DigiForest

Digital Analytics and Robotics for Sustainable Forestry

CL4-2021-DIGITAL-EMERGING-01
Grant agreement no: 101070405

DELIVERABLE 3.3

Per-platform Traversability-Aware Navigation in Forests

Due date: month 24 (Sept. 2024)
Deliverable type: R
Lead beneficiary: ETH Zurich

Dissemination Level: PUBLIC

Main author: Fan Yang, Matias Mattamala, Fang Nan, Cesar Cadena, Marco Hutter

Contents

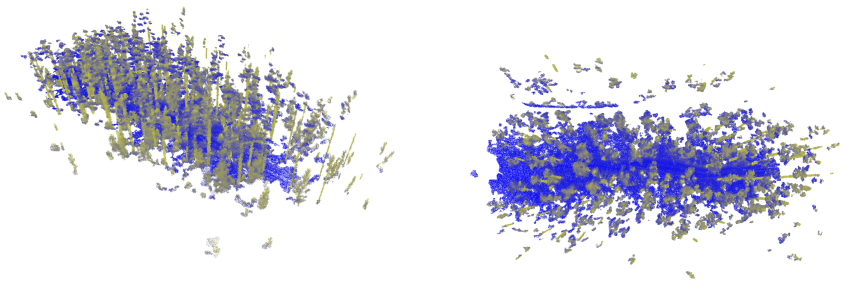
1 Introduction

Overview Autonomous robot navigation in forest terrain presents significant challenges due to the complex and variable nature of the environment. Understanding the diverse properties of the terrain is crucial for enabling robots to navigate effectively in such unstructured settings. This remains a major challenge in the field of robotics. Recent research has advanced our understanding from purely geometric analyses of terrain [17, 28, 12] to more nuanced semantic understandings [28, 4], which are critical for deploying different robotic platforms in off-road and forest environments. These developments include the integration of advanced sensor technologies [1] and machine learning [18, 7, 21, 2] algorithms to better interpret and respond to the dynamic and uneven terrains commonly found in forests.

Within the scope of the Digiforest project, two main ground robot platforms are employed: the SAHA forest machine [11] and the ANYmal legged robot [5]. The SAHA forest platform is primarily designed for tree harvesting, requiring it to navigate along forest trails, which demands robust terrain traversal capabilities. On the other hand, the ANYmal-legged robot is utilized for mapping forest areas beneath the tree canopy, necessitating its ability to maneuver through dense ground covers, such as high grass and uneven terrain.

To achieve autonomous navigation in these challenging environments, each robot platform must address three key aspects from a robotics perspective: Terrain Analysis, Planning and Navigation, and Driving/Locomotion Control. Terrain Analysis involves the use of both visual and non-visual sensors to classify and understand the terrain, allowing the robots to adapt their movement strategies accordingly. For instance, the ANYmal robot uses proprioceptive feedback to maintain stability and efficiency in various terrains [10, 16], including slipping and vegetated areas. Planning and Navigation focus on developing robust algorithms, such as those based on deep learning [29] and classic motion primitives approach [9], which allow the robots to make real-time decisions for obstacle avoidance and efficient pathfinding. Finally, Driving/Locomotion Control [26] is concerned with the execution of these plans, ensuring that the robots can follow the planned path or move effectively through the terrain by adjusting their gaits and adapting to the physical challenges presented by the environment.

In this report, we will delve into the technical details of each aspect of the different platforms, showcasing how these technologies enable the field deployment of robots in forests in Finland and Switzerland.



(a) (b)

Figure 1: Geometric-based terrain segmentation on fused point cloud, with colormap blue representing low-cost traversable points and with yellow representing high-cost non-traversable points

2 SAHA Robot - The Forest Harvester

2.1 Terrain Analysis

The terrain analysis module for the SAHA platform is designed to identify paths and areas that the vehicle can safely traverse. To ensure robust traversability and maintain the vehicle's operational safety, two key modules were developed.

In the first phase, terrain analysis is based on perceptual data, which can be derived from geometric or semantic information. This allows the system to make preliminary decisions about which areas are traversable and which are not. However, this approach assumes that the traversability is independent of the vehicle's current state. The determination of whether a path is safe to traverse relies on heuristic methods, meaning it does not fully account for the vehicle's kinematic and dynamic capabilities.

To address this limitation, the second phase introduces a more advanced, data-driven approach. This method leverages simulated data to accurately represent the vehicle's complete kinematic and dynamic characteristics. By integrating this simulation-based model with geometric inputs, we can assess traversability with respect to SAHA's specific kinematic and dynamic properties. This approach offers a more precise understanding of which areas the SAHA can traverse, factoring in the vehicle's kinodynamic behavior during movement, such as how it responds to varying terrain inclinations, obstacles, and surface conditions in deployment.

2.1.1 Heuristic-based traversability Cost

In this approach, we fused traversability costs estimated from both semantic and geometric data, using heuristic functions. For the geometric cost, we calculated it based on terrain elevation data obtained from LiDAR, normalizing the cost to a 0-1 interval for better comparison, as shown in Fig. 1. However, relying solely on geometric information, such as elevation, may not provide enough detail to differentiate between traversable and undesirable areas for the vehicle fully. For example, geometric data might miss contextual information like the presence of vegetation or obstacles that are critical for safe traversal.

To address this, we combined geometric costs with semantic costs. The semantic cost is derived from predefined terrain labels (e.g., grass, mud, rocks) and corresponding heuristic values for each terrain type. These labels were assigned based on prior

(a) (b)

Figure 2: Semantic-based terrain segmentation on RGB camera image input. a) shows the raw RGB camera input. b) shows the semantic segmentation for different reversibility classes.

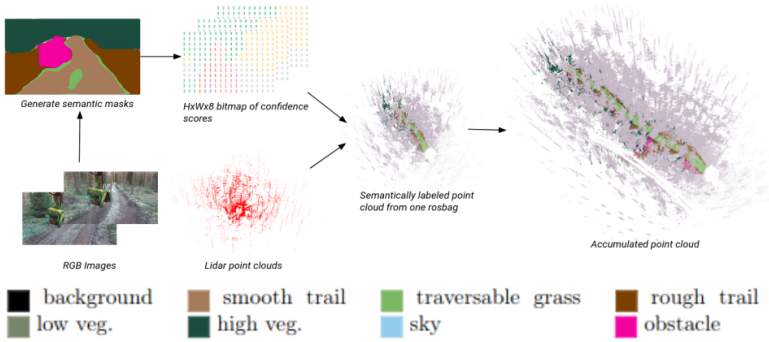


Figure 3: Point-to-Pixel projection process for generating a semantic-based traversability point cloud from RGB images.

knowledge and domain-specific experience, with semantic segmentation helping to better distinguish between different terrain types, as shown in Fig. 2

For the semantic segmentation task, we experimented with several network backbones and finally selected Segment Anything Model (SAM) [15], which performed best in terms of segmentation accuracy on off-road terrain datasets, as presented in Tab. 1. The evaluation focused on segmenting different terrain classes like road, obstacles, and rough terrain, which are critical for forest and off-road environments.

Next, we fused the semantic segmentation from RGB camera data with geometric LiDAR data through a Point-to-Pixel Projection method. This integration allowed us to leverage both the high spatial resolution of LiDAR and the contextual information from RGB imagery. The pipeline for this fusion process is shown in Fig. 3.

Finally, we fused the geometric and semantic costs using a K-means clustering approach, where each LiDAR point was grouped based on features like its relative height from the ground and the local height variations. This allowed us to better estimate terrain roughness and surface complexity. The semantic cost was also extrapolated to LiDAR points not directly covered by camera imagery, improving coverage and accuracy in estimating traversability. The result of this fused approach is depicted in Fig. 4.

	AP	AP50	AP75	APm	APl
Vitdet	28.881	47.574	30.666	1.744	31.466
SAM	**30.303**	52.072	28.696	4.858	31.202
DinoV2	28.961	50.935	28.614	1.594	32.063

Table 1: Average Precision (AP) score of different network backbone

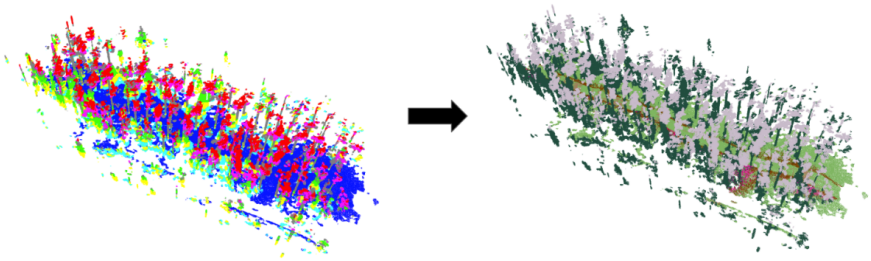


Figure 4: Fusion of traversability costs using K-means clustering from geometric and semantic data

2.1.2 Simulation-based Traversibilty Estimation

While the heuristic-based method provides reasonable results, it has a key limitation: it does not account for the specific kinematic and dynamic characteristics of the SAHA robot. This approach relies solely on heuristic values to estimate traversability, overlooking the real-time vehicle dynamics and control capabilities. To overcome this limitation, we explored a simulation-based approach that offers more accurate traversability analysis by incorporating the robot's kinematics and dynamics, as inspired by the work in [8].

For our simulation, we employed Nvidia Isaac Gym [19], a high-performance physics engine, to simulate the SAHA robot's full kinematic and dynamic behavior. This includes simulating key systems such as active suspension and chassis balancing, both of which are critical for navigating the uneven terrain commonly found in forest paths.

This approach returns to a focus on geometric data only. Given that the SAHA robot operates primarily on forest paths, a geometry-based approach is sufficient. Since the robot stays on well-defined trails, the added complexity of semantic segmentation—such as distinguishing between different vegetation types—is less critical. In fact, relying on semantic models introduces challenges like inconsistencies due to changing weather conditions and the requirement for specific terrain training data, which complicates system deployment and maintenance.

While a geometry-only method does present challenges, such as mistakenly identifying tall grass as an obstacle, these issues can be addressed by applying a low-pass filter to the sensor data. This filter smooths out the noise from vegetation, enabling the robot to better distinguish between actual obstacles and non-hazardous elements like grass. Despite this limitation, geometric features such as elevation, slope, and roughness are robust enough to ensure safe navigation on forest paths.

To replicate real-world conditions, we utilized surface terrain meshes from Swiss SURFACE3D [6], which included various terrain types such as slopes, rocky areas, and artificial constructions. These simulations allowed us to evaluate the SAHA robot's performance under a range of challenging conditions. We applied specific criteria

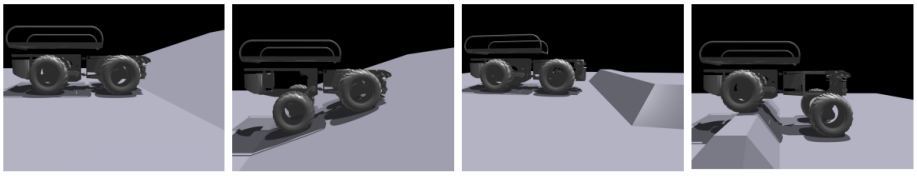


Figure 5: The simulated SAHA robot in Isaac Gym environment, with activated suspension, and balancing control

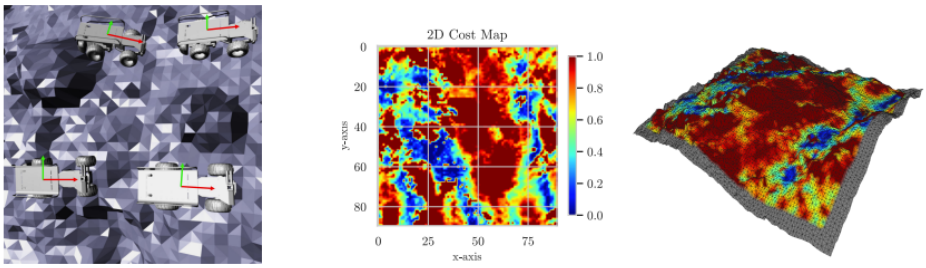


Figure 6: Traversability of the given terrain mesh. From left to right: robots end in different states, top-down view 2D cost map, 3D cost mesh visualization

to assess the robot's stability and control over different terrains. By focusing on geometric data and incorporating vehicle dynamics in the simulation, we achieved reliable traversability estimation without the complexities and potential inconsistencies of a semantic approach.

To quantify the traversability difficulty, we defined criteria based on the robot's physical response to the terrain, such as tilt angle, wheel slip, and suspension compression, listed in below:

- Tip-over instability
- Collision between robot and terrain
- Robot incapability of staying close to initial states

These criteria served as indicators for predicting the robot's stability and whether it could safely traverse a given area.

A simulated version of the SAHA robot, with its arm removed for simplicity, is shown in Fig. 6. By placing the robot at various points across the simulated terrain mesh and monitoring its behavior, we could assess the traversability of each area. The results, represented by triangular surface meshes, are shown in Figure 6, where we mapped the robot's stability across different terrain conditions.

2.2 Planning and Navigation

Based on the estimated traversability data, we implemented a motion-primitive-based method for motion planning and navigation for the SAHA robot [9]. The core of this approach involves generating a set of precomputed motion primitives, which serve as building blocks for the robot's navigation.

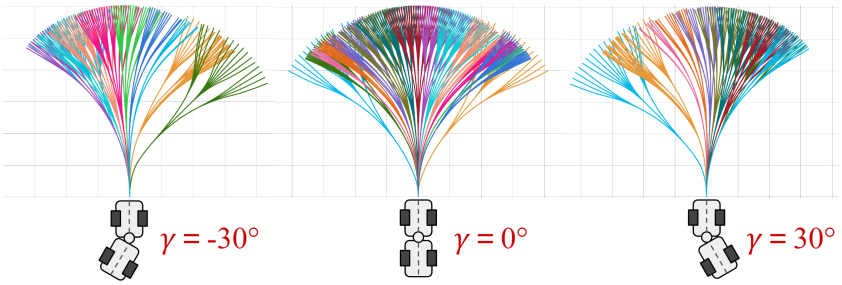


Figure 7: Example of the motion primitives for vehicles with steering angles equal -30 degrees, 0 degrees, and 30 degrees.

Firstly, the kinematic model of the SAHA robot is formulated. Assuming no sideslip conditions and considering the geometric constraints at points P_1 and P_2, the relationship between the front part's heading, velocity, steering angle, and steering velocity can be expressed as derived in [3]:

$$\dot{x}_1 = v_f \cos(\theta_1) \tag{1}$$

$$\dot{y}_1 = v_f \sin(\theta_1) \tag{2}$$

$$\dot{\theta}_1 = -\frac{v_f \sin(\gamma) + l_2\dot{\gamma}}{l_1 \cos(\gamma) + l_2}. \tag{3}$$

By organizing equations Eq. (1) to Eq. (3), we derive a kinematic model with four state variables, using the front center P_1 as the virtual control point:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \\ \dot{\theta}_1 \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} \cos(\theta_1) & 0 \\ \sin(\theta_1) & 0 \\ -\frac{\sin(\gamma)}{L} & -\frac{l_2}{L} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \dot{\gamma} \end{bmatrix}, \tag{4}$$

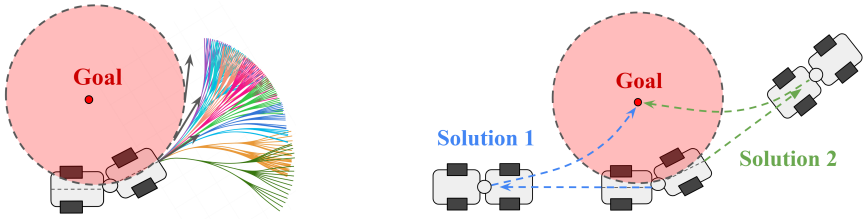
where $L = l_2 + l_1 \cos(\gamma)$ is defined as the effective length of the vehicle. This model considers the robot's turning radius, wheelbase, and moving velocity. Using this model, we conducted forward simulations to generate a diverse set of feasible motion primitives offline, as illustrated in Fig. 7. These primitives encapsulate the various maneuvers the SAHA can execute, such as turning, moving forward, or navigating inclines. During online operation, we use a grid-based collision detection method to evaluate potential paths against the current traversability estimates, ensuring that the selected path is free of obstacles.

The path selection process is guided by a heuristic-based reward function, which evaluates the feasibility of each path based on a combination of factors such as continuous velocity and difficulty of the terrain. Specifically, we calculate the comprehensive score S_{ij} for each control group of trajectories:

$$S_{ij} = \frac{\sum_{k=1}^{N_{ij}} s_k^{ij}}{N_{ij}}, \tag{5}$$

where N_{ij} is the total number of trajectory in control group Σ_{ij}. Each trajectory score s_k^{ij} is determined by

$$s_k^{ij} = (s^{dir} + \alpha \cdot s^{dist})^2 \cdot s^{vel} \cdot s^{state} \cdot s^{terrain} \cdot s^{p}. \tag{6}$$



(a) (b)

Figure 8: (a) shows the "Unreachable zone" for the SAHA. (b) depicts two potential solutions with bi-directional trajectories.

The score terms in (6) are given by:

$$s^{dist} = f^{dist} \propto (D_{max} - \sqrt{dx^2 + dy^2}) \tag{7}$$

$$s^{dir} = f^{dir} \propto (2\pi - |d\theta_1|), (2\pi - |d\theta_2|) \tag{8}$$

$$s^{state} = f^{state} \propto (2\gamma_{max} - |\gamma_k - \gamma_{now}|) \tag{9}$$

$$s^{vel} = f^{vel} \propto v_k \tag{10}$$

$$s^{terrain} = f^{terrain}(H_{max}) \tag{11}$$

$$s^{p} = f^{p}(diff_c), \tag{12}$$

where (dx, dy) is distance to the goal, D_{max} is a constant. $d\theta_1$ and $d\theta_2$ are direction differences toward the goal based on the heading and position of the trajectory τ_k^{ij}. γ_k is the initial steer angle for the trajectory τ_k^{ij}. v_k represents the velocity for the trajectory τ_k^{ij}. H_{max} is the maximum height of the terrain along the trajectory: $\max(height)$ in τ_k^{ij}. $diff_c$ is the distance between the end position of τ_k^{ij} and the end position of the last selected trajectory.

Additionally, to account for the SAHA's kinematic limitations—specifically its large turning radius—we developed a heuristic search strategy. This approach enables the robot to reach goals that would otherwise fall inside the robot's minimum turning radius, namely the "Unreachable zone", as shown in Fig. 8. This ensures that the SAHA can navigate tight or confined spaces that are common in forested environments.

2.3 Driving Control

While the selected motion primitive provides a kinematically feasible path, it cannot be directly used for control due to the discretized nature of the states in the primitive generation process. To bridge this gap, we implemented a lookahead sampling method. From the selected motion primitive, we continuously sample a lookahead point, which serves as a reference for real-time control.

For the control strategy, we designed a variant of the pure-pursuit method, inspired by Park's method [38]. Pure-pursuit is a widely used approach in mobile robot navigation, where the robot adjusts its heading to "pursue" a point ahead on the path. Our proposed version, however, is tailored to the SAHA's specialized kinematics, such as its active suspension system and large wheelbase, to ensure stable and smooth traversal over uneven terrain.

We also introduced a pose-stabilizing feedback controller that accounts for the robot's specific kinematic and dynamic properties. This controller ensures that the robot maintains a stable orientation and minimizes deviations from the desired path,

Table 2: Tracking Error of Two Different Controller - Unit: meter

	$\gamma = 0°$	$\gamma = 15°$	$\gamma = 30°$	Total
Pure-pursuit	0.0338	0.0424	0.0615	0.0459
Ours	**0.0322**	**0.0388**	**0.0448**	**0.0386**

Table 3: Comparison of Motion Planning Test Results

	Env 1		Env 2		Env 3		Env 4		Total	
	SR	SPL	SR	SPL	SR	SPL	SR	SPL	SR	SPL
Baseline	92%	0.8047	48%	0.4221	40%	0.3287	36%	0.3110	54%	0.4666
Ours (with p.p.)	**100%**	**0.9632**	96%	0.7451	92%	0.6928	64%	0.5426	88%	0.7359
Ours	**100%**	0.9570	**100%**	**0.7987**	**96%**	**0.7550**	**76%**	**0.5999**	**93%**	**0.7776**

even when encountering obstacles or uneven surfaces. The comparison between our method and the traditional pure-pursuit method is shown in Tab. 2, highlighting improvements in terms of path stability, execution time, and traversal accuracy.

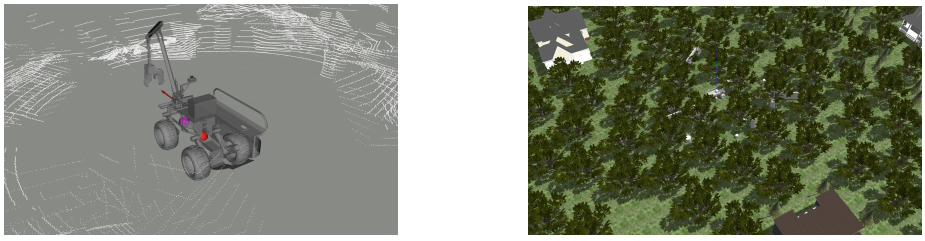
2.4 Experiments

We evaluated the traversability-aware navigation system for the SAHA robot in both simulated and real-world environments to assess the system's effectiveness and robustness.

2.4.1 Simulation Environment

In the simulation, we developed a detailed SAHA robot model shown in Figure 9, which captures the full kinematic structure of the robot. This model is integrated into the Gazebo physics engine, allowing us to simulate the robot's interaction with the environment, including dynamic forces acting on the robot's body and wheels, and real-time sensor inputs.

For the simulated forest environment, we utilized an open-sourced Gazebo forest world with varying density of trees and obstacles, providing a realistic representation of the forest environment, as shown in Figure 9. We conducted experiments comparing the performance of our proposed motion-primitive-based planner [9] and driving controller against a baseline method [30]. The baseline was a primitive-based reactive planner



(a) (b)

Figure 9: (a) shows the SAHA robot model in the Gazebo simulation environment, with white points representing simulated LiDAR data. (b) shows the simulated Gazebo environment for a forest.

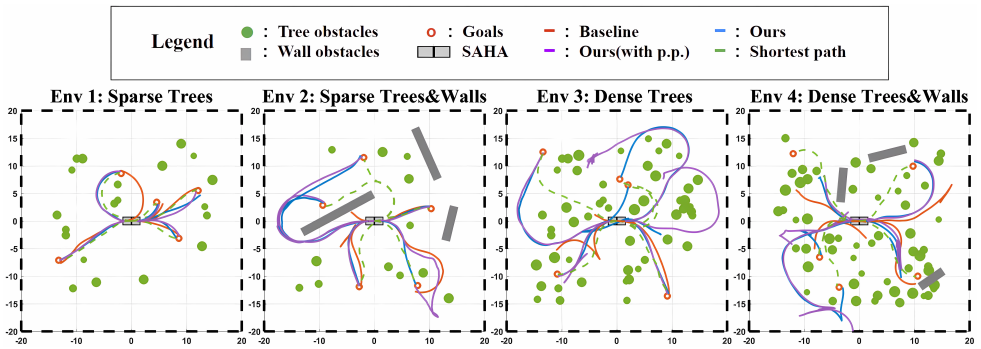


Figure 10: Example maps of four environment types: 1) sparse tree obstacles ($N_t = 20$), 2) sparse tree and wall obstacles ($N_t = 12$, $N_w = 3$), 3) dense tree obstacles ($N_t = 50$), and 4) dense tree and wall obstacles ($N_t = 55$, $N_w = 3$). The maps include example trajectories generated by different methods: optimal paths (green), our method (blue), the baseline method (red), and our method with the baseline controller (purple).

without kinematic or dynamic considerations. The performance was measured using several key metrics, including Success weighted by Path Length (SPL) and tracking error. As shown in Figure 10 and Table 3, our motion-primitive-based planner can achieve higher SPL scores compared to the baseline, indicating that our approach results in more successful navigation over longer distances. Additionally, the proposed driving controller demonstrated lower tracking errors as shown in Table 2, meaning the robot followed the planned path with greater accuracy, even in challenging terrain conditions.

2.4.2 Real-world Testing

For real-world testing, we deployed the SAHA robot in two forest environments: one in Finland and the other in Switzerland. In these tests, we integrated Graph-MSF [24] with Compslam [13] for state estimation. The system combined input from the onboard LiDAR and IMU sensors to estimate the robot's position and orientation in real-time. State estimation was critical for accurately localizing the robot within the environment and ensuring smooth navigation. Additionally, in collaboration with PreFor, we utilized a fast Iterative Closest Point (ICP) [31] for map registration. This allowed us to match real-time LiDAR scans with pre-built maps of the forest, ensuring precise localization during the robot's traversal, and navigation for a longer range. The real-world tests were conducted on both Evo, Finland, and Stein am Rhein, Switzerland's forest paths. Figure 11(a) shows the SAHA robot navigating through a Finnish forest, while Figure 11(b) captures its operation on Swiss forest paths. In each environment, the system demonstrated its capability to perform real-time traversability analysis and adapt to changing terrain conditions.

In the real-world experiments, the operator provided specific goal locations, and the robot autonomously navigated toward those goals, adjusting its path based on the dynamic terrain analysis. As shown in Figure 12, the system performed real-time traversability estimation, continuously updating the robot's path based on new data. Figure 13 provides a screenshot of the system navigating within a pre-built global point cloud map, showing the integration of both local and global navigation strategies.

Overall, the experiments demonstrate that our system's combination of terrain analysis, motion-primitive planning, and driving control can handle both simulated

(a) (b)

Figure 11: The SAHA robot navigating diverse forest environments. (a) The SAHA robot navigates through a forest path in Evo, Finland. (b) The SAHA robot in action on forest paths in Stein am Rhein, Switzerland.

and real-world forest environments effectively, achieving high performance in terms of path planning efficiency and safety.

3 ANYmal legged Robot

3.1 Terrain Analysis

Similar to the SAHA platform, the terrain analysis module for the legged robot platform ANYmal is designed to estimate traversable or steppable areas for the robot. The system utilizes multi-modal sensory inputs, such as LiDAR and RGB cameras, to extract information from both geometric and semantic perspectives, enabling the identification of traversable and non-traversable regions.

However, unlike the SAHA, which primarily operates on predefined forest paths, ANYmal is designed to traverse more challenging and unpredictable forest terrain. The terrain properties of the forest floor, such as loose soil, roots, and vegetation, are difficult to model using traditional heuristic functions. Therefore, the system must be adaptive, relying heavily on the robot's proprioceptive feedback, such as IMU data and joint torque measurements, to dynamically adjust its traversability estimates in real-time.

3.1.1 Adaptive Terrain Analysis Module

The first component of the terrain analysis module is an adaptive system that links the robot's current proprioceptive feedback to visual features extracted from its onboard cameras, enabling dynamic traversability estimation [21]. This system is trained using human demonstrations, where areas deemed traversable by humans are labeled and associated with corresponding visual features. These features are extracted using a pre-trained DINOv2 feature extractor [25], which captures high-level representations of the terrain. As illustrated in Fig. 14, during deployment, these demonstrated traversable areas serve as a prior, and the system continuously updates and refines its understanding of terrain features based on additional inputs collected throughout the mission. This adaptive learning allows the robot to handle the variability of the forest ground more effectively as it progresses.

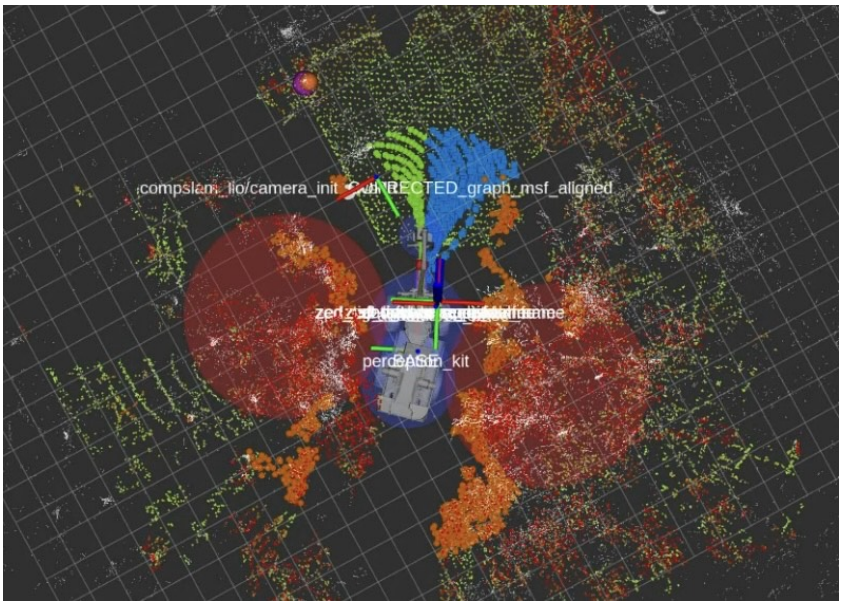


Figure 12: The figure illustrates the navigation system in action. The orange dot represents an obstacle detected by the terrain analysis module. The green path highlights the selected primitive groups leading to the goal (purple) among the candidate paths (blue). The two red circles indicate unreachable zones.

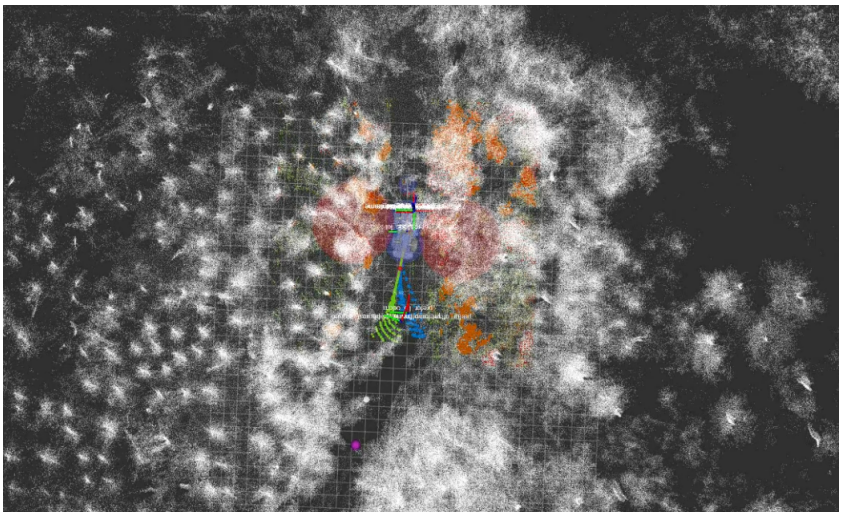


Figure 13: The figure illustrates an instance of global navigation on a forest path in Stein am Rhein, Switzerland. The goal (purple) is defined in the global map frame, while the white point cloud represents a segmentation of the pre-built global map.

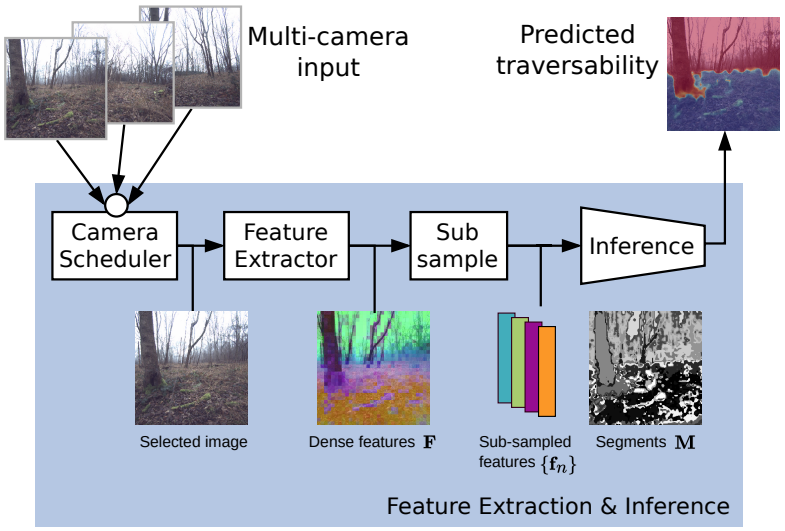


Figure 14: Feature Extraction and Inference Process: The camera scheduler selects a camera, whose RGB image is processed by the feature extractor using pre-trained models to extract dense visual features. The sub-sample module then refines these features into embeddings, which the inference module uses to predict terrain traversability.

3.1.2 Physics-based Terrain Estimation

In addition to the adaptive visual-based system, we introduce a more explainable method for estimating the physical properties of the terrain, such as friction and stiffness [2]. Unlike purely visual-based systems, this approach offers greater transparency and reliability by directly predicting the physical parameters that govern the robot's interaction with the ground.

Specifically, we propose training a physical decoder in a simulation environment to predict key parameters such as friction and stiffness from multi-modal inputs (both geometric and visual). This system is trained using simulated terrain data, where the ground truth for physical properties is known. Once trained, the decoder is able to label real-world terrain images with physical parameters in a self-supervised manner. This labeled data is then used to further train a visual network that can predict these physical properties from real-world images during deployment. As shown in Figure 9, the system provides dense predictions of terrain friction and stiffness, allowing the robot to make more informed decisions about which areas are safe to traverse, even in unknown or dynamically changing environments.

By combining both adaptive visual-based estimation and physics-based property analysis, the ANYmal robot gains a comprehensive understanding of the terrain. This dual approach improves the robot's ability to navigate complex, uneven surfaces commonly found on forest floors, making it better equipped to handle diverse and challenging environments.

3.2 Planning and Navigation

For the navigation task, the system integrates a reactive local planner [20] to generate SE(2) twist commands, which guide the robot toward its goal while avoiding

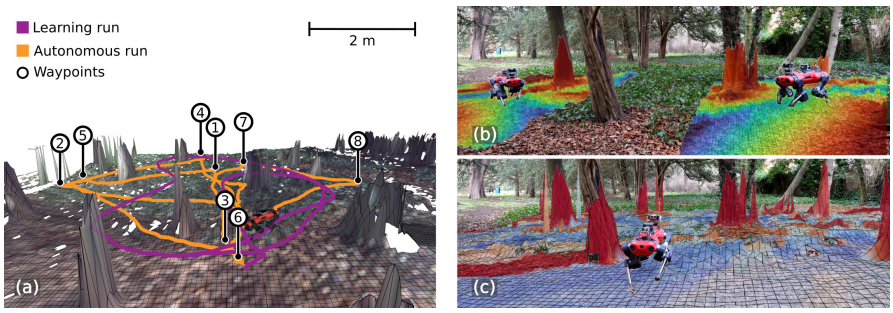


Figure 15: Point-to-point autonomous navigation: (a) After 2 minutes of teleoperation (magenta path), the robot achieved autonomous navigation in a woodland environment (orange path). (b) Examples of predicted traversability during autonomous operation. (c) 2.5D reconstruction of the test area, illustrating our approach's capabilities.

untraversable terrain. The twist commands represent the robot's velocity and angular motion in a 2D plane. The planner continuously updates the robot's traversable area based on the real-time traversability estimation introduced earlier. This ensures that the robot dynamically adapts to changing environmental conditions, such as variations in terrain geometry, obstacles, and vegetation, maintaining a safe and efficient trajectory toward the goal.

The local planner works together with global navigation strategies, ensuring that the robot not only avoids immediate obstacles but also stays aligned with long-term goals like reaching specific locations or exploration within the forest. By combining geometric terrain data (e.g., slopes and obstacles) and semantic terrain analysis (e.g., high grass or mud), the planner optimizes the robot's path, allowing it to navigate through unpredictable forest environments without getting stuck or veering off course, as shown in Figure. 9

3.3 Locomotion control

To ensure the robot can move robustly through uneven terrain, the system incorporates a reinforcement learning (RL)-based perceptive locomotion controller [23]. This controller is trained in a simulated environment that mimics the diverse terrain conditions encountered in forest settings. The simulated environment includes various terrain conditions, such as uneven ground, and rocky areas, enabling the controller to learn and generalize effective locomotion strategies for real-world deployments. By leveraging RL, the controller is capable of optimizing the robot's movements to balance stability, energy efficiency, and adaptability to dynamic terrain changes.

A metric-based elevation mapping method [22] provides a detailed environmental representation that is used as input to the locomotion controller. This elevation map offers precise information about the local terrain's surface, such as elevation differences, and slopes. The controller can adjust the robot's gait, foot placement, and balance by processing this map to ensure safe traversal over challenging ground, such as uneven rocks, roots, or muddy areas.

In addition to relying on elevation data, the RL policy also incorporates proprioceptive feedback from the robot itself, such as IMU data and joint torque measurements [23]. This enables the system to react to unforeseen terrain properties or sensor errors. For instance, when the robot detects false positive obstacle observations—such as tall grass that is mistakenly identified as a solid object—the system can override the

Mission				Robot setup		Metrics			
Campaign	**Date**	**ID**	**Area** $[m^2]$	**Hardware**	**Software**	**Mission time** $[s]$	**Distance traveled** $[m]$	**Area covered** [ha]	**Interv.** [#]
Evo, Finland	2023-05-03	M1	40×25	ANYmal C, Velodyne VLP-16	CompSLAM odometry	575.6	270.3	0.33	2
	2023-05-03	M2	40×25			432.4	233.6	0.32	0
	2023-05-04	M3	40×35			816.8⋆	301.1	0.37	7
	2023-05-04	M4	35×35			988.4	336.6	0.37	7
	2023-05-05	M5	70×25			1275.5	609.7	0.64	10
Wytham Woods, UK	2023-10-06	M6	20×20	ANYmal C, Velodyne VLP-16	VILENS odometry, VILENS-SLAM	436.9	215.0	0.29	2
Forest of Dean, UK	2023-02-19	M7	125×30	ANYmal D, Hesai QT64	VILENS odometry, VILENS-SLAM	1283.5	665.5	0.96	8

⋆ This mission was manually interrupted.

Table 4: Missions summary. We report the specifications of each mission executed across three campaigns in Finland and the UK, along with the robot setup used, as well as the main metrics reported on each mission, such as the MDBI and MTBI.

obstacle avoidance behavior and proceed smoothly. This adaptability is crucial in forest environments where dense vegetation can often confuse traditional obstacle detection systems.

By combining the RL-based locomotion controller with perceptive feedback and elevation mapping, the robot can effectively navigate complex, uneven terrain while maintaining robustness and safety, even in the face of unpredictable environmental conditions.

3.4 Experiments

Over the course of the last two years, the ANYmal robot has been deployed autonomously in different missions in Finland and the UK (Tab. 4). The objective has been assessing the previously described traversability and navigation modules in the context of autonomous forest inventory. Preliminary results have been reported in a non-peer reviewed paper presented in the ICRA 2024 Workshop on Field Robotics. A thorough analysis of these missions, field reports, and lessons learned is under preparation for a journal submission.

3.5 Evo Campaign

Our first campaign, in Finland, had the goal of evaluating the proposed autonomy system in forest environments. This was executed with an ANYmal C platform, using the Velodyne VLP-16 (30° Field of View) as the main environmental sensor. We used a state estimation setup derived from the CERBERUS SubT stack [27], which used CompSLAM [14] as the main odometry system. CompSLAM does not perform online pose graph optimization and loop-closure detection to ensure consistency.

Fig. 16 shows the autonomy performance reported in these missions (M1-M5), while Fig. 17 (a) shows an example from M2. Missions M1 and M2 were executed

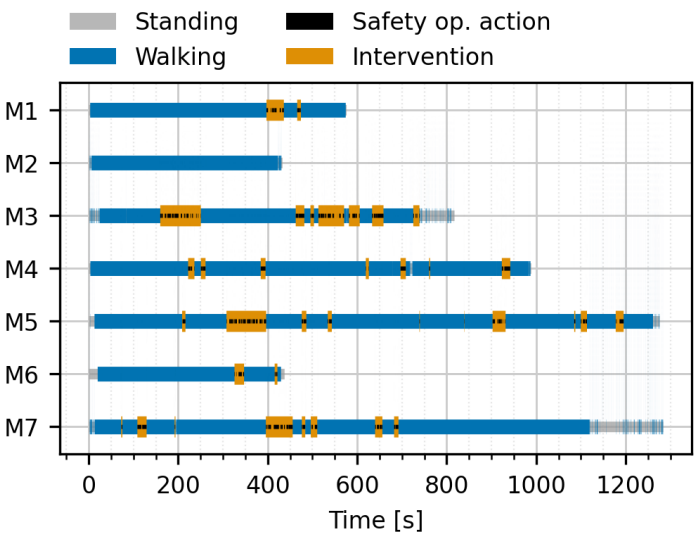


Figure 16: Summary of the seven survey missions executed across the three campaigns. We illustrate the periods where the robots operated fully autonomously, and where they required manual interventions from the safety operator.

Figure 17: Illustrative examples of the robot deployments during the different campaigns. (a) Coniferous forest in Evo, Finland. (b) Mixed forest in Wytham Woods, UK. (c) Oak forest in the Forest of Dean, UK.

in the same plots, obtaining consistent results. Mission M3 was executed in an area nearby, but with more challenging terrain, which required to interrupt the mission due to the robot getting trapped in a damp area. Missions M4 and M5 were executed in different areas of the forest, but achieving similar area coverage per time unit (1.5 ha), as shown in Tab. 4. In M5, we additionally observed severe tracking problems with the reference survey path due to drift in the state estimation system, which were more evident in this longer sequence (600 m long, compared to 300 m set for the other missions).

The missions lasted between 10–20 min, with the robot walking up to 600 m on the longest missions. About the autonomy performance, while we report the Mean Distance Between Interventions (MDBI) and Mean Time Between Interventions (MTBI) in Tab. 4, we found the fine grained analysis of the distribution of autonomy segments a more informative metric. This is shown in Fig. 18. With exception of mission M2, which was fully autonomous, in most of the other missions we reported a higher density of short autonomy segments ($\sim$10 m or $\sim$50 s). Many cases were due to short, frequent 'pushes' performed by the safety operator to move the local planner solution out of

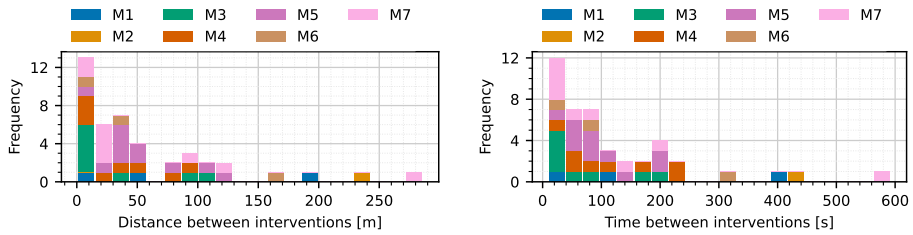


Figure 18: Distribution of distance and time between interventions for all missions. *Left:* Distance between interventions. *Right:* Time between interventions.

local minima, or driving the robot around a dead end.

We additionally reported how long the interventions took on each mission (Fig. 19). We observed that most of the interventions were ∼15 s, which is within the time required for the safety operator to get the robot out of dead ends that the mission and local planner interaction was not able to solve automatically. Only in specific cases—M3 and M5—, where the robot was deployed in considerably more challenging environments (damp terrain, short trees, higher density of bushes) we required longer interventions (>1 min) to move the robot to safer areas before continuing the mission.

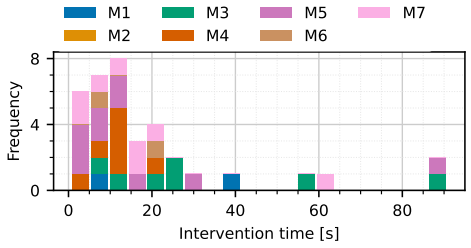


Figure 19: Duration of the interventions reported for each mission. Most of the interventions reported were short—20 s on average.

3.6 Wytham Woods Campaign

The second campaign was executed in the UK, in a mixed forest plot in Wytham Woods, near Oxford. The same ANYmal C platform and LiDAR setup was used in this campaign, as the goal was to improve the state estimation solution problems reported in the Evo campaign. For this, we transitioned to the proposed state estimation solution, which explicitly relied on a LiDAR-inertial odometry system along with an online pose graph SLAM to ensure consistency.

Fig. 17 (b) shows the testing environment, which was a clear sloped plot, with a few loose branches and twigs. We set a smaller 20 m × 20 m survey area, which corresponded to 0.29 ha covered for an effective 15 m LiDAR range. The reported coverage rate was consistent with the results obtained in the Evo campaign, above 1.5 ha/h. Only two short interventions were required (see Fig. 19, M6) to avoid falls due to loose branches, which trapped the robot's legs.

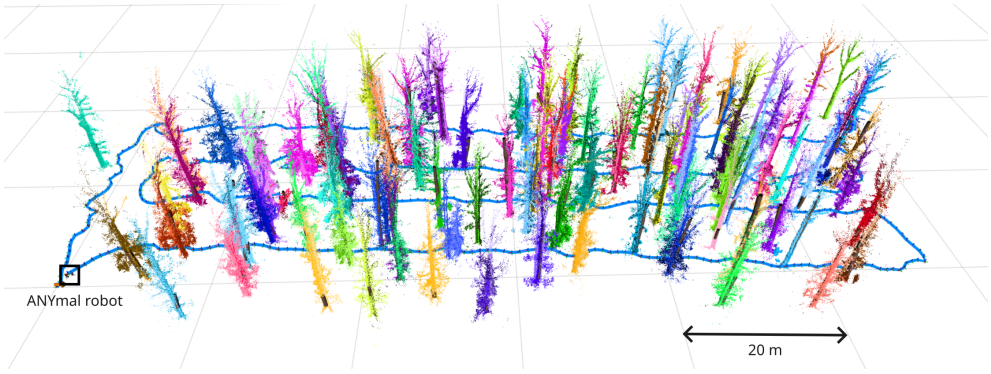


Figure 20: Illustrative example of the output produced by our system after an autonomous mission in the Forest of Dean, UK. The robot surveyed 0.96 ha in 20 min, and around 100 trees were segmented during online operation.

3.7 Forest of Dean Campaign

The last campaign, executed in the Forest of Dean in the UK, was used to test the integration of the forest analysis system towards online forest inventory. Additionally, we changed the robot platform from ANYmal C to ANYmal D—which has longer shanks, though the locomotion policy does not exploit this fact explicitly. We also improved the sensor payload, where we used a sensing unit with a dedicated computer and a Hesai QT64 LiDAR, which as a larger vertical Field of View (104°), enabling better coverage of the full trees. The state estimation and autonomy were left unchanged apart from the sensor configuration, as they were validated in the previous campaigns.

For the specific mission we executed, denoted M7, we specified a 125 m × 30 m survey area, which effectively corresponded to a 0.96 ha plot. The robot completed the mission in approximately 20 min, segmenting up to 100 trees, as shown in Fig. 20.

4 Conclusion and Future Works

In this report, we have successfully developed and evaluated terrain analysis modules for both the SAHA and ANYmal robots, integrating these with effective planning and control systems. The SAHA platform uses a motion-primitive-based planner combined with kinematic and dynamic simulations to assess traversability in forest environments, while the ANYmal legged robot employs adaptive and physics-based terrain analysis for more challenging forest floor navigation. Through a combination of simulation and real-world experiments in diverse forest environments, including Finland and Switzerland, we have demonstrated the effectiveness and robustness of these systems in addressing the unique challenges posed by forest terrains.

For future work, we aim to improve the robustness of the system by further benchmarking the performance of the different modules under varied environmental conditions. This includes comparing terrain analysis methods, evaluating control strategies, and integrating new sensory inputs. Another key area of focus will be expanding the system to support global mapping, enabling real-time map sharing between multiple robots. By integrating this with existing mapping systems, we can improve the overall efficiency and coordination of robotic deployments in forest environments. This will allow for more scalable and autonomous operations across large areas, with potential applications in forest management, environmental monitoring, and automated harvesting.

References

[1] Adekanmi Adeyinka Adegun, Serestina Viriri, and Jules-Raymond Tapamo. "Review of deep learning methods for remote sensing satellite images classification: experimental survey and comparative analysis". In: *Journal of Big Data* 10.1 (2023), p. 93.

[2] Jiaqi Chen et al. "Identifying Terrain Physical Parameters from Vision-Towards Physical-Parameter-Aware Locomotion and Navigation". In: *IEEE Robotics and Automation Letters* (2024).

[3] Peter I. Corke and Peter R. Ridley. "Steering kinematics for a center-articulated mobile robot". In: *IEEE Transactions on Robotics* 17.2 (2001), pp. 215–218. DOI: 10.1109/70.928568.

[4] Parker Ewen et al. "These maps are made for walking: Real-time terrain property estimation for mobile robots". In: *IEEE Robotics and Automation Letters* 7.3 (2022), pp. 7083–7090.

[5] Péter Fankhauser and Marco Hutter. "Anymal: a unique quadruped robot conquering harsh environments". In: *Research Features* 126 (2018), pp. 54–57.

[6] Federal Office of Topography swisstopo. *swissSURFACE3D Raster*. https://www.swisstopo.admin.ch/en/height-model-swisssurface3d-raster. Accessed on: 2024-09-10, ©swisstopo. 2024.

[7] Jonas Frey et al. "Fast traversability estimation for wild visual navigation". In: *arXiv preprint arXiv:2305.08510* (2023).

[8] Jonas Frey et al. "Locomotion policy guided traversability learning using volumetric representations of complex environments". In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2022, pp. 5722–5729.

[9] Jiangpeng Hu et al. "Motion Primitives Planning For Center-Articulated Vehicles". In: *arXiv preprint arXiv:2405.17127* (2024).

[10] Jemin Hwangbo et al. "Learning agile and dynamic motor skills for legged robots". In: *Science Robotics* 4.26 (2019), eaau5872.

[11] Edo Jelavic et al. "Harveri: A small (Semi-) Autonomous precision tree harvester". In: *Innovation in Forestry Robotics: Research and Industry Adoption, ICRA 2022 IFRRIA Workshop*. ETH Zurich, Institute of Robotics and Intelligent Systems. 2022.

[12] Zhuozhu Jian et al. "Putn: A plane-fitting based uneven terrain navigation framework". In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2022, pp. 7160–7166.

[13] Shehryar 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.

[14] Shehryar Khattak et al. "Complementary Multi–Modal Sensor Fusion for Resilient Robot Pose Estimation in Subterranean Environments". In: *Int. Conf. on Unmanned Aircraft Systems (ICUAS)*. 2020, pp. 1024–1029. DOI: 10.1109/ICUAS48674.2020.9213865.

[15] Alexander Kirillov et al. "Segment anything". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 4015–4026.

[16] Joonho Lee et al. “Learning quadrupedal locomotion over challenging terrain”. In: *Science robotics* 5.47 (2020), eabc5986.

[17] Abe Leininger et al. “Gaussian Process-based Traversability Analysis for Terrain Mapless Navigation”. In: *arXiv preprint arXiv:2403.19010* (2024).

[18] Zhizheng Liu et al. “Unsupervised continual semantic adaptation through neural rendering”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 3031–3040.

[19] Viktor Makoviychuk et al. “Isaac gym: High performance gpu-based physics simulation for robot learning”. In: *arXiv preprint arXiv:2108.10470* (2021).

[20] Matias Mattamala, Nived Chebrolu, and Maurice Fallon. “An efficient locally reactive controller for safe navigation in visual teach and repeat missions”. In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 2353–2360.

[21] Matías Mattamala et al. “Wild Visual Navigation: Fast Traversability Learning via Pre-Trained Models and Online Self-Supervision”. In: *arXiv preprint arXiv:2404.07110* (2024).

[22] Takahiro 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.

[23] Takahiro Miki et al. “Learning robust perceptive locomotion for quadrupedal robots in the wild”. In: *Science robotics* 7.62 (2022), eabk2822.

[24] Julian Nubert, Shehryar Khattak, and Marco Hutter. “Graph-based multi-sensor fusion for consistent localization of autonomous construction robots”. In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 10048–10054.

[25] Maxime Oquab et al. “Dinov2: Learning robust visual features without supervision”. In: *arXiv preprint arXiv:2304.07193* (2023).

[26] Jong Jin Park and Benjamin Kuipers. “A smooth control law for graceful motion of differential wheeled mobile robots in 2D environment”. In: *2011 IEEE International Conference on Robotics and Automation*. IEEE. 2011, pp. 4896–4902.

[27] Marco Tranzatto et al. “CERBERUS in the DARPA Subterranean Challenge”. In: 7.66 (2022). DOI: 10.1126/scirobotics.abp9742.

[28] Martin Wermelinger et al. “Navigation planning for legged robots in challenging terrain”. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2016, pp. 1184–1189.

[29] Fan Yang et al. “iplanner: Imperative path planning”. In: *arXiv preprint arXiv:2302.11434* (2023).

[30] Ji Zhang et al. “Falco: Fast likelihood-based collision avoidance with extension to human-guided navigation”. In: *Journal of Field Robotics* 37.8 (2020), pp. 1300–1313.

[31] Juyong Zhang, Yuxin Yao, and Bailin Deng. “Fast and robust iterative closest point”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.7 (2021), pp. 3450–3466.