	-	31.3
3D-MiniNet [30]	0.956	0.780	0.856	0.753	0.416	0.200	0.270	0.156	24.5
Our method	0.969	0.779	0.863	0.760	0.478	0.295	0.365	0.223	14.3

TABLE IV: Comparison results (Dataset B)

Method	Road Region				Obstacles				FPS
	Precision	Recall	F1score	IoU	Precision	Recall	F1score	IoU	
OG [34]	0.813	0.441	0.572	0.400	-	-	-	-	172.1
PF [35]	0.473	0.845	0.606	0.435	-	-	-	-	38.6
GA [36]	0.357	0.339	0.348	0.210	-	-	-	-	69.9
ED [37]	0.480	0.966	0.642	0.472	-	-	-	-	65.3
GndNet [38]	0.825	0.815	0.818	0.695	-	-	-	-	31.3
3D-MiniNet [30]	0.938	0.958	0.948	0.901	0.207	0.501	0.294	0.172	24.5
Our method	0.926	0.966	0.946	0.897	0.343	0.546	0.421	0.267	14.3

TABLE V: Key parameters used in comparison counterparts

Method	Parameters	Value
OG	Var_{thr}	0.3(m)
	$Mean_{thr}$	0.3(m)
PF	$Thre_{dist}$	0.1(m)
	$Thre_{seeds}$	1.2(m)
GA	Num_{lpr}	20
	$Thre_{angle}$	10°
ED	$Resolution$	0.5(m)
	h_{mn}	0.3(m)
Deep learning methods	$Resolution$	0.5(m)
	$Batch\ size$	4
Deep learning methods	$Learning\ rate$	0.001

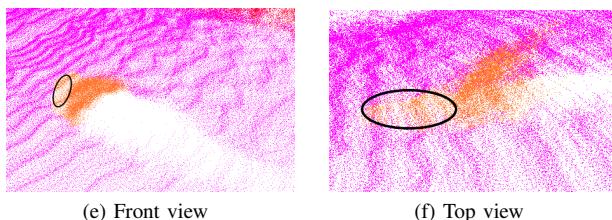


Fig. 7: The annotation error that can be identified from two different views (black ellipse).

2) *Network*: GndNet, 3DMiniNet and our model were implemented on NVIDIA GeForce RTX 3090. We set the max number of the training epoch to 40. Our model's learning rate and batch size are 0.001, 4, respectively.

C. Comparison results

Table III and Table IV present the comparison results. The first notable observation from Table III is that our method

surpasses all other methods in Dataset A regarding the identification performance of both road regions and obstacles. The superior discriminative ability of our utilized 3D-MiniNet in off-road environments significantly enhances the accuracy of identifying road regions. Comparing the performance of 3D-MiniNet with other traditional methods, we can observe its strength. Moreover, our M & R module also contribute to that by further checking the segmented road regions of 3D-MiniNet and exclude the misclassification. However, 3D-MiniNet also exhibits a weaker ability in identifying obstacles. Our M & R module addresses this limitation and further improves the accuracy of obstacle identification.

The second observation from Table IV is that, in Dataset B, our method performs better than other methods but is slightly inferior to 3D-MiniNet (by less than 0.2% F1 score) in identifying road regions. When comparing the scenarios in Dataset B (Fig. 6) with Dataset A (Fig. 5), we can see that Dataset B has more obstacles. These obstacles are distributed in different positions with different shapes and heights. Identifying these obstacles cannot solely rely on the segmentation results of 3D-MiniNet (as observed in the obstacle identification results), and our M & R module needs to make additional efforts in identifying the obstacle point clouds from the road regions. To filter out as many obstacle point clouds as possible, some road region point clouds may be sacrificed. However, considering the significant benefit of obstacle identification, the sacrifice in performance for road region identification is relatively insignificant.

The third observation highlights a significant improvement in obstacle identification for both Dataset A (9.5% F1score) and Dataset B (12.7% F1score). In these datasets, the boundaries between obstacles and roads are unclear, and

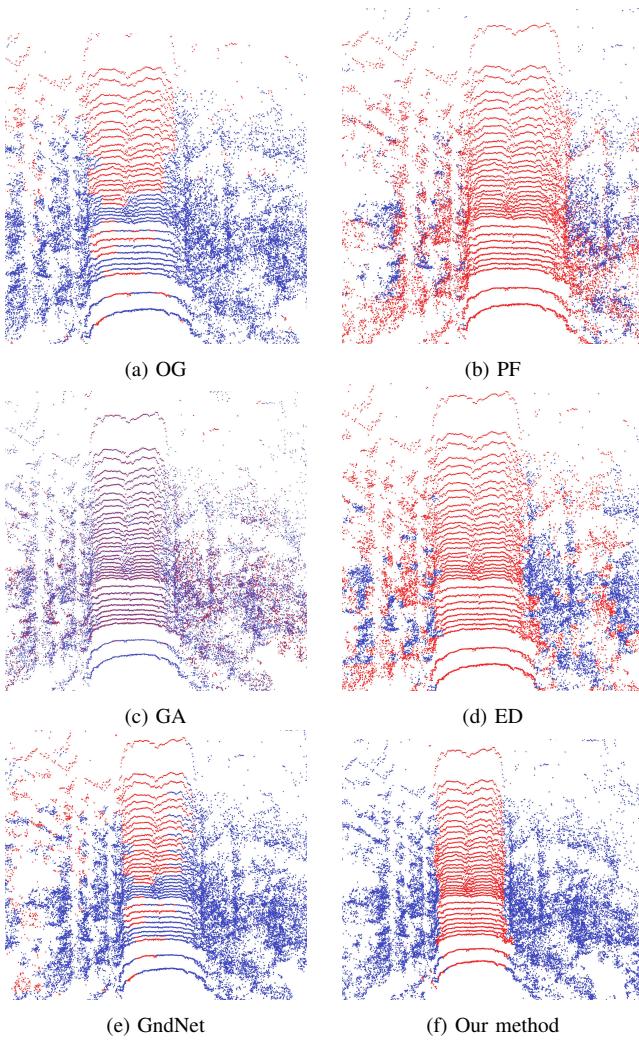


Fig. 8: Road regions detection performance of different methods (red denotes the road surface and blue denotes the vegetation the same below). The OG (a) and GA (b) methods exhibit poor performance due to unstable scans of LiDARs. In contrast, deep learning-based methods (e-f) show better results when compared to traditional methods, thanks to deep discriminative features.

the discriminative features are weak. The performance of 3D-MiniNet demonstrates that even deep features struggle to accurately identify them. However, with the introduction of the M & R module, we enhance the global features of roads through the modeling process IOGP, eliminate potential outliers, and reevaluate these outliers using a specifically designed composite filter to filter out potential obstacle point clouds.

Fig. 8 illustrates the identification performance of each method in Dataset A. Notably, OG and GndNet methods misclassify many road points as vegetation due to the height grids' inability to represent the discriminative feature of the road surface when the LiDAR cannot maintain a stable height from the ground. GA method assumes that the vehicle should drive on a smooth road surface, making it unsuitable for off-road environments, resulting in the worst recall rate. PF

and ED methods can extract the road surface despite some segmentation errors on the boundary between the road surface and vegetation. These methods improve with more iteration optimization (PF) or by selecting more robust features (ED) instead of height difference. Among the deep-learning based methods, the GNDNet is customized for ground extraction, it still falls short of our method as it cannot distinguish obstacles from road regions.

TABLE VI: Ablation study on Dataset A

Method	Obstacles				FPS
	Precision	Recall	F1score	IoU	
Net	0.416	0.200	0.270	0.156	24.5
Net+IOGP	0.066	0.422	0.114	0.060	17.5
Net+IOGP+EC	0.322	0.330	0.198	0.274	14.6
Net+IOGP+EC+WIM	0.478	0.295	0.365	0.222	14.3

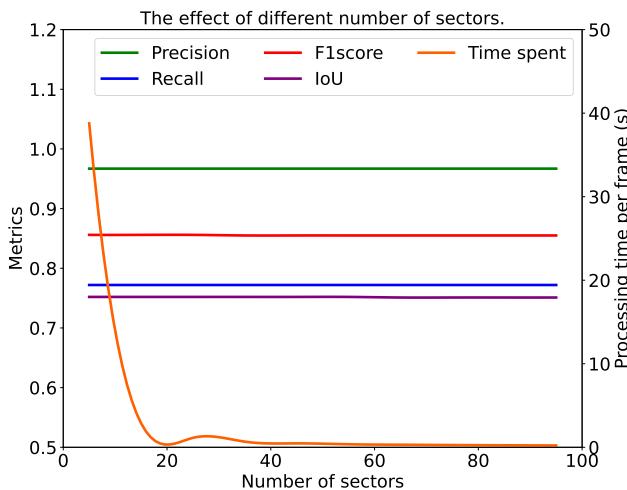
TABLE VII: Ablation study on Dataset B

Method	Obstacles				FPS
	Precision	Recall	F1score	IoU	
Net	0.207	0.501	0.294	0.172	24.5
Net+IOGP	0.066	0.763	0.122	0.065	17.5
Net+IOGP+EC	0.174	0.676	0.277	0.161	14.6
Net+IOGP+EC+WIM	0.343	0.546	0.421	0.267	14.3

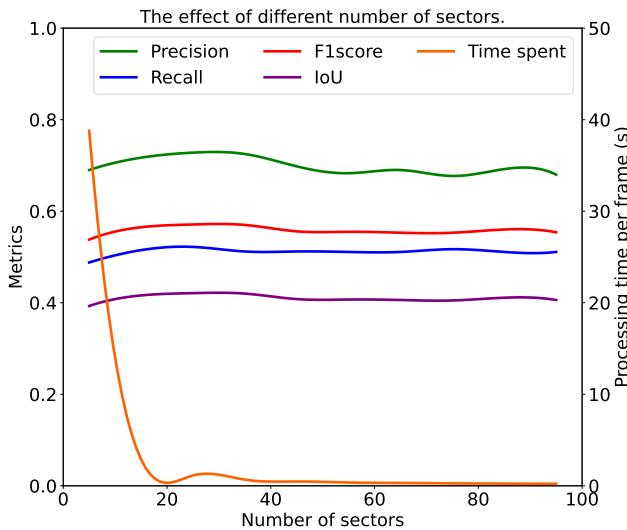
D. Ablation study

To investigate the individual effects of each modules of our method, we conducted an ablation study on dataset A and dataset B, as presented in Tab. VI and Tab. VII. Since the segmentation of road regions mainly depend on our customized network, we only compare the identification performance of obstacles with our M & R module. The results in Tab. VI and Tab. VII indicate that while IOGP can enhance the recall of obstacles, it also decreases the identification precision of obstacles. This is primarily due to the fact that IOGP is not trained offline, and misclassification is bound to occur during the online iteration optimization process. Additionally, IOGP only takes the distribution of z into account, which is not a reliable method for identifying obstacles. Nonetheless, IOGP indeed achieves that more obstacle point clouds are segmented from road regions, resulting in a significant increase in recall and narrowing down the filtering workspace for subsequent processing. The results also show that our composite filter can further refine the excluded point clouds of IOGP, improving performance without sacrificing efficiency. We will further analyze these two modules to better understand their individual contributions to the overall performance of our method.

1) *IOGP*: To demonstrate the modeling and filtering performance of IOGP, a comparison and ablation study experiment is conducted based on Dataset A, as shown in Tab. II. The accuracy of inlier judgement is used to measure the modeling ability, with the aim of including as many road region point clouds as possible. The results show that IOGP has similar performance to the RANSAC method and is second only to the original GP. However, the purpose of modeling is to assist in identifying obstacles, so the accuracy of inlier judgement alone is not enough to measure the whole performance.



(a) The impact of different sector numbers on road.



(b) The impact of different sector numbers on obstacles.

Fig. 9: The effect of different number of sectors. Increasing the number of sectors has been found to significantly reduce the time consumption, while maintaining the stability of performance.

For the filtering ability, the accuracy of outlier judgement is used, with the aim of excluding obstacle points from the segmented point clouds as many as possible. The results show that IOGP could significantly capture obstacle point clouds and improve recall by 5% over RANSAC and the original GP has little obstacle-identification ability. The low precision in outlier judgement is mainly due to the small number of obstacle point clouds compared to the road region point clouds.

Regarding modeling speed, the results indicate that the GP method is slower than the RANSAC method. However, with the acceleration module, IOGP achieves the highest modeling speed among all the methods. The discretization of the regions significantly reduces the size of the covariance matrix and

speeds up computation. The effect of different degrees of discretization on modeling accuracy is shown in Fig. 9. The time consumption decreases significantly with the increase of the number of sectors, but the identification accuracy is not sensitive to the number of sectors. Based on this experiment, each region is divided into 90 sub-sectors in the IOGP method.

2) *Composite filter*: The composite filter used in our method consists of two components: the Euclidean Cluster based filter (EC) and the well-trained IOGP model based filter (WIM). These filters are employed to filter out the possible road region point clouds in the excluded outliers during the IOGP modeling, leaving only the obstacle point clouds.

The EC filter significantly improves obstacle identification by utilizing local clustering features, which have superior discriminative performance in identifying obstacles. This leads to a remarkable 21.4% improvement in F1 score for Dataset A (Tab. VI) and a 15.5% improvement for Dataset B (Tab. VII). However, the clustering process is unable to identify road region point clouds that are in close proximity to obstacles and are not excluded during the IOGP process. To address this issue, a WIM module is employed to filter out these road region point clouds based on inlier judgment. This approach results in a 16.7% improvement in F1 score for Dataset A (Tab. VI) and a 23.5% improvement for Dataset B (Tab. VII).

Fig. 11(a-b) displays the point clouds of obstacles that were erroneously identified as road regions. Our EC filter effectively rectifies this issue by utilizing the clustering property of these point clouds. In Fig. 11(c), we observe that several road point clouds were mistakenly classified as obstacles, but our WIM filter successfully corrected this error.

E. Investigation on key parameters

In our method, the parameters in IOGP are just as important as the parameters in 3D-MiniNet. Therefore, we conducted an investigation into the effects of these parameters and their sensitivity to our method. Specifically, we looked at the model parameters of σ_f and σ_n and the decision parameters $Model_{th}$ and $Data_{th}$ in Eq. (2) - (4). In the experimental parameter selection, we set ten parameters at intervals of 0.1 from 0.1 to 1.0, the results are shown in Fig. 10 and Tab. VIII. One key observation is that the identification of road regions is not as sensitive to these parameters as the identification of obstacles. This is because there are typically fewer point clouds in obstacle regions, making them more prone to being affected by changes in the parameters. Another important finding is that the most sensitive parameter for obstacle identification is $Model_{th}$, which determines the inliers and outliers based on the distribution of z . However, we also found that our performance improves with increasing $Model_{th}$, making it easy to determine the optimal value for this parameter. Finally, we found that most of the parameters have a reasonable value between 0.4-0.6, indicating a range for parameter optimization.

It is noteworthy that the optimal hyperparameter values for achieving the best performance in road regions and obstacle identification may not be the same, as observed in Tab. VIII. But the final parameter selection in this paper is primarily guided by values that enhance the model's performance in

TABLE VIII: The impact of different values of key parameters on the F1 score of road regions and obstacle identification. In the table, $F1_x$ represents the F1 score value when the parameter is set to x.

	Key parameters	$F1_{0.1}$	$F1_{0.2}$	$F1_{0.3}$	$F1_{0.4}$	$F1S_{0.5}$	$F1_{0.6}$	$F1_{0.7}$	$F1_{0.8}$	$F1_{0.9}$	$F1_{1.0}$	Final values
Road regions	σ_f	0.857	0.857	0.857	0.857	0.857	0.851	0.851	0.851	0.851	0.851	$\sigma_f: 0.50$
	σ_n	0.854	0.856	0.856	0.857	0.851	0.851	0.848	0.844	0.844	0.844	$\sigma_n: 0.40$
	$Model_{th}$	0.844	0.844	0.851	0.851	0.858	0.858	0.858	0.858	0.858	0.858	$Model_{th}: 0.50$
	$Data_{th}$	0.840	0.849	0.853	0.855	0.856	0.858	0.859	0.859	0.859	0.859	$Data_{th}: 0.65$
Obstacles	σ_f	0.541	0.547	0.548	0.563	0.567	0.414	0.414	0.414	0.414	0.414	
	σ_n	0.468	0.504	0.522	0.534	0.413	0.422	0.325	0.272	0.272	0.272	
	$Model_{th}$	0.272	0.272	0.417	0.413	0.567	0.567	0.567	0.567	0.567	0.567	
	$Data_{th}$	0.348	0.443	0.468	0.508	0.529	0.567	0.565	0.527	0.525	0.524	

obstacle identification. The reasons for selecting parameters in this manner primarily stem from two considerations. Firstly, as observed in Fig. 10, the performance of road regions identification is less influenced by parameter values compared to obstacle identification. Secondly, this paper aims to use modeling to identify more misclassified obstacle points, making accurate obstacle identification more crucial. The specific selected parameter values are listed in Tab. VIII. Overall, our investigation into the effects of these parameters has helped us to better understand the sensitivity of our method and optimize its performance.

V. CONCLUSION

Our scheme mainly contributed to the traversable-region identification for the off-road driving and proposed an identification method with SOTA accuracy and a real-time speed. Compared to existing methods for identifying traversable regions, our approach has two distinct advantages. Firstly, it enables simultaneous identification of both road regions and obstacles, thus benefiting later decision-making and planning processes. Secondly, by integrating deep learning with traditional methods, our approach reduces the requirements for large learning models and maintains a balance between the accuracy and speed, which enables real-time applications. While our approach generates better segmentation results than many existing methods, it has a drawback of requiring annotations. As our method relies on deep features to achieve accurate road segmentation, a large amount of annotated data is needed to train the deep learning pipeline. Annotating point clouds is a labor-intensive task, and future research directions may involve developing more advanced unsupervised or semi-supervised methods to reduce the reliance on annotations.

ACKNOWLEDGMENT

This research is supported partly National Natural Science Foundation of China (No. 62232008), partly by Shenzhen science and technology program (JCYJ20210324122203009) partly by GuangDong Basic and Applied Basic Research Foundation (2020A1515110199), partly by Guangdong HUST Industrial Technology Research Institute, Guangdong Provincial Key Laboratory of Manufacturing Equipment Digitization (2020B1212060014), partly by China's University-Industry-Research Collaboration Fund for Innovation (2021ZYA03005) and partly by Key Laboratory of MIIT for Intelligent Products Testing and Reliability 2023 Key Laboratory Open Project Fund (No. CEPREI2023-02).

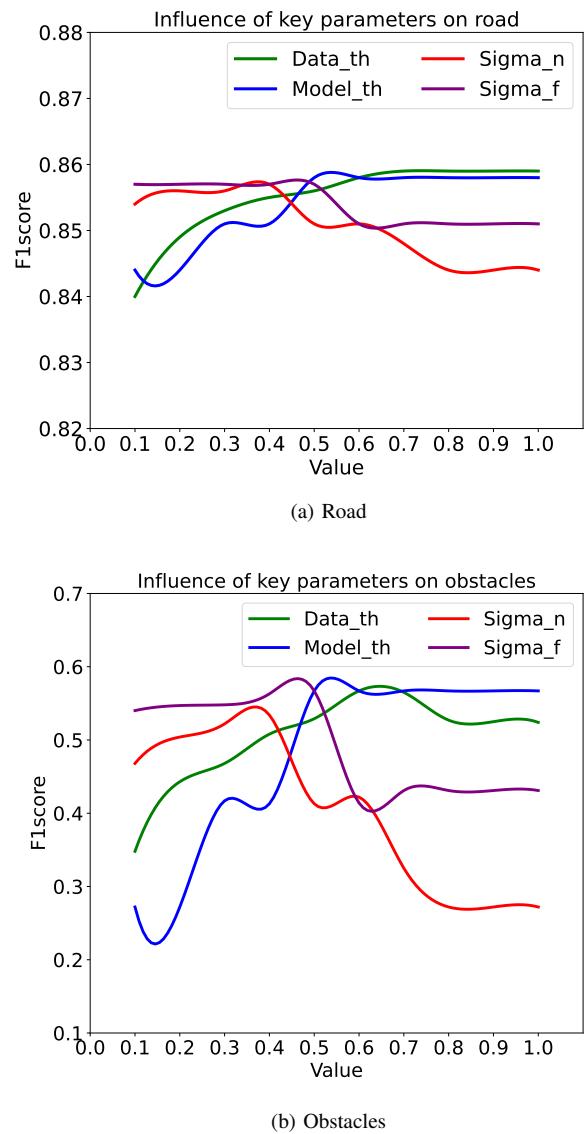


Fig. 10: Influence of key parameters on the performance of both obstacles and road regions identification.

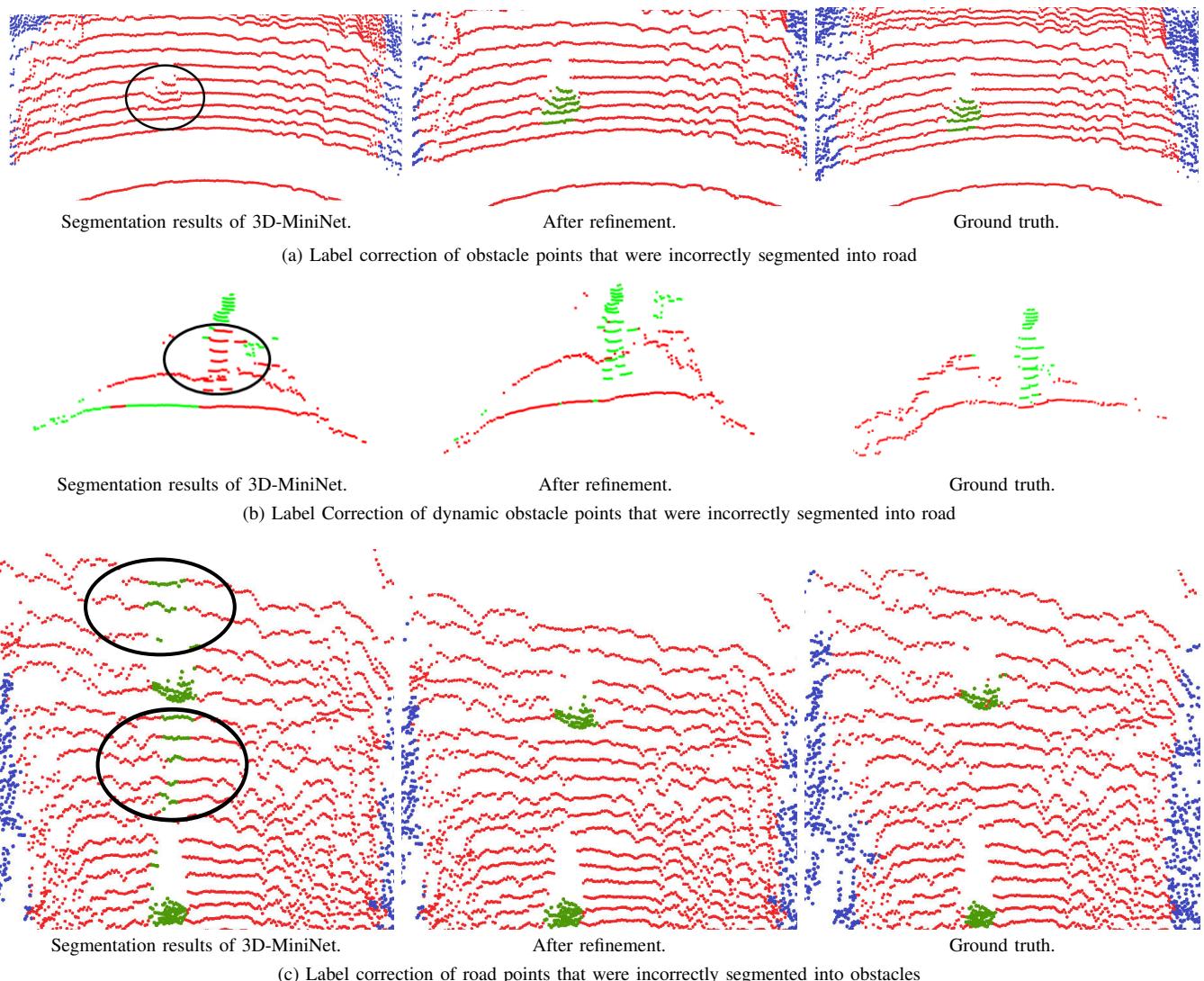


Fig. 11: Performance of our obstacle refinement module. Our refinement module has demonstrated remarkable success in correcting several mis-segmented obstacle point clouds by 3D-MiniNet, as objectively verified by comparing with the ground truth.

REFERENCES

- [1] Qingquan Li et al. “A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios”. In: *IEEE Transactions on Vehicular Technology* 63.2 (2013), pp. 540–555.
- [2] Guojun Wang et al. “Speed and accuracy tradeoff for LiDAR data based road boundary detection”. In: *IEEE/CAA Journal of Automatica Sinica* 8.6 (2020), pp. 1210–1220.
- [3] Yunxiao Shan et al. “Lidar-Based Stable Navigable Region Detection for Unmanned Surface Vehicles”. In: *IEEE Transactions on Instrumentation and Measurement* 70 (2021), pp. 1–13.
- [4] J. Ahtiainen, T. Stoyanov, and J. Saarinen. “Normal Distributions Transform Traversability Maps: LIDAR-Only Approach for Traversability Mapping in Outdoor Environments”. In: *Journal of Field Robotics* 34.3 (2017), pp. 600–621.
- [5] S. Zhou and K. Iagnemma. “Self-supervised learning method for unstructured road detection using Fuzzy Support Vector Machines”. In: *IEEE/RSJ International Conference on Intelligent Robots & Systems*. 2010.
- [6] Agneev Guin. *Terrain Classification to find Drivable Surfaces using Deep Neural Networks: Semantic segmentation for unstructured roads combined with the use of Gabor filters to determine drivable regions trained on a small dataset*. 2018.
- [7] Biao Gao et al. “Off-road drivable area extraction using 3D LiDAR data”. In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2019, pp. 1505–1511.
- [8] Zhe Chen, Jing Zhang, and Dacheng Tao. “Progressive lidar adaptation for road detection”. In: *IEEE/CAA Journal of Automatica Sinica* 6.3 (2019), pp. 693–702.

- [9] Mario Passani, J. Javier Yebes, and Luis M. Bergasa. “Fast pixelwise road inference based on Uniformly Reweighted Belief Propagation”. In: *2015 IEEE Intelligent Vehicles Symposium (IV)*. 2015, pp. 519–524. DOI: 10.1109/IVS.2015.7225737.
- [10] Giovani B. Vitor, Alessandro C. Victorino, and Janito V. Ferreira. “Comprehensive performance analysis of road detection algorithms using the common urban Kittiroad benchmark”. In: *2014 IEEE Intelligent Vehicles Symposium Proceedings*. 2014, pp. 19–24. DOI: 10.1109/IVS.2014.6856616.
- [11] Jiaming Han et al. “Robust and efficient vanishing point detection in unstructured road scenes for assistive navigation”. In: *Sensor Review* (2019).
- [12] Yifei Zhang and Shiyuan Wang. “A Robot Navigation System in Complex Terrain Based on Statistical Features of Point Clouds”. In: *IEEE Transactions on Intelligent Vehicles* (2023), pp. 1–17. DOI: 10.1109/TIV.2023.3297998.
- [13] Yeqiang Qian, John M Dolan, and Ming Yang. “DLT-Net: Joint detection of drivable areas, lane lines, and traffic objects”. In: *IEEE Transactions on Intelligent Transportation Systems* 21.11 (2019), pp. 4670–4679.
- [14] Xiaofeng Han et al. “Fully Convolutional Neural Networks for Road Detection with Multiple Cues Integration”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018, pp. 4608–4613. DOI: 10.1109/ICRA.2018.8460663.
- [15] Travis Manderson et al. “Learning to drive off road on smooth terrain in unstructured environments using an on-board camera and sparse aerial images”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 1263–1269.
- [16] Hanzhang Xue et al. “LiDAR-based Drivable Region Detection for Autonomous Driving”. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2021, pp. 1110–1116.
- [17] Jiangpeng Tao. *3D LiDAR based Drivable Road Region Detection for Autonomous Vehicles*. 2020.
- [18] Yunfeng Ai et al. “A Real-time Road Boundary Detection Approach in Surface Mine Based on Meta Random Forest”. In: *IEEE Transactions on Intelligent Vehicles* (2023), pp. 1–13. DOI: 10.1109/TIV.2023.3296767.
- [19] Yeqiang Qian et al. “Hy-Seg: A Hybrid Method for Ground Segmentation Using Point Clouds”. In: *IEEE Transactions on Intelligent Vehicles* 8.2 (2023), pp. 1597–1606. DOI: 10.1109/TIV.2022.3187008.
- [20] Hyunsuk Lee and Woojin Chung. “A Self-Training Approach-Based Traversability Analysis for Mobile Robots in Urban Environments”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 3389–3394. DOI: 10.1109/ICRA48506.2021.9561394.
- [21] Fengyu Xu et al. “A Framework for Drivable Area Detection Via Point Cloud Double Projection on Rough Roads”. In: *Journal of Intelligent & Robotic Systems* 102.2 (2021), pp. 1–19.
- [22] Shuo Gu et al. “3-D LiDAR + Monocular Camera: An Inverse-Depth-Induced Fusion Framework for Urban Road Detection”. In: *IEEE Transactions on Intelligent Vehicles* 3.3 (2018), pp. 351–360. DOI: 10.1109/TIV.2018.2843170.
- [23] Yingna Su et al. “An Illumination-Invariant Nonparametric Model for Urban Road Detection”. In: *IEEE Transactions on Intelligent Vehicles* 4.1 (2019), pp. 14–23. DOI: 10.1109/TIV.2018.2886689.
- [24] Juil Sock et al. “Probabilistic traversability map generation using 3D-LIDAR and camera”. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016, pp. 5631–5637.
- [25] Lupeng Zhou et al. “Terrain Traversability Mapping Based on LiDAR and Camera Fusion”. In: *2022 8th International Conference on Automation, Robotics and Applications (ICARA)*. IEEE. 2022, pp. 217–222.
- [26] Peizhou Ni et al. “Robust 3D Semantic Segmentation Based on Multi-Phase Multi-modal Fusion for Intelligent Vehicles”. In: *IEEE Transactions on Intelligent Vehicles* (2023), pp. 1–13. DOI: 10.1109/TIV.2023.3317784.
- [27] Fabio Ruetz et al. “Forest Traversability Mapping (FTM): Traversability estimation using 3D voxel-based Normal Distributed Transform to enable forest navigation”. In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2022, pp. 8714–8721. DOI: 10.1109/IROS47612.2022.9981401.
- [28] Wenquan Zhang et al. “LiSeg: Lightweight Road-object Semantic Segmentation In 3D LiDAR Scans For Autonomous Driving”. In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. 2018, pp. 1021–1026. DOI: 10.1109/IVS.2018.8500701.
- [29] Justin Liang et al. “Convolutional Recurrent Network for Road Boundary Extraction”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 9504–9513. DOI: 10.1109/CVPR.2019.00974.
- [30] Iñigo Alonso et al. “3D-MiniNet: Learning a 2D Representation From Point Clouds for Fast and Efficient 3D LIDAR Semantic Segmentation”. In: *IEEE Robotics and Automation Letters* 5.4 (2020), pp. 5432–5439. DOI: 10.1109/LRA.2020.3007440.
- [31] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. “Segnet: A deep convolutional encoder-decoder architecture for image segmentation”. In: *IEEE transactions on pattern analysis and machine intelligence* 39.12 (2017), pp. 2481–2495.
- [32] Bertrand Douillard et al. “On the segmentation of 3D LIDAR point clouds”. In: *2011 IEEE International Conference on Robotics and Automation*. IEEE. 2011, pp. 2798–2805.
- [33] R. D. Maesschalck, D. Jouan-Rimbaud, and D. L. Massart. “The Mahalanobis distance”. In: *Chemometrics and Intelligent Laboratory Systems* 50.1 (2000), pp. 1–18.

- [34] Sebastian Thrun et al. "Stanley: The robot that won the DARPA Grand Challenge". In: *Journal of field Robotics* 23.9 (2006), pp. 661–692.
- [35] Dimitris Zermas, Izzat Izzat, and Nikolaos Papantoniopoulos. "Fast segmentation of 3d point clouds: A paradigm on lidar data for autonomous vehicle applications". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 5067–5073.
- [36] Igor Bogoslavskyi and Cyrill Stachniss. "Efficient online segmentation for sparse 3D laser scans". In: *PFG—Journal of Photogrammetry, Remote Sensing and Geoinformation Science* 85.1 (2017), pp. 41–52.
- [37] Bo Zhou et al. "S4-SLAM: A real-time 3D LIDAR SLAM system for ground/watersurface multi-scene outdoor applications". In: *Autonomous Robots* 45.1 (2021), pp. 77–98.
- [38] Anshul Paigwar et al. "Gndnet: Fast ground plane estimation and point cloud segmentation for autonomous vehicles". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 2150–2156.
- [39] Jens Behley and Cyrill Stachniss. "Efficient Surfel-Based SLAM using 3D Laser Range Data in Urban Environments." In: *Robotics: Science and Systems*. Vol. 2018. 2018, p. 59.
- [40] Xiangtong Yao et al. "LiDAR based navigable region detection for unmanned surface vehicles". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 3754–3759.



Yunxiao Shan received the M.Sc. degree in machinery manufacturing and automation from the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun, China, and the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2010 and 2018, respectively. From 2015 to 2016, he was trained at Rutgers University, New Brunswick, NJ, USA. He joined Sun Yat-sen University as a Associate Professor in 2021. He also joined the team of marine intelligent unmanned equipment

in the Southern Marine Science and Engineering Guangdong Laboratory (zhuhai). He has contributed more than 20 publications on the research of autonomous systems. His research interests include the planning and control of autonomous vehicles, water surface intelligence, and research on autonomous surface vehicles.



Yao Fu is currently pursuing the M.Sc. degree in computer science and technology with the School of Artificial Intelligence, Sun Yat-sen University, Zhuhai, China. His research interests focus on point cloud processing and traversable area identification.



Xiangchun Chen received the M.S. degree from Army artillery college in 2006. He is currently an associate professor in the academy of artillery and air defense. His research interests include the high-mobility unmanned systems technology and applications.



Hongquan Lin received the M.Sc. degree in computer science and technology from the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China, in 2022. His research interests include object detection and tracking, and SLAM in autonomous surface vehicle areas.



Ziquan Zhang received the M.Sc. degree in computer science and technology from the School of Artificial Intelligence, Sun Yat-sen University, Zhuhai, China, in 2022. His research interests include object detection and tracking, and water surface intelligence.



Jun Lin serves as the deputy director of the Software and System Research Unit at the China CEPREI Laboratory. As a Professor of Engineering, he is entrusted with overseeing the scientific research management within the software and system domain of the laboratory. With over two decades of experience in the software and telecommunications industries, Lin Jun has dedicated his recent years to concentrated efforts in researching and providing services related to software quality and reliability. His focus extends to industrial software, industrial Internet, basic software, and software security & safety.



Kai Huang received the B.Sc. degree from Fudan University, China, in 1999, the M.Sc. degree from the University of Leiden, The Netherlands, in 2005, and the Ph.D. degree from ETH Zurich, Switzerland, in 2010. He joined Sun Yat-sen University as a Professor in 2015. He was appointed as the Director of the School of Data and Computer Science, Institute of Unmanned Systems, in 2016. He was a Senior Researcher with the Computer Science Department, Technical University of Munich, Germany, from 2012 to 2015, and a Research Group Leader with Fortiss GmbH, Munich, in 2011. His research interests include techniques for the analysis, design, and optimization of embedded systems, particularly in the automotive and robotic domains. He has been serving as a member of the Technical Committee on Cybernetics for Cyber-Physical Systems of the IEEE SMC Society since 2015. He was awarded the Program of Chinese Global Youth Experts 2014 and was granted the Chinese Government Award for Outstanding Self-Financed Students Abroad 2010. He was a recipient of the Best (Student) Paper/Candidates Awards IV 2018, ROBIO 2017 ESTC 2017, ESTMedia 2013, SAMOS 2009, ESTIMedia 2009, and a General Chairs' Recognition Award for Interactive Papers in CDC 2009.