

# Follow the Footprints: Self-supervised Traversability Estimation for Off-road Vehicle Navigation based on Geometric and Visual Cues

Yurim Jeon, E In Son and Seung-Woo Seo  
Seoul National University, Seoul, Republic of Korea  
{fabioisyo01, pingpang, sseo}@snu.ac.kr

**Abstract**—In this study, we address the off-road traversability estimation problem, that predicts areas where a robot can navigate in off-road environments. An off-road environment is an unstructured environment comprising a combination of traversable and non-traversable spaces, which presents a challenge for estimating traversability. This study highlights three primary factors that affect a robot’s traversability in an off-road environment: surface slope, semantic information, and robot platform. We present two strategies for estimating traversability, using a guide filter network (GFN) and footprint supervision module (FSM). The first strategy involves building a novel GFN using a newly designed guide filter layer. The GFN interprets the surface and semantic information from the input data and integrates them to extract features optimized for traversability estimation. The second strategy involves developing an FSM, which is a self-supervision module that utilizes the path traversed by the robot in pre-driving, also known as a footprint. This enables the prediction of traversability that reflects the characteristics of the robot platform. Based on these two strategies, the proposed method overcomes the limitations of existing methods, which require laborious human supervision and lack scalability. Extensive experiments in diverse conditions, including automobiles and unmanned ground vehicles, herbfields, woodlands, and farmlands, demonstrate that the proposed method is compatible for various robot platforms and adaptable to a range of terrains. Code is available at <https://github.com/yurimjeon1892/FtFoot>.

## I. INTRODUCTION

Traversability estimation is the task of predicting the terrain that a ground robot can traverse using data obtained from sensors. The most typically used sensors for this task are cameras and LiDAR, which provide two-dimensional images and three-dimensional (3D) point clouds, respectively. The prediction output is a traversability map, which is used to plan the robot’s driving path. As such, this task is crucial in ensuring the safe navigation of the robot. Failure of the traversability estimation may cause the robot to traverse the wrong path, thus potentially resulting in hazardous situations such as derailment, collision, or overturning. In particular, traversability estimation in off-road environments presents unique challenges that are not encountered in structured environments such as urban or indoor environments. These challenges include scattered obstacles and difficulty in distinguishing between traversable and non-traversable spaces in off-road terrains.

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2022-00207391, Development of Hashgraph-based Blockchain Enhancement Scheme and Implementation of Testbed for Autonomous Driving)

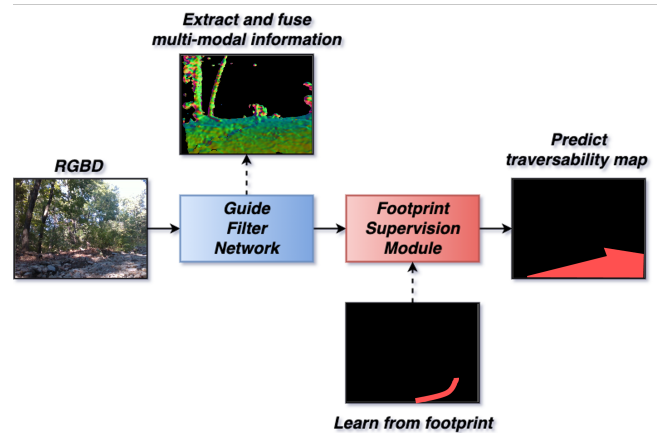


Fig. 1: **Overview of the proposed method** The proposed method consists of two components: a guide filter network optimized for feature extraction and fusion in off-road traversability estimation, and a footprint supervision module that learns traversability in a self-supervised manner from the robot’s path, referred to as the footprint.

Efforts have been expended continuously to estimate traversability in off-road environments. The first approach explicitly defines traversable spaces as human-supervised labels and employs rule-based [1][2][3][4] or learning-based [5][6] methods to predict them. The second approach is a self-supervised approach, that combines commonly used exteroceptive sensors (e.g., camera and LiDAR) with newly added proprioceptive sensors (e.g., IMU) [7][8][9][10][11]. The first approach is intuitive, but it requires laborious human supervision and fails to consider the robot platform in estimation. The second approach relies heavily on the proprioceptive sensor, which is sensitive to changes in the robot platform. Therefore, the traversability estimated by this method is applicable to only a few robot platforms and thus not scalable to other robot platforms.

The traversability of off-road terrains is not solely determined by obstacles or ground appearance. It is affected by multiple factors involving the off-road environment and robot platform. This study highlights three primary factors that affect a robot’s traversability in an off-road environment: surface slope (geometric cue), semantic information (visual cue), and robot platform. These three factors determine the traversability through interactions; In cases where the surface slope of the terrain sharply increases, the area acts as an

obstacle and becomes non-traversable. However, if the area's semantics can be penetrated, such as reeds, it becomes traversable. Lastly, even semantically non-traversable areas like rubble may still vary in traversability depending on the robot platform. For example, all-terrain vehicles can traverse over rubble, whereas small unmanned ground vehicles (UGVs) cannot do the same. Previous studies failed to predict robot-dependent traversability, by considering only geometric obstacles [4] or semantic information [6]. Studies focused on a robot platform have limitations in compatibility with different robot platforms [7][10].

We introduce two strategies to consider all three factors in traversability estimation. The first involves designing a neural network named as a **guide filter network (GFN)**, which extracts *surface* and *semantic information* from exteroceptive sensor data and integrates them to produce features optimized for traversability estimation. This is achieved by implementing an extraction network that extracts surface normals and semantic information from the input data, along with a fusion network composed of newly proposed guide filter layers, which offer advantages in the fusion of information from different modalities. The second strategy involves a **footprint supervision module (FSM)**, i.e., a self-supervision module that utilizes the footprint, which is the path traversed by the robot in pre-driving and thus incorporates the characteristics of the *robot platform*. This module enables a more scalable traversability estimation by providing a common traversability score that is compatible with diverse robot platforms and does not require laborious human supervision. Using these two strategies, we developed a self-supervised traversability estimation method applicable to various robot platforms based on the geometric and visual information of the surrounding environment. An overview of the proposed method is shown in Fig. 1.

We demonstrate the effectiveness of our method through experiments using the public off-road dataset RELIS-3D [12] and ORFD [13]. Our method successfully estimates the traversable space under various conditions, including different types of terrain, weather, lighting and robot platforms. The results show that the proposed method is a viable and optimized solution for estimating traversability in off-road environments.

The key contributions of this study are as follows:

- A novel approach for traversability estimation is proposed that considers critical factors that determine off-road traversability: surface slope, semantic information, and robot platform.
- A new network named the guide filter network, which is optimized for traversability estimation, is designed; it offers advantages in extracting and fusing information from different modalities.
- A footprint supervision module that can predict the robot-dependent traversability based on the robot's footprint in a self-supervised manner is developed to improve the versatility of the algorithm.
- The performance of the proposed method is demonstrated through experiments using datasets from various

terrains, weather, lighting and robot platforms.

## II. RELATED WORK

Early traversability estimation studies primarily focused on urban and indoor environments. In these structured environments, the definition of traversable space is clear. Therefore, the approach of predefining the traversable space for estimation demonstrated desirable performance. In an urban environment, a traversable space was typically defined as a free space, which is an empty space on a paved road without obstacles [14][15]. Similarly, in indoor environments, the presence of obstacles was a significant factor that determines traversability [16][17].

Early studies on off-road conditions used the same approach of predefining traversability. Rule-based methods extracted physical features such as roughness, slope, discontinuity, and hardness from input data to classify traversable spaces [1][2]. Another method used appearance features from the superpixels of an RGB image for classification [3]. Additionally, there is a method that pre-defined the characteristics of off-road obstacles in 3D data and segmented obstacles based on these characteristics [4]. A support vector machine-based method estimated traversability by classifying 3D point clouds into ground, vegetation, or objects using SVM [5]. A semantic segmentation-based method predicted semantic classes from RGB images, such as road, grass, tree, and obstacle, and estimated traversability based on these classes [6]. However, these methods required laborious human supervision and cannot predict traversability varies depending on the robot platform.

Subsequently, a self-supervised approach using proprioceptive sensor data from a robot's driving experience was introduced. There is a method that employed self-supervision, utilizing self-generated ground-truth cost derived from force-torque signals in a proprioceptive sensor to train the neural network model [7]. Other methods utilized self-supervision from IMU data [8] and the robot's velocity [9][10] to estimate traversability. Overall, these methods, use the robot's state as self-generated ground truth, making them incompatible with various robot platforms. Lastly, WayFAST [11], the closest method to ours, integrated RGB and depth images using the standard convolution layer. It used a narrow area around the robot's trajectory as self-generated ground truth. However, this method inefficiently integrated multi-modal information due to a standard convolution layer [18] and had a limitation of providing diverse path planning options due to a narrow ground truth space.

Herein, we introduce a new traversability estimation method that addresses the limitations of the previous methods. The proposed method can predict traversability in a self-supervised manner and does not require human supervision. Furthermore, our method considers the robot platform in traversability estimation while also providing universally applicable traversability results compatible with various robot platforms.

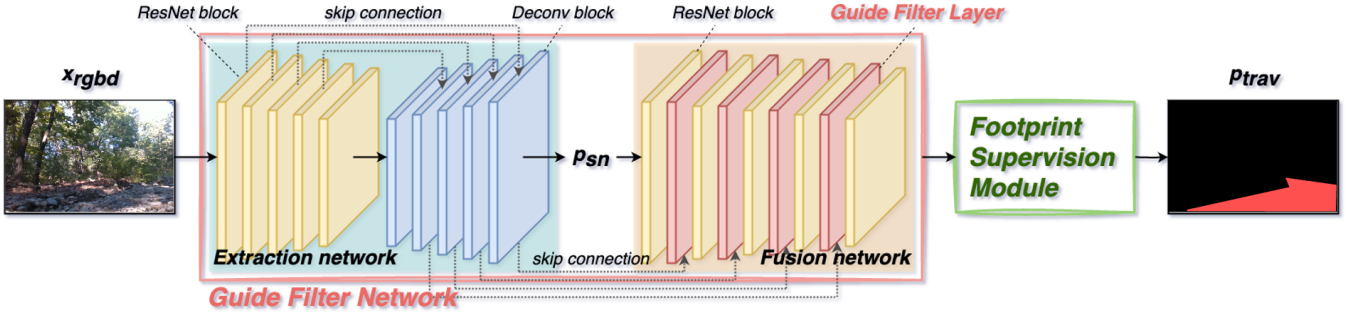


Fig. 2: **Structure of the proposed method** The guide filter network (GFN) consists of an extraction network and a fusion network. The extraction network estimates the surface normal image  $p_{sn}$  from the input RGB-D image  $x_{rgb-d}$ . Subsequently, a fusion network composed of guide filter layers integrates information from different modalities. Finally, the footprint supervision module (FSM) predicts the traversability map  $p_{trav}$ .

### III. METHODS

#### A. Overview

The structure of the proposed method is shown in Fig. 2. The network uses an RGB-D image denoted by  $x_{rgb-d}$  as the input. The input undergoes a sequence of ResNet [19] and deconvolution blocks. A skip connection is applied between ResNet block and deconvolution block to allow information flow between different layers, except for the first deconvolution block. In the skip connection, the output of the ResNet block is concatenated with the output of the previous deconvolution block and then input into the next deconvolution block. The output of the final deconvolution block is the surface normal image, denoted as  $p_{sn}$ . The surface normal image has a 3D surface normal value for each pixel in the image. The mean-squared error is used as the loss function for  $p_{sn}$ . A network from  $x_{rgb-d}$  to  $p_{sn}$  is referred to as the extraction network.

After the extraction network,  $p_{sn}$  is passed through a sequence of ResNet blocks and guide filter layers. The guide filter layer requires two inputs: the output of the previous ResNet block, which serves as a guidance image containing geometric information, and the output of the deconvolution block, which serves as a convolve image containing visual information. Through the guide filter layer, the network simultaneously considers both the surface slope contained in the geometric information as well as semantic information contained in the visual information. This network is referred to as the fusion network.

The output feature map is input into the footprint supervision module. The footprint supervision module is designed to estimate the entire traversable space using the footprint, which is a partial annotation of the entire traversable space. Consequently, a traversability map,  $p_{trav}$ , is generated as the final output.

#### B. Guide Filter Network (GFN)

Traversability estimation requires the consideration of geometric (e.g., surface slope) and visual cues (e.g., semantic information). The challenge lies in extracting appropriate features by integrating information from different modalities. A common approach for this integration is using a neural

network with standard convolutional layers. However, standard convolutional layers apply the same content-agnostic convolution filter to each image pixel, which leads to the limited feature representation [18][20]. This limitation becomes critical in off-road environments, where traversability depends on the interaction between geometric and visual cues.

Thus, we devised a new layer, named the guide filter layer, which can extract features via different filters for each location and content by simultaneously considering different modalities. The guide filter layer generates different input-conditioned filters for each pixel in an image. These spatially variant and content-dependent filters can improve the feature representation.

The remainder of this section is organized as follows. Section III-B.1 discusses the dynamic filter layer, which is the base of the guide filter layer. Section III-B.2 discusses the structure of the guide filter layer.

1) *Background: Dynamic filter layer:* The convolution operation of a dynamic filter layer can be formulated as follows:

$$F^*(i, j) = K_{ij} \otimes F(i, j), \quad (1)$$

where  $F \in \mathbb{R}^{c \times h \times w}$  is the input feature map and  $F^* \in \mathbb{R}^{c^* \times h^* \times w^*}$  is the output feature map.  $K_{ij} \in \mathbb{R}^{c^* \times c \times k \times k}$  is a generated filter of size  $k$  for pixel  $(i, j)$  and  $\otimes$  denotes the convolution operation. The filter  $K_{ij}$  can be decomposed [21] to reduce the number of parameters per filter as follows:

$$\begin{aligned} F'(i, j) &= K'_{ij} \otimes F(i, j) \\ F''(i, j) &= K''_{ij} \otimes F'(i, j), \end{aligned} \quad (2)$$

where  $F' \in \mathbb{R}^{c \times h' \times w'}$  is the intermediate feature map obtained by applying the spatially variant filter  $K'_{ij} \in \mathbb{R}^{1 \times 1 \times k \times k}$  to  $F$ . The final feature map  $F'' \in \mathbb{R}^{c' \times h' \times w'}$  is obtained by applying content-dependent filter  $K''_{ij} \in \mathbb{R}^{c' \times c \times 1 \times 1}$  to  $F'$ .

2) *Guide filter layer:* We developed a new layer, named the guide filter layer, which utilizes the basic structure of the dynamic filter layer for the effective fusion of different modalities. This layer requires two inputs, i.e., a guidance image and a convolve image, which represents different

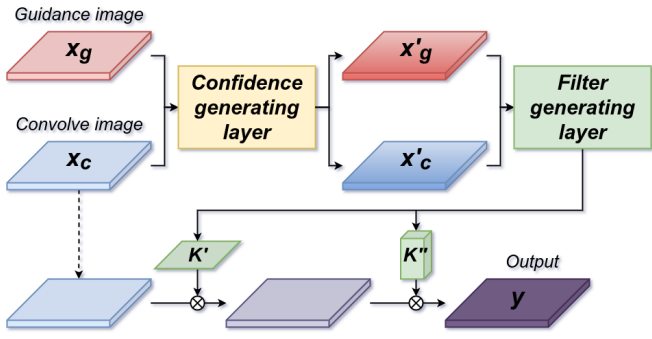


Fig. 3: **Structure of the guide filter layer** The guide filter layer takes two inputs: the guidance image  $x_g$  and the convolve image  $x_c$ . First, these images are fed into a confidence-generating layer, which produces weighted images  $x'_g$  and  $x'_c$ . These are then forwarded to the filter-generating layer to generate two decomposed filters,  $K'$  and  $K''$ . Finally, these filters are sequentially applied to the initial input  $x_c$ , resulting in the output  $y$ .

modalities. A guidance image is primarily used to generate a dynamic filter, which represents a surface normal image containing geometric information. A convolve image, which is an RGB image containing visual information, is then convolved with the dynamic filter for feature extraction. The guide filter layer regulates the effect of the guidance and convolve images on filter generation, thus allowing it to extract task-optimized features by considering both images. This approach allows the guide filter layer to efficiently integrate information from different modalities. The structure of the guide filter layer is shown in Fig. 3.

Two inputs of the guide filter layer, i.e., the guidance image  $x_g \in \mathbb{R}^{c \times h \times w}$  and convolve image  $x_c \in \mathbb{R}^{c \times h \times w}$ , are passed to the confidence-generating layer. The confidence-generating layer comprises a standard convolutional layer and a softmax layer. As output, a confidence score  $w_{conf} \in \mathbb{R}^{2 \times h \times w}$  is generated. The two inputs,  $x_g$  and  $x_c$ , are multiplied elementwise by  $w_{conf}$  to obtain weighted inputs  $x'_g$  and  $x'_c$ , respectively.

The weighted guidance image  $x'_g$  and convolve image  $x'_c$  are input to the filter-generating layer, which produces two types of filters per image pixel. The first filter is a spatially variant filter  $K'_{ij} \in \mathbb{R}^{1 \times 1 \times k \times k}$ , and the second is a content-dependent filter  $K''_{ij} \in \mathbb{R}^{c' \times c \times 1 \times 1}$ . The two generated filters are used sequentially for the convolution operation with  $x_c$  based on Eq. 2.

### C. Footprint Supervision Module (FSM)

Traversable space varies depending on the robot platform. Therefore, a recent approach utilizes a proprioceptive sensor, which measures the state of the robot, to train algorithms that can estimate robot-dependent traversability. However, the proprioceptive sensor is sensitive to minor changes in the robot platform, such as weight or sensor configurations. Thus, this approach makes the algorithm vulnerable to variations of the robot platform, hindering the scalability of the

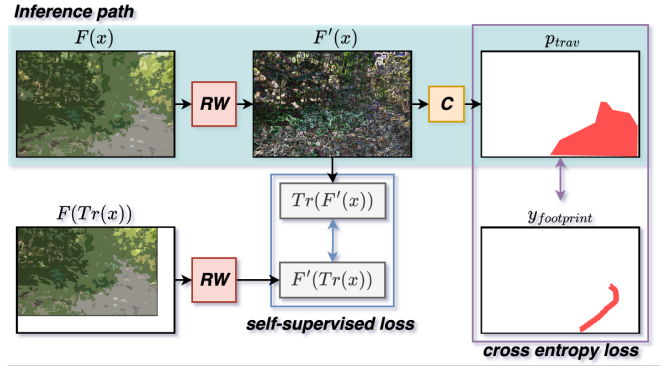


Fig. 4: **Structure of the footprint supervision module** The footprint supervision module takes an input feature map,  $F(x)$ . This module consists of a random walk (RW) and a convolutional layer (C).  $Tr$  represents the transformation function.  $p_{trav}$  is the final traversability map. The inference path is represented by the light blue box.

algorithm.

Our approach leverages the robot's trajectory in pre-driving as a footprint to capture the robot's response to the terrain. This enables us to predict the robot-dependent traversability, which reflects the characteristics of the robot platform. Additionally, this approach resolves the compatibility issue by estimating a common traversability score that can be applied to various robot platforms. This idea is implemented under the name, footprint supervision module (FSM). The footprint provides a partial annotation of the entire traversable space, and the footprint supervision module propagates information to neighboring areas with similar characteristics, thus enabling the estimation of the complete traversable space from partial annotation. The structure of this module is designed based on scribble-supervised semantic segmentation [22].

Sections III-C.1 and III-C.2 discuss the elements of the footprint supervision module. The structure of the footprint supervision module is illustrated in Fig. 4.

1) *Random walk*: The first element of the footprint supervision module is a random walk that propagates information around the footprint. The random walk has been used in graph theory [23]. In a graph  $G$  with  $n$  nodes, an affinity matrix  $A \in \mathbb{R}^{n \times n}$  can be defined, where  $A_{ij}$  represents the affinity between nodes  $i$  and  $j$ . Information propagation from a specific node within  $G$  can be simulated using  $A$ , and the diffusion of information over time can be modeled through repeated matrix multiplication.

The application of the random-walk operation to a neural network can be expressed as follows:

$$F' = \alpha AF + F, \quad (3)$$

where  $F$  denotes the input feature map and  $F'$  denotes the output. The affinity matrix  $A$  indicates the probability of a random walk and is defined as  $A = softmax(F^T F)$ . Finally,  $\alpha$  represents a learnable parameter that controls the degree of the random walk, which is the degree of

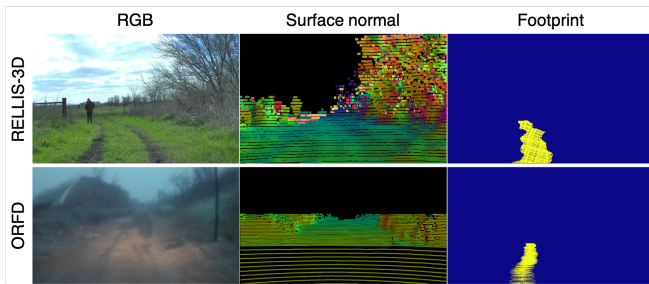


Fig. 5: **Examples of the dataset** Sample images from the RELLIS-3D and ORFD datasets.

information diffusion.

2) *Loss function: Self-supervised loss.* Self-supervised loss is widely used in unsupervised learning to yield consistent network outputs from similar features. It is defined as the difference between the transformation of the neural representation generated from input  $x$  and the neural representation generated from the transformation of  $x$ , and is expressed as follows:

$$L_{ss} = dist(Tr(F(x)), F(Tr(x))), \quad (4)$$

where  $F(x)$  is the feature map from input  $x$ ,  $Tr$  is the transformation function, and  $dist$  is the distance metric between the two neural representations. In this study,  $Tr$  is used as the horizontal flip and translation, and  $dist$  is used as the mean squared error.

**Cross entropy loss.** The cross-entropy loss is calculated between the robot’s footprint  $y_{footprint}$  and predicted traversability map  $p_{trav}$ . Because the footprint represents only a partial area of the entire traversable space, the negative labels in  $y_{footprint}$  include both positive and negative samples of the traversable space. Therefore, we assign weights of 1 and 0.1 to the losses on positive and negative labels, respectively.

## IV. EXPERIMENTS

### A. Dataset

The RELLIS-3D [12] dataset is an off-road dataset obtained using the Clearpath Robotics Warthog UGV platform. It provides RGB images captured using a Basler acA1920-50gc camera, point clouds from the Ouster OS1, and pose data from GPS. The dataset contains four data sequences, each comprising more than 2000 samples. We used sequences 0, 1, 2, and 3 for training (11497 samples) and sequence 4 for testing. Sequence 4 was split into three test routes, each containing 500, 500, and 700 frames, respectively.

The ORFD [13] dataset is another off-road dataset collected using the FAW Mazda Ruiyi 6 Coupe automobile platform. It includes RGB images and point clouds obtained using the Hesai Pandora all-in-one sensor kit. The dataset covers diverse terrains (woodland, farmland, and grassland), weather conditions (sunny, foggy, and snowy), and lighting variations (daylight, darkness, etc.). It provides 8392 samples for training, 1245 for validation, and 2193 for testing.

The footprint of the robot is generated based on the robot’s pose data. First, the robot’s trajectory is generated from the pose data, and projected onto the image plane. Next, the projected trajectory is transformed into a binary image, where areas the robot traversed are marked as 1, and others as 0. The example of the dataset is shown in Fig. 5.

### B. Experiments on Path Planning

1) *Experimental setting:* We evaluated the traversability estimation methods within the context of path planning on the RELLIS-3D dataset. Each method predicted a local cost map for each frame, and these local maps were accumulated to generate a global cost map. Here, a low cost indicates that the ground can be easily traversed by the robot, whereas a high cost indicates the opposite. Subsequently, paths were generated from the global cost map through the RRT\* algorithm [27]. The runtime was measured using an NVIDIA GeForce GTX 2080 Ti with an Intel Core i7-9700 K CPU @ 3.60 GHz.

We used a simulator to evaluate the suitability of the predicted global cost map for actual robot navigation. We generated 10 paths for each cost map, and tested whether the robot could actually follow these paths in a simulator. In the simulator, a virtual environment was implemented by converting the point cloud map to a mesh using Poisson surface reconstruction and integrating with the PyBullet physics engine [28].

The following methods were used in the experiment for comparison:

- WayFAST [11]: WayFAST uses RGB-D images as input and learns traversability through self-supervision from the driving experience.
- BEVNet [24]: BEVNet utilizes a neural network and supervised learning to predict geometric and semantic information from point clouds over time. It classifies terrain into four cost classes to generate traversability maps.
- GANav [25]: GANav is a supervised learning-based method that predicts semantic classes from RGB images and generates traversability maps through semantic class to traversability mapping.
- Agishev et al. [26]: Agishev et al. proposed a learning-based robot-terrain interaction model using RGB images and point clouds. The model leverages the robot’s inclination limits to predict traversability maps.

2) *Experimental results:* The quantitative evaluation results are presented in Tab. I. The success rate (SR) represents the ratio of successful driving trials out of 30 trials for each path. Cross-track error (CTE) is the difference between the position of the robot and the desired path while driving along the generated path. Finally, Hausdorff distance (HD) [29] was used to evaluate the similarity between the GT path and the generated path. For all of these metrics except SR, lower values indicate better performance. In the table, the symbol ‘-’ indicates that a path cannot be generated from the predicted cost map.

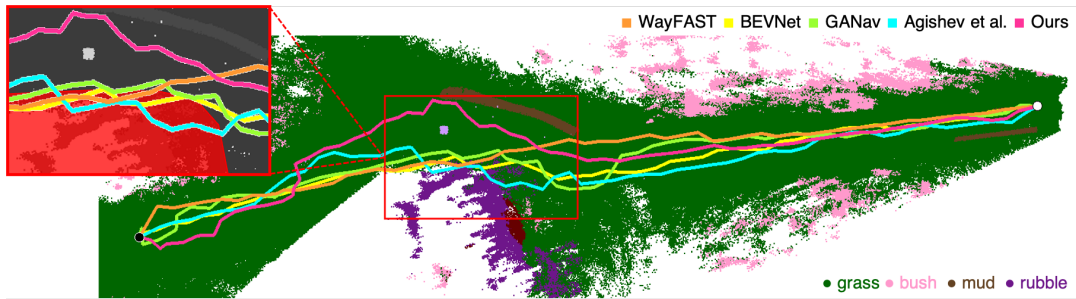


Fig. 6: Path generated on the RELIS-3D test route 1 The red area indicates dangerous terrain for driving due to rocks. Ours (pink line) is the only safe path that avoids this area.

TABLE I: Quantitative comparison of path planning on the RELIS-3D dataset

Method	Route 1			Route 2			Route 3			Runtime (ms)
	SR	CTE (m)	HD (m)	SR	CTE (m)	HD (m)	SR	CTE (m)	HD (m)	
WayFAST [11]	0.8	0.089	2.23	<b>1</b>	0.085	1.99	<b>1</b>	0.063	1.54	6.51
BEVNet [24]	0.63	0.116	3.04	<b>1</b>	0.093	3.13	0.7	0.137	2.32	86.31
GANav [25]	0.57	0.113	3.57	-	-	-	-	-	-	34.51
Agishev et al. [26]	0.5	0.131	2.43	0.67	0.134	2.39	0.87	0.113	2.73	60.17
Ours	<b>1</b>	<b>0.066</b>	<b>2.08</b>	<b>1</b>	<b>0.071</b>	<b>1.55</b>	<b>1</b>	<b>0.061</b>	<b>1.18</b>	16.5

TABLE II: Quantitative comparison of freespace detection on the ORFD dataset

Method	Supervision	Modality	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)	IoU (%)
OFF-Net [13]	Supervised	RGB+SN	96.1	94.5	91.1	92.8	86.6
WayFAST [11]	Self-supervised	RGB+D	67.1	44.6	<b>78.2</b>	56.8	39.6
Ours	Self-supervised	RGB+D	<b>85.6</b>	<b>76.8</b>	68.7	<b>72.5</b>	<b>56.9</b>

WayFAST showed the second-best performance after our method. However, it failed to effectively integrate geometric and visual information during the estimation process, leading to low cost estimates in hazardous areas. This resulted in a failure of the driving trial at Route 1. BEVNet, which relies solely on point clouds for traversability estimation, generated multiple impractical paths that cannot be followed in route 1 and 3. Similarly, GANav, which utilizes only RGB images, classified the puddle as a non-traversable space, through which the robot can easily pass; thus, a traversable space could not be secured and path generation failed in routes 2 and 3. Lastly, Agishev et al. also exhibited low performance, particularly in generating several impractical paths to follow. Our method outperforms the other methods for all of the metrics. This result indicates, in quantitative terms, that our fusion of geometric and visual cues was successful and that our method effectively learned the properties of the robot’s footprint. Furthermore, our method demonstrated the second fastest runtime after WayFAST, demonstrating its feasibility for real-time deployment in actual robots.

The qualitative comparison result is shown in Fig. 6. Our method ensured safety by taking a detour near non-traversable areas (marked in red). In contrast, other methods generated unsafe paths that pass through these areas.

### C. Experiments on Freespace Detection

Here, we evaluated the traversability estimation methods within the context of freespace detection on the ORFD dataset. We classified areas in the predicted cost map as freespace if their cost was less than 0.5. The method used

for comparison is WayFAST and OFF-Net [13]. OFF-Net uses RGB and surface normals with human-supervised labels for supervised learning in freespace detection. For quantitative evaluation, we used common metrics in freespace detection [13], including accuracy, precision, recall, F-score, and intersection of union (IoU). The evaluation results are presented in Tab. II.

Our method predicted a high cost for the boundaries of freespace, considering them as dangerous for navigation. This results in narrow freespace predictions, leading to a low recall value. However, avoiding these areas is crucial in real-world driving scenarios.

Overall, the performance of our self-supervised method is comparable to the supervised method, OFF-Net, and superior to another self-supervised method, WayFAST. This result demonstrates the validity of our method in pixel-level evaluation for the freespace detection. Additionally, it highlights the strength of our method in off-road environments where generating human-annotated ground truth is challenging.

## V. CONCLUSIONS

In this study, we addressed the off-road traversability estimation problem by considering surface slope, semantic information, and robot platform as essential factors. We introduced a novel approach that includes the *guide filter network*, integrating surface and semantic information, and the *footprint supervision module*, for self-supervised learning of robot-dependent traversability. Our optimized solution ensures safe robot navigation in diverse terrain and weather conditions, and is compatible with various robot platforms.

## REFERENCES

- [1] A. Howard and H. Seraji, "Real-time assessment of terrain traversability for autonomous rover navigation," in *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)*(Cat. No. 00CH37113), vol. 1. IEEE, 2000, pp. 58–63.
- [2] A. Howard, H. Seraji, and E. Tunstel, "A rule-based fuzzy traversability index for mobile robot navigation," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, vol. 3. IEEE, 2001, pp. 3067–3071.
- [3] D. Kim, S. M. Oh, and J. M. Rehg, "Traversability classification for ugv navigation: A comparison of patch and superpixel representations," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 3166–3173.
- [4] A. Talukder, R. Manduchi, A. Rankin, and L. Matthies, "Fast and reliable obstacle detection and segmentation for cross-country navigation," in *Intelligent Vehicle Symposium, 2002. IEEE*, vol. 2. IEEE, 2002, pp. 610–618.
- [5] M. Kragh, R. N. Jørgensen, and H. Pedersen, "Object detection and terrain classification in agricultural fields using 3d lidar data," in *Computer Vision Systems: 10th International Conference, ICVS 2015, Copenhagen, Denmark, July 6-9, 2015, Proceedings*. Springer, 2015, pp. 188–197.
- [6] D. Maturana, P.-W. Chou, M. Uenoyama, and S. Scherer, "Real-time semantic mapping for autonomous off-road navigation," in *Field and Service Robotics: Results of the 11th International Conference*. Springer, 2018, pp. 335–350.
- [7] L. Wellhausen, A. Dosovitskiy, R. Ranftl, K. Walas, C. Cadena, and M. Hutter, "Where should i walk? predicting terrain properties from images via self-supervised learning," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1509–1516, 2019.
- [8] G. G. Waibel, T. Löw, M. Nass, D. Howard, T. Bandyopadhyay, and P. V. K. Borges, "How rough is the path? terrain traversability estimation for local and global path planning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 16462–16473, 2022.
- [9] M. G. Castro, S. Triest, W. Wang, J. M. Gregory, F. Sanchez, J. G. Rogers III, and S. Scherer, "How does it feel? self-supervised costmap learning for off-road vehicle traversability," *arXiv preprint arXiv:2209.10788*, 2022.
- [10] A. J. Sathyamoorthy, K. Weerakoon, T. Guan, J. Liang, and D. Manocha, "Terrapn: Unstructured terrain navigation using online self-supervised learning," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 7197–7204.
- [11] M. V. Gasparino, A. N. Sivakumar, Y. Liu, A. E. Velasquez, V. A. Higuti, J. Rogers, H. Tran, and G. Chowdhary, "Wayfast: Navigation with predictive traversability in the field," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10651–10658, 2022.
- [12] P. Jiang, P. Osteen, M. Wigness, and S. Saripalli, "Rellis-3d dataset: Data, benchmarks and analysis," 2020.
- [13] C. Min, W. Jiang, D. Zhao, J. Xu, L. Xiao, Y. Nie, and B. Dai, "Orfd: A dataset and benchmark for off-road freespace detection," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2532–2538.
- [14] J. Yao, S. Ramalingam, Y. Taguchi, Y. Miki, and R. Urtasun, "Estimating drivable collision-free space from monocular video," in *2015 IEEE Winter Conference on Applications of Computer Vision*. IEEE, 2015, pp. 420–427.
- [15] S. Tsutsui, T. Kerola, S. Saito, and D. J. Crandall, "Minimizing supervision for free-space segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 988–997.
- [16] N. Hirose, A. Sadeghian, M. Vázquez, P. Goebel, and S. Savarese, "Gonet: A semi-supervised deep learning approach for traversability estimation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3044–3051.
- [17] N. Hirose, A. Sadeghian, F. Xia, R. Martín-Martín, and S. Savarese, "Vunet: Dynamic scene view synthesis for traversability estimation using an rgb camera," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2062–2069, 2019.
- [18] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool, "Dynamic filter networks," *Advances in neural information processing systems*, vol. 29, 2016.
- [19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [20] J. Zhou, V. Jampani, Z. Pi, Q. Liu, and M.-H. Yang, "Decoupled dynamic filter networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6647–6656.
- [21] J. Tang, F.-P. Tian, W. Feng, J. Li, and P. Tan, "Learning guided convolutional network for depth completion," *IEEE Transactions on Image Processing*, vol. 30, pp. 1116–1129, 2020.
- [22] Z. Pan, P. Jiang, Y. Wang, C. Tu, and A. G. Cohn, "Scribble-supervised semantic segmentation by uncertainty reduction on neural representation and self-supervision on neural eigenspace," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7416–7425.
- [23] L. Lovász, "Random walks on graphs," *Combinatorics, Paul erdos is eighty*, vol. 2, no. 1-46, p. 4, 1993.
- [24] A. Shaban, X. Meng, J. Lee, B. Boots, and D. Fox, "Semantic terrain classification for off-road autonomous driving," in *Conference on Robot Learning*. PMLR, 2022, pp. 619–629.
- [25] T. Guan, D. Kothandaraman, R. Chandra, A. J. Sathyamoorthy, K. Weerakoon, and D. Manocha, "Ga-nav: Efficient terrain segmentation for robot navigation in unstructured outdoor environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8138–8145, 2022.
- [26] R. Agishev, T. Petříček, and K. Zimmermann, "Trajectory optimization using learned robot-terrain interaction model in exploration of large subterranean environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3365–3371, 2022.
- [27] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.
- [28] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," <http://pybullet.org>, 2016–2021.
- [29] A. A. Taha and A. Hanbury, "An efficient algorithm for calculating the exact hausdorff distance," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 11, pp. 2153–2163, 2015.