

# 3D Traversability Analysis in Forest Environments based on Mechanical Effort

Afonso E. Carvalho<sup>§</sup>, João Filipe Ferreira<sup>†,§</sup>, and David Portugal<sup>§</sup>

<sup>§</sup>Institute of Systems and Robotics, University of Coimbra, Portugal

<sup>†</sup>School of Science and Technology, Nottingham Trent University, UK

**Abstract.** Autonomous navigation in harsh and dynamic 3D environments poses a great challenge for modern Robotics. This work presents a novel traversability analysis and path-planning technique that processes 3D pointcloud maps to generate terrain gradient information. An analysis of terrain roughness and presence of obstacles is applied on the perceived environment in order to generate efficient paths. These avoid major hills when more conservative paths are available, potentially promoting fuel economy and reducing the wear of the equipment and the associated risks. The proposed approach outperforms existing techniques based on results in realistic 3D simulation scenarios, which are discussed in detail.

**Keywords:** Forestry Robotics, Traversability Analysis, 3D Navigation, Path Planning, UGV, Mechanical Effort

## 1 Introduction

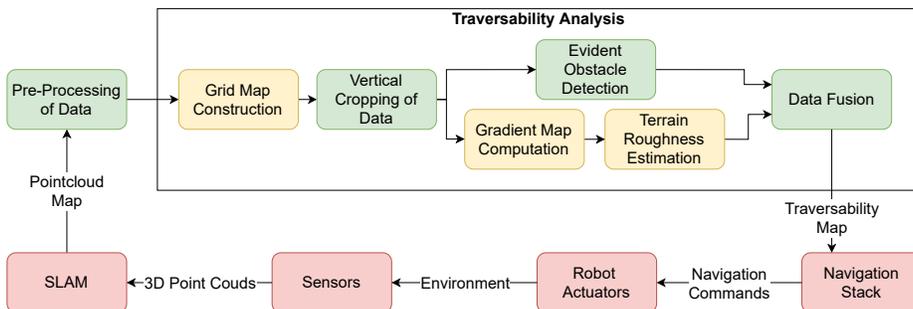
Forest maintenance is a labor-intensive, time-consuming, expensive, and ultimately hazardous activity, making it a prime candidate for automation. This requires reliable unsupervised navigation, which is a sizeable challenge considering that forests are highly unstructured, dynamic environments. As such, R&D efforts have been increasing, tackling key issues such as autonomous maintenance and management in fully [1] and semi-unstructured [2] forests, as well as 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 a UGV, avoiding steep sections of terrain to potentially improve fuel economy and reduce mechanical wear by minimizing the mechanical effort that the robot is subject to. For this purpose, a 2D costmap is used to represent the perceived environment, on which traversability analysis and successive path planning is then performed. The proposed method was tested and compared against existing techniques on a realistic 3D Gazebo<sup>1</sup> simulation environment providing a solid basis for discussion.

Existing literature on navigation and traversability analysis focuses on methods such as Elevation Maps [6], 2.5D-NDT [7], Octomaps [8] and fusion of complex physics-based planning with simplified approaches [9] to represent the fore-

---

<sup>1</sup> <http://gazebosim.org/>



**Fig. 1.** Proposed System architecture. Green boxes: novel modules; Yellow boxes: modified modules; Red boxes: off-the-shelf modules used “as is”.

seeable environment in a discrete way. Typically, traversability estimation is applied on these discrete representations by assigning costs to distinct portions of terrain. These include probabilistic [10], semantic segmentation [11] and neural network (NN) depth maps [12] approaches. Finally, classical planning algorithms, e.g. A\* [13], recent methods, e.g. D\* Lite [14], or those based on NNs [15], can generate a plan leading the agent to its target destination.

## 2 System Architecture

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

1. A pre-processing module responsible for down sampling the input data;
2. A traversability analysis block responsible for generating traversability maps;
3. A navigation technique that uses its contents to generate and execute paths.

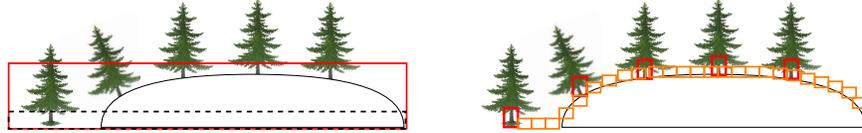
All modules have been developed using ROS<sup>2</sup> and run concurrently.

**Pre-Processing** In this module, a 3D global pointcloud 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 pointcloud 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 pointcloud 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 grid cell according to its location in space.

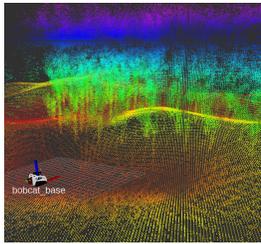
**Vertical Cropping of Data** Having then centered the decimated pointcloud in the robot’s frame of reference, all points that are outside of a configured

<sup>2</sup> Robot Operating System – <http://ros.org>.

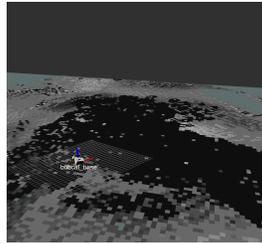


(a) Cropbox with a limit on the Z-axis. (b) Cell by cell analysis of vertical axis.

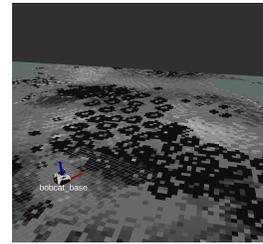
**Fig. 2.** Cropping heuristics. The one illustrated in (b) was ultimately selected.



(a) Pointcloud to be analyzed.



(b) Costmap before applying cropping heuristic.



(c) Costmap after applying cropping heuristic.

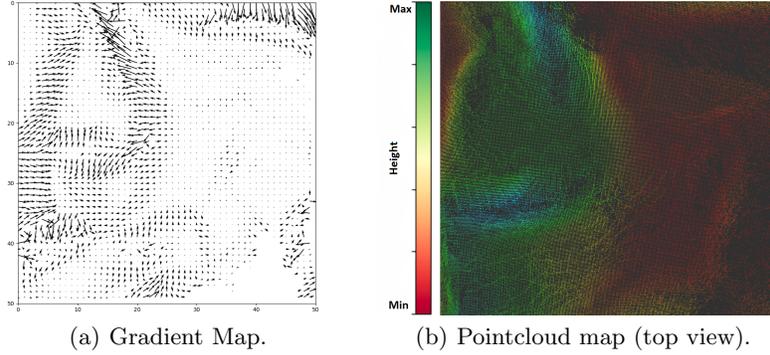
**Fig. 3.** Impact of applying the 3D vertical cropping heuristic on the 2D costmap representation (X-Y plane), which outlines existing obstacles more clearly (see black cells).

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 pointcloud 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 analyzed and consequently reduce the complexity and computational cost of the entire process.

**Gradient Map Computation** In this step, the terrain is analyzed using the gradient. For cells that contain more than one point, we apply the median on the set of points in that cell, and only then we apply the gradient to the resulting pointcloud, generating a map that estimates the overall roughness of the terrain.

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 gra-



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

gradient of a terrain and the energy consumption that it requires to be traversed, as it provides information of the direction of greatest variation.

Since it is not possible to employ a function that describes the entire point distribution, we make use of an approximation of the gradient, computed using second order accurate central differences, and the resulting gradient values are clipped to a configurable threshold, i.e.:

$$\nabla f(x_m, y_n) = \begin{cases} \nabla f(x_m, y_n), & \|\nabla f(x_m, y_n)\| < \nabla_{th} \\ \nabla_{th}, & \text{otherwise} \end{cases}, \quad (1)$$

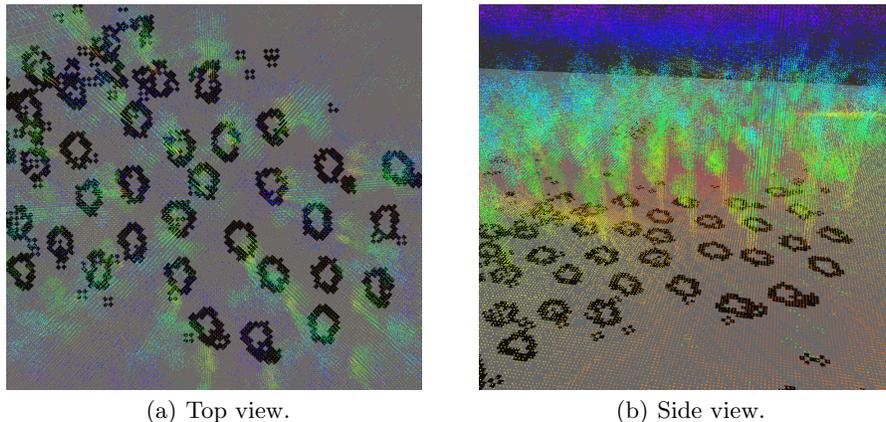
where  $\nabla f(x_m, y_m)$  represents the gradient in a given cell and  $\nabla_{th}$  represents the clipping threshold. This allows us to exclude both hills and depressions that exceed this configurable inclination limit. Fig. 4 illustrates this step.

**Evident Obstacle Detection** In this stage, we analyze the filtered pointcloud generated from the vertical heuristic applied previously (see Figs. 1 and 2(b)). We compute the mean, variance and range of the heights of all points that form each cell, and apply the following heuristic that marks a cell as an evident obstacle:

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

where  $\mu_{th}$ ,  $\sigma_{th}^2$  and  $\gamma$  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.

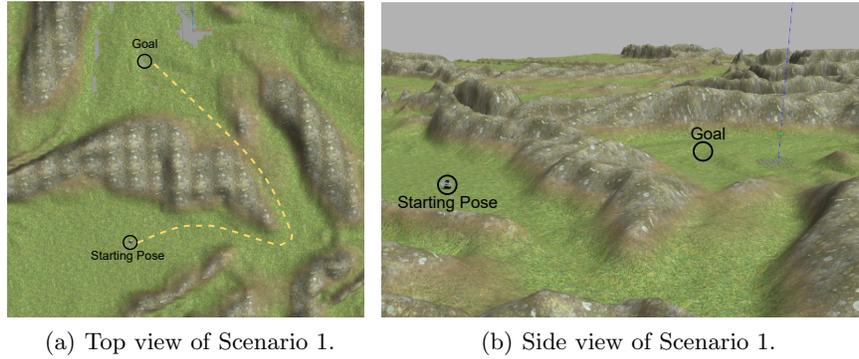
**Terrain Roughness Estimation** This module estimates terrain roughness using the previously generated gradient map. For this, we use the definition of mechanical effort  $\gamma = \|\nabla f(x, y)\| \cdot \cos \theta$  from [5], where  $\theta$  is the angle between the vector that connects the position of the robot with the position of the cell we are currently analyzing and the gradient vector in that cell.

The terrain roughness estimation module takes as input the gradient map and modifies it according to the mechanical effort equation, thus generating a 2D gridmap, in which the traversability cost increases as the robot’s direction of movement aligns with the gradient and vice-versa.

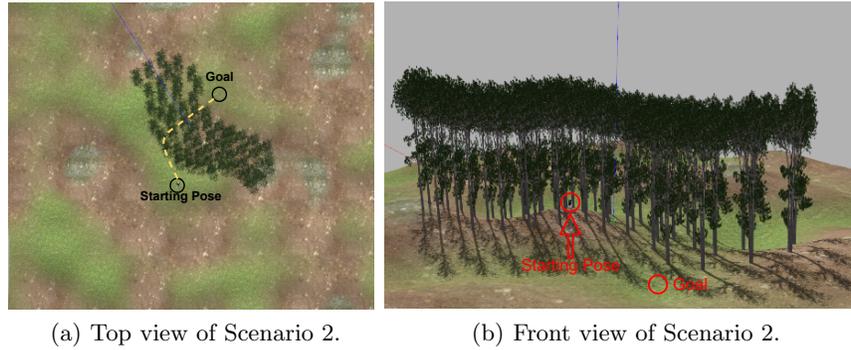
**Data Fusion** Finally, this module outputs a single 2D global costmap that incorporates all the useful information contained on the *evident obstacles costmap* and the *terrain roughness costmap*. All cells identified as evident obstacles are marked as non-traversable ( $cost = 100$ ) in the final 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.

### 3 Experimental Evaluation

We have designed three realistic simulation scenarios in Gazebo to evaluate the performance of the proposed method. Fig. 6 represents **Scenario 1**, which includes steep hills, making this an appropriate benchmarking scenario, rewarding path safety and mechanical effort over traveled distance and travel time, while still taking them into account. A second version of this scenario was designed, which we call **Scenario 1 remapped**, 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.



**Fig. 6.** First experimental scenario from two different perspectives, with starting pose, goal point and expected approximate path for our technique in yellow.



**Fig. 7.** Second experimental scenario from two different perspectives, with starting pose, goal point and expected approximate path for our technique in yellow.

Fig. 7 represents **Scenario 2**, 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 choosing a traversable path to do so.

We have defined the following **experimental objectives**:

1. Demonstrate that our work manages to create and execute paths with a lower energetic cost when compared to others.
2. Verify that our method outperforms the method in which it is based.
3. Show that our method can run in real time.

In order to quantify performance and compare methods, we have defined the following 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 2.

**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 1 and 2.

**Mean Map Generation Time** 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 2 and 3.

**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. This metric helps to demonstrate objective 1.

**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 ( $\theta$ ) values along the executed trajectory. It gives us an indication of the mean vertical travelling angle, which has a direct correlation with the effort of the chosen path. The candidate techniques should aim at minimizing it. This metric helps to demonstrate objective 1.

**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 2.

**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:

$$RDI = \frac{1}{1 + e^{-(0.5\phi - 13)}}, \quad (3)$$

which de-linearizes the penalty curve of the UGV’s inclination. Considering the roll angle limit to be  $35^\circ$ , it is almost equally safe to travel with between  $5^\circ$ - $10^\circ$  of inclination. However, the same does not apply to the  $5^\circ$  difference between  $25^\circ$ - $30^\circ$  of inclination, because the stability of the vehicle rapidly decreases when approaching its angular limits. The sigmoid filter in (3) describes this behavior.

**Pitch Danger Index** The mean percentage of pitch-related risk that the UGV takes during the execution of its path. Using the same rationale, the absolute pitch values are filtered through the following sigmoid function:

$$PDI = \frac{1}{1 + e^{-(0.25\theta - 6)}}. \quad (4)$$

**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).
- 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.
- The UGV rolled over.

This metric keeps track of failed runs, providing a solid measure of a system’s reliability and robustness, helping to demonstrate objective 2.

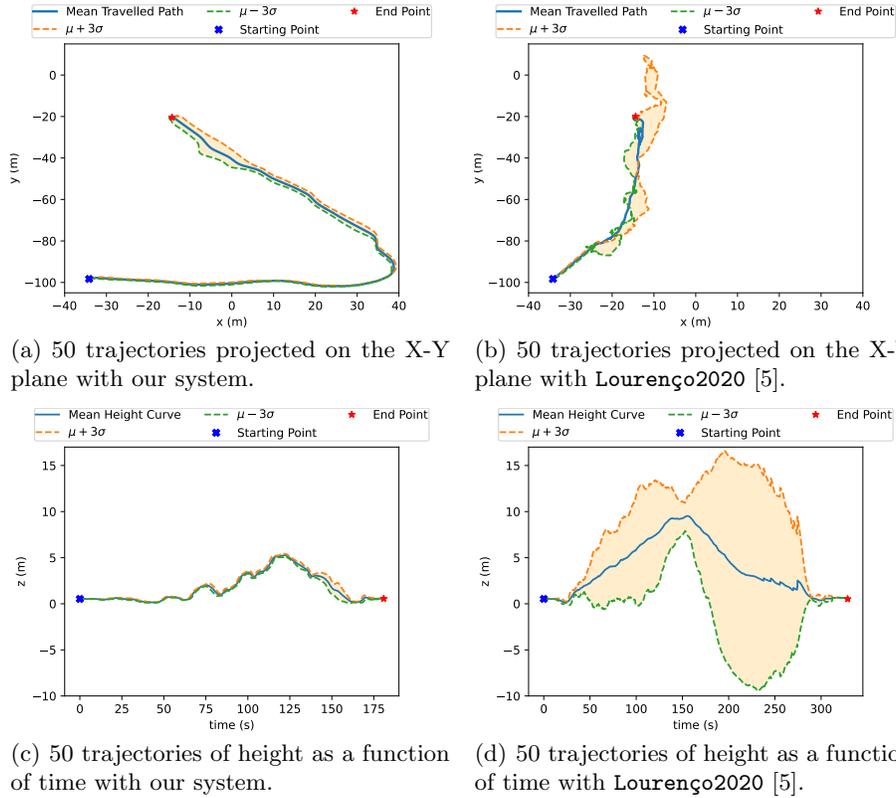
## 4 Results and Discussion

Our system is compared against Lourenço2020 [5], which introduced the original concept of mechanical effort, and `move_base_naive`, which applies the ROS navigation stack<sup>3</sup> with no observation sources and empty costmaps, following a straight path to the goal that, although expected to underperform other techniques, establishes a baseline for benchmarking. In this section, we discuss the findings of running the three methods on the scenarios described previously. To this end, 50 navigation trials were conducted in each scenario and a statistical analysis of the results is presented. In these experiments, our system uses a 200×200 m global costmap, while Lourenço2020 uses a 15×15 m local costmap.

Table 1 and Fig. 8 present the results of the experiments in Scenario 1. Instead of traversing straight to the goal, 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 system chooses the more conservative and levelled path available, which is around the hill. By doing so, it is able to significantly reduce the effort associated to that trajectory and potentially the fuel consumption, considering its relationship with terrain gradient (see [16]).

The pitch danger index is significantly lower than in other techniques, while the roll danger index is not that low, especially when compared to `move_base_naive`. This is justified since while the UGV travels around the hill, it 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, it only travels on average at 12.8% the maximum inclination allowed, while Lourenço2020 travels at a considerably higher 26.6%, because it detects the presence of a significant slope ahead, but is unable to find a better

<sup>3</sup> [http://wiki.ros.org/move\\_base](http://wiki.ros.org/move_base)



**Fig. 8.** Resulting plots from 50 runs in Scenario 1 (Fig. 6) with our system in the left, and Lourenço2020 [5] in the right.

path due to its short sensory horizon. Therefore, it travels sideways on the slope while attempting to find a better path, a behavior that is not ideal, considering that it tends to increase the danger and riskiness indexes and the elapsed time.

Table 2 shows the results from running the same experiment in the remapped Scenario 1. We can see that by slightly increasing the slope angles, the other systems perform drastically worse, while ours presents similar results to Scenario 1. 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 objectives 1 and 2.

Lastly, Table 3 and Fig. 9 presents the results of the experiments in Scenario 2 (Fig. 7), 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.

**Table 1.** Results from 50 runs in Scenario 1 (Fig. 6). Best values highlighted in bold.

Metrics	Lourenço2020	move_base_naive	Our System
Failure Rate (%)	84.000	<b>2.000</b>	<b>2.000</b>
Mean Effort ( $^{\circ}$ )	18.661 $\pm$ 1.828	12.805 $\pm$ 0.205	<b>5.601 <math>\pm</math> 0.060</b>
Pitch Danger Index (%)	38.592 $\pm$ 4.892	25.406 $\pm$ 0.335	<b>2.814 <math>\pm</math> 0.142</b>
Up Variation (m)	15.240 $\pm$ 1.886	8.877 $\pm$ 0.036	<b>8.183 <math>\pm</math> 0.088</b>
Travelled Distance (m)	112.387 $\pm$ 8.066	<b>84.958 <math>\pm</math> 0.068</b>	171.887 $\pm$ 0.343
Path Riskiness Index (%)	4.291 $\pm$ 4.094	<b>0.517 <math>\pm</math> 0.650</b>	0.932 $\pm$ 0.212
Roll Danger Index (%)	26.580 $\pm$ 4.648	<b>0.042 <math>\pm</math> 0.137</b>	12.843 $\pm$ 0.695
Mean Map Generation Time (seconds @ points)	8.2 @ 40.1k	N/A	18.3 @ 287k
Elapsed Time (s)	343.582 $\pm$ 14.886	<b>87.014 <math>\pm</math> 12.689</b>	174.368 $\pm$ 4.184

**Table 2.** Results from 50 runs in Scenario 1 remapped. Best values highlighted in bold.

Metrics	Lourenço2020	move_base_naive	Our System
Failure Rate (%)	100.000	90.000	<b>10.000</b>
Mean Effort ( $^{\circ}$ )	N/A	17.172 $\pm$ 0.902	<b>7.004 <math>\pm</math> 0.301</b>
Pitch Danger Index (%)	N/A	32.131 $\pm$ 1.199	<b>5.807 <math>\pm</math> 0.653</b>
Up Variation (m)	N/A	11.586 $\pm$ 0.050	<b>10.319 <math>\pm</math> 0.476</b>
Travelled Distance (m)	N/A	<b>87.873 <math>\pm</math> 0.399</b>	175.689 $\pm$ 1.894
Path Riskiness Index (%)	N/A	12.558 $\pm$ 0.853	<b>2.839 <math>\pm</math> 1.134</b>
Roll Danger Index (%)	N/A	<b>0.807 <math>\pm</math> 1.413</b>	12.788 $\pm$ 2.581
Mean Map Generation Time (seconds @ points)	8.9 @ 40.5k	N/A	18.3 @ 287k
Elapsed Time (s)	N/A	<b>25.299 <math>\pm</math> 2.510</b>	173.037 $\pm$ 31.038

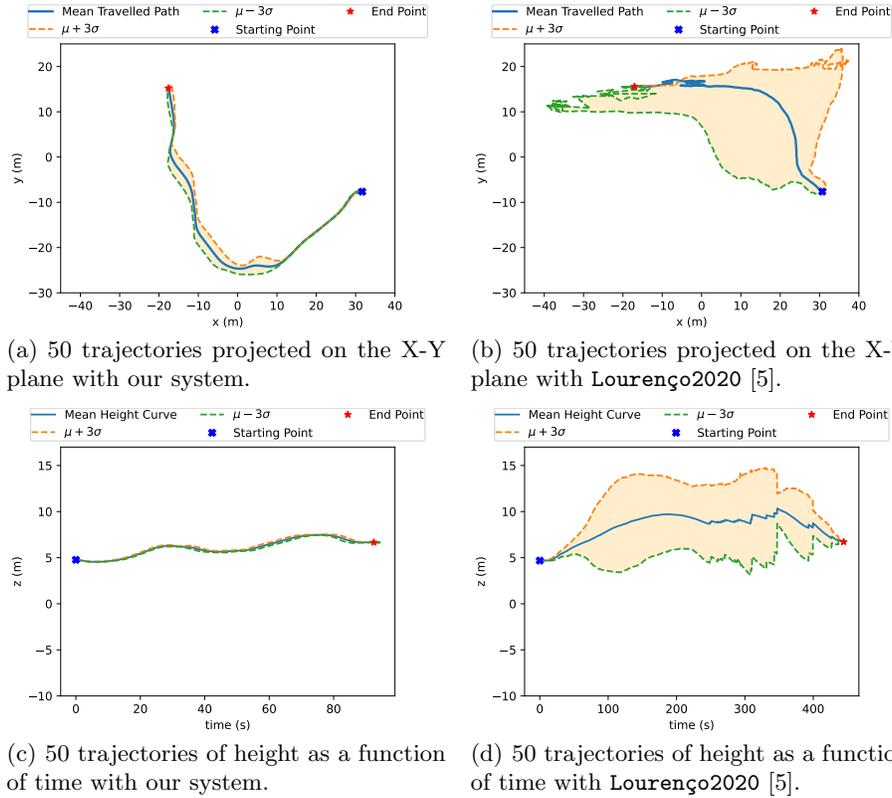
**Table 3.** Results from 50 runs in Scenario 2 (Fig. 7). Best values highlighted in bold.

Metrics	Lourenço2020	move_base_naive	Our System
Failure Rate (%)	58.000	92.000	<b>6.000</b>
Mean Effort ( $^{\circ}$ )	11.024 $\pm$ 3.500	15.188 $\pm$ 0.100	<b>3.595 <math>\pm</math> 0.066</b>
Pitch Danger Index (%)	9.326 $\pm$ 8.016	23.300 $\pm$ 0.565	<b>0.794 <math>\pm</math> 0.015</b>
Up Variation (m)	7.879 $\pm$ 0.496	7.035 $\pm$ 0.066	<b>3.714 <math>\pm</math> 0.045</b>
Travelled Distance (m)	80.226 $\pm$ 9.643	<b>58.299 <math>\pm</math> 0.497</b>	85.586 $\pm$ 0.493
Path Riskiness Index (%)	1.353 $\pm$ 5.998	0.767 $\pm$ 0.701	<b>0.000 <math>\pm</math> 0.000</b>
Roll Danger Index (%)	0.297 $\pm$ 0.297	0.909 $\pm$ 0.469	<b>0.020 <math>\pm</math> 0.001</b>
Mean Map Generation Time (seconds @ points)	12.1 @ 42.4k	N/A	20.2 @ 358k
Elapsed Time (s)	340.362 $\pm$ 123.149	182.514 $\pm$ 193.306	<b>88.266 <math>\pm</math> 12.011</b>

Our system substantially outperforms the remaining ones in every single metric except in the travelled distance, which was slightly higher<sup>4</sup>. However, that small increase in travel distance resulted in a significant difference in the mean effort and vertical variation, the main variables of interest for our objective, therefore posing a very reasonable trade-off.

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

<sup>4</sup> A video of our system in Scenario 2 is available at <https://youtu.be/2wSMTPBgOZU>.



**Fig. 9.** Resulting plots from 50 runs in Scenario 2 (Fig. 7) with our system in the left, and Lourenço2020 [5] in the right.

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.

## 5 Conclusion

This work proposes a novel traversability analysis and path planning technique based on the mechanical effort concept introduced in [5]. It categorizes terrain according to the effort required to traverse it, while identifying key evident obstacles, thus generating efficient paths that avoid obstacles and major hills, potentially minimizing fuel consumption. A concurrent pipeline of obstacle detection and terrain roughness estimation is implemented and the system is optimized to execute in real time while performing a global analysis. This potentially allows a robot to plan in real-time far beyond its observable range if given an *a priori* map of the environment. Finally, the implemented system has been tested against other methods in a 3D realistic simulation engine, yielding very positive results and proving to be a strong competitor against other techniques.

We have accomplished the experimental objectives proposed, while also successfully addressing one of the weaknesses stated by Lourenço et al. [5], i.e. the map generation frequency was improved by 3 to 5 times for the scenarios tested.

## Acknowledgements

We thank **Gonçalo Martins** for the invaluable insight and active support. 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.

## References

1. M. S. Couceiro, D. Portugal, J. F. Ferreira, and R. P. Rocha, “SEMFIRE: Towards a new generation of forestry maintenance multi-robot systems,” *2019 IEEE/SICE Int. Symposium on System Integration, SII 2019*, pp. 270–276, 2019.
2. A. Mowshowitz, A. Tominaga, and E. Hayashi, “Robot navigation in forest management,” *Journal of Robotics and Mechatronics*, vol. 30, no. 2, pp. 223–230, 2018.
3. N. Shalal, T. Low, C. McCarthy, and N. Hancock, “Orchard mapping and mobile robot localisation using on-board camera and laser scanner data fusion—part b: Mapping and localisation,” *Computers and Electronics in Agriculture*, 2015.
4. K. H. Choi, S. K. Han, S. H. Han, K.-H. Park, K.-S. Kim, and S. Kim, “Morphology-based guidance line extraction for an autonomous weeding robot in paddy fields,” *Computers and Electronics in Agriculture*, vol. 113, 2015.
5. D. Lourenço, J. F. Ferreira, and D. Portugal, “3D Local Planning for a Forestry UGV based on Terrain Gradient and Mechanical Effort,” *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. P. Fankhauser, M. Bloesch, and M. Hutter, “Probabilistic terrain mapping for mobile robots with uncertain localization,” *IEEE RAL*, vol. 3, no. 4, 2018.
7. S. Yang, S. Yang, and X. Yi, “An Efficient Spatial Representation for Path Planning of Ground Robots in 3D Environments,” *IEEE Access*, vol. 6, 2018.
8. A. Hornung et al., “OctoMap: An efficient probabilistic 3D mapping framework based on octrees,” *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.
9. P. Krüsi, P. Furgale, M. Bosse, and R. Siegwart, “Driving on Point Clouds: Motion Planning, Trajectory Optimization, and Terrain Assessment in Generic Nonplanar Environments,” *Journal of Field Robotics*, vol. 34, no. 5, pp. 940–984, 2017.
10. P. Fankhauser and M. Hutter, “A Universal Grid Map Library: Implementation and Use Case for Rough Terrain Navigation,” in *ROS Vol. 1*, Springer, 2016.
11. K. Usui, “Data augmentation using image-to-image translation for detecting forest strip roads based on deep learning,” *Int. Journal of Forest Engineering*, 2020.
12. L. Tai, S. Li, and M. Liu, “A deep-network solution towards model-less obstacle avoidance,” *IEEE IROS 2016*, pp. 2759–2764, 2016.
13. P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
14. D. Osmankovic, A. Tahirovic, and G. Magnani, “All terrain vehicle path planning based on D\* lite and MPC based planning paradigm in discrete space,” *IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics, AIM*, pp. 334–339, 2017.
15. N. Ormahony et al., “Deep Learning for Visual Navigation of Unmanned Ground Vehicles : A review,” *29th Irish Signals and Systems Conference, ISSC 2018*, 2018.
16. G. Svenson and D. Fjeld, “The influence of road characteristics on fuel consumption for logging trucks,” *HVTT12: 12th International Symposium on Heavy Vehicle Transport Technology*, vol. 31, no. 5, pp. 526–536, 2012.