IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

Seeing Through the Grass: Semantic Pointcloud Filter for Support Surface Learning

Anqiao Li, Chenyu Yang, Jonas Frey†, Joonho Lee, Cesar Cadena, and Marco Hutter

***Abstract*—Mobile ground robots require perceiving and understanding their surrounding support surface to move around autonomously and safely. The support surface is commonly estimated based on exteroceptive depth measurements, e.g., from LiDARs. However, the measured depth fails to align with the true support surface in the presence of high grass or other penetrable vegetation. In this work, we present the semantic pointcloud filter (SPF), a convolutional neural network (CNN) that learns to adjust LiDAR measurements to align with the underlying support surface. The SPF is trained in a semi-self-supervised manner and takes as an input a LiDAR pointcloud and RGB image. The network predicts a binary segmentation mask that identifies the specific points requiring adjustment, along with estimating their corresponding depth values. To train the segmentation task, 464 distinct images are manually labeled into rigid and non-rigid terrain. The depth estimation task is trained in a self-supervised manner by utilizing the future footholds of the robot to estimate the support surface based on a Gaussian process. Our method can correctly adjust the support surface prior to interacting with the terrain and is extensively tested on the quadruped robot ANYmal. We show the qualitative benefits of SPF in natural environments for elevation mapping and traversability estimation compared to using raw sensor measurements and existing smoothing methods. Quantitative analysis is performed in various natural environments, and an improvement by 48% RMSE is achieved within a meadow terrain.**

***Index Terms*—Legged Robots, Deep Learning for Visual Perception, Field Robots**

I. INTRODUCTION

A Comprehensive understanding of the robot's surroundings is a key element in achieving autonomous robot navigation in outdoor environments. Typically, ground robots are equipped with exteroceptive sensors such as LiDARs and depth cameras to directly capture the structure of the 3D environment. The captured geometrical information can be accumulated in a 2D occupancy map [1], [2], 2.5D elevation map [3], [4], or 3D voxel map [5] representation. These maps provide information about the support surface (the rigid surface that can provide support for the robot during traversing) and obstacles for downstream tasks such as motion planning [6], [7], traversability assessment [8], [9], and trajectory planning [10], [11]. However, these conventional approaches are based on the assumption that the world is rigid, i.e., the robot will step on the terrain or collide with the obstacles rather than moving or deforming them. While the assumption holds in most structured environments, it is not the case for unstructured natural environments with vegetation or soft terrain. Most navigation planning and control pipelines often treat the penetrable vegetation as rigid, which leads to sub-optimal locomotion and navigation. For example, a robot may try to step over vegetation or avoid non-existing obstacles without the capability to identify the support surface. In contrast, humans easily traverse natural terrains by associating semantic information with the terrain property. For example, before walking on grass, humans can anticipate from visual information that the high grass is penetrable and correctly adjust their gait.

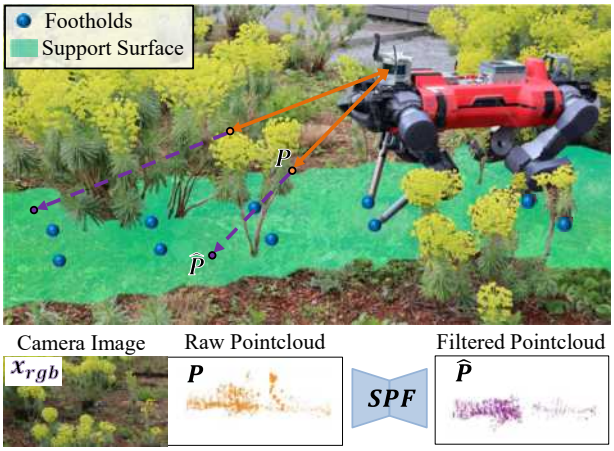

Fig. 1. A systematic sketch for Semantic Pointcloud Filter (SPF). From its onboard LiDAR sensors, the raw pointcloud P (orange) is generated. The raw pointcloud captures the structure of vegetation, which is undesirable for the reconstruction of the support surface. Utilizing the semantics extracted from onboard sensors, our proposed SPF filters the pointcloud by adjusting the depth of each point in the cloud. The filtered pointcloud $\hat{P}$ (magenta) is flatter and depicts the support surface.

In this work, we present a novel approach to enhance the environment perception capabilities of robots by learning the support surface from RGB images and LiDAR, thereby overcoming the limitations of prior geometric methods. Instead of directly acquiring a fused support surface representation, such as an elevation map or voxel map, we opt to refine raw sensor data for broader downstream application adaptability. Specifically, we propose a multi-modal model that filters pointcloud measurements by jointly performing semantic segmentation and depth estimation. The semantic segmentation

Manuscript received: May, 8, 2023; Revised August, 12, 2023; Accepted September, 8, 2023.

This paper was recommended for publication by Editor Sven Behnke upon evaluation of the Associate Editor and Reviewers' comments.

This work was supported by the Swiss National Science Foundation (SNSF) through project 188596, the National Centre of Competence in Research Robotics (NCCR Robotics), the European Union's Horizon 2020 research and innovation program under grant agreement No 101016970, No 101070405, and No 852044, and an ETH Zurich Research Grant. Jonas Frey is supported by the Max Planck ETH Center for Learning Systems.

All authors are with the Robotic Systems Lab, ETH Zurich, 8092 Zurich, Switzerland. {anqiali, chenyang, jonfrey, jolee, cesarc, mhutter} @ethz.ch

† Jonas Frey is with the Max Planck Institute for Intelligent Systems, 72076 Tübingen, Germany

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

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

module identifies the regions of the support surface, whose depth is then adjusted by the depth estimation module to align with the support surface. Considering the inherent challenge associated with annotating depth data, the depth estimation component employs a self-supervised learning approach by leveraging supervision signals generated by foothold positions. Whereas the semantic segmentation component is trained in a supervised manner using hand-labeled data.

We deploy our method on the legged robot ANYmal and validate it in a range of outdoor environments, including Grassland, Forest, and Hillside, against existing methods. Additionally, we show the benefit of our method for multiple robotic downstream tasks, including elevation mapping and traversability estimation. Our results highlight our approach's practicality and potential applicability in real-world scenarios.

Our main contributions are the following:

1) A novel Semantic Pointcloud Filter architecture that enables the filtering of raw LiDAR data to achieve alignment with the underlying support surface.
2) A semi-self-supervised learning framework that jointly performs support surface segmentation and depth estimation. Our framework utilizes manually labeled segmentation masks and support surface depths annotated by foothold positions.
3) Real-world experiments using the legged robot ANYmal, showcasing the benefits of the SPF for elevation mapping and traversability estimation.

II. RELATED WORK

A. Support Surface Estimation

Support surface estimation focuses on determining stable surfaces for robot mobility. In the situation where, e.g., sparse vegetation occludes the support surface, the vegetation can be filtered out by regarding them as outliers. This problem can be tackled by methods using Kalman Filters [4], [12] and other heuristics [7], [13].

However, these methods assume that the underlying surface is mostly flat. One can get a more realistic and robust estimation of the support surface by explicitly modeling the vegetation and the support surface. Various probabilistic models capturing both vegetation and ground characteristics [14], [15] have been proposed. They use proprioceptive and exteroceptive information to estimate support surfaces.

Rather than using heuristic modeling, Šalanský et al. [16] introduced a learning-based method to filter a 2.5D map, while Wellington et al. [17] employ a learning approach to filter a voxel map. In our research, instead of using a fused representation such as an elevation map or voxel map, we refine the raw sensor measurements directly. This allows our approach to generalize more effectively for a wider range of downstream applications.

B. Self-supervised Depth Estimation

Supervised depth estimation has a significant body of work [18], [19], but requires pixel-wise depth labeling. Self-supervised learning has been applied to monocular depth estimation, achieving better performance than supervised methods [20]. Further research incorporates semantic information [21], but monocular depth estimation remains challenging due to its ill-posed nature.

Our task also relates to the monocular depth completion, where the goal is to utilize sparse LiDAR pointcloud as additional input to predict the full geometry of the scene [22]. Self-supervised methods have been widely used for depth completion tasks [23], [24].

C. Multi-modal Semantic Segmentation

Image-based semantic segmentation has been successfully applied to various robotic applications, including terrain classification [25], [26]. To further improve the precision, various studies have explored the fusion of RGB and depth data for multi-modal semantic segmentation [27], [28]. In our task, we employ semantic segmentation as a means of aiding the estimation of the depth of the support surface. Beyond image space segmentation, pointcloud filtering methods like [29], [30] directly segment the pointcloud into the ground and objects. These methods are often used in large-scale terrain modeling. Whereas, our method is designed for real-time filtering of the on-board LiDAR.

III. METHOD

The overview of our approach is given in Fig. 2. We define two labels for the supervision of the model: the support surface depth estimation (SSDE) label, used to train the depth estimation module, and the support surface segmentation (SSSeg) label, used for the segmentation module. The whole process comprises three stages: In the first stage, SSDE labels are generated from robot trajectories in a self-supervised manner, and SSSeg labels are manually labeled. Then, the SPF is trained to perform jointly depth estimation and semantic segmentation. Finally, the pointcloud predicted by the SPF can be used for downstream applications.

A. Problem Definition

As shown in Fig. 1, given a camera image $\mathbf{x}_{rgb} \in \mathbb{R}^{m \times n \times 3}$ and raw pointcloud $P \in \mathbb{R}^{3 \times k}$, our objective is to derive the filtered pointcloud $\hat{P}$, using our SPF, $\hat{P} = f_{\text{SPF}}(P, \mathbf{x}_{rgb})$. Each point in the filtered pointcloud, aligns with the support surfaces or the impenetrable obstacles.

B. SSDE Label Generation

We extract the foothold positions of the robot and approximate the support surface using a Gaussian process (GP) similar to [15]. Then we generate pixel-wise depth labels by projecting the reconstructed support surface into the image.

1) Foothold extraction: By employing accurate localization method [31], [32], we assume the foothold positions serve as ground truth samples for the support surface. From the dataset, foot position trajectories are obtained using forward kinematics, comprising discrete height points over time. Footholds are extracted, assuming foot-ground contact occurs at local height minima in the *world frame*.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

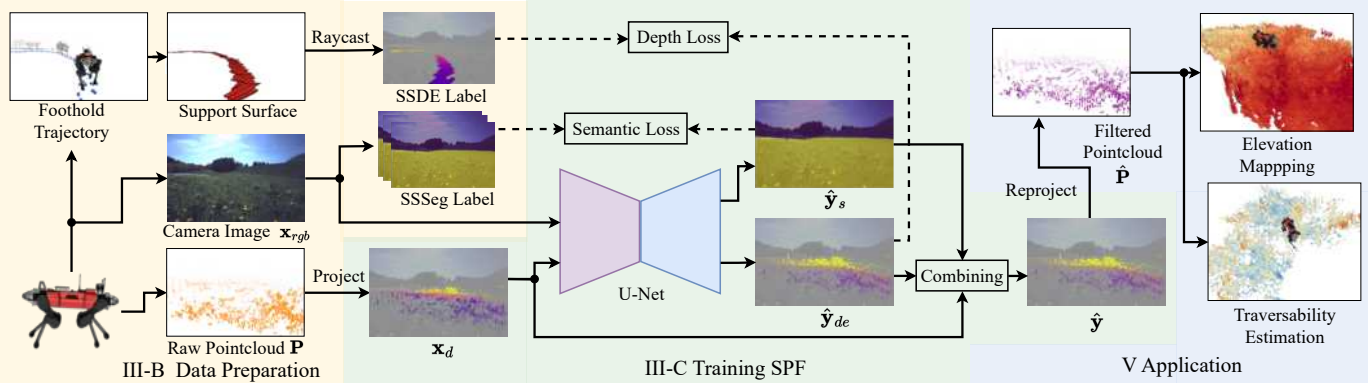

Fig. 2. Overview of the system and corresponding sections. The training data is generated from pre-recorded data, consisting of raw pointcloud and camera images acquired from onboard sensors of the robot and robot trajectories obtained through exteroceptive observations. The support surface can be reconstructed from footholds extracted from the trajectories. When training the SPF, the raw pointcloud data P is first projected onto image space as $\mathbf{x}_d$ and concatenated with the camera image. The U-net-based neural network predicts jointly the depth (using supervision of support surface depth estimation (SSDE) labels) and semantic segmentation (using supervision of hand-labeled semantic segmentation (SSSeg) labels). The final output $\hat{\mathbf{y}}$ is given by combining $\mathbf{x}_d$ and the predicted depth estimation $\hat{\mathbf{y}}_{de}$ based on the predicted semantic segmentation mask $\hat{\mathbf{y}}_s$, as shown in Eq. 1. The filtered pointcloud $\hat{P}$ is reprojected from $\hat{\mathbf{y}}$ into 3D and used for two downstream tasks: elevation mapping and traversability estimation.

Specifically, to extract footholds, a large temporal window W_B with duration t_B is constructed, and a short temporal window W_S with duration t_S is built around a point within W_B (Fig. 3a). Mean foot heights within W_S and W_B are calculated, yielding $mean_S$ and $mean_B$, and the maximum height change within W_S is denoted as r_S. A point is identified as a foothold if $mean_S \leq mean_B$ and $r_S \leq r_t$, where r_t is a threshold. This is applied iteratively for each point in W_B. After evaluating all points, W_B is moved forward, and the process is repeated. Footholds are denoted as $F_i \in \mathbb{R}^3, i \in \{1 \dots M\}$, with M the number of extracted footholds.

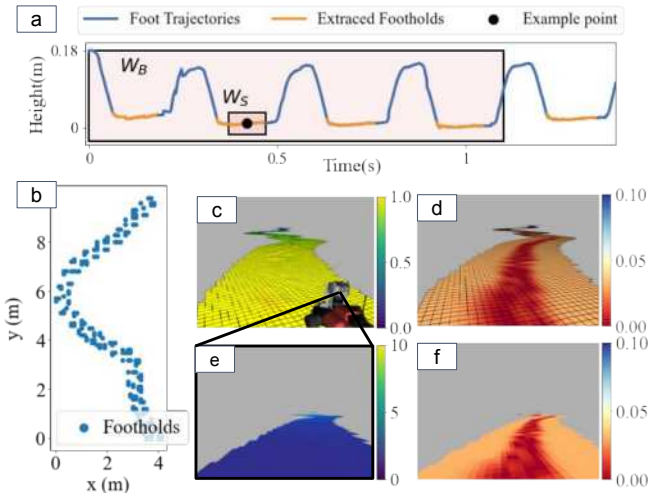

Fig. 3. The reconstruction of the support surface. (a) The method to extract footholds. (b) Top view of the sparse grid map with the extracted foothold positions. (c) Reconstructed support surface using Gaussian Processes (GP) with color-coded elevation. (d) The corresponding variance of the reconstruction. The red area is with lower variance than the orange area. (e) The sparse depth image of the reconstructed support surface with the distance from the camera focal point being color-coded. (f) The corresponding variance of e in the image plane.

2) Support surface reconstruction: We extrapolate the positions of the extracted footholds in the *world frame* to generate the support surface as done by Homberger et al. [15]. The foothold extrapolation method is valid under the assumption that the support surface is continuous and smooth. We bin the footholds into small overlapping tiles of a grid map (Fig. 3b) and model them locally as a GP with radial basis function (RBF) kernel [33]. For the overlapped regions, the height value is determined by calculating the mean GP prediction, while the variance is set to the maximum variance.

3) Support Surface Projection: To generate training labels for SPF depth estimation, we reproject the generated support surface (Sec. III-B2) into all captured camera images along the trajectory. By raycasting the generated height grid map (Fig. 3c), we obtain correspondences between the pixels on the image plane and grid map cells. Given the correspondences, we can generate a support surface depth image capturing the distance from the camera center to the support surface (Fig. 3e), as well as project the variance of the Gaussian process σ into the image plane (Fig. 3f). The variance of the Gaussian Process does not capture the uncertainty with respect to the distance from which the support surface is observed. We found experimentally that using a more sophisticated formulation incorporating the observation distance during training did not increase the performance. The variance is used to mask the region of valid depth information using a threshold parameter τ_{max}, resulting in the support surface label $\mathbf{y}_{de}$ used as a target during training.

C. Semantic Pointcloud Filter

To leverage image-based models, P is projected to the camera image plane to obtain a sparse depth image $\mathbf{x}_d \in \mathbb{R}^{m \times n}$. We use an image processing neural network to predict the filtered depth, $\hat{\mathbf{y}} = h(\mathbf{x}_{rgb}, \mathbf{x}_d)$. The filtered depth image is reprojected to the space as the filtered pointcloud $\hat{P}$.

1) Network Structure: The neural network model follows an encoder-decoder structure with skip connections, as proposed in [34]. The input to the network is a stacked tensor of the sparse depth image $\mathbf{x}_d$ and the camera image $\mathbf{x}_{rgb}$. The encoder employs a pretrained EfficientNet-B5 [35] as the backbone, where the first convolution layer is adapted to accommodate 4 input channels. Following two convolution

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

layers with 2048 output channels, the decoder consists of a sequence of 5 standard feature upsampling blocks, with nearest-neighbor upsampling blocks, each containing two groups of convolution layers.[1]

The decoder simultaneously generates the support surface depth $\hat{\mathbf{y}}_{de}$, and the binary semantic segmentation mask $\hat{\mathbf{y}}_s$ with two classes: *rigid obstacles* and *support surface*. The final output $\hat{\mathbf{y}}$ combines the support surface depth estimation $\hat{\mathbf{y}}_{de}$ with the raw pointcloud $\mathbf{x}_d$

$$\hat{\mathbf{y}} = \mathbf{x}_d \cdot [\hat{\mathbf{y}}_s == rigid] + \hat{\mathbf{y}}_{de} \cdot [\hat{\mathbf{y}}_s == support] \tag{1}$$

with the square brackets denoting the Iverson brackets, where they serve to express logical conditions, and the multiplication and addition operators represent element-wise operations.

2) Training: The network is trained to minimize the total training loss $\mathscr{L}_{\text{total}}$, given by the weighted sum of the support surface depth estimation loss $\mathscr{L}_{\text{de}}$ and semantic segmentation loss $\mathscr{L}_{\text{seg}}$:

$$\mathscr{L}_{\text{total}} = \mathscr{L}_{\text{seg}} + w\mathscr{L}_{\text{de}} \tag{2}$$

where w is used to balance the loss contributed by depth estimation and semantic segmentation. $\mathscr{L}_{\text{de}}$ of prediction $\hat{\mathbf{y}}_{de}$ is computed as the mean squared error (MSE) with respect to SSDE labels and is weighted by the pixel-wise variance σ_i of the support surface.

$$\mathscr{L}_{\text{de}} = \sum_{i \in C_{\text{ss}}} \sigma_i^{-2} \left(\hat{\mathbf{y}}_{de,i} - \mathbf{y}_{de,i}\right)^2 \tag{3}$$

C_{ss} is set of pixels where $\mathbf{y}_{de}$ has valid values. Meanwhile, $\mathscr{L}_{\text{seg}}$ is computed by employing the cross-entropy loss with respect to SSSeg labels, which is commonly used for semantic segmentation tasks.

IV. IMPLEMENTATION DETAILS

A. SSDE Generation

When generating the SSDE labels, we set the length of W_B to 1.5 s, the length of S_B to 0.07 s, and r_t to 0.015 m. In the RBF kernel, l is set to $1e^{-5}$. For the depth image of the reconstructed terrain, we only use the region with a variance lower than $\tau_{max} = 0.03$ as our SSDE labels.

B. Network Training

The neural network is trained for 9000 steps ($\sim$1 hour) using a batch size of 6 on a single NVIDIA GeForce RTX3090. To train the network, we use the AdamW optimizer [36] and schedule the learning rate following [37]. Setting w to 0.02 achieves the best performance and best trades off the semantic segmentation and depth estimation performance. During training for data augmentations, we first flip $\mathbf{x}_{rgb}$ and $\mathbf{x}_d$ horizontally with a probability of 0.5. Secondly, the input images are randomly cropped from a shape of 540 $\times$ 720 to 352 $\times$ 704. The output depth estimation $\hat{y}_{de}$ and segmentation mask $\hat{y}_s$ are of the same size as the input image.

Fig. 4. Example images from different environments.

V. EXPERIMENTS

A. Dataset Overview

The data is collected using the legged robot ANYmal, controlled by a human operator, in various outdoor environments in Perugia, Italy and Hönggerberg, Switzerland. ANYmal is equipped with a Robosense RS-Bpearl 32 beam Dome-LiDAR and an Alphasense Core 0.4 MPix RGB camera unit. The camera images and LiDAR data are collected at a rate of 10 Hz, and downsampled during post-processing to 2 Hz, with image resolution at 540 $\times$ 720. We recorded six 10-20 minutes trajectories from Perugia, Italy, classified into three environments: Grassland, Hillside, and Forest. Each environment contains two trajectories.

The training set and testing set each contain three trajectories, one from each of these three environments. For each trajectory, We sampled the images at a 3.5-second interval to avoid repetition (one out of every 7 images). Due to the different lengths of the trajectories, the training set contains 464 images, and the testing set contains 304 images. Each image is paired with the latest pointcloud by their hardware timestamp and is compensated for the ego-motion using the precise state estimation [31], [32]. The images are manually annotated into two classes: *support surface* and *rigid obstacles* for semantic segmentation. The remaining 6/7 of the images from the three training trajectories comprise the validation set. The model is tuned by considering the performance of the downstream applications on the entire training sequence, which combines the training and validation sets. The following results are all reported on the testing set. Additionally, a trajectory was collected from Hönggerberg, Switzerland, which stands in contrast to the wild setting of other trajectories. We labeled 50 images from this trajectory to test the model's robustness to domain shift. Notably, although the SPF is trained with offline data, it could be deployed with a frequency of 6-7 Hz on a Jetson Orin GPU on the ANYmal robot. Examples of images from the onboard camera are shown for the different environments in Fig. 4. The dataset is made publicly available.

B. Support Surface Depth Estimation

The result presented in Fig. 5 shows the distribution of the absolute errors of pointcloud depth in relation to the distance. These errors are computed with respect to the SSDE

[1]The detailed structure of the network can be accessed in the public code. https://sites.google.com/leggedrobotics.com/semantic-pointcloud-filter

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

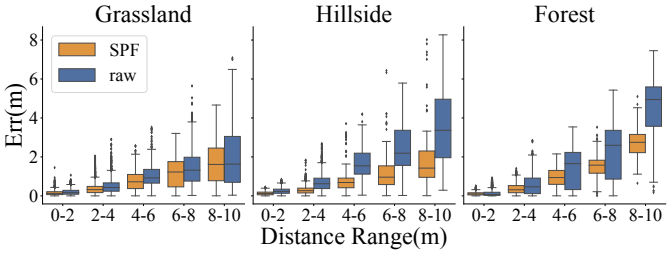

Fig. 5. Comparison of the absolute error of the depth estimation with respect to the distance in the different environments in the testing set.

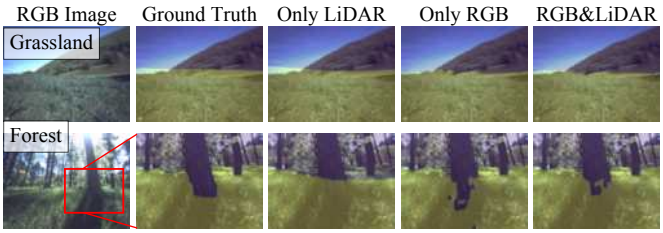

Fig. 6. The impact of varying modality input on semantic segmentation. In a relatively simple environment with no rigid obstacles, the segmentation performance of all examined models appears to be comparable (1st row). However, in more complex environments (2nd row), the model relying solely on pointcloud input struggles to accurately segment the *rigid obstacles* near the ground (the trunk), while the model utilizing only RGB images is adversely impacted by the tree's shadow.

labels from the camera and are binned by the distance. We compare the absolute error of the raw pointcloud and the multi-modal SPF-filtered pointcloud. The result demonstrates that the performance of the raw pointcloud worsens from Grassland to Hillside, and further in Forest environments. The filtered pointcloud also exhibits this trend, indicating that the quality of the raw pointcloud directly affects the performance of our filtered pointcloud. Notably, in all environments, within any distance bins, the filtered pointcloud outperforms the raw pointcloud. The error increases with respect to the distance, which is common in monocular depth estimation tasks. Additionally, the reconstructed support surface, while being locally consistent in proximity to the robot, becomes less accurate at longer ranges due to localization drift.

C. Ablation Study

The first ablation assesses the impact of providing varying input modalities, *Only LiDAR* refers to a model solely employing LiDAR pointcloud as input, *Only RGB* indicates using solely the RGB image, *RGB&LiDAR* involves utilizing both RGB&LiDAR image as model inputs. The second investigates the use of hand-labeled SSSeg labels in the training of the semantic segmentation task. For rigors analysis, we train all models 10 times with varying random seeds and report the standard deviation.

We evaluate depth estimation and semantic segmentation separately for each model. To measure the performance of depth estimation, we use the following metrics:

1) Average Relative Error (REL): $\frac{1}{n}\sum_p^n \frac{|y_p - \hat{y}_p|}{y}$

2) Root Mean Squared Error (RMSE): $\sqrt{\frac{1}{n}\sum_p^n (y_p - \hat{y}_p)^2}$

where all metrics are commonly used to evaluate monocular depth estimation performance [37]. The metrics are evaluated with respect to all the validation pixels of the SSDE label within 5m. The result is shown in Table I.

We present an ablation comparison for support surface depth estimation in Table I. Our multi-modal model (*RGB&LiDAR*) and single-modal model (*Only RGB* and *Only LiDAR*) outperform the raw pointcloud. This observation indicates our models' reliability across all trained environments, even when provided with *Only RGB* or *Only LiDAR* input. Table 1 indicates advantages for both RGB and LiDAR modalities under different conditions. We hypothesise RGB images offer clearer features within the forest environment, aiding distinction from grass. LiDAR is more robust in scenarios with a large variety of objects, such as hillsides, where we hypothesize that RGB information is more susceptive to domain shift. Performance improves on average by combining these modalities into a multi-modal model. This multi-modal model is remaining within 0.02 m of the best ablations across scenarios. Conversely, single-modal models may experience over 20% RMSE increase compared to multi-modal models in less favorable environments, emphasizing the multi-modal design's enhanced robustness.

The REL metric, being more sensitive to near-object errors, reveals the *Only LiDAR* model's efficiency in depth estimation close to the robot. We think this can be attributed to the fact that the sparse LiDAR pointcloud can directly measure the ground surface in sparse vegetation in the proximity of the robot. For more distant assessments, the ground surface is occluded by vegetation and less likely to be measured by LiDAR, where RGB can be beneficial.

We evaluate the semantic segmentation of ours and the ablated models with the mean intersection of union (mIoU) metric, which is a widely adopted measurement in the field of semantic segmentation tasks [38]. We show the results for two exemplary inputs in Fig. 6, and provide numerical mIoU values in Table II. Surprisingly, our results in Table. II indicate that the multi-modal model, as well as the single-modality models, demonstrate comparable performance in mIoU. This can be attributed to two factors. First, as illustrated in Fig. 6, the majority of the image is relatively straightforward to segment. The key challenge lies in accurately determining the boundaries of the support surface, which represents only a minor portion in comparison to the entire label. This causes the mIoU to be less sensitive to the segmentation of the area of interest. Secondly, the result demonstrates that semantic information is derived from both the RGB image's visual data and the LiDAR point cloud. While RGB images provide more explicit semantic information than pointcloud, they can suffer from poor lighting conditions (as demonstrated in Fig. 6), making it difficult to generalize on our small dataset. Consequently, incorporating geometric information into our model can potentially enhance its robustness. Considering all three environments, our ablation shows that multi-modal and single-modal perform on par for semantic segmentation.

We also investigate the necessity of manually labeled seg-

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

TABLE I
EVALUATION OF SUPPORT SURFACE DEPTH ESTIMATION.

	Grassland		Hillside		Forest		All	
Method	RMSE↓	REL↓	RMSE↓	REL↓	RMSE↓	REL↓	RMSE↓	REL↓
Raw Pointcloud	0.503	0.15	0.710	0.20	0.656	0.15	0.63	0.17
Only RGB	0.46 ± 0.03	0.15 ± 0.06	0.37 ± 0.03	0.11 ± 0.04	**0.35 ± 0.03**	**0.09 ± 0.02**	0.40 ± 0.03	0.12 ± 0.03
Only LiDAR	0.45 ± 0.03	0.11 ± 0.01	**0.29 ± 0.02**	**0.08 ± 0.01**	0.46 ± 0.04	0.10 ± 0.01	0.41 ± 0.01	0.10 ± 0.03
RGB&LiDAR	**0.37 ± 0.02**	**0.11 ± 0.01**	0.31 ± 0.04	0.10 ± 0.01	0.37 ± 0.02	0.10 ± 0.01	**0.35 ± 0.01**	**0.10 ± 0.01**

TABLE II
SUPPORT SURFACE SEMANTIC SEGMENTATION IN MIOU.

	Grassland	Hillside	Forest	All
No hand-label	43.93±0.93	48.61±0.96	50.26±1.19	45.99±0.68
Only RGB	89.12±1.33	**83.71±2.67**	96.90±0.85	89.02±0.72
Only LiDAR	**90.77±0.21**	83.28±0.89	96.50±0.07	89.14±0.33
RGB&LiDAR	90.46±0.35	82.83±0.92	**97.29±0.17**	**89.42±0.38**

TABLE III
EVALUATION OF DOMAIN SHIFT IN HÖNGGERBERG

Method	RMSE↓	REL↓	mIoU↑
Raw Pointcloud	0.344	0.086	
Only RGB	0.537±0.04	0.173±0.01	91.88±0.95
Only LiDAR	**0.209±0.02**	**0.070±0.01**	92.85±1.92
RGB&LiDAR	0.307±0.03	0.108± 0.01	**93.55±0.82**

mentation labels by comparing it to solely employing labels generated based on the valid regions of the SSDE label C_{ss} and considering other regions as rigid obstacles. This label does not reflect the scene well given that all untraversed regions are regarded as rigid obstacles. As shown in Table II, the models with hand-labeled data significantly outperforms the model trained without hand-labels.

D. Domain Shift

We further test our model in the Hönggerberg dataset, which contains more structured obstacles like buildings and railings. The result is shown in Table. III. Notably, in the urban setting, the *Only Lidar* model performs best in the depth estimation. The reduced accuracy of *Only RGB* model arises from visual discrepancies in the RGB Image due to unfamiliar obstacles. Thus the depth estimation from RGB images is not reliable anymore. In contrast, the geometric information conveyed by LiDAR point clouds remains relatively stable. However, in semantic segmentation, the multi-modal input still has the best performance, indicating that, in our data setting, the semantic segmentation is more robust to the domain shift than the depth estimation. For the paper consistency and model performance on the trained environments, we choose *LiDAR&RGB* for the testing of downstream applications.

E. Elevation Mapping

To validate SPF's practical benefits, we input the filtered pointcloud into an elevation mapping algorithm [3], which fuses the pointcloud with the robot pose to generate an $8\,\text{m}\times 8\,\text{m}$ grid map at $0.04\,\text{m}$ resolution. We compare our method with three baselines: *Raw* generates maps using the same elevation mapping algorithm but inputs the raw pointcloud. *Smooth* involves applying a smooth operation [3] on the elevation map generated from *Raw*. For these two baselines, we only feed the points within the camera's field of view (FoV) into the algorithm. An illustration of FoV is presented in Fig. 8. Moreover, *Foothold GP* refers to the GP completion from the footholds where the robot traversed. We use the same kernel as the one in support surface reconstruction. This baseline exemplifies methods relying solely on proprioceptive observation.

The result in Fig. 7 shows our method and two baseline methods applied to three elevation mapping samples. In each image, we illustrate two robots (past and future), which allow us to validate the elevation mapping accuracy by examining the foothold alignment with the estimated terrain surface. Sample (a) features high grass. The elevation map constructed with our SPF matches the robot's foot height, while the maps from the two baselines are higher than the robot's leg. Smoothing the raw elevation map reduces its root-mean-square error (RMSE) by 12.6%, while our method reduces it by 56.0%. Sample (b), set in a forest with minimal grass, also demonstrates our method's superior performance in representing support surfaces and recognizing obstacles. Sample (c), taken from Hönggerberg Zurich, highlights a domain shift from nature to an urban setting. Similar to (a), our method reduces the RMSE by 42.0% compared to the raw elevation map and accurately captures the fence's structure. These comparisons demonstrate that our SPF effectively and robustly distinguishes vegetation from rigid obstacles, correctly lowering the perceived vegetation height while preserving the structure of rigid obstacles.

For a quantitative comparison over the testing trajectory in Perugia Grassland, we compare the constructed elevation map with the SSDE labels described in Sec. III-B. Specifically, elevation maps are generated at a rate of 10 Hz as the trajectory is replayed. On each update of the maps, an error is computed between the new map and the SSDE labels. Then we transform these error maps into the robot's frame, accumulate them, and compute the RMSE and error variance, as depicted in Fig. 8. In the testing trajectory, the robot was mostly walking forward with some turning motions. (The forward direction is shown in the bottom right of Fig. 8.) Moreover, the elevation perceived in the front goes towards the back of the robot's frame as the robot moves forward. Therefore, the upper parts (the parts in the FoV) of each subfigure correspond to the "prediction"

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

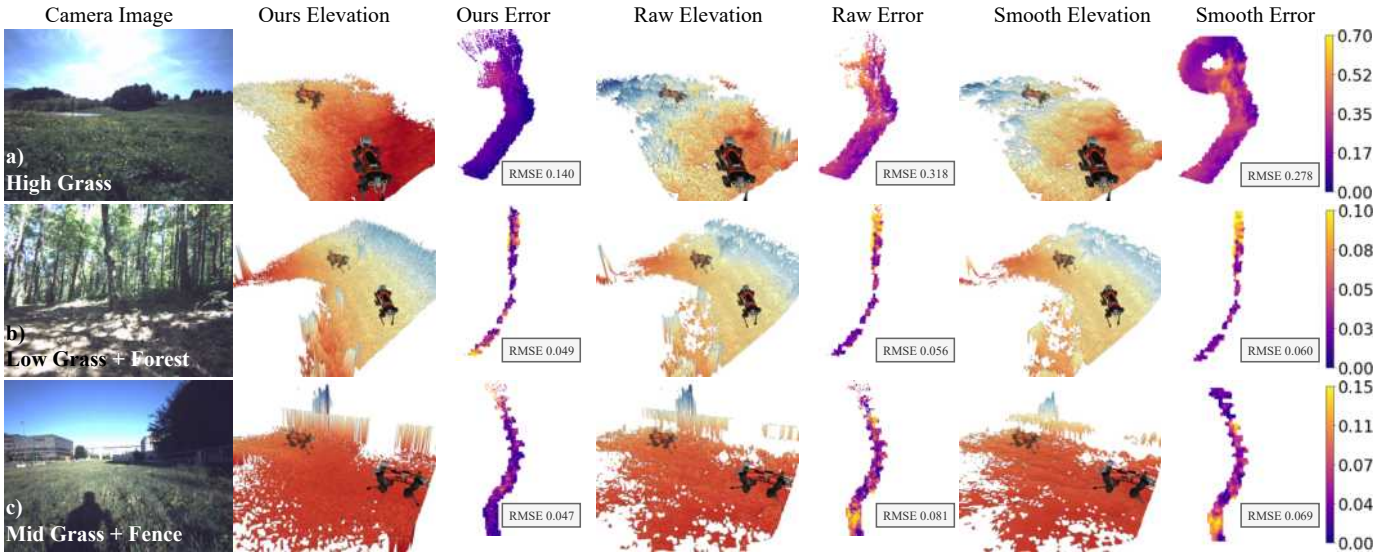

Fig. 7. Comparison of elevation mapping generated from different methods in different environments. On the leftmost side are the onboard camera images. For each method, we provide the generated elevation map and the error map (RMSE) from a top-down perspective with respect to the support surface for this region of the trajectory. We compare our method (*Ours*) to the raw pointcloud (*Raw*) and the raw pointcloud with additional smoothing (*Smooth*). The colorbars for error maps are placed on the right, with brighter colors indicating higher error. Our method outperforms the baselines in various environments, such as high grass hills (a), forests (b), and urban grasslands with impenetrable fences (c).

of elevation height, while the lower parts correspond to the mapping result fused over several frames.

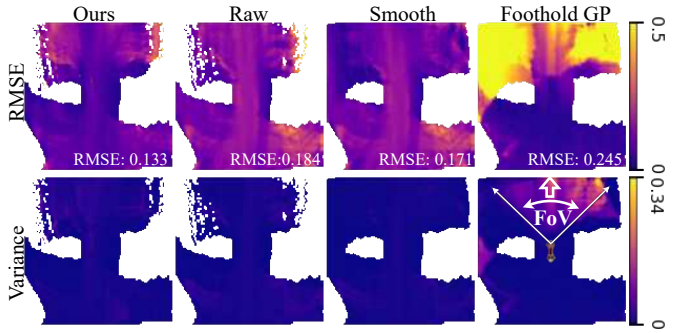

Fig. 8. The errors between elevation maps generated using our method and baselines. In the first row, the value of each pixel is the RMSE of that pixel over the testing trajectory, expressed in meters. The variances (second row) of the errors are also provided. These figures are accumulated in the robot frame. The forward direction and the field of view (FoV) of the robot are shown in the bottom right figure. White areas indicated that no errors are collected in those cells).

The elevation map constructed with our filtered pointcloud has an RMSE of 0.133 m, which is about 48% less than the 0.183 m of the raw baseline. Smoothing results in an RMSE of 0.170 m, which is only a 7% improvement. Our method has a lower error at both the front and back of the robot. The high error value on the top left is exceptional and is caused by overexposure when the robot leaves tree shades. As the footholds are exact samples of the support surface, foothold GP achieves almost zero RMSE in the areas under the robot and behind the robot. However, it predicts the height of the untraversed area in the front much worse than the methods with exteroceptive observations. The standard deviations of the errors are shown in the second row. The error from the smooth baseline is more stable, resulting in a lower standard deviation.

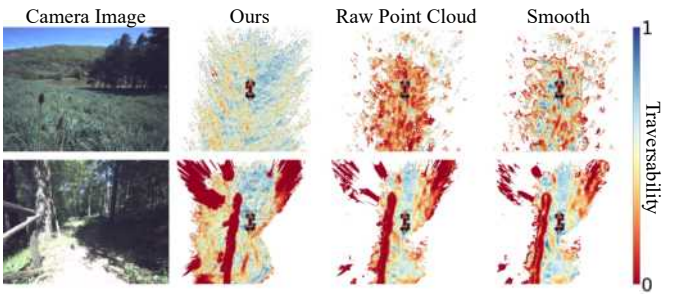

Fig. 9. Comparison of traversability estimation using the raw, smoothed, and our filtered pointcloud. The traversability is color-coded, where blue indicates traversable and red untraversable. Our method correctly predicts the traversability in the meadow and forest environment. Using the raw or smoothed pointcloud prohibits motion planning in the meadow.

Both our method and the raw pointcloud have comparable standard deviations for their errors. Please note that, thanks to the modularity design of our SPF, it is possible to leverage both pointcloud filtering and smoothing to obtain predictions that are more accurate and stable.

F. Traversability Estimation

Filtering out noisy high grass and revealing true support terrain is beneficial for an accurate traversability estimation. We estimate the traversability using the method described in [39]. Elevation maps constructed in Sec. V-E are used as the input to this algorithm. Two samples of traversability estimation in the natural environment are shown in Fig. 9. The grid maps on the right rows are colored to represent traversability, where high traversability corresponds to blue, and low traversability corresponds to red. When the robot is in the vegetation, our method recognizes vegetation and estimates the underlying support surface leading to the correct traversability estimate.

On the other hand, directly using the pointcloud or the smoothed version leads to incorrect traversability rendering motion planning and navigation impossible. When the robot is in the forest, our method and baselines all yield reasonable traversability maps.

VI. CONCLUSION

In this work, we present our semantic pointcloud filter (SPF) trained in a semi-self-supervised manner, which can accurately adjust the pointcloud to the support surface on a vegetation-occluded terrain, leveraging semantics from the camera images and pointcloud. Using pointcloud filtered by SPF, we improved elevation mapping and traversability estimation performance compared to existing baseline methods, potentially increasing autonomy for a variety of robotic systems within natural environments.

While multiple real-world experiments demonstrated the efficacy of the proposed method, limitations exist, such as errors in filtered depth estimation under adverse lighting conditions and the possibility of inaccuracy due to domain shifts in vegetation species and seasons. Future research on integrating anomaly detection into SPF can allow us to estimate the reliability of the predicted support surfaces and address safety challenges induced by a learned module.

REFERENCES

[1] E. Marder-Eppstein *et al.*, "The office marathon: Robust navigation in an indoor office environment," in *International Conference on Robotics and Automation*, 2010.

[2] S. Macenski, F. Martin, R. White, and J. Ginés Clavero, "The marathon 2: A navigation system," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[3] Miki *et al.*, "Elevation mapping for locomotion and navigation using gpu," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 2273–2280.

[4] P. Fankhauser, M. Bloesch, and M. Hutter, "Probabilistic terrain mapping for mobile robots with uncertain localization," *IEEE Robotics Autom. Lett.*, vol. 3, no. 4, pp. 3019–3026, 2018.

[5] H. Oleynikova *et al.*, "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[6] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, p. eabk2822, 2022.

[7] Jenelten *et al.*, "Tamols: Terrain-aware motion optimization for legged systems," *IEEE Transactions on Robotics*, 2022.

[8] J. Frey, D. Hoeller, S. Khattak, and M. Hutter, "Locomotion policy guided traversability learning using volumetric representations of complex environments," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022.

[9] J. Frey, M. Mattamala, N. Chebrolu, C. Cadena, M. Fallon, and M. Hutter, "Fast Traversability Estimation for Wild Visual Navigation," in *Proceedings of Robotics: Science and Systems*, 2023.

[10] H. Oleynikova, Z. Taylor, R. Siegwart, and J. Nieto, "Safe local exploration for replanning in cluttered unknown environments for micro-aerial vehicles," *IEEE Robotics and Automation Letters*, 2018.

[11] T. Dang *et al.*, "Graph-based subterranean exploration path planning using aerial and legged robots," *Journal of Field Robotics*, vol. 37, no. 8, pp. 1363–1388, 2020, wiley Online Library.

[12] P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, and R. Siegwart, "Robot-centric elevation mapping with uncertainty estimates," in *Mobile Service Robotics*. World Scientific, 2014, pp. 433–440.

[13] J. Z. Kolter, Y. Kim, and A. Y. Ng, "Stereo vision and terrain modeling for quadruped robots," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 1557–1564.

[14] C. K. Wellington, A. C. Courville, and A. Stentz, "A generative model of terrain for autonomous navigation in vegetation," *The International Journal of Robotics Research*, vol. 25, pp. 1287 – 1304, 2006.

[15] T. Homberger, L. Wellhausen, P. Fankhauser, and M. Hutter, "Support surface estimation for legged robots," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8470–8476.

[16] V. Šalanský *et al.*, "Pose consistency kkt-loss for weakly supervised learning of robot-terrain interaction model," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5477–5484, 2021.

[17] C. Wellington and A. Stentz, "Learning predictions of the load-bearing surface for autonomous rough-terrain navigation in vegetation," *Field and Service Robotics: Recent Advances in Reserch and Applications*, pp. 83–92, 2006.

[18] I. Laina *et al.*, "Deeper depth prediction with fully convolutional residual networks," in *2016 Fourth International Conference on 3D Vision (3DV)*, 2016, pp. 239–248.

[19] B. Li, Y. Dai, and M. He, "Monocular depth estimation with hierarchical fusion of dilated cnns and soft-weighted-sum inference," *Pattern Recognition*, vol. 83, pp. 328–339, 2018.

[20] H. Zhou, D. Greenwood, and S. Taylor, "Self-supervised monocular depth estimation with internal feature fusion," in *32nd British Machine Vision Conference 2021, BMVC 2021*, p. 378.

[21] V. R. Kumar *et al.*, "Syndistnet: Self-supervised monocular fisheye camera distance estimation synergized with semantic segmentation for autonomous driving," *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 61–71, 2021.

[22] Uhrig *et al.*, "Sparsity invariant cnns," in *2017 international conference on 3D Vision (3DV)*. IEEE, 2017, pp. 11–20.

[23] F. Ma *et al.*, "Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3288–3295.

[24] Choi *et al.*, "Selfdeco: Self-supervised monocular depth completion in challenging indoor environments," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 467–474.

[25] Valada *et al.*, "Deep auxiliary learning for visual localization and odometry," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 6939–6946.

[26] B. Kuang *et al.*, "Semantic terrain segmentation in the navigation vision of planetary rovers—a systematic literature review," *Sensors*, vol. 22, no. 21, p. 8393, 2022.

[27] G. Rizzoli, F. Barbato, and P. Zanuttigh, "Multimodal semantic segmentation in autonomous driving: A review of current approaches and future perspectives," *Technologies*, vol. 10, no. 4, 2022.

[28] Dolz *et al.*, "Hyperdense-net: A hyper-densely connected cnn for multi-modal image segmentation," vol. 38, no. 5, pp. 1116–1126, 2019.

[29] Zhang *et al.*, "An easy-to-use airborne lidar data filtering method based on cloth simulation," *Remote Sensing*, vol. 8, no. 6, 2016. [Online]. Available: https://www.mdpi.com/2072-4292/8/6/501

[30] T. J. Pingel *et al.*, "An improved simple morphological filter for the terrain classification of airborne lidar data," *Isprs Journal of Photogrammetry and Remote Sensing*, vol. 77, pp. 21–30, 2013.

[31] Bloesch *et al.*, "State estimation for legged robots-consistent fusion of leg kinematics and imu," *Robotics*, vol. 17, pp. 17–24, 2013.

[32] Khattak *et al.*, "Complementary multi–modal sensor fusion for resilient robot pose estimation in subterranean environments," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2020, pp. 1024–1029.

[33] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning.*, ser. Adaptive computation and machine learning. MIT Press, 2006.

[34] Ronneberger *et al.*, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Navab *et al.*, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.

[35] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[36] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[37] S. Farooq Bhat, I. Alhashim, and P. Wonka, "Adabins: Depth estimation using adaptive bins," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 4008–4017.

[38] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[39] B. Yang *et al.*, "Real-time optimal navigation planning using learned motion costs," in *IEEE International Conference on Robotics and Automation, ICRA*. IEEE, 2021, pp. 9283–9289.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.