

# Three-dimensional Terrain Aware Autonomous Exploration for Subterranean and Confined Spaces

Héctor Azpúrua, Mario F. M. Campos, and Douglas G. Macharet

**Abstract**—Despite the advances in autonomous navigation and motion planning, there are still several challenges to overcome, especially for confined or underground spaces. Confined scenarios present challenges such as lack of global or accurate external localization, uneven and slippery terrains, and multi-level stages. Exploring and mapping unknown unstructured environments is a fundamental step into the safe and efficient accomplishment of real-world tasks such as search and rescue missions or the autonomous inspection of dangerous areas. This paper proposes a novel three-dimensional autonomous exploration method for ground robots that considers the terrain traversability combined with the frontier expected information gain as a metric for the next best frontier selection in GPS-denied, confined spaces. Safe paths for navigation and frontier extraction are calculated iteratively from multiple 3D map representations such as octrees and meshes. Results in realistic simulated underground scenarios from the DARPA subterranean challenge demonstrate the technique’s feasibility, achieving a more reliable and faster exploration rate over competing approaches.

## I. INTRODUCTION

Confined areas are common in many industries: routine inspections, exploration, and surveillance are tasks that could benefit from using autonomous mobile robots. However, despite the advances in the field, there are still challenges to overcome for robot operation in those situations. Some of the principal challenges ground robots face in subterranean or enclosed scenarios lie in the lack of external or global localization, communication interference, and complex and rugged terrain topography, which means that long-run teleoperation is not feasible. These particular characteristics force ground autonomous exploring devices to use terrain-aware path planning methods that could generate safer paths and minimize risks during robot locomotion. These hard challenges are gaining attention from the robotic academy and industry: even the current DARPA (SubT) Challenge (2021), a highly regarded robotic competition, is currently focused on underground environments [1].

Confined areas are not only challenging for robots, but they also present hazardous situations for humans, such as gases, risk of entanglement, lack of oxygen, and poisonous animals, so using autonomous robots instead of human personnel improves operational safety. Traditional exploration methods work well for indoor, controlled scenarios where the ground is flat, and 2D map representations will suffice.

The authors are with the Computer Vision and Robotics Laboratory (VeRLab), Department of Computer Science, Universidade Federal de Minas Gerais, MG, Brazil. {hector.azpuru, mario, doug}@dcc.ufmg.br.

Héctor Azpúrua is also with Instituto Tecnológico Vale (ITV), Ouro Preto, Brazil. hector.azpuru@itv.org

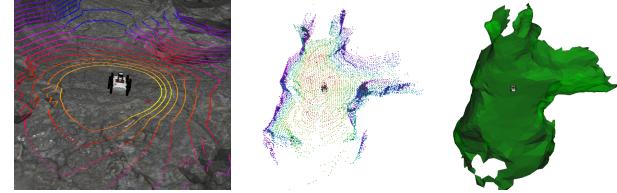


Fig. 1: Simulated robotic platform performing an iteration of the exploration pipeline. From left to right: the simulation environment and robot platform, the resulting point-cloud for the LiDAR SLAM algorithm, and the reconstructed mesh of the environments used for navigation and frontier selection.

For complex scenarios, such as subterranean caves, those 2D methods cannot capture enough details of the environment to be successful for planning and exploration. Thereof, we present a complete pipeline for autonomous three-dimensional exploration tailored for subterranean confined spaces and environments alike.

This work tackles the three-dimensional exploration problem using an information-theoretic approach that considers terrain traversability and expected mutual information gain as metrics for frontier selection. Unlike traditional techniques, we use multiple 3D map representations such as octrees and meshes to extract valuable data from the terrain to estimate safer, obstacle-free, and less bumpy navigation paths (Fig. 1); hence, the paths are generated by a weighted graph search algorithm combining multiple terrain metrics. In this sense, the proposed terrain-aware exploration algorithm considers if a frontier is reachable and the safest way to reach it. The frontier’s information gain is estimated over the reconstructed scenario by measuring the expected information sensed by the robot. In summary, the main contributions of this work are: (i) an information-theoretic method for 3D exploration that uses both the volumetric data and terrain traversability to estimate the fitness of a frontier; and (ii) a novel approach to generate a traversability map for ground robots considering maximum slope angles but respecting reachable multi-level stages.

## II. RELATED WORKS

Exploration is fundamental when dealing with coverage and mapping of an unknown region. However, despite the great variety of approaches, most of the literature’s works focus on generating 2D maps, which are not adequate when considering unstructured and uneven environments.

The works that deal with 3D maps are mainly based on the use of Octomaps [2] or other types of voxelization algo-

rithms [3]. Octomaps provide an efficient 3D representation based on Octrees that can shape complex 3D objects [4]. On the other hand, methods such as elevation maps or 2.5D mapping [5] are used for local planning in rugged terrains, but computational cost rapidly degrades with the map size. Those methods also have trouble dealing with roofed confined spaces. Another type of map representation with high description capabilities are Meshes, which are a collection of vertices, edges, and faces that define the shape of a polyhedral object. Compared to 2.5D solutions, full 3D mesh surfaces allow for planning and navigation in arbitrary complex environments, including multi-level environments [6], [7]. In the present work, we used octrees and meshes for their particular characteristics of great 3D representation, and in the case of meshes, their inherent capabilities to represent angles and slopes.

Exploration methods often use some kind of frontier-based selection [8], considering metrics such as distance [9], [10], 2D information gain [11], among others [12]. Confined spaces bring an extra level of difficulty for robotic exploration, given their enclosed nature. Thrun *et al.* [13] showed the inspection of an abandoned mine with a terrestrial robot, but in this case, the scenario was predominantly flat, and a 2D planner was used for navigation. Wang *et al.* [14] proposed a method for autonomous exploration of indoor spaces using a UAV navigating using an Information Potential Field (IPF) that repels obstacles while guiding the robot to the most informative frontier. The method calculates the information gain in a 2D slice of the 3D Octomap. In [15], the authors proposed a local and global 3D exploration planner for UAVs in confined spaces that considers the volumetric gain of frontiers. The environment is modeled as a graph, and feasible paths are generated using the RRT\* and Dijkstra algorithms. Papachristos *et al.* proposed an autonomous aerial exploration method for subterranean caves using multiple sensors like LiDAR, cameras, and IMU [16]. In [17], the authors proposed a 3D exploration method for confined spaces that uses raycasting to detect flyable frontiers and a semantic classification for frontier selection.

This work proposes a three-dimensional exploration method for ground robots that considers the terrain traversability and maximizes the information gain of visiting a frontier using the expected sensor information. We explored popular 3D map representation methods to navigate the robot over a traversability graph using an optimal graph search algorithm. The graph is generated by the online reconstruction of a 3D mesh of the environment from a LiDAR SLAM algorithm.

### III. PROBLEM FORMULATION

We tackle the problem of an autonomous exploration with no prior information in confined environments, such as subterranean mines and caves. The task will be executed by a single ground robot  $R$ , where its pose  $k$  is represented by a configuration  $\mathbf{q}_k \in \text{SE}(3)$ .

The robot must map a static environment  $\mathcal{E} \in \mathbb{R}^3$ , which poses critical challenges for the navigation, for example, obstacles, uneven terrains, and narrow passages. We assume

the robot is the only moving agent in the environment. Let  $\mathcal{M}$  be a three-dimensional occupancy grid representation of  $\mathcal{E}$ , generated by the observations of a 3D range sensor. The map will be initially set to  $\mathcal{M} = \mathcal{E}_{\text{unknown}}$ , as we do not assume any previous information on the environment. Space already explored ( $\mathcal{E}_{\text{known}}$ ) can be either mapped into  $\mathcal{M}_{\text{free}}$  (visited cells that do not contain any obstacle at the time of measurement) or  $\mathcal{M}_{\text{occupied}}$  (cells with more than 0.5 probability of occupation given the sensor model).

Given an initial configuration  $\mathbf{q}_i \in \mathcal{M}_{\text{free}}$ , to reach a goal position  $\mathbf{p}_g$  we must define a safe and efficient continuous path  $\tau : [0, 1] \rightarrow \mathcal{M}_{\text{free}}$ , such that  $\tau(0) = \mathbf{q}_i$  and  $\tau(1) = \mathbf{p}_g$ .

Finally, a fundamental aspect of the exploration task is the selection of a location to visit. Therefore, based on the concept of frontier, given a collection of reachable frontiers  $\mathcal{F}$ , we must select the one that looks more promising to aggregate information to our map.

**Problem 1** (Three-dimensional Terrain Aware Exploration). *Given a ground robot  $R$  in a confined static environment  $\mathcal{E}$ . The problem consists of efficiently build a map  $\mathcal{M}$  (3D occupancy grid) of the environment. For that, we must:*

- Create and maintain updated a set  $\mathcal{F}$  of all current identified reachable frontier regions in  $\mathcal{E}$ ;
- Select a frontier  $f_i \in \mathcal{F}$  that maximizes the information gain and is relatively close to the robot's current position;
- Determine a feasible and collision-free path  $\tau$  that drives the robot to an open area near the selected frontier.

### IV. METHODOLOGY

The proposed methodology considers a 3D 360° range sensor, such as a multi-line LiDAR, and an IMU to perform the SLAM used to generate a 3D occupational grid [18], [19]. A mesh reconstructed from the estimated occupational grid serves as input for generating a traversability graph with the robot's reachable regions. The information gain and the path cost are calculated for every extracted frontier using both the octree and reconstructed mesh. Finally, the selected frontier is navigated by the robot following an estimated safe path, and this cycle repeats until there are no more frontiers to explore or any other stop condition is met. A high-level description of the proposed method for exploration is depicted in Fig. 2.

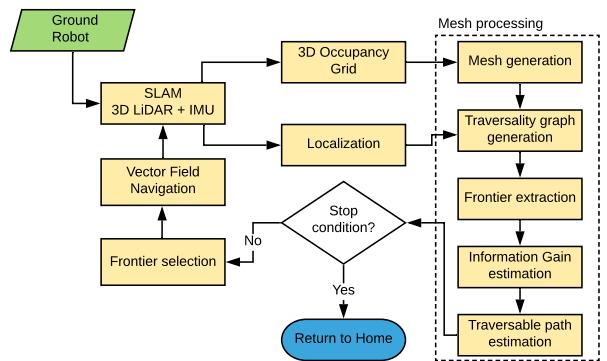


Fig. 2: High-level description of the proposed multi-step procedure for autonomous exploration.

### A. Mesh and Traversability Graph generation

Estimating reachable and non-reachable regions is particularly crucial for ground robots performing exploration tasks autonomously in uneven terrains. To generate a traversability map, we iteratively transform the octree 3D map of the environment into a mesh  $\mathbb{M}$  in order to estimate slopes. The mesh computation procedure consists of an automatic surface reconstruction pipeline optimized for subterranean caves and is able to work online<sup>1</sup>. The slopes are estimated by extracting the  $z$  normal vector ( $\vec{n}_i^z$ ) for every face of the mesh. We use a normal estimation method inspired by Hoppe *et al.* [20]. However, in our case, the Minimum Spanning Tree (MST) generated over the Riemannian Graph is based on the robot's position to guarantee that outliers from the point cloud do not result in reconstruction problems.

The mesh has a collection of faces  $\mathbb{F}$ , and it is filtered considering the maximum slope angle traversable by the robot ( $\theta_{max}$ ). We generate a graph  $\mathcal{G} = (V, E)$ , where the vertices  $V$  are the centroids of the mesh faces, and the edges  $E$  connect neighboring faces. The remaining unconnected traversable stages (such as stairs or similar structures) can be connected via a *bumpiness* threshold, calculated from a 3D sphere with a fixed radius from the closest vertices between the sub-graphs using a KD-tree search. Finally, we inflate borders and obstacles on  $\mathcal{G}$  to prevent paths from being too close to dangerous areas. The complete process is summarized in Algorithm 1.

---

#### Algorithm 1: Traversability graph estimation

---

```

 $\mathbb{M} \leftarrow \text{generateMesh}(\mathcal{M})$ 
for  $i \leftarrow 1$  to  $|\mathbb{F}|$  do
    if  $\vec{n}_i^z < \theta_{max}$  then
         $\hat{\mathbb{M}} \leftarrow \mathbb{M} \setminus \{\mathbb{F}_i\}$             $\triangleright$  Remove face
    end
end
 $\mathcal{G} \leftarrow \text{graphFromFaceCentroids}(\hat{\mathbb{M}})$ 
 $\mathcal{G} \leftarrow \text{KDTreeTraversablePlatformConnect}(\mathcal{G})$ 
 $\mathcal{G} \leftarrow \text{removeNonConnectedComponents}(\mathcal{G})$ 
 $\mathcal{G} \leftarrow \text{inflateBorders}(\mathcal{G})$ 
return  $\mathcal{G}$ 

```

---

Both slope and bumpiness thresholds are conservative constants that might be estimated and assigned via experimental evaluation of obstacles bypassing by the robot platform.

### B. Reachable frontiers extraction

The traversability graph  $\mathcal{G}$  and the complete mesh  $\mathbb{M}$  are used to estimate the map frontiers. The known traversable areas that are neighbors of unexplored regions, including regions edging the map's borders without being an obstacle, are considered frontiers. Therefore a face in the mesh is

<sup>1</sup><https://github.com/verlab/mesh-vr-reconstruction-and-view>

considered as a frontier if:

$$\text{isFrontier}(\mathbb{M}, i) = \begin{cases} 1, & \text{if } |N_{\mathbb{F}}(i)| \leq 2 \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

where  $N_{\mathbb{F}}(i)$  is the set of neighboring faces of face  $i$ .

The extracted frontiers that do not belong to the traversability graph are removed. This way, unreachable frontiers are eliminated from the pipeline. The frontier points are clustered into groups by their Euclidean distance using the Density-based spatial clustering algorithm (DBSCAN) [21], considering a distance ( $eps$ ) and minimum group size ( $C_{min}$ ). Finally, we use the KD-tree search to determine the visit location of a frontier cluster as the closest reachable point in  $\mathcal{G}$  to the cluster's centroid. The frontier estimation process is described in Algorithm 2, where  $\mathcal{B}$  are the raw mesh frontiers,  $\mathcal{C}$  the clusters of reachable frontiers, and  $n$  a reachable node from  $G$  used as frontier visit point.

---

#### Algorithm 2: Reachable frontiers extraction

---

```

 $\mathcal{B} \leftarrow \{\}$                                  $\triangleright$  Raw mesh frontiers
for  $i \leftarrow 1$  to  $|\mathbb{F}|$  do
    if  $\text{isFrontier}(\mathbb{M}, i)$  then
         $\mathcal{B} \leftarrow \mathcal{B} \cup \mathbb{F}_i$ 
    end
end
 $\mathcal{B}' \leftarrow \text{traversabilityFilter}(V, \mathcal{B})$ 
 $\mathcal{F} \leftarrow \{\}$                                  $\triangleright$  Reachable frontiers centroids
 $\mathcal{C} \leftarrow \text{DBSCAN}(\mathcal{B}', eps, C_{min})$ 
for  $c \in \mathcal{C}$  do
     $n \leftarrow \text{getCentroidReachableNode}(\mathcal{G}, c)$ 
     $\mathcal{F} \leftarrow \mathcal{F} \cup n$ 
end
return  $\mathcal{F}$ 

```

---

### C. Path cost estimation

The paths to reach the frontiers are determined by using the well-known Dijkstra shortest path algorithm over the traversability graph. We define a path  $\tau = \{n_1, \dots, n_{goal}\}$  as an ordered subset of neighboring nodes from  $\mathcal{G}$ , where  $n_1$  is the robot current pose  $q$  and  $n_{goal}$  is the location of a frontier  $f \in \mathcal{F}$ . We use a metric that performs a linear combination of the normalized Euclidean distance, terrain traversability, and energy consumption, as proposed in [22]. The final path cost is given by:

$$C(\tau) = \sum_{n \in \tau} [0.25 * N_d D(n) + 0.5 * N_t T(n) + 0.25 * N_e E(n)], \quad (2)$$

where  $N_d$ ,  $N_t$  e  $N_e$  are normalization coefficients.

The Euclidean distance  $D(n)$  is the cumulative sum of the 3D Euclidean distance for neighboring nodes in the path.

The traversability metric is based on the positive angle  $T(n)$  between the normal  $\vec{n}_i^z$  of face  $i$  and the Z-axis ( $\vec{Z}$ ):

$$T(n) = \arccos \left( \frac{|\vec{n}_i^z \cdot \vec{Z}|}{\|\vec{n}_i^z\| \|\vec{Z}\|} \right). \quad (3)$$

The energy consumption is estimated as a linear regression of the battery consumption, terrain inclination, friction coefficient, robot mass, and angle  $\theta$  between the vectors linking neighboring mesh centers:

$$E(n) = \left( \frac{E_{\text{mean}} \alpha(n)}{2\pi} \right) + (a\theta(n) + b)D(n), \quad (4)$$

where  $E_{\text{mean}}$  is the mean battery consumption while turning  $2\pi$  rad and  $a, b$  are the linear regression parameters. The quantities  $D(n)$  and  $\alpha(n)$  are estimates of the angular and linear displacements, respectively, when the robot moves between nodes  $n$  and  $n + 1$ .

Finally, the robot navigates the path to visit a frontier using a vector field navigation algorithm [23] with a potential-field based obstacle avoidance.

#### D. Information-theoretic frontier selection

To estimate the utility of a frontier, we use the mutual information gain metric over the 3D map  $\mathcal{M}$ , and the cost of the path  $C(\tau)$  to reach the frontier. The mutual information uses the probability of the octree cells to calculate the current entropy [24] of the map and then compares it with the expected entropy of the map after performing a virtual exploration at the selected frontier. The entropy is defined as:

$$H(\mathcal{M}) = - \sum_{i,j,k} p_{ijk} \log(p_{ijk}), \quad (5)$$

where  $\mathcal{M}$  is the current map, and  $p_{ijk}$  is the outcome of the Bernoulli random variable representing the cell occupation.

The mutual information  $I(\mathcal{M}, f_i)$  is used to obtain the expected information gain by visiting frontier  $f_i$ , i.e.:

$$I(\mathcal{M}, f_i) = H(\mathcal{M}) - H(\mathcal{M}|f_i), \quad (6)$$

where  $H(\mathcal{M}|f_i)$  is the expected new entropy.

The *virtual exploration* phase uses the sensor model maximum range and field of view to project rays in the current map  $\mathcal{M}$  considering the vehicle will be in the centroid of each frontier in  $\mathcal{F}$ . Since we assume a sensor with a  $360^\circ$  horizontal FoV, the robot's orientation is not used for calculations, only the mesh face's inclination (expected robot's roll and pitch) at the visiting site is used for transformation purposes.

Given the current map  $\mathcal{M}$  state, the projected rays give a reasonable estimate of the maximum free volume the robot could sense of a frontier. Rays that hit walls or fall within the range of known empty cells ( $\mathcal{M}_{\text{known}}$ ) do not increment much information. We consider rays that escape known areas and approach the maximum sensor range without hitting an obstacle as more informative. Since the measurements from these types of multi-line LiDAR sensors results in a sparse Octree, we performed a filling step at the virtual map using a uniform sampling of the mesh surfaces  $\mathbb{F}$ , preventing the projected rays from escaping through walls and solid objects over the occupancy grid  $\mathcal{M}$ .

Finally, the metric for selecting the best frontier is:

$$c^* = \arg \max_{\forall f \in \mathcal{F}} \frac{MI(\mathcal{M}, f) + e}{C(\tau^f)}, \quad (7)$$

where  $\tau^f$  is a path from the robot's current position to frontier  $f$ , and  $e$  is a small tolerance constant.

## V. EXPERIMENTS AND RESULTS

### A. Experimental setup

The experiments were performed in simulated scenarios using CoppeliaSim v4.0.0, with ROS Kinetic running in Ubuntu 16.04. The laptop used for the experiments has an Intel Xeon E3-1200 CPU and 16GB of memory. We validated the performance of the proposed exploration pipeline using three scenarios: (a) a single-level cave map extracted from the DARPA SubT challenge, (b) a multi-level cave map also from the DARPA challenge, and (c) a synthetic simplified cave map in a flat environment. The scenarios can be observed in Fig. 3. The subterranean environments present realistic challenges for locomotion, such as uneven terrains, multi-level platforms, rocks, and other obstacles. The EspeleoRobô robot [22] were used for the experiment, and it was equipped with a simulated Ouster OS1 LiDAR sensor and an IMU. For all experiments, the maximum slope is defined as  $\theta_{\text{max}} = 30^\circ$  and the bumpiness as 25cm. The code implementation of the exploration pipeline is made available online<sup>2</sup>.

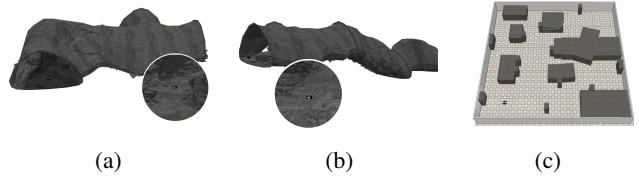


Fig. 3: Simulated environments used for experimental evaluation: (a) a single-level cave map extracted from the DARPA SubT challenge, (b) a multi level cave map also from the DARPA challenge, and (c) a synthetic simplified cave map in a flat environment [25].

### B. Traversability graph generation

An example of the traversability estimation with a threshold of  $\theta_{\text{max}} = 30^\circ$  over a synthetic multi-level scenario is depicted in Fig. 4. The stair-like structure gives a good reference to reachable and non-reachable multi-level platforms. Hence a connected graph is generated for every traversable platform, but as the connection between platforms has a greater inclination than the established threshold, the platforms are not initially connected (Figs. 4a-b). After the bumpiness post-processing step, the first two platforms within the traversable range are connected to the main base (Figs. 4c-d).

### C. Frontier extraction

An example of the frontier extraction over a simulated DARPA SubT cave section using the mesh and the traversability graph can be observed in Fig. 5. In Fig. 5a, the red squares mark the expected locations of ground frontiers, and in Fig. 5b, the traversability graph that leads to the frontiers visit points

<sup>2</sup>[https://github.com/ITVRoC/espeleo\\_planner](https://github.com/ITVRoC/espeleo_planner)

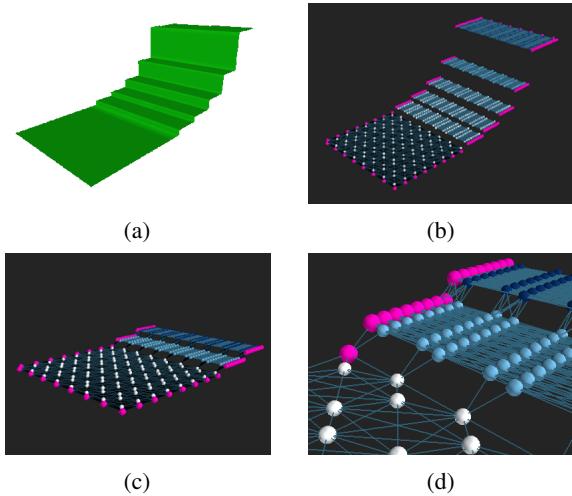


Fig. 4: Ground traversability map generation: (a) synthetic mesh reconstruction with slopes at different heights, (b) traversability graph, (c) traversable map with bumpiness threshold applied, and (d) zoom of the connected edges.

in pink. This double verification reduces false positives and guarantees that the estimated frontiers can be reachable by the ground robot.

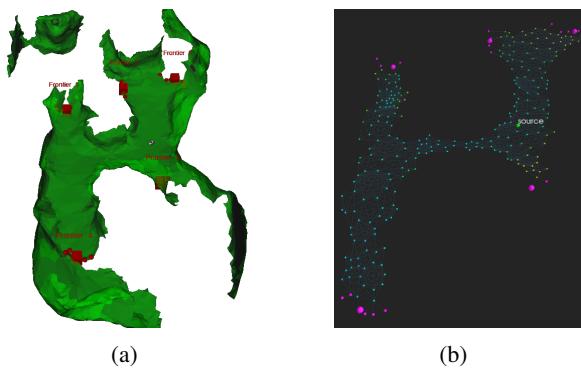


Fig. 5: Frontier estimation for the reconstructed mesh and traversability graph. (a) Generated mesh with extracted frontiers (red), and (b) traversability graph with the remaining frontiers from the mesh that are also within the graph (pink).

#### D. Octree filling

The raw map generated by the LeGO-LOAM mapping algorithm [18] is a sparse octree that is not directly suitable for many operations evolving terrain analysis and information estimation. Given the high number of spots in  $\mathcal{M}$  with missing  $\mathcal{M}_{occupied}$  cells, as shown by Fig. 6a, the ray-tracing algorithm will not work as best because the rays could pass through the missing cells and give a wrong volumetric measurement. The mesh reconstruction procedure will fill those small missing parts of the map and generate continuous surfaces that we sub-sample and use to fill the octree, as shown in Fig. 6b-c.

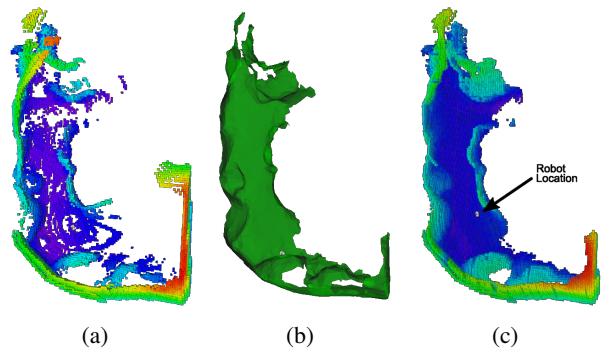


Fig. 6: Filling missing parts of the octree with mesh information, preventing the projected rays from escaping for holes in walls and solid objects: (a) raw estimated octree from the LiDAR SLAM algorithm, and (b) reconstructed mesh, and (c) the filled virtual octree.

#### E. Complete autonomous exploration

The exploratory efficiency over the cave scenarios (Fig. 3) was validated by comparing the resulting mapped point clouds generated iteratively by the SLAM algorithm with a reference map. Small discrepancies in the point clouds could occur since only the embedded robot sensors were used for localization. Therefore, the comparison method *sim* is defined as:

$$sim(a, b) = RMSE(a, b) * 0.5 + RMSE(b, a) * 0.5, \quad (8)$$

where  $a$  and  $b$  are the generated pointcloud and reference cloud, and the Root Mean Square Error (RMSE) between the clouds is given by:

$$RMSE(a, b) = \sqrt{\frac{\sum_{i=0}^{|b|} dist(a_i, b_i)^2}{|a|}}. \quad (9)$$

The resulting maps and the reconstructed meshes for every scenario could be observed in Fig. 7.

We validated multiple frontier and path selection metrics: (i) our proposed approach using the information gain and traversability path generation of the frontiers to select the next visit point, (ii) a greedy selection of the closest next frontier using the path that gives the smallest Euclidean distance instead of the terrain aware one, and (iii) random selection of the frontier using a terrain aware path. All methods use the traversable graph to generate the paths, but in the case of the greedy approach, only the shortest distances are prioritized regardless of terrain roughness or energy consumption. The results of this analysis can be observed in Fig. 8, where the displayed lines are the mean RMSE for every timestep of ten repetitions per experiment. The timesteps were estimated using only the time the robot moved and did not include the paths' computing time. The proposed metric (blue) converges to a lower error rate at every environment than the other metrics, even if the more terrain-friendly paths generated by the proposed metric are longer than the shortest euclidean path. An interesting effect of the greedy metric is that the shortest path usually increments the chances of entering a riskier area, also increasing the chances of entrapment or

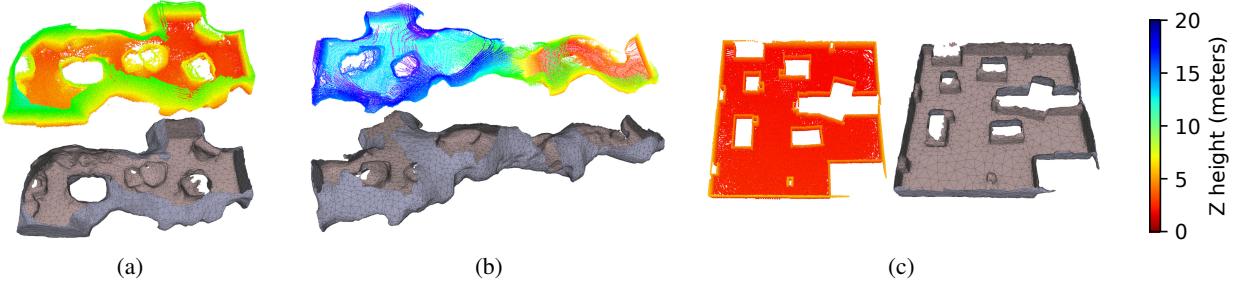


Fig. 7: Estimated point-cloud (colored) and final reconstructed mesh (brownish) for the cave environments: (a) a single-level cave, (b) a multi-level cave map, and (c) a synthetic cave map. The color gradient of the point-cloud represents the height variation (up to 20m).

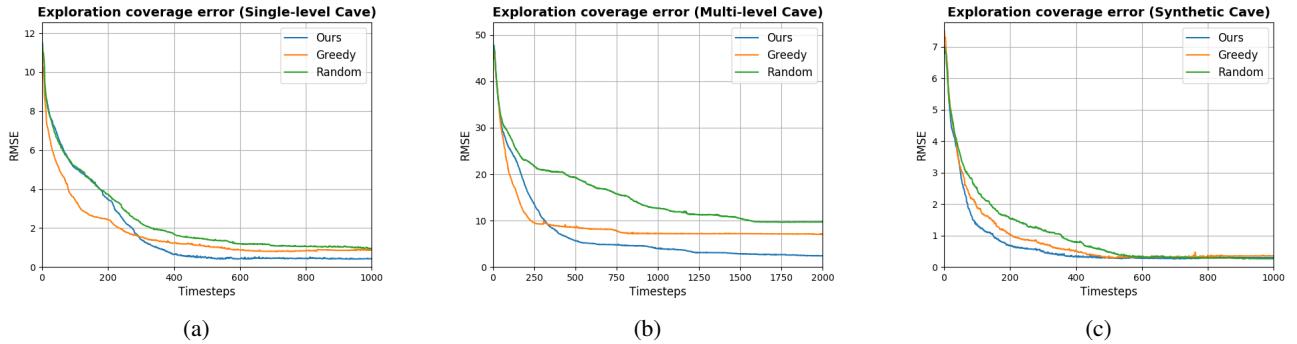


Fig. 8: Exploration error (RMSE) mean of ten runs comparing the real-time SLAM point cloud with a reference map for the: (a) single-level cave, (b) multi-level cave, and (c) synthetic cave. All methods use the traversability graph to generate the paths. Only the greedy approach prioritizes solely Euclidean distance instead of a smooth, safer path.

collisions. Table I depicts the maximum processing time taken by the mesh reconstruction algorithm at the final step of the exploration. Showing that the proposed method could be used online with only small pauses between planning and execution.

TABLE I: Mean execution time (10 runs) of the mesh reconstruction method for all testing environments' final state.

Map	Num. Points	Execution time	$\sigma$
Single-level Cave	122084	30.143s	$\pm 1.31$
Multi-level Cave	142124	38.054s	$\pm 1.06$
Synthetic Cave	42000	11.648s	$\pm 0.72$

## VI. CONCLUSIONS AND FUTURE WORK

This paper describes a pipeline for a terrain aware autonomous exploration in subterranean and confined spaces. We proposed a novel approach for exploration that combines the cost of traversing rugged terrains and the expected information gathered by visiting a frontier. Our proposed method uses octrees and meshes to calculate the most informative frontiers and generate safe paths that consider terrain traversability, distance, and energy consumption. Unlike traditional exploration methods, the proposed method works in complex 3D environments without assuming any priors over the map structure. The information gain of a frontier is calculated using information-theoretic approaches by projecting a 3D range sensor rays and estimating the best-case scenario of

the robot's information visiting the frontier. The proposed pipeline works for outdoor and confined scenarios alike since it does not need any external localization. The mutual information metric shows increased performance in more comprehensive environments where there are many possible visiting locations and obstacles. In this sense, the algorithm could detect the small information frontiers and prevent the robot from wasting time and energy visiting them. In small scenarios, the benefit of selecting a frontier only by information and traversability decreases, and its similar to the most straightforward nearest frontier selection. Real-world validation should be a straightforward transition; however, point cloud noise and size could be a focal point to consider for mesh generation.

Future work will focus on improving the map information estimation using the topology of known frontier borders to learn realistic 3D map expansions instead of using the sensor model alone. Another key area for future work is cooperation. As the path's cost depends on particular robot characteristics, heterogeneous teams could inspect different parts of the map by defining different mapping constants. Finally, more efficient mesh generation methods should also be studied.

## ACKNOWLEDGMENT

This work was developed with the support of Instituto Tecnológico Vale (ITV). Partially financed by CAPES - Finance Code 001, CNPq, FAPEMIG, and Vale S.A.

## REFERENCES

- [1] B. Allen, "Unearthing the subterranean environment." [Online]. Available: <https://www.subtchallenge.com/>
- [2] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous robots*, vol. 34, no. 3, pp. 189–206, 2013.
- [3] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning," in *2017 Ieee/rsj International Conference on Intelligent Robots and Systems (iros)*. IEEE, 2017, pp. 1366–1373.
- [4] J. Wilhelms and A. Van Gelder, "Octrees for faster isosurface generation," *ACM Transactions on Graphics (TOG)*, vol. 11, no. 3, pp. 201–227, 1992.
- [5] P. Fankhauser and M. Hutter, "Anymal: a unique quadruped robot conquering harsh environments," *Research Features*, no. 126, pp. 54–57, 2018.
- [6] S. Pütz, T. Wiemann, J. Sprickerhof, and J. Hertzberg, "3d navigation mesh generation for path planning in uneven terrain," *IFAC-PapersOnLine*, vol. 49, no. 15, pp. 212–217, 2016.
- [7] S. Pütz, J. S. Simón, and J. Hertzberg, "Move base flex," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3416–3421.
- [8] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*. IEEE, 1997, pp. 146–151.
- [9] J. Faigl, M. Kulich, and L. Přeučil, "Goal assignment using distance cost in multi-robot exploration," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 3741–3746.
- [10] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, "Collaborative multi-robot exploration," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 1. IEEE, 2000, pp. 476–481.
- [11] J. M. Pimentel, M. S. Alvim, M. F. Campos, and D. G. Macharet, "Information-driven rapidly-exploring random tree for efficient environment exploration," *Journal of Intelligent & Robotic Systems*, vol. 91, no. 2, pp. 313–331, 2018.
- [12] M. Prágr, P. Cizek, J. Bayer, and J. Faigl, "Online incremental learning of the terrain traversal cost in autonomous exploration," in *Robotics: Science and Systems*, 2019.
- [13] S. Thrun, S. Thayer, W. Whittaker, C. Baker, W. Burgard, D. Ferguson, D. Hahnel, D. Montemerlo, A. Morris, Z. Omohundro *et al.*, "Autonomous exploration and mapping of abandoned mines," *IEEE Robotics & Automation Magazine*, vol. 11, no. 4, pp. 79–91, 2004.
- [14] C. Wang, L. Meng, T. Li, C. W. De Silva, and M. Q.-H. Meng, "Towards autonomous exploration with information potential field in 3d environments," in *2017 18th International Conference on Advanced Robotics (ICAR)*. IEEE, 2017, pp. 340–345.
- [15] T. Dang, S. Khattak, F. Mascarich, and K. Alexis, "Explore locally, plan globally: A path planning framework for autonomous robotic exploration in subterranean environments," in *2019 19th International Conference on Advanced Robotics (ICAR)*. IEEE, 2019, pp. 9–16.
- [16] C. Papachristos, S. Khattak, F. Mascarich, and K. Alexis, "Autonomous navigation and mapping in underground mines using aerial robots," in *2019 IEEE Aerospace Conference*. IEEE, 2019, pp. 1–8.
- [17] A. Akbari, P. Chhabra, U. Bhandari, and S. Bernardini, "Intelligent exploration and autonomous navigation in confined spaces," in *Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020.
- [18] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4758–4765.
- [19] G. Pereira, L. V. do Carmo Matos, H. Azpurua, G. Pessin, and G. Medeiros Freitas, "Investigação de técnicas lidar slam para um dispositivo robótico de inspeção de ambientes confinados," in *2020 Congresso Brasileiro de Automática (CBA)*. CBA, 2020.
- [20] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," in *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, 1992, pp. 71–78.
- [21] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN revisited, revisited: why and how you should (still) use DBSCAN," *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 3, pp. 1–21, 2017.
- [22] H. Azpurua, F. Rocha, G. Garcia, A. S. Santos, E. Cota, L. G. Barros, A. S. Thiago, G. Pessin, and G. M. Freitas, "EspeleoRobô - a robotic device to inspect confined environments," in *2019 19th Int. Conf. on Advanced Robotics (ICAR)*. IEEE, Dec. 2019.
- [23] V. M. Gonçalves, L. C. A. Pimenta, C. A. Maia, B. C. O. Dutra, and G. A. S. Pereira, "Vector fields for robot navigation along time-varying curves in n-dimensions," *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 647–659, Aug 2010.
- [24] C. E. Shannon, "A mathematical theory of communication," *Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.
- [25] A. Howard and N. Roy, "The Robotics Data Set Repository (Radish)," 2003. [Online]. Available: <http://radish.sourceforge.net/>