Overcoming Nature: Perception for Autonomous Navigation in Dense Vegetation*

Lukas Wimmer[1], Andre Koczka[2], and Gerald Steinbauer-Wagner[2]

***Abstract*— Autonomous navigation in densely vegetated off-road environments remains challenging because conventional geometric perception often treats traversable vegetation as non-traversable obstacles. In this work, we present a modular semantic–geometric perception pipeline for vegetation-aware navigation. The approach combines camera-based semantic data with LiDAR to generate a local grid map containing geometric and semantic information. A subsequent filtering stage uses this representation to correct vegetation-induced artifacts in standard elevation maps while preserving rigid obstacles for navigation. The system is designed to be portable across multiple robot platforms and sensor configurations. The pipeline was evaluated in challenging alpine off-road environments on three robot platforms, indicating improved distinction between traversable vegetation and solid obstacles and supporting more reliable navigation in dense natural environments.**

I. Introduction and Motivation

Navigation in challenging, unstructured environments has seen significant progress in the last decades [1], [2], [3]. However, navigating in dense vegetation remains a challenging problem. Forest trails, alpine terrain, and overgrown vegetation contain a mixture of obstacles such as trees, rocks, bushes, and tall grass. While some of these objects must be avoided, others are physically traversable, creating a fundamental ambiguity for perception systems. Recent work has explored learning-based approaches that estimate terrain traversability directly from multi-modal sensory inputs [4], [3]. While these methods can capture complex relationships between geometry and semantics, they often require large training datasets and are difficult to adapt across different robot platforms and sensor configurations. In this work we explore a hybrid perception approach that combines geometric terrain modeling with semantic information obtained from image segmentation. Instead of predicting traversability directly through end-to-end learning, semantic information is used to refine and correct geometric terrain representations, enabling the robot to distinguish between traversable vegetation and solid obstacles.

The proposed system integrates camera-based semantic segmentation with LiDAR measurements to generate a semantic elevation map that can be directly used by a navigation stack. The contributions of this work can be summarized as follows:

- A semantic grid mapping method with distinguished perception sectors
- A probabilistic semantic-obstacle fusion scheme
- A solution that corrects vegetation-induced disturbances and overhanging-branch artifacts in elevation maps.
- Implementation and evaluation on different robot systems

The key contribution is a sector-based semantic grid representation and map-merging strategy that corrects standard elevation maps in dense vegetation.

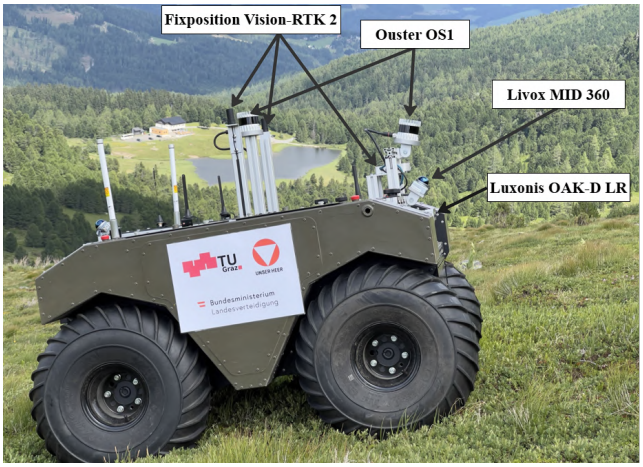


Fig. 1. Warthog and its sensor setup shown on the mountain in the Seetaler Alps.

II. Related Work

Reliable navigation in unstructured off-road environments requires perception systems that can deal with both terrain geometry and semantics. Prior work relevant to this problem can broadly be grouped into three areas: semantic segmentation for outdoor perception, datasets for off-road scene understanding, and learning-based traversability estimation.

A. Semantic segmentation

Semantic segmentation has become a key component for understanding the environment in robotics. Encoder–decoder convolutional neural networks such as U-Net [5] have been widely used due to their ability to preserve spatial detail through skip connections. More recent architectures such as DeepLabV3 [6] improve contextual reasoning using dilated convolutions and spatial pyramid pooling. In outdoor

*This work was funded by the Austrian defense research program FORTE of the Federal Ministry of Finance (BMF) under the project PATH.

[1]wimmer1luk@gmail.com,
[2]{akoczka,gerald.steinbauer-wagner}@tugraz.at, Institute of Software Engineering and Artificial Intelligence, Graz University of Technology, Graz, Austria.

robotics, segmentation has also been extended to multi-modal perception systems. For example, AdapNet [7] introduces adaptive fusion of different sensor modalities to improve robustness under changing environmental conditions. Transformer-based models such as Mask2Former [8] further improve segmentation accuracy by modeling long-range dependencies and have recently achieved strong performance in complex natural scenes. This is directly relevant to our work, since semantic segmentation provides the object-level scene understanding needed to distinguish traversable vegetation from rigid obstacles before fusion with LiDAR geometry.

B. Training data

The availability of suitable training data is critical for training semantic perception in natural environments. Early datasets such as Freiburg Forest [9] provided multimodal data for forested terrain but remain relatively small. Larger datasets such as RUGD [10] expanded coverage to diverse outdoor terrain types and improved training opportunities for modern segmentation models. In more recent work, datasets such as WildScenes [11] provide large-scale annotated data collected in natural environments and include both image and LiDAR information, enabling combined geometric and semantic perception. These datasets are relevant because the quality of vegetation-aware obstacle reasoning strongly depends on how well semantic models generalize to natural off-road scenes.

C. Traversability estimation

Several works investigate traversability estimation for off-road navigation. Learning-based approaches such as TerraPN [4] use self-supervised learning to infer terrain cost from the robot's interaction with the environment. Other methods integrate semantic information with geometric terrain representations to improve navigation in rough terrain. For example, Lee et al. [3] combine elevation maps with semantic terrain classes and model uncertainty using Gaussian processes for safer motion planning. While these methods provide powerful terrain understanding, they typically require extensive training data or complex learning pipelines. These works motivate our approach by showing the value of combining semantics and geometry, but unlike end-to-end traversability prediction, our method explicitly refines a geometric terrain representation.

In this work we explore a hybrid perception approach that combines LiDAR-based terrain mapping with semantic information obtained from image segmentation. Instead of learning traversability directly, semantic cues are used to refine geometric environment representations, enabling the robot to distinguish between traversable vegetation and solid obstacles in densely vegetated environments.

III. Pipeline overview

This work proposes a system that constructs a semantically informed local map for off-road navigation by combining camera-based semantic perception with LiDAR-based terrain mapping. This section provides an overview of the proposed pipeline.

First, RGB images are processed by a semantic segmentation model to obtain pixel-wise class predictions and confidence estimates. These semantic predictions are then associated with LiDAR measurements through an image–LiDAR fusion stage, resulting in a semantic point cloud where each valid 3D point is associated to a semantic class label and confidence value. Next, the raw LiDAR measurements are used to estimate a ground elevation layer in a local robot-centered grid map. This layer provides an approximation of the terrain surface and is smoothed to reduce the influence of outliers and vegetation-induced disturbances. Based on this ground estimate, the vertical space above the terrain is partitioned into three regions:

- Ground sector, representing terrain and low vegetation that can typically be traversed
- Obstacle sector, containing objects that may obstruct the robot
- Sky sector, representing structures above the robot height such as overhanging branches

Using the semantic point cloud together with the sector decomposition, semantic observations are accumulated in the grid map to estimate ground and obstacle classes for each cell. These class estimates are fused over time using a probabilistic log-odds formulation, which improves robustness against noisy single-frame predictions.

While this branch estimates the semantic class of each cell, a second branch uses LiDAR measurements to update the geometric occupancy of the environment. In particular, LiDAR measurements are used to update probabilistic obstacle and sky layers. The obstacle layer combines geometric measurements with semantic obstacle classes to determine whether a cell contains a rigid obstacle, while the sky layer captures structures above the robot height, such as overhanging vegetation.

Finally, the resulting semantic grid representation is combined with a conventional elevation map to produce the map suitable for off-road navigation. Traversable vegetation can therefore be treated as passable terrain, while rigid objects remain represented as obstacles. This results in a local environment model that is both geometrically consistent and semantically aware, allowing the robot to navigate through dense vegetation more effectively and traverse through vegetation where pure geometry-based elevation mapping would not provide a passable map. The flow of the entire pipeline is shown in Figure 4.

IV. Methodology

In this Section, we describe the key algorithmic components of the proposed perception framework. As outlined in Section III, the pipeline consists of semantic image processing, image–LiDAR fusion, semantic grid mapping, and final elevation-map correction.

A. Image-LiDAR Semantic Fusion

The first step combines semantic information from the camera with LiDAR data. The segmentation module produces a pixel-wise semantic prediction along with a confidence value for each pixel. LiDAR points are transformed from the LiDAR frame into the camera frame according to

$$p_{\text{cam}} = T_{\text{lidar}\to\text{cam}} \cdot p_{\text{lidar}}$$

where, $T_{\text{lidar}\to\text{cam}}$ denotes the rigid transformation between the LiDAR and camera frames. Before projection into the image plane, points outside the camera field of view are removed using the angular constraints

$$\theta_{\text{horizontal}} = \arctan 2(x_{\text{cam}}, z_{\text{cam}}),$$

$$\theta_{\text{vertical}} = \arctan 2(y_{\text{cam}}, z_{\text{cam}}),$$

where $\theta_{\text{horizontal}}$ and θ_{vertical} denote the horizontal and vertical viewing angles of the point relative to the optical axis of the camera. x_{cam}, y_{cam} and z_{cam} are the coordinates of the point in the camera frame. If either angle exceeds half of the corresponding camera field of view, the point is discarded. The remaining points are then projected into the image plane using the intrinsics of the camera. Each point is associated with the semantic class and confidence value of the corresponding image pixel. Points with low semantic confidence are rejected. The result of this step is a semantic point cloud, in which each relevant 3D LiDAR point is enriched with a semantic label and a confidence score.

B. Ground Elevation Estimation

To reason about objects relative to the terrain, a grid-based ground elevation layer is created using the LiDAR point cloud. Each grid cell stores an estimate of the ground height. Let $\hat{h}^-$ denote the previous height estimate of a cell, q a LiDAR point falling into that cell, and $h(q)$ the height along the z-axis of that point in the map frame. The updated ground estimate is computed using the minimum-height rule

$$\hat{h}^+ = \min(\hat{h}^-, h(q))$$

where $\hat{h}^+$ is the new ground height estimate after incorporating the current measurement. To reduce the effect of measurement noise and outliers, the ground height estimate is smoothed using a mean-radius filter applied over a circular neighborhood around each grid cell. The resulting smoothed ground elevation layer is later used as the reference surface for the subsequent sector decomposition.

C. Height-Based Semantic Sectoring

Once the smoothed ground elevation layer is available, the vertical space above the terrain is divided into three sectors: (1) Ground, (2) Obstacle, and (3) Sky. For a point q with height coordinate $h(q)$, its sector assignment is determined relative to the local ground elevation of the corresponding grid cell. See an illustration of this in Figure 2. Two height thresholds are used for this decomposition. The first is the obstacle threshold, which separates low structures that are still considered traversable from objects that may obstruct the robot. The second is the robot height, which separates potential obstacles from structures above the robot, such as overhanging branches. Thus, points between the ground elevation and the obstacle threshold are assigned to the Ground sector, points between the obstacle threshold and the robot height are assigned to the Obstacle sector. The Sky sector extends from the robot height to a predefined maximum measurement height. Separate semantic class layers are maintained for the Ground and Obstacle sectors so that traversable surface classes and rigid obstacle classes can be treated differently in later stages. Since not every cell of an object is actually hit by a laser beam, the resulting layers for obstacle and sky are not very dense.

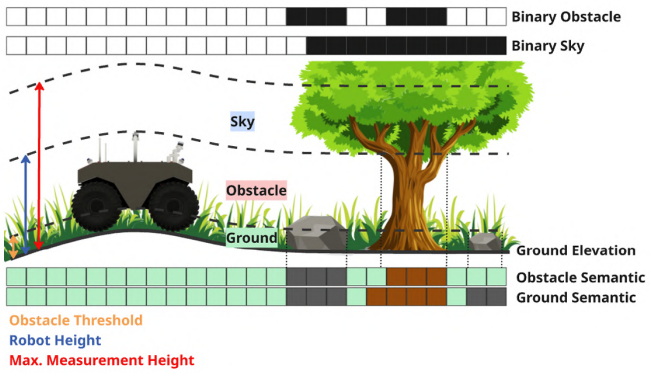


Fig. 2. Illustration of the Layers from the Semantic Grid Mapper. The Ground Elevation Layer represents the ground elevation of the environment, excluding the objects. Sectors are defined using the height thresholds: Obstacle Threshold, Robot Height, and Maximum Height. Semantic Layers represent the semantic object information of the Ground and Obstacle sectors. The Binary Obstacle Layer is created by combining the semantics from the Obstacle Semantic Layer and the LiDAR measurements.

D. Probabilistic Semantic Fusion

For each grid cell (i, j), hit counters are maintained for all semantic classes observed within the corresponding sector. Let $n_c(i, j)$ denote the number of hits belonging to class c in cell (i, j) and let C denote the set of all semantic classes. The total number of hits in the cell is defined as

$$N(i, j) = \sum_{c \in C} n_c(i, j).$$

The probability that cell (i, j) belongs to class c is computed as

$$p_{i,j}(c) = \frac{n_c(i, j)}{N(i, j)}.$$

Since this probability only represents the current observation, measurements from multiple semantic clouds are fused over time using a log-odds formulation: $L(p) = \log\left(\frac{p}{1-p}\right)$. Using this representation, the belief for class c in cell (i, j) is updated incrementally as

$$\mathcal{L}_t(i, j, c) = \mathcal{L}_{t-1}(i, j, c) + L(p_{i,j}(c))$$

where $p_{i,j}(c)$ is the probability derived from the most recent semantic cloud, t denotes the current update step.

The final class of the cell is determined by selecting the class with the highest log-odds value.

E. Probabilistic Obstacle and Sky Layers

To determine whether a grid cell contains a rigid obstacle, geometric LiDAR observations are combined with the semantic obstacle class layer in a probabilistic occupancy update. Let $L_t(i,j)$ denote the log-odds occupancy value of cell (i,j) at time step t, and $L_{t-1}(i,j)$ the previous value. Similar to the approach in the book Probabilistic Robotics [12] we employ a probabilistic approach for obstacle detection. However, since we are working in the 3D space without ray tracing, we simplify the probabilistic updates to only use the probabilities p_{hit}, indicating an obstacle was hit, and p_{miss} for every cell, independent of whether the cell was hit or not. The update is computed as

$$\mathcal{L}_t(i,j) = \mathcal{L}_{t-1}(i,j) + L(p_{\text{hit}}(i,j)) + L(p_{\text{miss}}(i,j)),$$

where p_{hit} and p_{miss} are tunable parameters. This formulation allows cells to be reinforced as occupied when obstacle evidence is present and gradually cleared when later measurements indicate free space. In parallel, a sky layer is created to represent objects located above the robot height. Unlike the obstacle layer, the sky layer does not depend on semantic obstacle classes and is derived only from the geometric LiDAR observations. This layer is later used to correct elevation-map artifacts caused by overhanging structures such as branches or foliage.

F. Final Semantic Map Representation

The semantic grid mapper produces a robot-centric map containing the smoothed ground elevation layer, semantic ground and obstacle class layers, and probabilistic obstacle and sky layers. From these layers, binary obstacle and sky representations are derived.

In the final stage, this semantic representation is transformed into a conventional elevation map. Traversable ground classes are used to replace vegetation-induced disturbances with the smoothed ground estimate, while the sky layer enables correction of elevated ground cells caused by overhanging branches or foliage (see Figure 3). The resulting local map is therefore both geometrically consistent and semantically aware. This allows the navigation stack to treat traversable vegetation as passable terrain while preserving rigid objects such as rocks or tree trunks as obstacles.

V. Implementation

A. System Architecture

The proposed system is implemented as a modular ROS 2 perception pipeline running on Ubuntu 22.04. The architecture follows the stages introduced in the methodology, namely semantic segmentation, image–LiDAR fusion, semantic grid mapping, and elevation-map merging. Each stage is realized as an independent software component, allowing the system to be adapted to different robot platforms and sensor configurations. This modular design was used to deploy the system on three off-road robot platforms with different sensor setups.

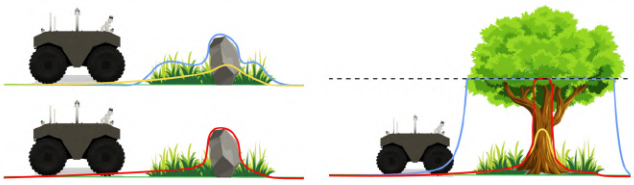


Fig. 3. Elevation Merging: (left) shows an example of the elevation merging for ground surfaces. The blue line represents the original elevation map, the yellow line represents the Ground Elevation Layer, and the red line represents the corrected elevation map; (right) shows the merging process for trees or overhanging branches. The dotted black line represents the maximum height for the sensor measurement.

TABLE I

Runtime comparison of the evaluated segmentation models for single-threaded and multithreaded execution.

Model	1 Thread		Multithreading	
	Avg. Time (ms) ↓	CPU Util. (%) ↓	Avg. Time (ms) ↓	CPU Util. (%) ↓
DeepLabV3	42.40	44.6	41.58	134.28
Mask2Former Swin-L	83.60	78.85	82.10	151.00
Mask2Former ResNet-50	48.40	51.75	47.80	142.30

B. Practical Realization

The semantic segmentation stage is implemented as a ROS 2 wrapper around pretrained segmentation models. In this work, DeepLabV3 [6], Mask2Former with Swin-L [8], and Mask2Former with ResNet-50 [8] were evaluated. To ensure comparable runtime behavior across different camera systems, incoming images are resized to a fixed resolution before inference. The output consists of a pixel-wise semantic prediction and a confidence image, which is later used to reject uncertain semantic assignments. Table I shows a comparison of average execution time of the semantic segmentation and CPU utilization.

Although Mask2Former with a Swin-L backbone is the most computationally demanding model in Table I, it was selected for all evaluation scenarios due to its consistently strong performance reported by Vidanapathirana et al. [11]. The image–LiDAR fusion stage combines segmented images with LiDAR measurements to generate a semantic point cloud. Each relevant 3D point is associated with the semantic class and confidence value of the corresponding image pixel. In addition, the implementation can publish a projection image of LiDAR points in the image plane, which is useful for sensor setup and calibration verification. For reliable projection a calibration between LiDAR and camera is done on all robotic platforms using the approach of Koide et al. [13].

The semantic point cloud and raw LiDAR measurements are integrated into a robot-centered rolling grid map [2]. This map maintains multiple layers for terrain height, semantic ground classes, semantic obstacle classes, and probabilistic obstacle and sky layer. The implementation supports a variable number of LiDAR inputs and different sensor geometries.

In the final stage, the semantic grid map is combined with a conventional elevation map [1]. This stage corrects grass-induced disturbances and overhanging branches represented

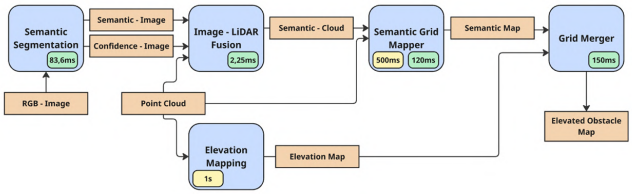


Fig. 4. Performance Overview of the System. The green boxes in the modules are the execution times. The yellow boxes represent the execution period.

Fig. 5. Example conditions on Route 2 of the autonomous navigation evaluation. The challenges of this route include the steep incline leading up the mountain, the overhanging branches, and the vegetation on the path.

as protrusions on the standard elevation map. This results in a final map which is intended for direct usage by the navigation stack. Before conducting the tests, the system has been profiled offline using ROS 2 bagfiles before field deployment. This step was important to ensure safe real-time navigation in real scenarios. For profiling a system with an NVIDIA RTX4070 Super GPU and an AMD Ryzen 9 5900X processor have been used. The average CPU utilization was measured using the pidstat command from the Linux sysstat package. Figure 4 shows the timing of each component.

The merged elevation map can be published at 1 Hz in the worst case, which is sufficient for off-road navigation, typically at 1 to 1.5 $\frac{m}{s}$.

VI. Experiments and Results

The proposed perception pipeline is evaluated at both the component level and the system level. The goal is to assess whether the semantic–geometric representation improves the separation of traversable vegetation and rigid obstacles, reduces terrain artifacts in the elevation map, and supports autonomous navigation in challenging off-road environments. As the pipeline consists of many different components, the evaluation is divided into three parts:

- Obstacle Precision Assessment (teleoperated)
- Merged Map Quality Assessment (teleoperated)
- Autonomous Navigation (fully autonomous)

The experiments were conducted on three robotic platforms with different sensor configurations in representative alpine and forest environments containing dense vegetation, uneven terrain, steep slopes, and overhanging branches.

A. Setup

The Warthog platform shown in Figure 1 was equipped with multiple LiDAR sensors, including Ouster OS1 units

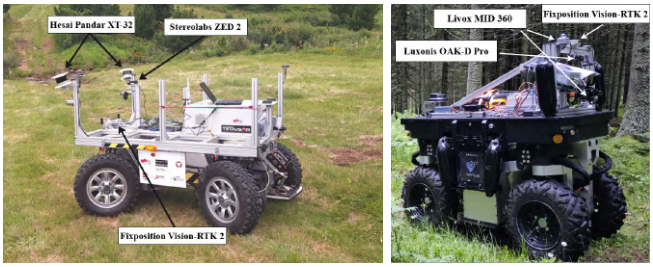


Fig. 6. The setup of Mercator (left) and Artus[1](right)

TABLE II
Results of the obstacle precision evaluation.

	Scenario 1	Scenario 2
Cell Resolution [m]	0.1	0.1
Total Area [m^2]	187	225
Ground Truth Traversable [m^2]	115.55	137.26
Total Detected Obstacles [m^2]	25.83	8.55
Intersecting Obstacles ↓ [m^2]	5.92	1.83
Intersecting ↓ [%]	22.92	21.4

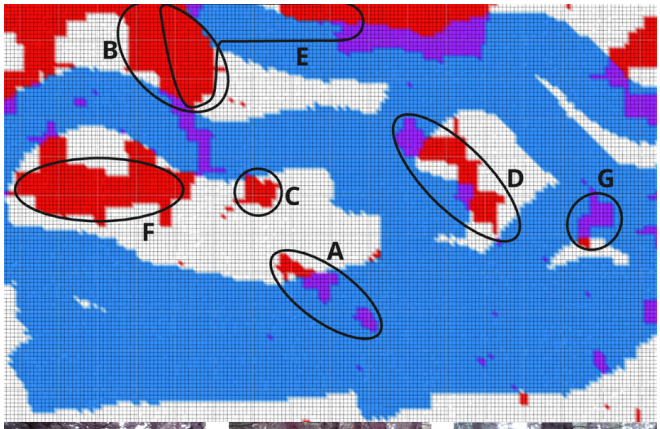
and a front-mounted Livox MID-360, a front mounted Luxonis OAK-D LR, and used an onboard PC with an Intel Core i7-13700HX, 64 GB RAM, and an NVIDIA GeForce RTX 4070 Mobile. The Mercator platform, shown in Figure 6 left, used a Stereolabs ZED camera together with two front-mounted Hesai Pandar XT-32 LiDARs and an onboard PC with an AMD Ryzen 9 3900X, 64 GB RAM, and a NVIDIA GeForce RTX 2070 SUPER Evo. The Artus platform, shown in Figure 6 right, uses a Luxonis Oak-D Pro camera and two Livox MID-360 LiDARs, one dedicated to frontal obstacle detection and one to sky-layer estimation, together with an Intel Core i7-13700HX, 64 GB RAM, and an NVIDIA GeForce RTX 4070 Mobile. All three systems use a Fixposition GNSS positioning system with RTK correction, which provides high accuracy.

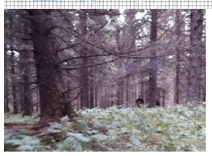
For the component-level experiments, the obstacle precision evaluation was carried out with the Warthog platform due to its strong stability in off-road terrain. The autonomous navigation evaluations were distributed across the three platforms depending on route difficulty and platform suitability.

B. Component-Level Evaluation

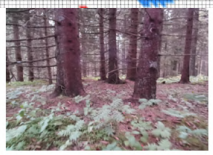
1) Obstacle precision: The first evaluation assesses how reliably the proposed obstacle representation separates true obstacles from traversable regions. For this evaluation, the robot is driven manually. The robot's footprint projected onto the grid map is used as a proxy for traversable ground truth: regions covered by the robot are considered traversable, while obstacle detections intersecting those regions are counted as false positives. Cells classified as obstacles outside the traversed region are treated as true positives. This evaluation was carried out in two forest scenarios containing trees, dense bushes, grass, and overhanging branches. Scenario 1 is shown in Figure 7 as an example.

[1]Artus robotic platform – CharismaTec og. https://charismatec.at/en/projects/





A: Tree with overhanging branch



B: Trees

C: Trees

D: Tree with overhanging branch

E: Set of trees with branches

F: Set of trees with branches

Fig. 7. Evaluation Visualization Map. Blue: Area covered by the robot, assumed to be traversable. Red: Detected obstacles. Purple: Intersection between the detected Obstacles and the robot. The sections B, C, and F show true obstacles. A and D show intersections due to overhanging branches. G marks a false positive.

Table II shows the quantitative results of both scenarios. In Scenario 1, the system detected $25.83\mathrm{m}^2$ of obstacles, of which $5.92\mathrm{m}^2$ intersected with traversable ground, corresponding to 22.92% of false positives. In Scenario 2, the detected obstacle area was smaller at $8.55\mathrm{m}^2$, with $1.83\mathrm{m}^2$ intersecting the traversable region, corresponding to 21.4% of false positives. The results indicate that the perception pipeline detects relevant obstacles reliably while keeping the proportion of obstacle detections in traversable regions at a comparable and moderate level in both scenarios. Most traversable regions remain free of persistent false obstacle detections. Remaining false positives mainly arise in two situations: first, when overhanging branches intersect the robot's height although they are correctly detected as obstacles in the obstacle sector, and second, when small vegetation structures or thin branches are captured by LiDAR but not consistently segmented by the vision model.

2) Merged Map Evaluation: The second evaluation investigates whether the semantic map improves the quality of the terrain representation after merging with the standard elevation map. This is measured by comparing the number of cells with slope greater than 45° inside the traversed area for the standard elevation map and the merged map. These slope masks are obtained by estimating surface normals and

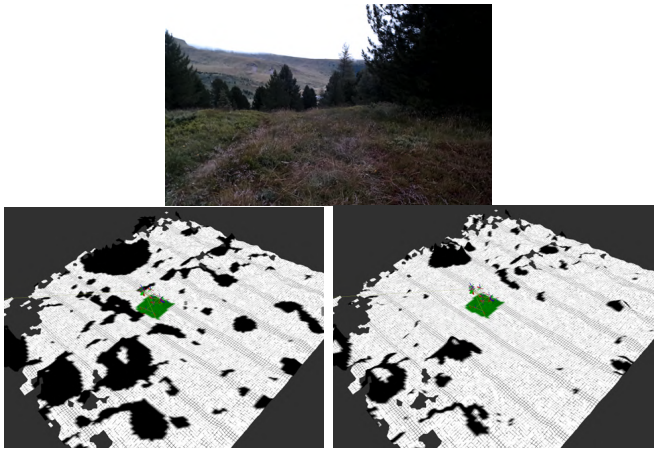


Fig. 8. Merged Map Evaluation: the upper picture shows the point of view of the robot traversing the environment; (left) shows the Masked Slope of the standard Elevation Map, which contains many marked regions due to the vegetation; (right) shows the Masked Slope of the MergedMap; Black regions in the Slope Masks represent a slope greater than 45°. The robot's footprint is marked in green.

computing the local slope for each map cell. Figure 8 shows the reduction of vegetation-induced elevation artifacts after merging. The number of traversed cells with slope greater than 45° in the merged map is reduced to approximately one tenth of the corresponding number in the standard elevation map. This indicates that replacing vegetation-heavy elevation values with the semantically informed ground estimate produces a substantially cleaner map for local planning. Qualitatively, the merged map preserves the underlying terrain structure while suppressing artifacts caused by grass tufts or overhanging foliage.

C. Autonomous Navigation Evaluation

The final evaluation considers the complete pipeline, used in a navigation architecture. Three routes were prepared in alpine off-road terrain, including forest areas with dense vegetation, steep inclines with overhanging branches, and open alpine sections with bushes and grass. Route 1 has been chosen to test transitions from normal roads to dense forest, Route 2 features steep terrain and overhanging branches, and Route 3 contains variations of bushes, grass, and small trees in open terrain. The routes were represented by GPS waypoints, which had been recorded using a mobile Fixposition sensor setup. During the trials, each robot localized using the onboard Fixposition GNSS system with RTK correction. The navigation task was defined as following a sequence of prerecorded GPS waypoints. The proposed perception pipeline provided the local traversability representation in the form of the merged semantic elevation map, which was used by the navigation stack for local planning, using a 2.5D planning approach described in [14]. Localization, waypoint management, and low-level motion execution were kept unchanged across experiments. The main variable under evaluation was therefore the terrain representation produced by the proposed semantic-geometric mapping approach. To

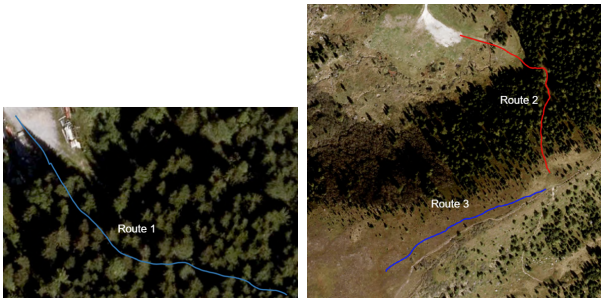


Fig. 9. Prerecorded Waypoints: (left) shows Route 1 through the forest road; (right) shows the two alpine paths. The first one in red (Route 2) is along a steep hiking path. The second one in blue (Route 3) leads uphill through a less overgrown area. All routes are located in the Seetaler Alps Military Training Area in Austria.

TABLE III

MANUAL INTERVENTIONS DURING THE AUTONOMOUS NAVIGATION EVALUATION ACROSS ALL ROUTES.

Route	Platform	Safety Interventions ↓	Functional Interventions ↓
Route 1	Artus	2	0
	Warthog	0	0
Route 2	Mercator	0	6
	Warthog	0	0
Route 3	Mercator	0	2
	Warthog	0	0

evaluate practical navigation performance, we count the number of manual interventions, distinguishing between safety interventions, which prevent dangerous maneuvers such as tip-over situations, and functional interventions, which free the robot when it can no longer continue autonomously. The first route was evaluated using the robots Artus[2]and Warthog, the remaining routes were tested with the robots Mercator and Warthog, since their wide construction and low center of gravity decrease the risk of tipping over. The results are listed in Table III.

For Route 1, shown in Figure 9 left, both Artus and Warthog managed to pass the forest entrance, which was considered the hardest part. This route has an approxiamte length of 180m. Warthog completed the route without incidents, while Artus required two manual slowdowns to avoid tip-over on steep and uneven terrain. These interventions were caused by Artus' narrow footprint and high center of gravity rather than by failures in the perception system itself. Using the standard elevation map instead of the proposed method, the robots could not traverse narrow passages at all, as they were represented as non-passable terrain. In contrast the proposed method allowed for safe traversal in these situations.

Route 2 shown in Figure 9 right in red, was the most challenging route. It has a length of 418m and follows a narrow, steep track overgrown with dense grass and bushes and includes overhanging branches. Snippets of this are shown in Figure 5. Plain elevation mapping represented the overhanging branches as solid elevations, which prevented planning entirely, whereas the proposed method corrected the map so that a path could still be planned underneath them. For Mercator, a total of six interventions were required, none of which were safety-critical. Two of these interventions were caused by overhanging branches. Although the Grid Merger is designed to clear regions that appear as hills in the Elevation Map due to overhanging branches, that are actually higher than the robot, in these cases, the branches were still within the robot's height. This prevented the Grid Merger from flattening the area. Warthog managed to autonomously drive the entire second route without any interventions. Its slightly smaller height and footprint allowed it to pass all tree branches without problems. The challenge in Route 3 shown in Figure 9 right in blue, lies in the trees and bushes scattered across the open area leading to the mountain summit. It has a length of 380m and connects directly to Route 2. Although the segmentation model correctly classifies the larger trees, it can struggle to distinguish between small trees and bushes. Often, this type of vegetation is classified as grass or bushes in the distance. This can lead to the behavior where obstacles appear a few meters in front of the robot, requiring the planner to replan the trajectory. For the Warthog, this did not pose any problems since the robot can turn in place and move around obstacles of this kind. The Mercator's double Ackermann mechanism does not allow for on-the-spot turns, therefore interventions were necessary.

Overall, the route-based evaluation further shows that the approach transfers across different environments and robot platforms. The results demonstrate, that the main remaining difficulties occur in challenging edge cases such as dense overhanging vegetation, severe terrain irregularities, and situations where thresholding or segmentation errors lead to incomplete obstacle representation. The autonomous navigation experiments demonstrate that the proposed semantic–geometric map is suitable for practical off-road navigation and supports more reliable behavior in dense natural environments than conventional elevation-only terrain representations.

VII. CONCLUSION AND FUTURE WORK

This work presented a modular semantic–geometric perception pipeline for autonomous navigation in densely vegetated off-road environments. By combining LiDAR-based elevation mapping with semantic information from image segmentation, the system improves the distinction between traversable vegetation and rigid obstacles. Real-world tests show that this approach supports more reliable planning in dense natural terrain and that the merging stage reduces vegetation-induced artifacts in standard elevation maps while preserving relevant obstacles for navigation. The approach was evaluated on three robot platforms in challenging alpine environments, demonstrating practical feasibility across different sensor configurations. Fully autonomous route-based evaluation showed improved navigation in vegetation compared with conventional elevation mapping.

[2]Artus robotic platform – CharismaTec og. https://charismatec.at/en/projects/

Future work includes software optimization, improved obstacle mapping, better ground-elevation estimation, and integration of depth cameras. In particular, the current implementation could be consolidated to reduce system overhead. More advanced obstacle representations need to be investigated to model more nuanced traversability cases, such as low-visibility situations. Moreover, synchronized RGB-D sensing could further improve the alignment between geometry and semantics, leading to higher quality maps.

REFERENCES

[1] P. Fankhauser, M. Bloesch, and M. Hutter, "Probabilistic terrain mapping for mobile robots with uncertain localization," *IEEE Robotics and Automation Letters (RA-L)*, vol. 3, no. 4, pp. 3019–3026, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8392399

[2] P. Fankhauser and M. Hutter, "A Universal Grid Map Library: Implementation and Use Case for Rough Terrain Navigation," in *Robot Operating System (ROS) – The Complete Reference (Volume 1)*, A. Koubaa, Ed. Springer, 2016, ch. 5. [Online]. Available: http://www.springer.com/de/book/9783319260525

[3] H. Lee, J. Kwon, and C. Kwon, "Learning-based uncertainty-aware navigation in 3d off-road terrains," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 10 061–10 068. [Online]. Available: https://ieeexplore.ieee.org/document/10161543

[4] 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. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9981942

[5] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-24574-4_28

[6] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017. [Online]. Available: https://arxiv.org/abs/1706.05587

[7] A. Valada, J. Vertens, A. Dhall, and W. Burgard, "Adapnet: Adaptive semantic segmentation in adverse environmental conditions," in *IEEE International Conference on Robotics and Automation (ICRA)*, 05 2017, pp. 4644–4651. [Online]. Available: https://doi.org/10.1109/ICRA.2017.7989540

[8] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 1290–1299. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2022/html/Cheng_Masked-Attention_Mask_Transformer_for_Universal_Image_Segmentation_CVPR_2022_paper.html

[9] A. Valada, G. Oliveira, T. Brox, and W. Burgard, "Deep multispectral semantic scene understanding of forested environments using multimodal fusion," in *International Symposium on Experimental Robotics (ISER)*, 03 2017, pp. 465–477. [Online]. Available: https://doi.org/10.1007/978-3-319-50115-4_41

[10] M. Wigness, S. Eum, J. G. Rogers, D. Han, and H. Kwon, "A rugd dataset for autonomous navigation and visual perception in unstructured outdoor environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 5000–5007. [Online]. Available: https://ieeexplore.ieee.org/document/8968283

[11] K. Vidanapathirana, J. Knights, S. Hausler, M. Cox, M. Ramezani, J. Jooste, E. Griffiths, S. Mohamed, S. Sridharan, C. Fookes, and P. Moghadam, "Wildscenes: A benchmark for 2d and 3d semantic segmentation in large-scale natural environments," *The International Journal of Robotics Research*, vol. 44, no. 4, p. 532–549, Sep. 2024. [Online]. Available: http://dx.doi.org/10.1177/02783649241278369

[12] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, ser. Intelligent Robotics and Autonomous Agents series. MIT Press, 2005. [Online]. Available: https://books.google.at/books?id=2Zn6AQAAQBAJ

[13] K. Koide, S. Oishi, M. Yokozuka, and A. Banno, "General, single-shot, target-less, and automatic lidar-camera extrinsic calibration toolbox," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 11 301–11 307. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10160691

[14] A. Koczka, "Investigating 2.5d path-planning methods for autonomous mobile robots in off-road scenarios," Master's thesis, Graz University of Technology, February 2025. [Online]. Available: https://doi.org/10.3217/9yba9-avz88