Robotics and Autonomous Systems 171 (2024) 104560

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

3D traversability analysis and path planning based on mechanical effort for UGVs in forest environments

Afonso E. Carvalho [a,*], João Filipe Ferreira [a,b], David Portugal [a]

[a] Institute of Systems and Robotics, University of Coimbra, Portugal
[b] Computational Intelligence and Applications Research Group, Department of Computer Science, School of Science and Technology, Nottingham Trent University, UK

ARTICLE INFO

Keywords:
Forestry robotics
Traversability analysis
Path planning
Mechanical effort
UGV
3D navigation

ABSTRACT

Autonomous navigation in rough and dynamic 3D environments is a major challenge for modern robotics. This paper presents a novel traversability analysis and path planning technique that processes 3D point cloud maps to compute terrain gradient information and detect the presence of obstacles to generate efficient paths. These avoid unnecessary slope changes when more conservative paths are available, potentially promoting fuel economy, reducing the wear on the equipment and the associated risks. The proposed approach is shown to outperform existing techniques both in 3D simulation scenarios as well as in a real forest dataset, in which it also generates paths that are comparable to the ones drawn by humans with different backgrounds and expertise levels.

1. Introduction

Forest maintenance is a labour-intensive, time-consuming, expensive, and ultimately dangerous activity, and therefore automation, and consequently robotics, have been studied and tested over the past few years as a viable and desirable solution to overcome these issues [1]. However, forestry robotics requires reliable unsupervised navigation, which is a sizeable challenge considering that forests are highly unstructured, dynamic environments. As such, state-of-the-art techniques have tackled key issues such as autonomous maintenance and management in fully [1] and semi-unstructured [2] forests, as well as in more delicate environments including orchards [3] and plantation fields [4].

This work proposes an innovative path planning technique that uses the concept of mechanical effort [5] to generate efficient paths for an unmanned ground vehicle (UGV). Steep terrain sections are avoided to potentially improve fuel economy and reduce the mechanical effort that the robot is subject to, thus minimizing the wear on the physical components of the robot. For this purpose, a 2D costmap is used to represent the perceived environment, on which traversability analysis and subsequent path planning are performed. The proposed method is tested and compared preliminarily with existing techniques in a 3D Gazebo simulation environment on several scenarios with different characteristics and associated complexity. Afterwards, a study is carried out in which a set of participants use real-world data from a forest scenario to perform a path planning task to further assess the effectiveness of our technique's planning capability when compared to human reasoning, thus providing a solid basis for discussion.

We demonstrate that our method successfully deals with all the tests performed, yielding positive results overall, outperforming other techniques and even the paths generated by the volunteers of the user study, based on the proposed performance metrics.

2. Related work

In this section, we present an overview of relevant work available in the literature in relevant fields, such as environment representation, motion planning and terrain classification, aiming to provide a foundation for understanding the current landscape of research, highlight emerging trends, and identify potential research opportunities.

2.1. Environment representation and mapping

The choice of a representation whose balance between computational efficiency and fidelity is increasingly important for perception and path planning in outdoor environments [6]. Recent endeavours in large-scale and compact models make use of efficient 3D probabilistic data representations such as Octomap [7], where the entire space is discretized according to a 3D grid, in which each cell holds the occupancy probability of that portion of space. Even though this technique

* Corresponding author.
E-mail address: afonso.carvalho@isr.uc.pt (A.E. Carvalho).

https://doi.org/10.1016/j.robot.2023.104560

Available online 14 October 2023

capitalizes on efficient 3D data structures, it still lacks the scalability required for very large workspaces. Another example is elevation mapping [8], which is a 2.5D technique that discretizes the environment into a 2D grid and assigns a height value to each cell, making it a computationally lighter algorithm, at the cost of a lower representation fidelity. In an attempt to overcome some of the shortcomings of 2.5D techniques, Yang et al. [9] present 2.5D-NDT maps which, although still being considered a 2.5D technique, have the ability to represent obstacles such as overhangs and bridges, which is a key improvement.

Other techniques include OVPC meshes [10], which creates a water tight mesh containing traversability information; or fusion of complex physics-based planning with simplified approaches [11], which explicitly address and filter dynamic entities, instead of using traditional volumetric occupancy methods with large processing footprint. Furthermore, relevant work in environment representation explores hierarchical, multi-resolution, and spatio-temporal mapping approaches [12, 13].

2.2. Motion planning and traversability analysis

One of the most seminal works in this field of research is presented in [11]. It describes a motion planner that computes trajectories in the full 6D space of robot poses by identifying the geometry and traversability of surfaces, represented as point clouds without explicitly reconstructing surfaces. The efficiency of the technique is demonstrated in several experimental scenarios, proving to be of particular relevance, given the results obtained, as well as the ability to avoid heavy surface reconstruction computations.

The classical combination of local planning strategies with Vector Field Histogram (VFH) [14] or Dynamic Window Approach (DWA) [15] coupled with global planning search algorithms like A* [16,17] or Dijkstra [18], as well as sampling-based planning techniques, such as Rapidly-exploring Random Trees (RRTs) [19], are still a reference nowadays. However, there are several challenges for fully autonomous navigation in rough outdoor terrains, such as lack of temporal or spatial consistency in the theatre of operations, sensor occlusion or hardware degradation, forcing the robot to plan under a high level of uncertainty. Clearly, there is a shortage of work that tackles autonomous UGV path planning in forest environments considering aspects such as locomotion, maneuverability and mechanical effort.

Nevertheless, recent work specifically applied to rough terrain navigation has been proposed, such as [20], in which the authors present a solution based on fast D* lite as global planner and MPC planning paradigm to solve a constrained optimal control problem for local planning, which accounts for the dynamic model of a vehicle and imposes a variety of constraint models such as obstacles and slippages in real time.

Sebastian et al. [21] proposed a novel physics-based path planner that uses D* Lite and takes into account variables such as terrain slippage, slope, actuator limits and the dynamics of the robot to generate an efficient path.

Jian et al. [22] recently proposed a plane-fitting based uneven terrain navigation (PUTN) method. The method proposes Plane Fitting RRT* (PF-RRT*), which is an improved version of the classic RRT algorithm. Gaussian process regression is used to generate traversability information for the trajectory, and finally nonlinear model predictive control is used for local path planning. The performed experiments reveal a robust method for navigating in rough terrain.

Other recent motion planning techniques based on a variety of concepts have emerged, such as neural networks [23].

In a very recent survey [24] regarding traversability analysis for autonomous ground vehicles, the authors presented various methods which they grouped into vision-based (e.g. [25]), LiDAR-based (e.g. [26,27]), alternative sensor-based (e.g. [28,29]) and fusion-based methods, such as [30]. They provide a thorough discussion regarding the different methods, sensors and challenges on traversability analysis. Of relevance to this work, they concluded that visual-based methods suffer from a few disadvantages, such as a low possibility of generalization when performing feature extraction, and intra-class variability that is hard to represent with specific features. For that reason, deep semantic segmentation networks have been gaining popularity, as they can extract high-level features from the input data. Furthermore, it was concluded that LiDAR-based approaches remain as one of the most powerful strategies for navigation, considering its capability to accurately represent the metric space. However, these approaches suffer from severe limitations when used in classification of geometrically similar surfaces.

2.3. Terrain classification and energy consumption estimation

In the field of autonomous robots, accurate terrain classification involves the categorization of different terrain types based on their physical properties and characteristics, playing a crucial role in enabling robot navigation and effective interaction with the environment.

Exteroceptive data collected by sensors such as cameras and depth sensors provide rich visual information that can be utilized for terrain classification. Computer vision techniques, including image processing, feature extraction, and machine learning algorithms, are commonly employed to analyse visual data and classify terrain based on colour, texture, shape, or other visual cues [31,32].

Prágr et al. [33] proposed a relevant method that uses aerial imagery to build a terrain model that is then used for the extraction of terrain feature where ground units will be navigating. Based on previous traversal experience of ground units in similar environments, the traversal cost model that was learned is employed to predict the traversal cost of the new environment and to plan an efficient path that connects all points of interest. The method achieves promising results in multiple scenarios, where the cost prediction results are visually coherent, and the paths mainly avoid higher cost terrains whenever possible.

Proprioceptive data collected by sensors such as IMUs is also widely used for terrain classification [34–36]. For instance, Faigl et al. [35] utilizes self-organizing maps coupled with unsupervised learning to estimate the traversal cost of the terrain experienced by an hexapod walking robot. Using proprioceptive signals, such as energy consumption and robot stability, combined with exteroceptive information collected by an RGB-D camera, the authors manage to predict the traversal cost of seen but not yet visited terrain.

Waibel et al. [36] present a comparison between two different approaches for estimating the roughness of the terrain: One solely based on LiDAR, which computes terrain cost directly from point clouds using a statistical approach; the other, a learning-based approach, which predicts an IMU output from geometric point cloud data using a 2D-Convolutional-LSTM neural network. Results indicate that both approaches demonstrate versatility for local and global planning, with the learning-based approach not heavily relying on specific terrain features and avoiding the need for threshold tuning on the costmap, thus enhancing its generality. In most cases, the learning-based approach either matches or surpasses the performance of other approaches. Nevertheless, both methods achieve similar results and significantly outperform an occupancy grid by a considerable margin.

Regarding the estimation of energy consumption, Wei et al. [37] proposed a learning-based method that takes terrain geometry and robot motion direction as input, yielding the expected energy consumption. This method manages to obtain a prediction error of 12% with relation to the ground truth, which outperforms by 10% a baseline physics-based method from the literature.

Eder et al. [38] propose a pipeline for traversability analysis and cost estimation for a robotic system on various terrain classes that uses locomotion data, multidimensional terrain classes, and environment segmentation to compute robot-dependent traversability costs.

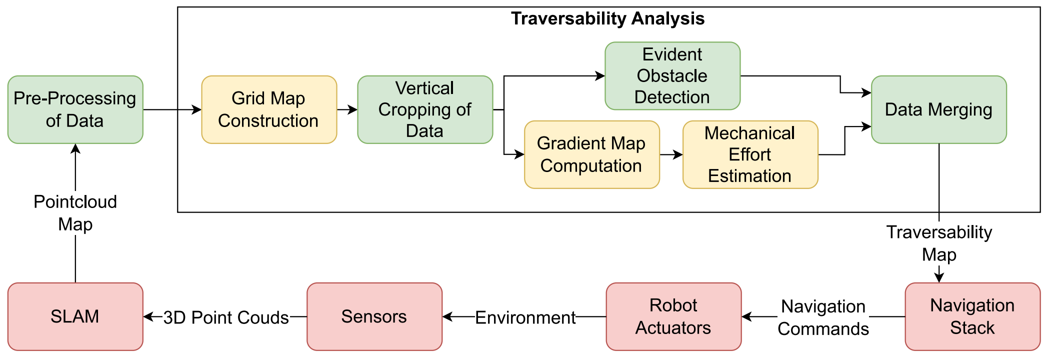


Fig. 1. Proposed System architecture. Green boxes: novel modules; Yellow boxes: modified modules; Red boxes: off-the-shelf modules used "as is". (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

While the experiments showed that the results are dependent on the quality and quantity of recorded locomotion data, the computed costs reflect the capabilities of the robotic systems well, thus confirming the usefulness of the proposed pipeline.

2.4. Statement of contributions

This paper addresses the lack of work in the literature that focus on the analysis and minimization of the traversing effort associated with a path, contributing with a method that:

- aims at minimizing the mechanical effort experienced by a UGV that traverses a forest environment;
- potentially improves fuel efficiency by reducing the overall inclination that the robot is subject to while performing the generated path, considering the strong correlation between terrain gradient and fuel consumption [39] and also that the proposed method generates paths over sections of terrain with reduced gradient;
- runs in real-time while being processing efficient, leading to an increase in data throughput when compared to [5], a point that was left open in the referred work. This improvement is achieved by optimizing the existing software, solving the major processing bottlenecks revealed.

To assess the method's capabilities and performance, a carefully designed experimental setup is conducted using both simulated and real-world environments. The method is compared against existing traversability and motion planning techniques in the literature using recognized performance metrics, and against human planning and traversability reasoning. This article is the extension of an earlier work [40], yet it clearly distinguishes itself from the previous publication by presenting an extended literature review, more extended testing in a simulated environment using an additional strategy based on elevation maps [8], and finally a new set of experiments using real-world forest data.

3. System architecture

Fig. 1 presents an overview of the developed architecture, consisting of three main stages:

1. A pre-processing module responsible for down sampling the input data, thus effectively reducing the amount of points to be further analysed by a series of techniques;
2. A traversability analysis block – the main contribution of this work – responsible for generating 2D global traversability maps based on the mechanical effort concept;
3. An off-the-shelf navigation technique that uses the contents of the traversability map to generate and execute paths.

All modules have been developed using Robot Operating System (ROS) [41] and run concurrently. Our contribution is comprised of the implementation of the green and yellow boxes shown in Fig. 1, as well as overall system integration.

Pre-processing. In this module, a 3D global point cloud map of the environment available *a priori* (whose generation is out of the scope of this work) is filtered to remove points outside the robot's workspace. The raw point cloud is decimated using a 3D voxel grid with a configurable downsample factor, which significantly reduces the number of points (up to 6.5×), depending on density.

Grid map construction. This module is responsible for creating a 2D grid map. It takes as input the decimated point cloud that has been processed before and creates a grid of configurable size and resolution where each 3D point is projected and assigned to a particular 2D grid cell according to its location in space.

Vertical cropping of data. Having then centred the decimated point cloud in the robot's frame of reference, all points that are outside of a configured 3D box with no limit along the Z vertical axis are filtered. This is depicted in Fig. 2(a), where we can observe that if we also crop along the Z-axis up to a certain height, it would result in the elimination of relevant information for path planning, such as trees on higher ground. Afterwards, the point cloud is filtered vertically by performing a cell-wise analysis, where the lowest point (i.e. ground level) in each cell is found, and only points that stand up to a given amount above it are included. This heuristic is demonstrated in Fig. 2(b).

In Fig. 3, we illustrate the impact of applying the heuristic on the generated costmap. The main goal is to reduce the number of points to be analysed and consequently reduce the complexity and computational cost of the entire process.

Gradient map computation. In this step, the terrain is analysed using the gradient at each point. To achieve this goal, we first make sure that every cell of the grid contains only one point. If that is not the case, we use the *median* of the set of points in that particular cell. We chose the median instead of other statistical measures, considering that it is more suitable for filtering outliers, and provides an approximation of the true height of that particular cell.

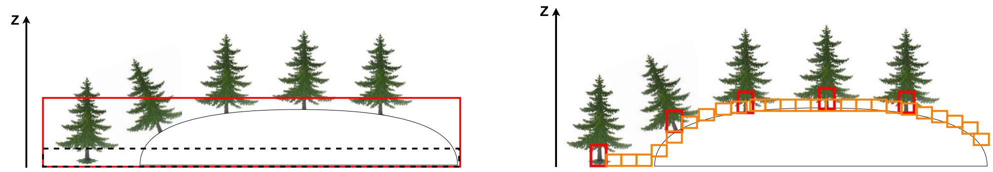
Considering that our goal is to minimize traversing effort and fuel consumption, it is important to note that there is a strong correlation between the gradient of a terrain and the energy consumption required to traverse it [39,42], since the gradient provides information about the direction of greatest height variation.

We compute second order accurate central differences for the interior points,

$$\frac{\partial f}{\partial \lambda} = \frac{f(\lambda + h) - f(\lambda - h)}{2h}, \tag{1}$$

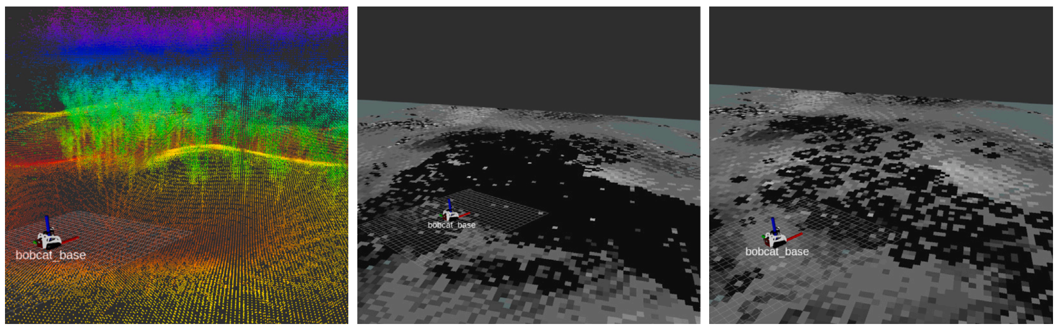
and first order accurate one-side differences at the boundaries:

$$\frac{\partial f}{\partial \lambda} = \frac{f(\lambda + h) - f(\lambda)}{h}, \tag{2}$$



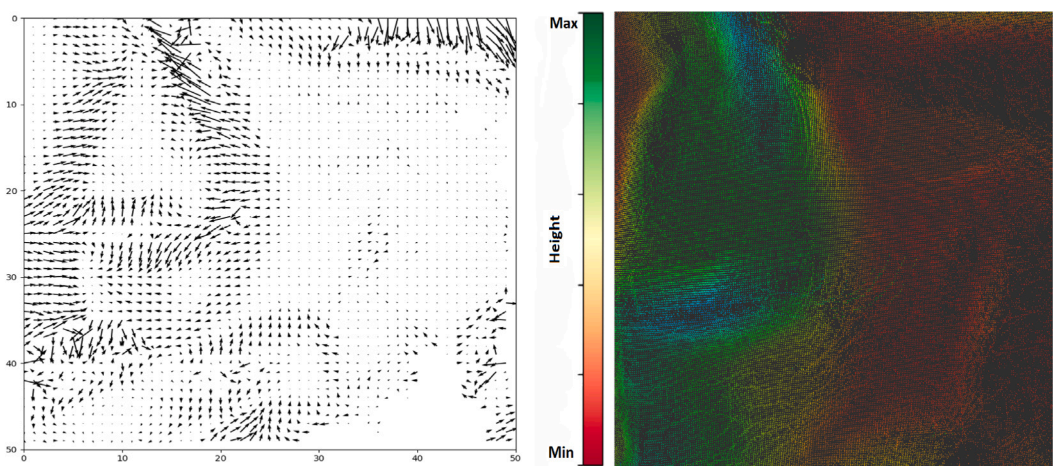
(a) Cropbox with a limit on the Z-axis. (b) Cell by cell analysis of vertical axis.

Fig. 2. Cropping heuristics. The one illustrated in (b) was ultimately selected.



(a) Point cloud to be analyzed. (b) Costmap before applying cropping heuristic. (c) Costmap after applying cropping heuristic.

Fig. 3. Impact of applying the 3D vertical cropping heuristic on the 2D costmap representation (X–Y plane), which outlines existing obstacles more clearly (see black cells).



(a) Gradient Map. (b) Point cloud map (top view).

Fig. 4. Output representation of the gradient analysis stage.

for the left boundaries and

$$\frac{\partial f}{\partial \lambda} = \frac{f(\lambda) - f(\lambda - h)}{h}, \tag{3}$$

for the right boundaries, for each of the dimensions of the gradient, resulting in gradient values that are then clipped to a configurable threshold, i.e.:

$$\nabla f(x_m, y_n) = \begin{cases} \nabla f(x_m, y_n), & \|\nabla f(x_m, y_n)\| < \nabla_{th} \\ \nabla_{th}, & \textbf{otherwise}, \end{cases} \tag{4}$$

where $\nabla f(x_m, y_m)$ represents the gradient in a given cell and ∇_{th} represents the clipping threshold. This allows us to exclude both hills and depressions that exceed this configurable inclination limit. Fig. 4 illustrates this step.

Evident obstacle detection. In this stage, we analyse the filtered point cloud generated from the vertical cropping heuristic applied previously (see Fig. 2). We compute the mean, variance and range of the heights of all points that form each cell, and determine if a cell is to be marked as an evident obstacle by computing:

$$obstacle = \begin{cases} \text{true}, & \begin{cases} \mu_{(x,y)} > \mu_{th} \\ \sigma^2_{(x,y)} > \sigma^2_{th} \\ (max_{(x,y)} - min_{(x,y)}) > \gamma \end{cases} \\ \text{false}, & \textbf{otherwise}, \end{cases} \tag{5}$$

where μ_{th}, σ^2_{th} and γ are the given thresholds for the mean, variance and range of the sample, respectively. By combining these, we are able to predict with a high degree of certainty the presence of obstacles.

Fig. 5 illustrates the output of this step in a forest scenario. It becomes clear that the algorithm considers tree trunks as evident obstacles while discarding tree tops, thus generating a traversable 2D gridmap that would otherwise be unattainable.

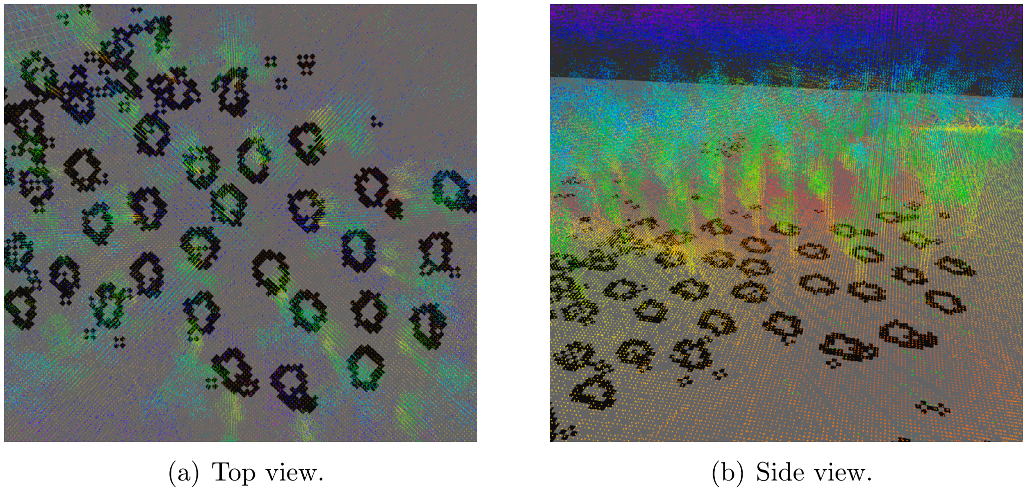


Fig. 5. Evident obstacles detection on a forest scenario.

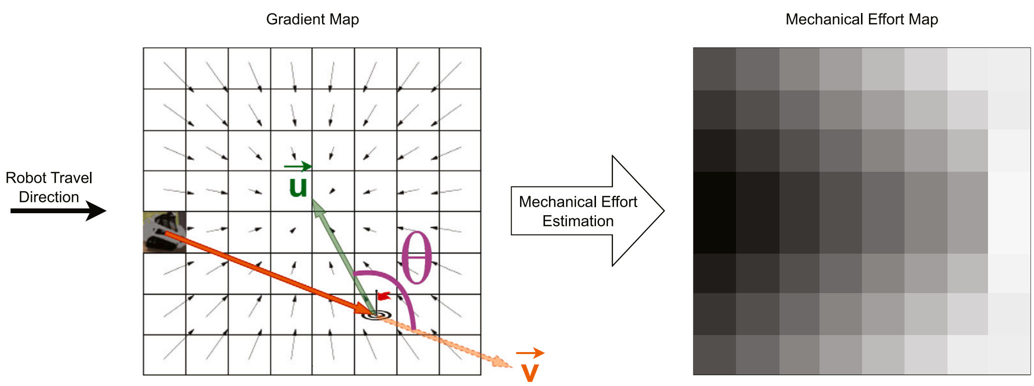


Fig. 6. Illustration of the mechanical effort estimation output on a centred hill in which darker colours represent harder to traverse areas and vice versa. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Mechanical effort estimation. This module estimates the mechanical effort of the terrain based on the previously generated gradient map. In general, when considering a 3D point cloud representing the environment, the gradient along x and y provides very useful information about the structure of the environment. For this, we use the definition of **mechanical effort** presented in Eq. (6):

$$\gamma = \|\nabla f(x, y)\| \cdot \cos\theta, \tag{6}$$

in which θ is the angle between two vectors ($\vec{v}$ and $\vec{u}$ in Fig. 6):

- $\vec{v}$ is the vector that connects the position of the robot with the position of the cell we are currently analysing;
- $\vec{u}$ is the gradient vector in the cell we are currently analysing.

The mechanical effort estimation module takes as input the gradient map (in the case of Fig. 6, of a hill that is equally steep from every angle) and modifies it according to the mechanical effort equation. The output is a 2D gridmap that considers the position of the robot with relation to each cell of the gradient map, where the traversability cost increases as the direction of the robot's motion aligns with the gradient and vice versa.

This map is re-calculated online according to the mechanical effort equation (Eq. (6)), as the robot's position changes due to its movement, and consequently so does the θ angle (see Fig. 6).

Data merging. This module outputs a single 2D global costmap that contains all the useful information from the *evident obstacles costmap* and the *mechanical effort costmap*. All cells identified as evident obstacles are marked as non-traversable ($cost = 100$) in the final costmap, and the remaining ones are marked with an integer ranging from 0 to 99, where 50 represents traversing even terrain, 0 represents the lowest possible traversing cost, and 99 the maximum possible cost of traversing a cell that is not an obstacle.

Notice how the mechanical effort values obtained during the above step can be negative, but the final costmap values range from 0 to 100. This is done by normalizing the mechanical effort values during the conversion to the 2D costmap.

In our model, 50 represents even terrain and 0 represents the steepest descent that is possible to accomplish without exceeding the angular limits of the robot, considering that gradient is the main contributor to fuel consumption [39] and that travelling downhill requires less energy than straight or uphill paths [42].

Remaining (red) blocks of the architecture. The costmap generated in the data merging step is fed into the ROS navigation stack [43]. We use the `move_base` software with a static layer that directly interprets the global costmap and plans accordingly using A* as the global planner, and trajectory rollout as the local planner, thus generating the appropriate velocities commands to follow the planned path.

There are more sophisticated navigation frameworks available, particularly for rough terrain applications. However, since our focus is particularly on the perception axis of outdoor navigation, we have used `move_base` for the actuation axis, as it is a widely tested off-the-shelf method that plans directly on 2D costmaps — the output format of the proposed technique.

Robot control is managed by the local planner which, in spite of planning on an empty local costmap, still has to apply control in order to follow the global path, thus naturally affecting the results, as a consequence of the robot's actuation over the terrain. Importantly, the classical ROS `move_base` software supports only 2D navigation, therefore it would not be applicable in a forest scenario, considering its limitations with floor plans that are not levelled or constant.

The robot actuators receive these velocities and turn them into actual robot motion, consequently changing the scene that the sensors perceive. In turn, the sensors generate new point clouds that are processed by the used Simultaneous Localization and Mapping (SLAM) approach which, in turn, re-injects the updated point cloud map into the system, thus closing the loop.

As for incorporating SLAM updates, which is out of the scope of this work, we would recalculate the global costmap when a new point cloud is originated by the SLAM method in use. However, for simplicity, in all the experiments reported (see Section 4) we utilize a previously mapped representation of the environment instead.

4. Experimental evaluation

We have designed an experimental setup consisting of two main parts: a set of simulated experiments in three scenarios in Gazebo [44] to evaluate the performance of planning and actuation with the proposed method, as described in Section 4.1. Furthermore, we designed an experiment using a real-world, high-quality forest dataset [45] with the participation of 30 volunteers, in which we compared the planned path of our method with their intuition, as well as to other planning techniques, described in more detail in Section 4.2. The dataset mentioned before was collected in Montmorency Forest, Canada by researchers at the Laval University, using a Clearpath A200 Husky mobile robot equipped with a Velodyne HDL32 LiDAR sensor, an Xsens MTI-30 IMU and wheel encoders.

It is important to highlight that comparing techniques against an optimal global path is unfeasible, as it would have to satisfy all the proposed metrics. There are, however almost-optimal paths which can minimize a given set of metrics, as we discuss later on.

4.1. Experimental setup in simulation

In order to set up the simulation scenarios, we use Gazebo's built-in feature of rendering the ground from a heightmap to build two different floors, as well as trees with accurate collision model as obstacles. Furthermore, we use a modified version of the Clearpath Husky robot (which is openly available[1]) with two VLP-16 LiDARs on top. The sensors have no noise in their measurements and error-free localization is provided by the simulator.

Fig. 7 illustrates `Scenario 1`, featuring gentle hills and valleys with the starting pose and the goal being on opposite sides of a hill, as shown in Fig. 7(b). Since most of this scenario is traversable without exceeding the angular limits of the robot, a myriad of different paths can be successfully executed. In Fig. 7(a), we have represented in yellow the approximate path followed by our technique when minimizing the mechanical effort. This path is represented to provide some insight into the reasoning behind the choice of each scenario. Fig. 8 represents `Scenario 2`, which includes steep hills, meaning that the traversing cost across the hills will be significantly higher than flatter terrain, and so will the respective risks. A second version of this scenario was designed, which we call `Scenario 2.1`, having the same spatial configuration, but with a 20% increase in height values, leading to a major difference: some slopes exceed the angular limits of the UGV, becoming untraversable. This slight variation is critical, considering that it marks the difference between steep slopes, and slopes that are mostly impossible to safely traverse.

Fig. 9 represents `Scenario 3`, which besides hilly terrain, also includes a forest of fully grown pine trees. This poses an added challenge to the candidate techniques, as the UGV needs to be able to safely navigate within the trees without colliding with them, while still choosing an efficient path to do so.

Throughout the simulated experiments, we have selected $\mu_{th} = 1$, $\sigma^2_{th} = 0.5$ and $\gamma = 1$ as thresholds for the evident obstacles detector and a value of $\nabla_{th} = 0.55$ for the gradient clipping threshold. These are typical values having in mind the characteristics of the simulated scenarios where the experiments take place, such as the height of the trees; being used throughout all scenarios and runs.

Our method, Mechanical Effort Based Traversability (MEBT), is compared against `Lourenço2020` [5], which originally introduced the concept of mechanical effort, applying it in a local planning approach, in which the path with the lowest perceived mechanical effort is chosen on-the-go and constantly updated as the robot moves through the workspace. We also use `move_base_naive` for comparison, which applies the ROS navigation stack with no observation sources and empty costmaps, following a straight path to the goal that, although expected to underperform against other techniques, establishes a baseline for benchmarking. Finally, we also compare our method against `Elevation Mapping` [8] by extracting an occupancy grid from the elevation map of the terrain.

These methods were chosen for experimental comparison for a couple of reasons. First of all, they are open source and can be configured to run in our experimental setup. Moreover, they are recent methods and relevant to the topic of this work. In addition to the solutions used in this comparison, we also attempted to compare our method against the solution presented in [9]; however, although the code is publicly available, the algorithm could not be reproduced and applied due to a set of unsolvable support issues.

Although all of the compared methods use `move_base`[2] to generate the plans and execute them, `Lourenço2020` uses Dijkstra as a global planner and Trajectory Rollout [46] as a local planner according to the author's original configuration. The remaining solutions use A* as a global planner and Trajectory Rollout as the local planner, all with the same configurations whenever possible. The key difference between the approaches tested is the way that they represent the traversable environment, and all efforts were made to guarantee that the techniques are given the same exact conditions for comparison, namely:

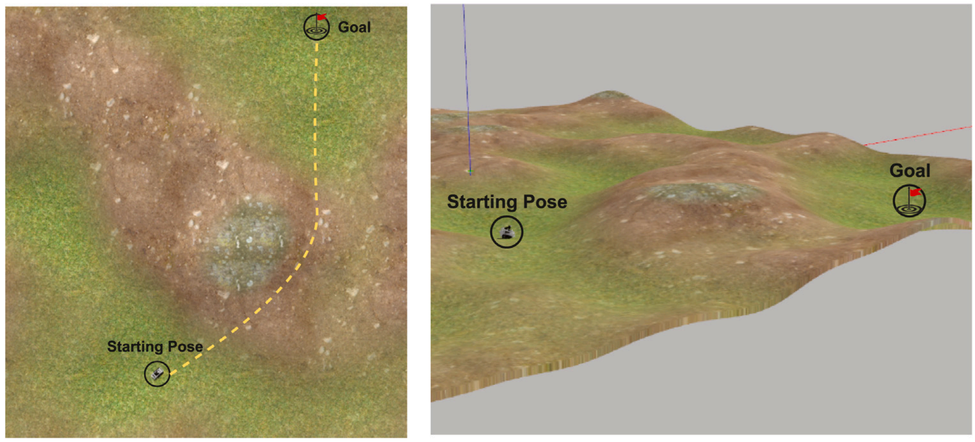
- The global costmaps are derived from the same 3D point cloud that represents the environment, having the same size and resolution, so that the techniques have the same known horizon;
- Both are used with empty local costmaps;
- The elevation map is generated offline from a point cloud of the entire terrain, limited to the same height threshold as our method and later converted to a 2D global costmap of the defined size using original tools from the authors of [8].

To provide the experiments with statistical relevance, accounting for the non-repeatable nature of simulations, 50 navigation trials were conducted in each scenario and for each approach (a total of 800 independent runs). Both our system and `Elevation Mapping` use a 200×200 m global costmap, while `Lourenço2020` uses a 15×15 m local costmap, as per the author's original configuration.

It is worth mentioning that, even though the known horizon of `Lourenço2020` is significantly shorter than our method's, we consider relevant to perform this comparison, as it was the first algorithm proposed using the mechanical effort concept that we also use, and to perceive the improvement in performance upon this technique.

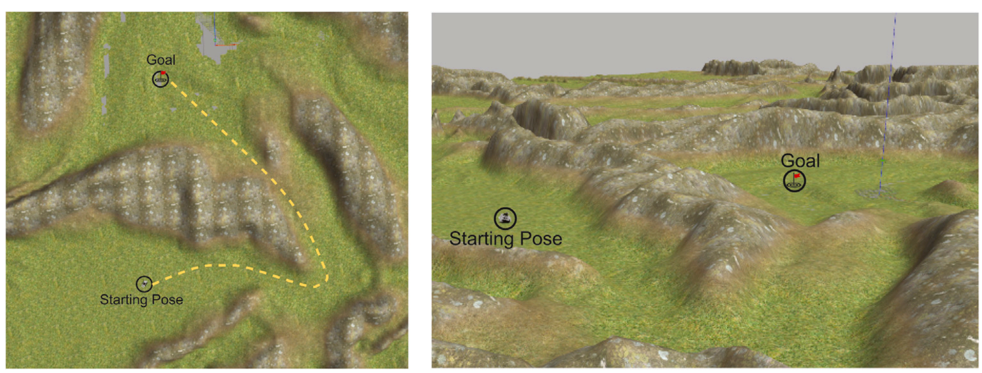
[1] https://github.com/husky/husky

[2] http://wiki.ros.org/move_base



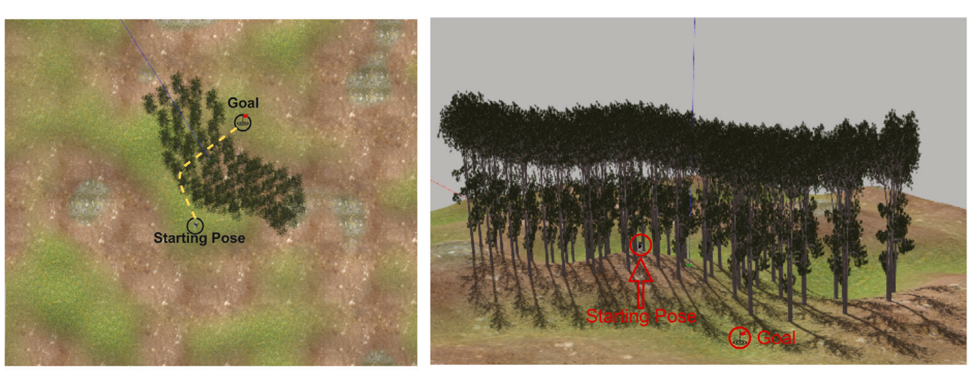
(a) Top view of Scenario 1. (b) Isometric view of Scenario 1.

Fig. 7. First experimental scenario from two different perspectives, with starting pose, goal point and expected approximate path for our technique in yellow. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) Top view of Scenario 2. (b) Isometric view of Scenario 2.

Fig. 8. Second experimental scenario from two different perspectives, with starting pose, goal point and expected approximate path for our technique in yellow. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) Top view of Scenario 3. (b) Front view of Scenario 3.

Fig. 9. Third experimental scenario from two different perspectives, with starting pose, goal point and expected approximate path for our technique in yellow. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

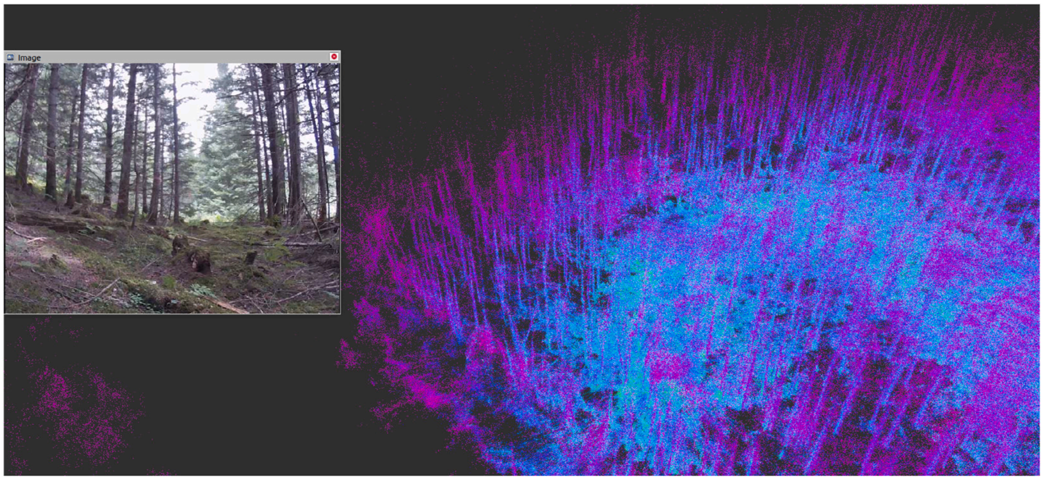


Fig. 10. Illustration of the real forest environment used in the experiments. Point cloud map in the right, and an RGB image in the left.

We have defined the following **Simulation Experimental Objectives**:

S1 Demonstrate that our work manages to create and execute paths with a lower energetic cost, quantitatively defined by metrics described in Section 4.3, when compared to other works in the literature.
S2 Verify that our method outperforms the approach that originally proposed the concept of mechanical effort of terrain traversability.
S3 Show that our method can run in real time.

4.2. Experimental setup with real-world data

In addition to the simulation tests, we performed experiments using the aforementioned forest dataset, with the participation of 30 volunteers that felt comfortable operating a computer mouse, aged between 12 and 60 years old, with backgrounds ranging from robotics and engineering to agriculture and forest maintenance. In our study, none of the participants had prior knowledge about the required task that could affect the results. The participants were divided into 3 groups of 10 people each:

- Non-expert group — Participants who were not from an engineering or forestry background;
- Intermediate group — Participants who had experience in navigating through forests (e.g. forest maintenance workers, lumberjacks, scouts), while still not having any knowledge about robotics and path planning;
- Expert group — Participants who work as robotics engineers, with experience in path planning and/or forestry robotics.

With these characterizations, we intend to understand whether different groups follow distinct approaches to the same problem and, additionally, to which group our technique is closer in terms of performance. In addition, we analyse how the algorithmic approaches of `move_base_naive`,[3] `elevation mapping` and `MEBT` (our method) compare against the different groups, using the same dataset for all trials.

For the real-world experiments, we have used $\mu_{th} = 1$, $\sigma^2_{th} = 0.5$ and $\gamma = 1.5$ as thresholds for the evident obstacles detector and a value of $\nabla_{th} = 1$, following the same rationale as previously for the simulated experiments.

Given that we are now dealing with data collected in a real forest scenario (see Fig. 10), we are able to further test the consistency of our method when compared to the simulated scenarios. Since this work mainly addresses traversability analysis and global planning, these experiments focus on the planning phase of the broader navigation problem. As such, we do not consider the actuation/locomotion phase, as it strongly depends on local planning.

At the start of each trial, the age and background of the volunteer were registered, and a script (see Appendix) containing the necessary explanations and instructions were read by the experimenter, while a top-down view of an elevation map (see Fig. 11) is presented. The volunteer was then asked to draw a continuous path, from start to finish, which he/she believed to be the most efficient while trying to minimize the mechanical effort, after this concept had been properly defined. Volunteers had the possibility of repeating the process as many times as necessary until they were fully satisfied with the path they had laid out.

Using an image manipulation tool,[4] we collected the drawn paths and processed them with a path traversing algorithm in order to obtain the coordinates of all the pixels in each path from one end to the other. From there, the pixels were converted into (x, y) coordinate values, and the height values (z) were obtained from the elevation map that is provided to the users. Metrics were then extracted from the obtained real-world (x, y, z) coordinates of the drawn paths.

We have defined the following **Real-World Data Experimental Objectives**:

R1 Demonstrate that our method is applicable both in simulation and with real world data;
R2 Verify that the method succeeds in generating paths that are more efficient than, at least, the non-expert and intermediate volunteer groups;
R3 Show that the method outperforms `elevation mapping` using the expert group results as a standard, and relevant performance metrics.

4.3. Performance metrics

In order to quantify performance and compare methods, we have defined the following performance metrics:

Elapsed time. The total time that it takes for the UGV to go from the starting pose to the end goal. A method should minimize the elapsed time without compromising other more important factors. This metric helps to sustain objective S2 from the simulation experiments.

[3] Only as a baseline for comparisons between techniques.

[4] https://www.gimp.org

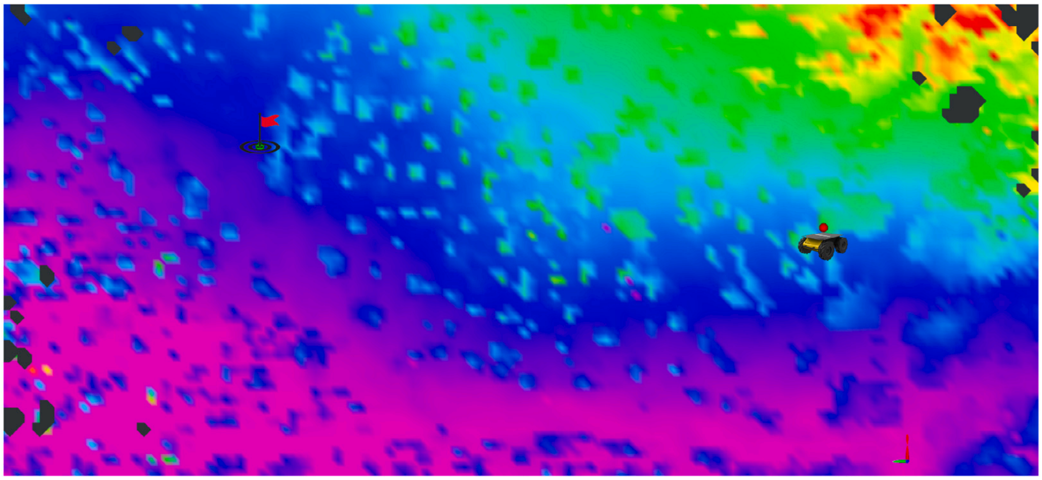


Fig. 11. Path drawing environment used in the volunteer study. According to the rainbow scale, magenta represents the lower terrain, followed by blue, cyan, green, yellow and red as elevation rises. The robot represents the starting point of the path to be traversed, while the flag represents the final goal. The image represents a $55\,\mathrm{m} \times 25\,\mathrm{m}$ ($1375\,\mathrm{m}^2$) area of forest terrain. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Travelled distance. The distance travelled by the UGV from the starting pose to the end goal. A technique should be rewarded for reducing the travelled distance, with some exceptions such as not compromising/worsening some other important metrics. This metric helps demonstrating objectives S1 and S2 from the simulation experiments and objectives R1, R2 and R3 from the real data experiments.

The total travelled distance is computed using the robot's positions along the path, according to:

$$D = \sum_{k=1}^{N} \sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2 + (z_k - z_{k-1})^2}, \tag{7}$$

where x_τ, y_τ, z_τ are the robot's positions coordinates in each of the axis (x,y,z), respectively, with τ representing the current (k) and previous ($k-1$) iterations, and N the total number of iterations.

Mean Map Generation Time (MMGT). The time it takes to generate a new costmap. For the purpose of this work, we consider that any approach runs in real time if a new map is generated before the UGV reaches the end of the current one. This represents the difference between the UGV navigating known or unknown terrain. This metric helps demonstrating objectives S2 and S3 from the simulated experiments.

Positive height variation. The cumulative vertical distance travelled by the UGV. In general, a technique should attempt to minimize this metric, as the positive gradient of the terrain is the main factor affecting fuel economy [39].

Each iteration of this metric is calculated according to:

$$\beta_k^+ = \begin{cases} z_k - z_{k-1}, & z_k - z_{k-1} > 0 \\ 0, & \text{otherwise}, \end{cases} \tag{8}$$

being the total positive height variation given by:

$$\beta^+ = \sum_{k=1}^{N} \beta_k^+. \tag{9}$$

This metric helps to demonstrate objective S1 from the simulated experiments and objectives R1, R2 and R3 from the real data experiments.

Mean effort. The mean effort that a technique faces while executing the given path. This metric is measured by averaging all the absolute UGV's pitch orientation (θ) values along the executed trajectory, according to:

$$\Psi = \frac{\sum_{k=1}^{N} \|\theta_k\|}{N}, \tag{10}$$

in which Ψ represents the mean effort, θ represents the pitch value, N represents the total number of iterations and k the current iteration.

Considering that it gives an indication of the mean vertical travelling angle, which has a direct correlation with the effort of the chosen path, and that even downhill paths represent a positive fuel consumption, according to the properties of a general UGV without regenerative braking, the candidate techniques should aim at minimizing it.

Naturally, this metric helps to demonstrate objective number S1 from the simulated experiments.

Path riskiness index. The path riskiness index metric indicates how risky the travelled path is. In this context, we define a "risky" situation as one where the UGV travels while exceeding any of its angular limits, and we quantify it as the percentage of time that the UGV exceeded said limits in the travelled path. This metric should be minimized and it helps demonstrating objective S2 from the simulated experiments.

Roll danger index. The mean percentage of roll-related risk that the UGV takes during the execution of its path. The absolute roll values are filtered with the following sigmoid function:

$$RDI = \frac{1}{1 + e^{-(0.5\phi - 13)}}, \tag{11}$$

which de-linearizes the penalty curve of the UGV's inclination. Considering the roll angle limit to be 35°, it is almost equally safe to travel with between 5°-10° of inclination. However, the same does not apply to the 5° difference between 25°-30° of inclination, because the stability of the vehicle rapidly decreases when approaching its angular limits.

The sigmoid values in Eq. (11) were iteratively refined and selected so that the shape and inflection points of the curve reflected this behaviour. This metrics helps demonstrating objective R2.

Pitch danger index. The mean percentage of pitch-related risk that the UGV takes during the execution of its path. Using the same rationale, and thus also supporting objective R2, the absolute pitch values are filtered through the following sigmoid function:

$$PDI = \frac{1}{1 + e^{-(0.25\theta - 6)}}. \tag{12}$$

Failure rate. We define the failure rate as the percentage of times that the path planner aborts the whole mission due to one of the following reasons:

- It has not been able to follow the selected path (e.g. an internal error in the navigation module).

Table 1
Results from 50 runs in scenario 1 (Fig. 7).

Metrics	Lourenço2020 [5]	move_base_naive	MEBT	Elevation map [8]
Failure rate (%)	10.000	8.000	8.000	**0.000**
Mean effort (°)	11.701 ± 1.064	12.520 ± 0.047	**10.202 ± 0.423**	10.636 ± 0.040
Pitch danger index (%)	11.290 ± 2.863	13.758 ± 0.136	8.796 ± 1.141	**8.790 ± 0.140**
Up variation (m)	10.433 ± 0.938	9.058 ± 0.017	8.666 ± 0.559	**7.808 ± 0.032**
Travelled distance (m)	112.839 ± 8.005	**89.645 ± 0.053**	106.388 ± 4.073	91.160 ± 0.386
Path riskiness index (%)	0.112 ± 0.732	**0.000 ± 0.000**	0.068 ± 0.201	**0.000 ± 0.000**
Roll danger index (%)	1.903 ± 1.428	**0.016 ± 0.001**	1.888 ± 0.652	0.035 ± 0.017
Mean map gen. time (secs @ pts (pts/sec))	11.3 @ 37.6k (3.3k)	N/A	15.4 @ 210k (13.6k)	N/A
Elapsed time (s)	279.612 ± 49.750	**87.031 ± 23.229**	111.326 ± 12.938	94.303 ± 0.527

- It has not been able to create a valid or feasible plan to the targeted goal.
- The specified timeout for the mission has been exceeded. We have used a timeout of 350 s in our experiments.
- The UGV rolled over.

This metric keeps track of failed runs regardless of the fail source, considering that we want to verify the overall robustness of the proposed system instead of only the traversability module, thus providing a solid measure of a system's reliability and robustness, helping to demonstrate objective S2 from the simulated experiments.

Regarding the experiments with the real dataset and considering that we only have the (x, y, z) coordinates along the planned paths, the metrics described above that rely on robot actuation do not apply. To evaluate these experiments, we make use of the **Travelled Distance**, the **Positive Height Variation** and the **Difference From the Best Volunteer's Path (DFBVP)** as quantitative metrics. This last metric is the percentage difference of the two last metrics between each of the compared techniques and the volunteer's path which generated the lowest value for both the travelled distance and the height variation metrics. We also qualitatively compare paths generated with our technique to the paths of the participant groups and to the ones from the remaining techniques, in order to understand to which group it is closer to in terms of the morphology of the path and how it compares against other techniques.

For all experiments, only successful runs count towards the metrics' scores.

5. Simulated results and discussion

In this section, we discuss the findings of running the four methods described on the simulation scenarios defined previously. These results were also partially presented as previous work in [40].

As a preliminary note, a major contributing factor to the number of failures obtained, is the adoption of the standard `move_base` behaviour that does not initiate path re-planning when a planned path is aborted due to various reasons, leading to an increase in failure rate in some cases. However, it is important to emphasize that this behaviour helps to mitigate the adverse effect that re-planning could have in the adopted metrics, e.g. by leading the robot through atypical paths or deteriorating some metrics' values, and possibly hindering the study at hand.

Also, it becomes clear that our technique is able to process significantly more points per second than the earlier technique described in [5]. This has been achieved by optimizing the existing software, as mentioned in the contributions of this work (cf. Section 2). A processing bottleneck was unveiled by a profiling tool, and it was subsequently solved by utilizing efficient iterators instead of resource-intensive nested loops. In connection with this, throughout the experiments, the MMGT metric helps to demonstrate the improved efficiency of our method, whom is now able to process several times more points per second in larger maps than the previous work based on mechanical effort.

The results of the first experiment performed, in Scenario 1 (Fig. 7), are presented in Table 1. This is the scenario with the slightest hills that are generally traversable and with no obstacles, thus representing little to no risk for the robot.

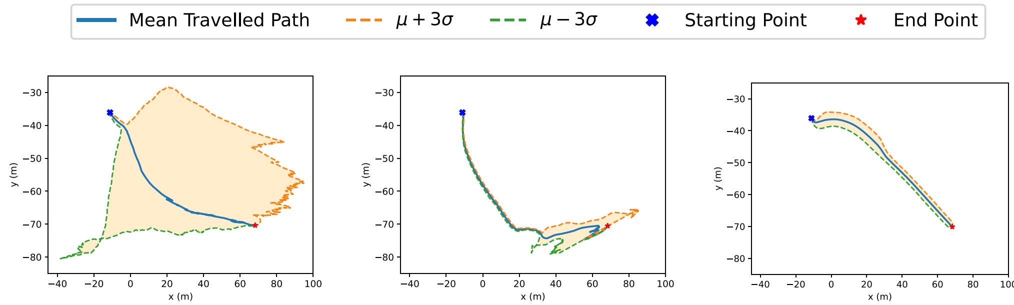
Although the paths generated in Scenario 1 differ significantly between our method and `elevation map` (as shown in Fig. 12), the differences in the metrics of all systems, with the exception of elapsed time, are relatively small. This is an expected result given the scenario complexity considerations mentioned above. Nonetheless, as shown in Table 1, even though our method has succeeded in generating paths that resulted in the lowest mean effort among all the techniques, which we prioritize in this work, move_base_naive and elevation map actually produce better results in all the other metrics. These results are not too significant, however, for a couple of reasons. For instance, as previously mentioned, this scenario represents little to no risk for the robot independently of the chosen path, which makes the relatively small differences in the risk metrics negligible. Specifically regarding the roll danger index, it is noteworthy that all values are negligible considering that even the worst performing system only travelled, on average, with an inclination of about 1.9% of the limit.

Table 2 and Fig. 13 present the results in Scenario 2. Instead of choosing straighter paths, like the remaining techniques did, which would represent a shorter elapsed time and travel distance at the cost of major risk (especially pitch related risk) and significantly higher mechanical effort, our method chooses a longer path with less relief, which is around the hill. By doing so, it is able to significantly reduce the effort at the cost of travelled distance, and potentially the fuel consumption associated with that trajectory, given that it is much strongly correlated with terrain gradient than with travelled distance (see [39]).

The pitch danger index is significantly lower than in other techniques, while the roll danger index is not so low, especially when compared to `move_base_naive` and `elevation map`. This is justified since, while travelling around the hill, the UGV tries to minimize the travelled distance without incurring in excessive risk, thus travelling on side slopes for the majority of the path, due to the morphology of the terrain in that area. By doing so, our method manages to greatly reduce the effort associated with its path while only travelling on average at 12.8% of the maximum sideways inclination allowed (given by the roll danger index), while `Lourenço2020` travels at a considerably higher 26.6%, since it detects the presence of a significant slope ahead, but is unable to find a better path due to its short planning horizon. Therefore, it travels sideways on the slope while attempting to find a better path, a behaviour that is not ideal, considering that it tends to increase the danger and riskiness indexes and the elapsed time.

One way to further reduce this danger index would be to parameterize the method to be more conservative with the inclinations. However, this would most likely lead to inferior results in other metrics, such as mean effort, travelled distance and elapsed time.

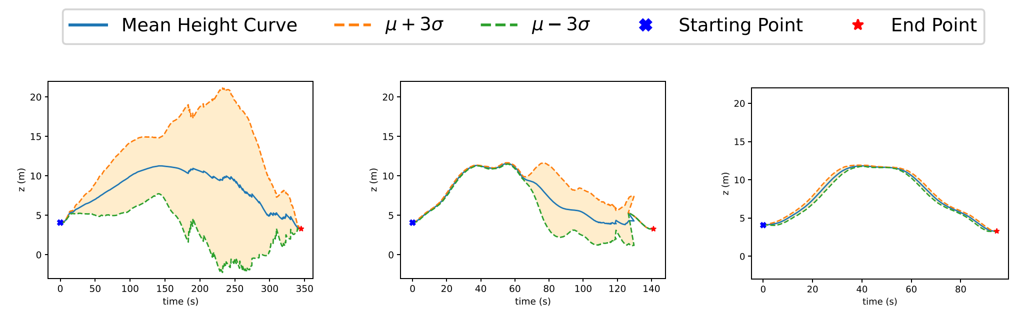
`Elevation map`, on the other hand, is the method which yields the lowest roll and path danger indexes at the cost of ascending major slopes and dealing with major pitch-related inclination, which in these experiments is not the ideal behaviour.



(a) 50 paths produced by the system from [5] projected on a 2D plane.

(b) 50 paths produced by MEBT projected on a 2D plane.

(c) 50 paths produced by elevation mapping projected on a 2D plane.

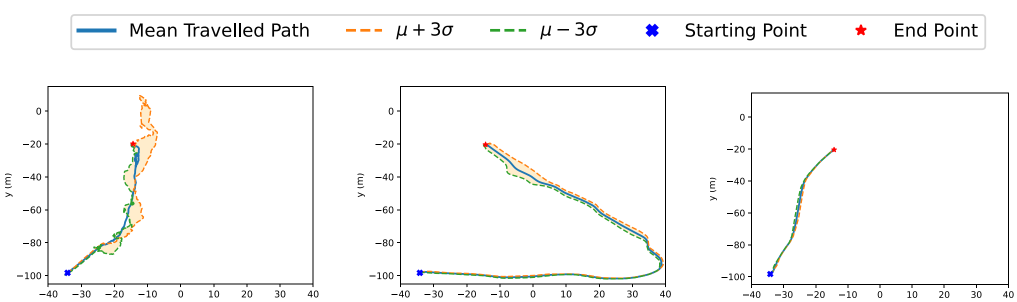


(d) Elevation as a function of time for 50 paths produced by the system from [5].

(e) Elevation as a function of time for 50 paths produced by MEBT.

(f) Elevation as a function of time for 50 paths produced by elevation mapping.

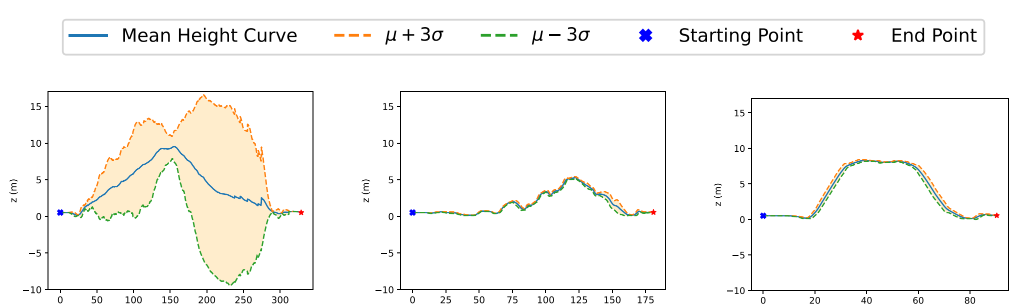
Fig. 12. Resulting plots from 50 runs in Scenario 1 (Fig. 7) with `Lourenço2020` [5] in the left, MEBT in the middle and `elevation map` in the right.



(a) 50 paths produced by the system from [5] projected on a 2D plane.

(b) 50 paths produced by MEBT projected on a 2D plane.

(c) 50 paths produced by elevation mapping projected on a 2D plane.



(d) Elevation as a function of time for 50 paths produced by the system from [5].

(e) Elevation as a function of time for 50 paths produced by MEBT.

(f) Elevation as a function of time for 50 paths produced by elevation mapping.

Fig. 13. Resulting plots from 50 runs in Scenario 2 (Fig. 8) with `Lourenço2020` [5] in the left, MEBT in the middle and `elevation map` in the right.

Table 2
Results from 50 runs in scenario 2 (Fig. 8). Best values highlighted in bold.

Metrics	Lourenço2020	move_base_naive	MEBT	Elevation map [8]
Failure rate (%)	84.000	**2.000**	**2.000**	4.000
Mean effort (°)	18.661 ± 1.828	12.805 ± 0.205	**5.601 ± 0.060**	12.547 ± 0.148
Pitch danger index (%)	38.592 ± 4.892	25.406 ± 0.335	**2.814 ± 0.142**	24.022 ± 0.316
Up variation (m)	15.240 ± 1.886	8.877 ± 0.036	**8.183 ± 0.088**	8.885 ± 0.049
Travelled distance (m)	112.387 ± 8.066	**84.958 ± 0.068**	171.887 ± 0.343	85.672 ± 0.077
Path riskiness index (%)	4.291 ± 4.094	0.517 ± 0.650	0.932 ± 0.212	**0.292 ± 0.532**
Roll danger index (%)	26.580 ± 4.648	0.042 ± 0.137	12.843 ± 0.695	**0.020 ± 0.012**
Mean map gen. time (secs @ pts (pts/sec))	8.2 @ 40.1k (4.9k)	N/A	18.3 @ 287k (15.7k)	N/A
Elapsed time (s)	343.582 ± 14.886	87.014 ± 12.689	174.368 ± 4.184	**86.695 ± 16.368**

Table 3
Results from 50 runs in scenario 2.1. Best values highlighted in bold.

Metrics	Lourenço2020	move_base_naive	MEBT	Elevation map [8]
Failure rate (%)	100.000	90.000	**10.000**	86.000
Mean effort (°)	N/A	17.172 ± 0.902	**7.004 ± 0.301**	16.022 ± 0.405
Pitch danger index (%)	N/A	32.131 ± 1.199	**5.807 ± 0.653**	29.652 ± 0.484
Up variation (m)	N/A	11.586 ± 0.050	**10.319 ± 0.476**	11.124 ± 0.063
Travelled distance (m)	N/A	**87.873 ± 0.399**	175.689 ± 1.894	88.004 ± 0.311
Path riskiness index (%)	N/A	12.558 ± 0.853	**2.839 ± 1.134**	11.138 ± 1.255
Roll danger index (%)	N/A	0.807 ± 1.413	12.788 ± 2.581	**0.345 ± 0.674**
Mean map gen. time (secs @ pts (pts/sec))	8.9 @ 40.5k (4.6k)	N/A	18.3 @ 287k (15.7k)	N/A
Elapsed time (s)	N/A	**25.299 ± 2.510**	173.037 ± 31.038	29.740 ± 3.072

Table 4
Results from 50 runs in scenario 3 (Fig. 9). Best values highlighted in bold.

Metrics	Lourenço2020	move_base_naive	MEBT	Elevation map [8]
Failure rate (%)	58.000	92.000	**6.000**	32.000
Mean effort (°)	11.024 ± 3.500	15.188 ± 0.100	**3.595 ± 0.066**	7.219 ± 0.187
Pitch danger index (%)	9.326 ± 8.016	23.300 ± 0.565	**0.794 ± 0.015**	3.484 ± 0.095
Up variation (m)	7.879 ± 0.496	7.035 ± 0.066	**3.714 ± 0.045**	4.823 ± 0.141
Travelled distance (m)	80.226 ± 9.643	**58.299 ± 0.497**	85.586 ± 0.493	63.434 ± 1.149
Path riskiness index (%)	1.353 ± 5.998	0.767 ± 0.701	**0.000 ± 0.000**	**0.000 ± 0.000**
Roll danger index (%)	0.297 ± 0.297	0.909 ± 0.469	**0.020 ± 0.001**	0.233 ± 0.035
Mean map gen. time (secs @ pts (pts/sec))	12.1 @ 42.4k (3.5k)	N/A	20.2 @ 358k (17.7k)	N/A
Elapsed time (s)	340.362 ± 123.149	182.514 ± 193.306	88.266 ± 12.011	**58.712 ± 16.341**

Table 3 shows the results from running the same experiment in Scenario 2.1. We can see that by slightly increasing the slope angles, the other systems perform drastically worse, while ours presents similar results to Scenario 2. The large failure rates observed are no surprise, as this scenario penalizes any technique that chooses to ignore its large hills, reinforcing the importance of avoiding the steepest sections of terrain and ultimately showing the robustness and reliability of our system, while also supporting simulation objectives S1 and S2.

Lastly, Table 4 and Fig. 14 present the results of the experiments in Scenario 3[5] (Fig. 9), arguably the most demanding scenario, considering not only that the UGV must avoid every tree in its path, but it should also choose the path with the lowest effort associated. The paths chosen by our technique do not lead to any situations where the inclination of the terrain is such that the UGV exceeds the roll and pitch limits, i.e. the path riskiness index is always zero.

In general, our method substantially outperforms the remaining ones in every metric except travelled distance, which is slightly higher. The small increase in travel distance results in a significant difference in the mean effort and up variation, the main variables of interest in our study for safe and efficient terrain traversability, therefore posing a very reasonable trade-off.

Notice, however, how the approach based on elevation mapping was also superior to the other techniques, apart from ours, by a considerable margin.

As stated before, the consistency and robustness of our technique is patent in the results presented, especially in the failure rate and mean effort. Moreover, our system significantly outperforms Lourenço2020 [5] in Mean Map Generation Time. This metric is expressed in `seconds @ points`, providing the time it took to generate each map and the number of points that it contains, on average.

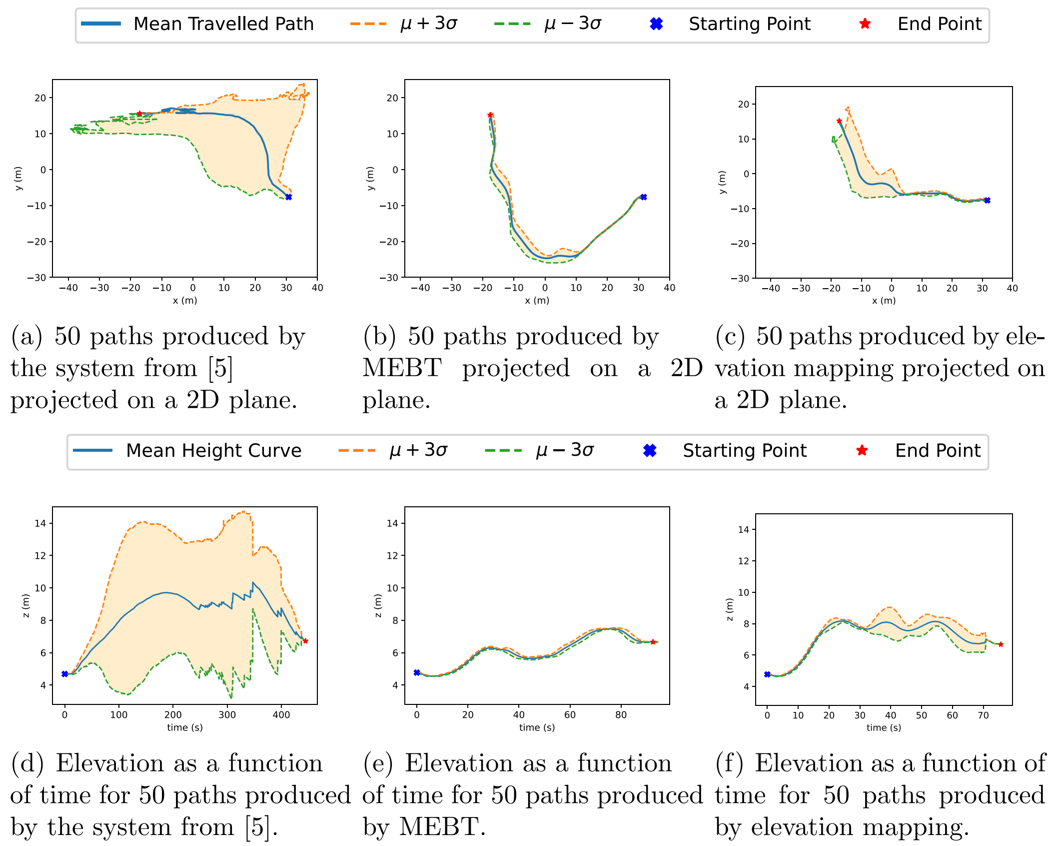
6. Real-world results and discussion

Figs. 15–17 represent the resulting paths from all the users in the real data experiments. As expected, the **non-expert group** is the one with the highest variation in path planning strategy, followed by the **intermediate group** who show a more systematic and conservative approach, vastly prioritizing descents and clear paths over cluttered ones. Finally, the **expert group** shows the highest mastery in all aspects of the study, producing very similar paths to one another with the exception of a single outlier.

In Fig. 17, we can clearly observe the resemblance between the path generated with **MEBT** and most paths from the expert group. This is quantitatively confirmed by Table 5, which reveals that our technique apparently follows an approach that is similar to the average approach of the participants taken as the gold standard[6] (i.e., the group

[5] A video of our system in Scenario 3 is available at https://youtu.be/2wSMTPBg0ZU.

[6] "Gold standard" in this context means the best possible benchmark available under reasonable conditions (*cf.* Oxford Dictionary of English, "*a thing of superior quality which serves as a point of reference against which other*



(a) 50 paths produced by the system from [5] projected on a 2D plane.

(b) 50 paths produced by MEBT projected on a 2D plane.

(c) 50 paths produced by elevation mapping projected on a 2D plane.

(d) Elevation as a function of time for 50 paths produced by the system from [5].

(e) Elevation as a function of time for 50 paths produced by MEBT.

(f) Elevation as a function of time for 50 paths produced by elevation mapping.

Fig. 14. Resulting plots from 50 runs in Scenario 3 (Fig. 9) with `Lourenço2020` [5] in the left, MEBT in the middle and `elevation map` in the right.

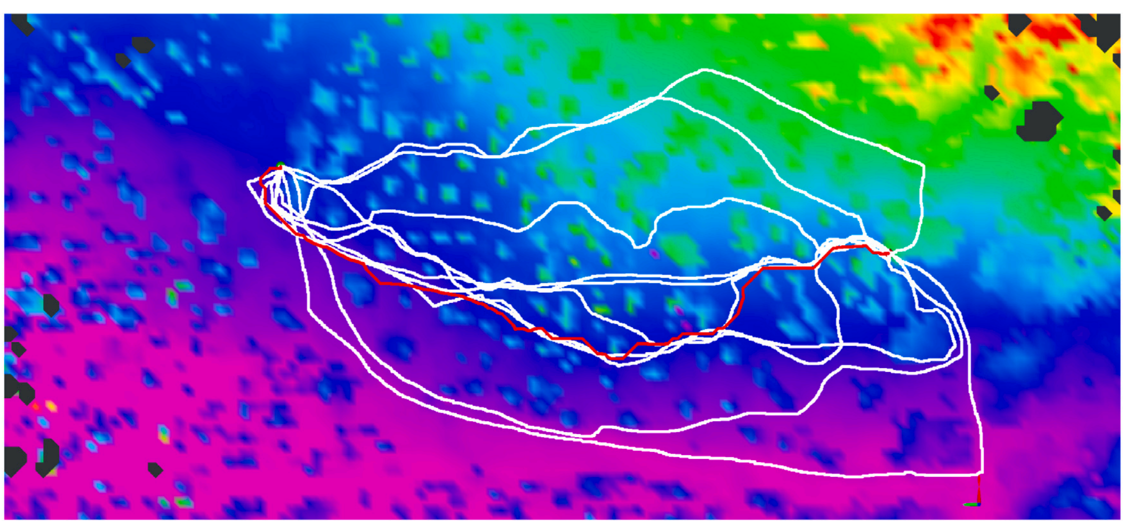


Fig. 15. Resulting paths of the non-expert group, with our technique's path shown in red. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

of experts) closer than any of the other methods, given the chosen metrics. We consider the average approach as the mean curve that can be visually inferred from the representation of all the chosen paths from a group of participants, in this case the experts. This curve is calculated point by point by considering all the values from each expert's path within the same vertical line.

This fact has a few important implications:

1. It suggests that, on average, participants (and, in particular, those who have experience in driving vehicles in woodlands)

things of its type may be compared"), as opposed to an optimal path, which is unfeasible to obtain under this study's circumstances.

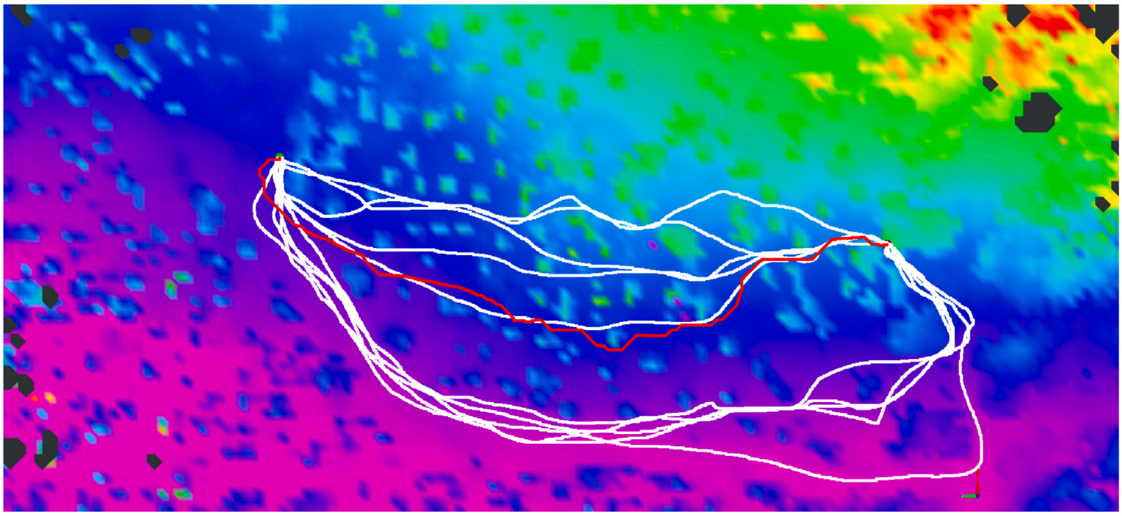


Fig. 16. Resulting paths of the intermediate group, with our technique's path shown in red. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

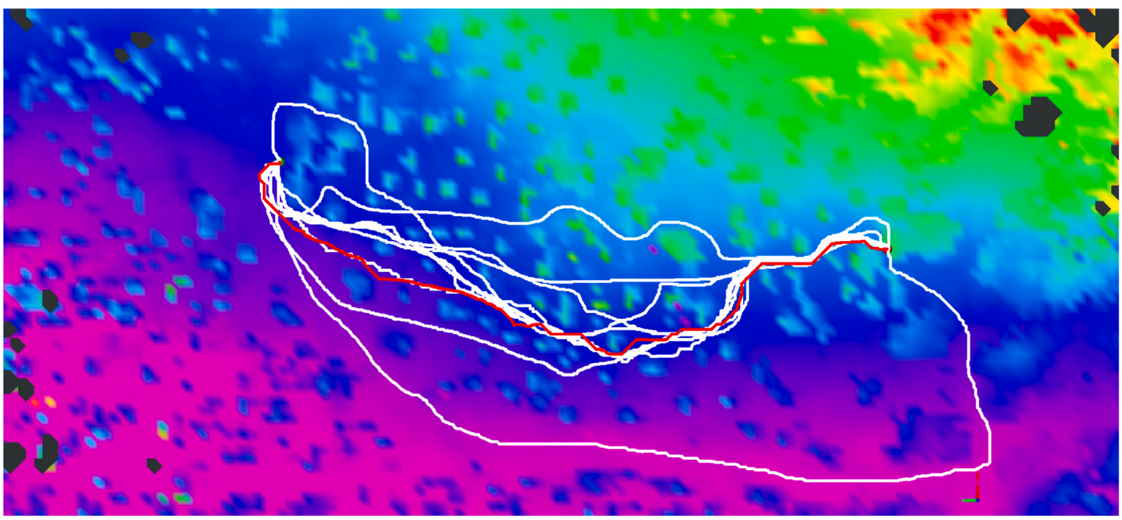


Fig. 17. Resulting paths of the expert group, with our technique's path shown in red. Note the single outlier, which is the path drawn closer to the bottom of the figure. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 5
Quantitative results from the user study in terms of positive height variation (Up variation) and travelled distance ($D_{Travelled}$), including percentage difference from each technique to the mean of the expert group (Diff_{MEG}) and to the best volunteer's path (Diff_{BVP}).

Approaches	Up variation (m)	Diff_{MEG} (%)	Diff_{BVP} (%)	$D_{Travelled}$ (m)	Diff_{MEG} (%)	Diff_{BVP} (%)
Non-expert group	7.448 ± 2.984	+25.16%	+116.95%	59.817 ± 9.259	+8.48%	+23.28%
Intermediate group	7.352 ± 1.905	+23.54%	+114.16%	60.141 ± 7.412	+9.07%	+23.95%
Expert group	5.951 ± 2.054	0.00%	+73.35%	55.141 ± 8.080	0.00%	+13.64%
MEBT	**4.743**	**−20.29%**	+38.16%	**50.690**	**−8.07%**	+4.47%
move_base_naive	26.321	+342.30%	+666.71%	88.903	+61.23%	+83.22%
Elevation map 1	10.370	+74.26%	+202.07%	67.709	+22.79%	+39.54%
Elevation map 2	14.492	+143.52%	+322.138%	67.231	+21.93%	+38.56%
Teleoperated	9.288	+56.07%	+170.55%	64.818	+17.55%	+26.44%

seem to be taking mechanical effort into account in their path planning, which would validate our primary assumption.

2. Although the empirical conditions of the experiment,[7] may also have an effect on human path planning, mechanical effort might not be the only issue being factored in by participants, as our method achieves superior scores on average with the benchmarking metrics. In any case, one can safely infer that the results indicate that it seems to be one of the most important factors.

Fig. 18 shows the resulting paths of all the methods compared, as well as the **teleoperated path** – the one generated by the human operator during the collection of the dataset, from the same origin to the same goal as all the remaining ones – and the **best volunteer's path** drawn by a user during our study, which is the one with the lowest values for both performance metrics considered. Notice how

[7] E.g. the robot is not actually being driven by the participants *in loco* and these do not have access to a quantitative representation of the relief.

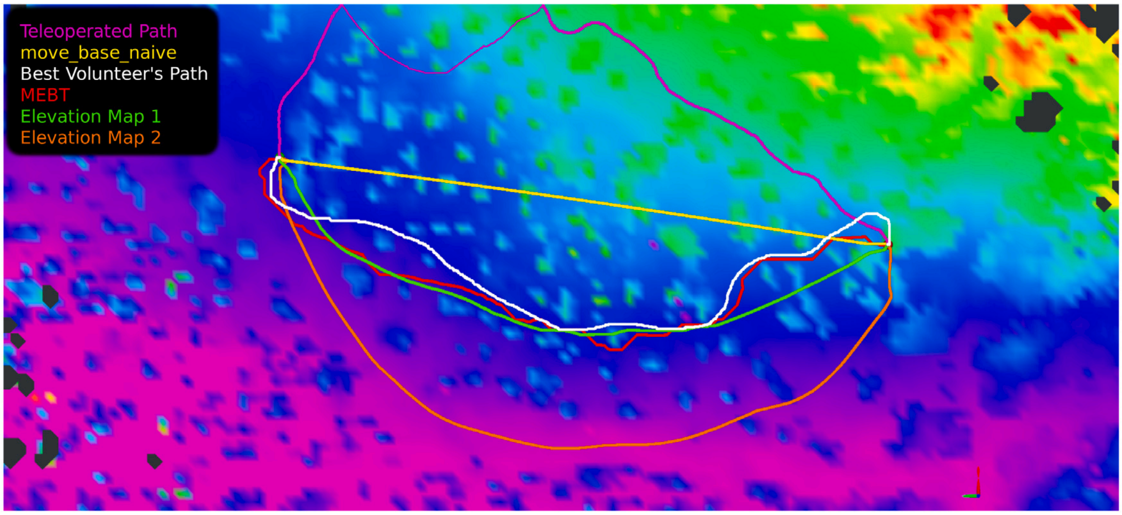


Fig. 18. Overview of paths generated by each of the methods under consideration: The **teleoperated path** (pink) results from the remote control of the robot by a human operator, as recorded in the dataset; **move_base_naive** (yellow) plans a straight path to the goal (see Section 4.1); the **best volunteer's path** (white) is the one that scored both the least up variation and travelled distance from all volunteers' paths; **MEBT**'s (our method) path (red), and finally; **elevation map**'s paths (green and orange) with two different configurations (best viewed in colour). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

there are two different paths generated with `elevation mapping`. They simply serve to show two different configurations, one more conservative than the other, and the respective metrics' values.

Furthermore, Table 5 shows the results of up variation and travelled distance obtained by all methods and the mean results for each user group, as well as the percentage difference from each method to the mean expert group (MEP) and the best volunteer's path (BVP). In this section, the system from [5] is not used, mainly due to its poor results in the simulation study and to the fact that it has a limited planning horizon, therefore it relies on actuation to generate a full path from starting to end point, which would not be possible using a dataset. As we can see, our method manages to greatly outperform all the others with respect to both metrics, which would make for a safe assumption that, if the path from our technique had been drawn by a human participant, they would probably have a background framed in the expert group.

These results are partially predictable, considering that **move_base_naive** would generate a straight path to the goal with no regards to terrain relief and obstacles, which penalizes both metrics — the up variation since it climbs all obstacles in its path, and the travelled distance as it is computed along the 3 spatial dimensions. In the case of the **elevation map**, as we stated earlier, we are planning on an occupancy grid extracted from the entire elevation map, and thus the global planner tries to avoid the major hill constituted by the overall terrain slope and not so much all the individual obstacles.

When it comes to the teleoperated path, it interestingly goes through the higher part of the terrain and still manages to obtain a better score according to our metrics than both the `elevation map` and `move_base_naive`'s baseline. That happens due to the fact that an expert human operator is teleoperating the robot in the forest environment, thus successfully avoiding all the obstacles and probably even the unnecessary changes in elevation (considering the expert group's behaviour described above), while not necessarily choosing the most efficient overall path, since it was not its main mission.

A key point to consider is that even though the path generated by `move_ base_naive` would never be executed in reality, we believe once again that analysing a baseline as a trivial straight path can be relevant to provide context, and even a worst-case scenario as lower bound for comparisons.

In a broader context, Fig. 19 provides a visual representation of the occlusion types and obstacles encountered in the dataset, as well as the output of our main stages: mechanical effort estimation (Fig. 19(c)) and evident obstacles detection (Fig. 19(d)). The outputs were overlaid on the elevation map of the terrain in order to highlight the overall consistency of our results, demonstrating higher cost near areas with high probability of containing obstacles. It is worth mentioning that Figs. 19(a) and 19(b) were captured by the onboard RGB camera as the robot traversed the actual path during dataset collection.

On a final note, the path generated with our method ranked in the top 5 of all paths analysed in the experiments, having only 4 expert paths with superior score according to the metrics considered. The best volunteer's path achieved an up variation of 3.433 metres and a travelled distance of 48.522 m. This represents a +38.16% and +4.47% difference for our method with each metric, respectively. Although these values may seem high, especially for the up variation, by analysing Table 5 it becomes clear that our method greatly outperformed others for both metrics when compared to the best volunteer's path.

This experiment's vital importance comes not only from the fact that we compare our technique and validate it against expert human beings and other approaches, but also from the fact that all the data is taken from a real-world forest dataset, further proving its effectiveness and applicability.

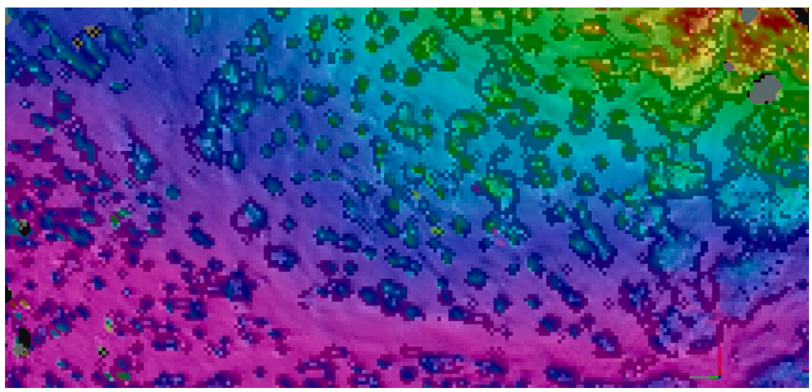
7. Conclusion

This paper proposes a novel technique for traversability analysis and path planning based on the concept of mechanical effort. It categorizes terrain according to the effort required to traverse it, while identifying key evident obstacles. This generates efficient paths that avoid obstacles and major hills, potentially minimizing fuel consumption. Given an *a piori* 3D point cloud map of the environment generated with a supporting SLAM algorithm, a concurrent pipeline of obstacle detection and mechanical effort estimation is implemented and the system is optimized to run in real time while performing a global analysis. This allows a robot to plan far beyond its observable range. We achieve the proposed experimental goals and, at the same time, successfully address one of the weaknesses mentioned by Lourenço et al. [5], i.e., the map generation frequency is improved by 3–5 times for the tested scenarios.

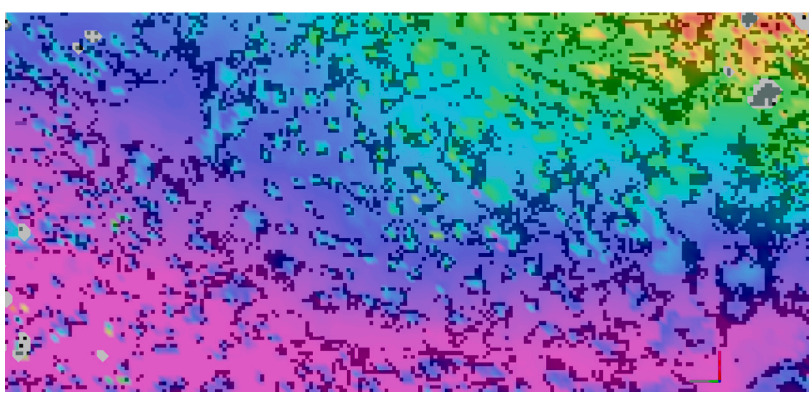
The proposed method has been tested both against other methods in a 3D simulation engine, yielding very positive results and proving to be a strong competitor against other techniques, as well as through an experiment with high quality data from a new and reputable real-world

(a) Illustration of non-regular obstacles in the environment.

(b) Illustration of dense foliage in the sensors' fields of view.



(c) Mechanical effort estimation output overlaid on the elevation map.



(d) Evident obstacles detection output overlaid on the elevation map.

Fig. 19. Output of the mechanical effort estimation on the Montmorency dataset, with examples showing the unstructured nature of the dataset.

forest dataset, which also provided unequivocal results regarding the effectiveness of our technique.

There are, however, still issues and improvements that we consider opportune to address in future work, such as:

- Further improve MEBT by using more efficient data structures and, eventually, parallelizing part of the process using a Graphics Processing Unit (GPU);
- Experiment with different SLAM approaches so that the processes of mapping, localization, traversability analysis and path planning can all occur online and ideally in real time;
- Improve our method's architecture, so that semantic-metric information can be taken into account when computing the mechanical effort associated with the perceived terrain, e.g. defining no-go areas and differentiating traversable low vegetation from other non-traversable obstacles with similar size;
- Expand the work to consider dynamic obstacles, eventually by coupling the proposed approach with a local planning method;
- Recalculate only the portion of the map that can have influence on the current path;
- Validate the improved architecture with a robot, thus also performing real-world navigation;
- Improve our comparison's methodology so that, aside from multiple scenarios, we consider multiple random pairs of start-origin poses.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

We thank Gonçalo Martins for the invaluable insight and active support. We would also like to express our gratitude to the data providers of the Montmorency dataset, namely Jean-François Tremblay, Martin Beland, François Pomerleau, Richard Gagnon, and Philippe Giguère. Their exceptional work in curating and making this dataset publicly available empowered us to validate our technique using real-world forest data.

This work was supported by the Safety, Exploration and Maintenance of Forests with Ecological Robotics (SEMFIRE, ref. CENTRO-01-0247-FEDER-03269) research project co-funded by the Agência Nacional de Inovação within the Portugal2020 programme, as well as the Semi-Autonomous Robotic System For Forest Cleaning and Fire Prevention (SafeForest, ref. CENTRO-01-0247-FEDER-045931) research project, co-funded by the Agência Nacional de Inovação within the CMU Portugal Large Scale Collaborative Projects program. Support for this research was also provided by the Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) through the Carnegie Mellon Portugal Affiliated Ph.D. Program under Grant PRT/BD/153920/2022 and the Scientific Employment Stimulus 5th Edition, contract ref. 2022.05726.CEECIND.

Appendix. Briefing and instructions provided to the user study volunteers

Greetings.

I am conducting a comparative study between a traversability analysis method I developed and the intuition of volunteers faced with a navigation problem in a three-dimensional terrain.

Essentially, we have a coloured representation of the general relief of the terrain which, when viewed from above, allows us to perceive the relative elevation of each section based on its colour and respective intensity. In this case, we are using the rainbow scale, i.e., the lower ground is represented by shades of magenta that change to blue, then cyan, then green, and so on as the elevation increases.

If we look at the image again, we see that we are dealing with a portion of relatively flat terrain with scattered obstacles (magenta and blue area) and another part of terrain that is steeper (blue/cyan/green area), also with scattered obstacles marked with different colours.

The red marker represents the robot's starting point, while the green marker represents the goal point. What I ask you is to draw a path that connects the red marker to the green one, that goes through the areas you consider to be of less effort for the robot, avoiding all obstacles. Here, we consider effort as the amount of energy that the robot must expend to traverse a given zone. For instance, the greater the slope of a hill we are going up, the greater the effort associated with that section of path. It should also be noted that for the same distance, travelling downhill requires less effort than travelling straight across the terrain.

There is no trial limit for this experience and you can request a restart at any time.

Thank you.

References

[1] M.S. Couceiro, D. Portugal, J.F. Ferreira, R.P. Rocha, SEMFIRE: Towards a new generation of forestry maintenance multi-robot systems, in: 2019 IEEE/SICE Int. Symposium on System Integration, SII 2019, 2019, pp. 270–276, http://dx.doi.org/10.1109/SII.2019.8700403.

[2] A. Mowshowitz, A. Tominaga, E. Hayashi, Robot navigation in forest management, J. Robot. Mechatron. 30 (2) (2018) 223–230, http://dx.doi.org/10.20965/jrm.2018.p0223.

[3] N. Shalal, T. Low, C. McCarthy, N. Hancock, Orchard mapping and mobile robot localisation using on-board camera and laser scanner data fusion–part B: Mapping and localisation, Comput. Electron. Agric. (2015).

[4] K.H. Choi, S.K. Han, S.H. Han, K.-H. Park, K.-S. Kim, S. Kim, Morphology-based guidance line extraction for an autonomous weeding robot in paddy fields, Comput. Electron. Agric. 113 (2015) http://dx.doi.org/10.1016/j.compag.2015.02.014, URL https://www.sciencedirect.com/science/article/pii/S0168169915000563.

[5] D. Lourenço, J.F. Ferreira, D. Portugal, 3D local planning for a forestry UGV based on terrain gradient and mechanical effort, in: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Workshop on Perception, Planning and Mobility in Forestry Robotics, WPPMFR 2020, Las Vegas, NV, USA, Oct 29, 2020.

[6] I.A. Bârsan, P. Liu, M. Pollefeys, A. Geiger, Robust dense mapping for large-scale dynamic environments, in: 2018 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2018, pp. 7510–7517.

[7] A. Hornung, K.M. Wurm, M. Bennewitz, C. Stachniss, W. Burgard, Octomap: An efficient probabilistic 3D mapping framework based on octrees, Auton. Robots 34 (3) (2013) 189–206.

[8] P. Fankhauser, M. Bloesch, M. Hutter, Probabilistic terrain mapping for mobile robots with uncertain localization, IEEE Robot. Autom. Lett. 3 (4) (2018) 3019–3026.

[9] S. Yang, S. Yang, X. Yi, An efficient spatial representation for path planning of ground robots in 3D environments, IEEE Access 6 (2018) http://dx.doi.org/10.1109/ACCESS.2018.2858809.

[10] F. Ruetz, E. Hernández, M. Pfeiffer, H. Oleynikova, M. Cox, T. Lowe, P. Borges, Ovpc mesh: 3d free-space representation for local ground vehicle navigation, in: 2019 International Conference on Robotics and Automation, ICRA, IEEE, 2019, pp. 8648–8654.

[11] P. Krüsi, P. Furgale, M. Bosse, R. Siegwart, Driving on point clouds: Motion planning, trajectory optimization, and terrain assessment in generic nonplanar environments, J. Field Robotics 34 (5) (2017) 940–984, http://dx.doi.org/10.1002/rob.21700.

[12] D. Droeschel, S. Behnke, Efficient continuous-time SLAM for 3D lidar-based online mapping, in: 2018 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2018, pp. 5000–5007.

[13] S. Macenski, D. Tsai, M. Feinberg, Spatio-temporal voxel layer: A view on robot perception for the dynamic world, Int. J. Adv. Robot. Syst. 17 (2) (2020) 1729881420910530.

[14] J. Borenstein, Y. Koren, The vector field histogram - fast obstacle avoidance for mobile robots, IEEE J. Robot. Autom. 7 (3) (1991) 278–288.

[15] D. Fox, W. Burgard, S. Thrun, The dynamic window approach to collision avoidance, IEEE Robot. Autom. Mag. 4 (1) (1997) 23–33.

[16] P.E. Hart, N.J. Nilsson, B. Raphael, A formal basis for the heuristic determination of minimum cost paths, IEEE Trans. Syst. Sci. Cybern. 4 (2) (1968) 100–107.

[17] N.J. Nilsson, Principles of Artificial Intelligence, Springer Science & Business Media, 1982.

[18] E.W. Dijkstra, et al., A note on two problems in connexion with graphs, Numerische Math. 1 (1) (1959) 269–271.

[19] S.M. LaValle, Rapidly-Exploring Random Trees: A New Tool for Path Planning, Technical Report, Computer Science Department, Iowa State University (TR 98–11), Ames, IA, USA, 1998.

[20] D. Osmankovic, A. Tahirovic, G. Magnani, All terrain vehicle path planning based on D* lite and MPC based planning paradigm in discrete space, in: IEEE/ ASME Int. Conf. on Advanced Intelligent Mechatronics, AIM, IEEE, 2017, pp. 334–339, http://dx.doi.org/10.1109/AIM.2017.8014039.

[21] B. Sebastian, P. Ben-Tzvi, Physics based path planning for autonomous tracked vehicle in challenging terrain, J. Intell. Robot. Syst. 95 (2019) 511–526.

[22] Z. Jian, Z. Lu, X. Zhou, B. Lan, A. Xiao, X. Wang, B. Liang, 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.

[23] N. Ormahony, et al., Deep learning for visual navigation of unmanned ground vehicles : A review, in: 29th Irish Signals and Systems Conference, ISSC 2018, 2018, http://dx.doi.org/10.1109/ISSC.2018.8585381.

[24] P. Borges, T. Peynot, S. Liang, B. Arain, M. Wildie, M. Minareci, S. Lichman, G. Samvedi, I. Sa, N. Hudson, M. Milford, P. Moghadam, P. Corke, A survey on Terrain traversability analysis for autonomous ground vehicles: Methods, sensors, and challenges, Field Robot. 2 (1) (2022) 1567–1627, http://dx.doi.org/10.55417/fr.2022049.

[25] S. Palazzo, D.C. Guastella, L. Cantelli, P. Spadaro, F. Rundo, G. Muscato, D. Giordano, C. Spampinato, Domain adaptation for outdoor robot traversability estimation from RGB data with safety-preserving loss, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2020, pp. 10014–10021.

[26] G. Reina, A. Leanza, A. Milella, A. Messina, Mind the ground: A power spectral density-based estimator for all-terrain rovers, Measurement 151 (2020) 107136.

[27] C. Goodin, L. Dabbiru, C. Hudson, G. Mason, D. Carruth, M. Doude, Fast terrain traversability estimation with terrestrial lidar in off-road autonomous navigation, in: Unmanned Systems Technology XXIII. Vol. 11758, SPIE, 2021, pp. 189–199.
[28] J. Ahtiainen, T. Peynot, J. Saarinen, S. Scheding, A. Visala, Learned ultra-wideband RADAR sensor model for augmented LIDAR-based traversability mapping in vegetated environments, in: 2015 18th International Conference on Information Fusion, Fusion, IEEE, 2015, pp. 953–960.
[29] C. Winkens, F. Sattler, D. Paulus, Hyperspectral terrain classification for ground vehicles, in: VISIGRAPP, 5: VISAPP, 2017, pp. 417–424.
[30] F. Vulpi, A. Milella, R. Marani, G. Reina, Recurrent and convolutional neural networks for deep terrain classification by auton. robots, J. Terramech. 96 (2021) 119–131.
[31] T. Manderson, S. Wapnick, D. Meger, G. Dudek, Learning to drive off road on smooth terrain in unstructured environments using an on-board camera and sparse aerial images, in: 2020 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2020, pp. 1263–1269.
[32] F. Ebadi, M. Norouzi, Road terrain detection and classification algorithm based on the color feature extraction, in: 2017 Artificial Intelligence and Robotics, IRANOPEN, IEEE, 2017, pp. 139–146.
[33] M. Prágr, P. Váňa, J. Faigl, Aerial reconnaissance and ground robot terrain learning in traversal cost assessment, in: International Conference on Modelling and Simulation for Autonomous Systems, Springer, 2019, pp. 3–10.
[34] A. Krebs, C. Pradalier, R. Siegwart, Comparison of boosting based terrain classification using proprioceptive and exteroceptive data, in: Experimental Robotics: The Eleventh International Symposium, Springer, 2009, pp. 93–102.
[35] J. Faigl, M. Prágr, On unsupervised learning of traversal cost and terrain types identification using self-organizing maps, in: Artificial Neural Networks and Machine Learning–ICANN 2019: Theoretical Neural Computation: 28th International Conference on Artificial Neural Networks, Munich, Germany, September 17–19, 2019, Proceedings, Part I, Springer, 2019, pp. 654–668.
[36] G.G. Waibel, T. Löw, M. Nass, D. Howard, T. Bandyopadhyay, P.V.K. Borges, How rough is the path? Terrain traversability estimation for local and global path planning, IEEE Trans. Intell. Transp. Syst. 23 (9) (2022) 16462–16473.
[37] M. Wei, V. Isler, Predicting energy consumption of ground robots on uneven terrains, IEEE Robot. Autom. Lett. 7 (1) (2021) 594–601.
[38] M. Eder, R. Prinz, F. Schöggl, G. Steinbauer-Wagner, Generating robot-dependent cost maps for off-road environments using locomotion experiments and earth observation data, in: 2022 Sixth IEEE International Conference on Robotic Computing, IRC, IEEE, 2022, pp. 172–176.
[39] G. Svenson, D. Fjeld, The influence of road characteristics on fuel consumption for logging trucks, in: HVTT12: 12th International Symposium on Heavy Vehicle Transport Technology. Vol. 31, no. 5, 2012, pp. 526–536, URL http://road-transport-technology.org/conferenceproceedings/hvtt-12/.
[40] A.E. Carvalho, J.F. Ferreira, D. Portugal, 3D traversability analysis in forest environments based on mechanical effort, in: 17th International Conference on Intelligent Autonomous Systems, IAS-17, 2022, pp. 457–468.
[41] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A.Y. Ng, et al., ROS: An open-source robot operating system, in: ICRA Workshop on Open Source Software. Vol. 3, Kobe, Japan, 2009, p. 5.
[42] G. Fontaras, N.G. Zacharof, B. Ciuffo, Fuel consumption and CO2 emissions from passenger cars in Europe – laboratory versus real-world emissions, Prog. Energy Combust. Sci. 60 (2017) 97–131, http://dx.doi.org/10.1016/j.pecs.2016.12.004.
[43] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, K. Konolige, The office marathon: Robust navigation in an indoor office environment, in: 2010 IEEE International Conference on Robotics and Automation, IEEE, 2010, pp. 300–307.
[44] N. Koenig, A. Howard, Design and use paradigms for Gazebo, an open-source multi-robot simulator, in: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems. Vol. 3, IROS IEEE Cat. No. 04CH37566, IEEE, 2004, pp. 2149–2154.
[45] J.-F. Tremblay, M. Béland, F. Pomerleau, R. Gagnon, P. Giguère, Automatic 3D mapping for tree diameter measurements in inventory operations, J. Field Robot. Special issue: Field Serv. Robot. (2020) http://dx.doi.org/10.1002/rob.21980.
[46] B.P. Gerkey, K. Konolige, Planning and control in unstructured terrain, in: ICRA 2008 Workshop on Path Planning on Costmaps, IEEE, 2008.

Afonso Carvalho completed his Integrated Master's Degree in Electrical and Computers Engineering in December 2021 at the University of Coimbra, in Portugal. During this time he attended multiple courses and workshops, being a summer course in the Technical University of Crete the most relevant. He worked as a fellow researcher at the Institute of Systems and Robotics in the SEMFIRE project. He is now pursuing his Ph.D. within the CMU Portugal Affiliated Program on the subject of Efficient Large-Scale Mapping and Navigation in Forest Environments and providing support to the SafeForest project. His main areas of expertise are forestry robotics, mobile robotics, traversability analysis and path planning.

João Filipe Ferreira is a Senior Lecturer in Computer Science at Nottingham Trent University (NTU) since 2018. He received the Engineering degree, the M.Sc. degree, and the Ph.D. degree in Electrical Engineering and Computers from the University of Coimbra in 2000, 2005 and 2011, respectively. He is as an integrated member of the Institute of Systems and Robotics (ISR), a research institute of the University of Coimbra, and also as a member of the Computational Intelligence and Applications group (CIA) at NTU. His main research interests concern the broad scientific themes of Artificial Perception and Cognition. Within these themes, the following topics receive his main focus: probabilistic modelling of perception, perception and sensing for field robotics in precision forestry and agriculture, and bioinspired perception and cognition for social robotics, co-robotics and human–robot interaction. He is a member of the IEEE Robotics and Automation Society (RAS) since 2012, and a member of the IEEE RAS Technical Committee on Cognitive Robotics since 2015, and member of the Technical Committee on Agricultural Robotics, TC AgRA, since 2019. He has been involved in several FP6, FP7 and H2020 European projects developed in consortium over the past few years and was the PI of the nationally funded research project CASIR — Coordinated Control of Stimulus-Driven and Goal-Directed Multisensory Attention Within the Context of Social Interaction with Robots (FCT Contract PTDC/EEI-AUT/3010/2012). He was recently Co-Lead Investigator for the FEDER/PORTUGAL 2020-funded projects SEMFIRE (Safety, Exploration and Maintenance of Forests with the Integration of Ecological Robotics — CENTRO-01-0247-FEDER-032691) and CORE (Centre of Operations for Rethinking Engineering — CENTRO-01-0247-FEDER-037082), both running from 2018 to 2021. He is currently a part of the SafeForest (Semi-Autonomous Robotic System for Forest Cleaning and Fire Prevention — CENTRO-01-0247-FEDER-045931) senior research team.

David Portugal completed his Ph.D. degree on Robotics and Multi-Agent Systems at the University of Coimbra in Portugal, in March 2014. His main areas of expertise are cooperative robotics, multi-agent systems, simultaneous localization and mapping, field robotics, human–robot interaction, sensor fusion, metaheuristics, and graph theory. After his Ph.D., he spent 4 years outside academia, and he is currently working as an Assistant Researcher at the Institute of Systems and Robotics and as an Invited Assistant Professor at the University of Coimbra. He has been involved in several local and EU-funded research projects in Robotics and ICT, such as CHOPIN, TIRAMISU, Social Robot, CogniWin, GrowMeUp, STOP, CORE, SEMFIRE, WoW, 5GSmartFact, and TRUSTID. He has co-authored over 85 research articles included in international journals, conferences and scientific books.