frontiers | Frontiers in Robotics and AI

TYPE Original Research
PUBLISHED 03 October 2023
DOI 10.3389/frobt.2023.1249586

Check for updates

OPEN ACCESS

EDITED BY
Matthew Gombolay,
Georgia Institute of Technology,
United States

REVIEWED BY
Claudio Rossi,
Polytechnic University of Madrid, Spain
Krzysztof Tadeusz Walas,
Poznań University of Technology, Poland

*CORRESPONDENCE
Christoffer Heckman,
✉ christoffer.heckman@colorado.edu

RECEIVED 28 June 2023
ACCEPTED 06 September 2023
PUBLISHED 03 October 2023

CITATION
Miles MJ, Biggie H and Heckman C (2023), Terrain-aware semantic mapping for cooperative subterranean exploration.
Front. Robot. AI 10:1249586.
doi: 10.3389/frobt.2023.1249586

Terrain-aware semantic mapping for cooperative subterranean exploration

Michael J. Miles[1], Harel Biggie[2] and Christoffer Heckman[2]*

[1]Department of Mechanical Engineering, University of Colorado Boulder, Boulder, CO, United States,
[2]Department of Computer Science, University of Colorado Boulder, Boulder, CO, United States

Navigation over torturous terrain such as those in natural subterranean environments presents a significant challenge to field robots. The diversity of hazards, from large boulders to muddy or even partially submerged Earth, eludes complete definition. The challenge is amplified if the presence and nature of these hazards must be shared among multiple agents that are operating in the same space. Furthermore, highly efficient mapping and robust navigation solutions are absolutely critical to operations such as semi-autonomous search and rescue. We propose an efficient and modular framework for semantic grid mapping of subterranean environments. Our approach encodes occupancy and traversability information, as well as the presence of stairways, into a grid map that is distributed amongst a robot fleet despite bandwidth constraints. We demonstrate that the mapping method enables safe and enduring exploration of subterranean environments. The performance of the system is showcased in high-fidelity simulations, physical experiments, and Team MARBLE's entry in the DARPA Subterranean Challenge which received third place.

KEYWORDS

subterranean environments, field robots, semantic grid mapping, terrain mapping, cooperative exploration, DARPA subterranean challenge

1 Introduction

Robots have proliferated in structured and well-controlled environments such as those in warehousing and retail due to the simplicity and consistency of such environments. However, unstructured and subterranean environments remain a challenge for most autonomous robots due to their complex topology, large size, and lack of infrastructure or available prior knowledge of the environment. Such environments are especially challenging for mobile, ground-based robots, which must contend with rugged terrain (i.e., obstacle-laden indoor environments, rough or off-road outdoor environments) during their operation. The complexity of this problem is further exacerbated in high-risk scenarios, e.g., autonomous search and rescue operations, in which a team of robots must quickly and continuously explore such environments with limited inter-robot communication abilities. Thus, the mapping solution used by ground robots to understand their environments is critical to the task of autonomous, multi-agent exploration. An effective mapping solution for such scenarios should generate high-fidelity reconstructions of the environment which include information that may aid in the operation, such as terrain features and objects or structures of interest. Furthermore, the solution should facilitate rapid construction and efficient sharing to enable collaborative exploration.

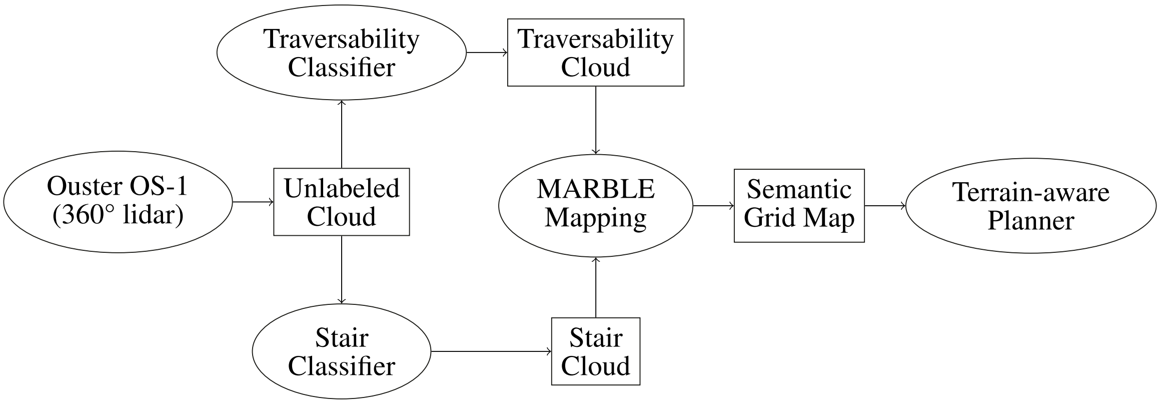

FIGURE 1
Overview of Team MARBLE's terrain-aware navigation solution. Ellipses denote software packages. Rectangles denote data products.

Early approaches to robot mapping focused primarily on occupancy grid mapping, in which an environment is discretized into a 2D (or later, 3D) grid. Each grid cell independently models the probability that the area/volume to which it corresponds consists of free or occupied space. These approaches generally model occupancy by integrating range and bearing measurements over time via a binary Bayes filter. One of the first of these approaches was that of Moravec and Elfes (1985) which utilizes an array of twenty-four sonar transducers covering 360° around a mobile robot to generate a 2D occupancy grid map. Payeur et al. (1997) makes the extension of occupancy grid maps from 2D to 3D computationally tractable by introducing a closed-form approximation of the occupancy probability function and storing the cells in a multi-resolution octree. Despite their early beginnings, occupancy grid mapping approaches remain at the forefront of many modern robot mapping solutions. Hornung et al., 2013 introduces OctoMap, an extremely popular (Lluvia et al., 2021) open-source mapping package that features an octree-based data structure and lossless compression for highly efficient occupancy grid mapping.

Occupancy grid maps are efficient at mapping the general occupancy state of an environment but do not scale well for high-resolution mapping of surfaces, which are of particular interest in terrain traversability mapping. Oleynikova et al. (2017) introduces Voxblox which utilizes a Truncated Signed Distance Field (TSDF) -based map representation and voxel hashing for fast queries, however, this approach uses large amounts of memory, which restricts sharing with other robots in the fleet, especially for large environments. Shan et al. (2018) leverages probabilistic inference to build terrain maps from sparse lidar data by modeling traversability as a Bernoulli-distributed random variable, however, this approach requires training, which can become inaccurate in unknown environments. Labbé and Michaud (2019) presents RTAB-Map which maintains pointcloud maps via a memory-efficient loop closure approach, however, memory inefficiency in mapping still makes this approach prohibitive for sharing dense information.

Some approaches combine the simplicity of grid mapping with the height resolution of surface maps to yield 2.5D elevation maps, wherein the value of each horizontal grid cell corresponds to the probabilistic height of the surface at that location. Hebert et al. (1989) presents an early such approach that fused measurements from a scanning lidar imager into an elevation map for mapping potential footholds for a quadrupedal robot. Fankhauser et al., 2014; Fankhauser and Hutter 2016 extends this idea with GridMap, an open-source mapping package that generates multi-layer elevation and traversability maps in a robot-centric approach for a quadrupedal robot. Hines et al. (2021) utilizes GridMap for traversability-mapping in conjunction with an occupancy grid map which features novel "virtual surfaces" to encourage safe exploration amidst negative obstacles as part of Team CSIRO Data61's entry in the DARPA Subterranean Challenge. Fan et al. (2021) also utilizes a similar elevation map in conjunction with a Conditional-Value-at-Risk metric for traversability-mapping as part of Team CoSTAR's entry in the DARPA Subterranean Challenge. Haddeler et al. (2022) presents a method similar to our own which classifies semantic traversability from geometric pointcloud features and incorporated the information into a 2D elevation map.

In addition to traversability, the classification and mapping of objects or structures of interest, such as stairways, is also highly valuable. This is particularly important in the context of urban search and rescue where the ability to identify and navigate stairways is a critical feature of autonomous rescue systems. Harms et al. (2015) presents an approach to stair detection which utilized a stereo-vision camera system and a convolutional kernel to extract the alternating concave and convex edges of the steps of a stairway. Sánchez-Rojas et al. (2021) presents a hybrid visual and geometric approach to stair detection as well as a fuzzy logic controller for the alignment of robots to the detected stairway. (Westfechtel et al., 2016, Westfechtel et al., 2018) introduces StairwayDetection, an open-source package that directly classifies the various parts of stairways using a graph-based approach, including the stair tread, riser, and railing segments.

In this work, we present a terrain-mapping system based on semantically-encoded 3D grid maps. In addition to occupancy, the traversability of the terrain as well as the presence of stairways are geometrically classified and probabilistically mapped. Furthermore, a low-bandwidth method for sharing these maps amongst a heterogeneous fleet of quadrupedal and wheeled vehicles is shown. This mapping system was used in conjunction with

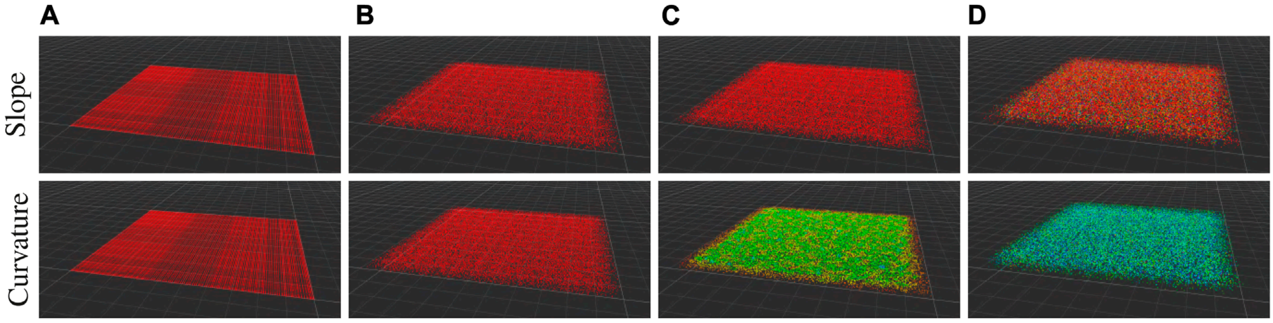

FIGURE 2
A typical pointcloud generated for the noise analysis in Table 1 for each parameter set **(A–D)**. The pointcloud has a side length of 10 m and a horizontal resolution of 5 cm. The point colors represent the cost terms of Eq. 3: the slope cost term on the top row, where red points indicate a cost of 0.0 and blue points indicate a curvature cost of 20.0; and the curvature cost term on the bottom row, where red points indicate a cost of 0.0 and blue points indicate a curvature cost of 0.7.

TABLE 1 Noise analysis of a planar pointcloud, as in Figure 2. Four sets of parameters describing the standard deviation of the points in meters in the *x*, *y*, and *z*-axes are used: A (0.0, 0.0, 0.0), B (1.0, 1.0, 0.0), C (1.0, 1.0, 0.10), and D (1.0, 1.0, 0.5). The mean and standard deviation of the slope cost and curvature cost terms of Eq. 3 in each pointcloud over 20 samples of each parameter set are shown. Throughout this work, we use a slope gain c_{slope} of 20.0 and a curvature gain c_{curv} of 2.0.

Average traversability cost by point position noise				
Cost term	A	B	C	D
Slope	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	2.254 ± 0.029
Curvature	0.0 ± 0.0	0.0 ± 0.0	0.097 ± 0.005	0.364 ± 0.001

a terrain-aware path planner as part of Team MARBLE's entry into the DARPA Subterranean Challenge in which they took third place (Biggie et al., 2023). This opportunity provided an extremely unique and realistic environment in which to field deploy and validate the proposed terrain-mapping system. We hypothesize that the terrain-mapping system described herein, which is made possible with several novel and open-source subsystems in elegant coordination, constitutes an effective method to enable safe and enduring exploration in large, unstructured, subterranean environments.

2 Materials and methods

Briefly, the terrain-mapping system, called *MARBLE Mapping*, operates as follows. Classifications of terrain traversability and the presence of stairways are performed independently on unlabeled 3D pointclouds from geometric features and the resultant labeled pointclouds are spatially and temporally integrated into a 3D grid map. A path planner uses this semantic grid map to evaluate candidate plans for compatibility with the given platform and explore the subterranean environment. As the grid map is updated, differential maps containing compressed occupancy, traversability, and stair-presence information are queued and shared with other robot agents whenever a communication link is established. Upon receiving differential maps from external agents, these maps are fused into the recipient agent's local grid map. Furthermore, a powerful lidar-inertial odometry solution called LIO-SAM (Shan et al., 2020) is used for online localization. The system operates on the Robot Operating System (ROS1) message-passing framework (Quigley et al., 2009) for internal communication and a custom UDP-based transport layer (Biggie and McGuire, 2022) for multi-agent communication.

While this work will be primarily concerned with the classification and mapping of traversability and stairways, differential map sharing and terrain-aware path planning are major considerations and will be used to evaluate the mapping method described here. The proposed method and its context in Team MARBLE's terrain-aware navigation solution is illustrated in Figure 1.

2.1 Occupancy classification and mapping

Our semantic occupancy mapping pipeline is based on the framework provided in Hornung et al. (2013). Utilizing a time-of-flight lidar, this framework leverages pointcloud data to construct a 3D voxel-based representation of the surrounding environment. Points in the pointcloud constitute occupied space and the space between the sensor and the point is considered free. These occupancy classifications are probabilistically fused into the grid map.

The framework scales incredibly well due to the underlying octree data structure which provides efficient data storage and queries. An octree is a hierarchical structure for storing volumetric data contained in a 3D grid. Each volumetric cell ("voxel") in the grid is represented by a node of the octree and can be recursively combined to represent coarser volumes of decreased resolution ("pruned") or recursively subdivided to represent finer volumes of increased resolution ("expanded"). The primary benefit of the octree data structure is that it does not need to maintain the position of a voxel or its size explicitly. Instead, this information is reconstructed from inexpensive traversal of the octree, which results in a highly memory-efficient map. This low-memory representation is fundamental to sharing these maps between robot agents via a low-bandwidth communication network.

Occupancy grids suffer from uncertainty arising from several sources: sensor noise, localization drift, misalignment between multiple sensors and/or robots, etc. This motivates the use of a probabilistic occupancy representation to classify the occupancy state of an environment; in this work, we use the binary Bayes filter (Eqs. 5 and 6) as was done in (Moravec and Elfes 1985; Thrun 2002).

The posterior probability $P_{occ}(v|z_{occ,1:t})$ that some voxel v is occupied, given sensor measurements $z_{occ,1:t}$ is estimated via:

$$\begin{aligned} &P_{occ}\left(v|z_{occ,1:t}\right) \\ &\quad= \left[1+\frac{1-P_{occ}\left(v|z_{occ,t}\right)}{P_{occ}\left(v|z_{occ,t}\right)}\frac{1-P_{occ}\left(v|z_{occ,1:t-1}\right)}{P_{occ}\left(v|z_{occ,1:t-1}\right)}\frac{P_{occ}\left(v\right)}{1-P_{occ}\left(v\right)}\right]^{-1}, \\ &\quad\text{where } P_{occ}\left(v|z_{occ,1:t}\right)\in(0,1). \end{aligned} \tag{1}$$

A voxel with an occupancy probability of greater than the prior $P_{occ}(v)$ is considered occupied, and a voxel is considered free space if it has an occupancy probability of less than the prior. An occupancy probability equal to the prior implicitly defines a voxel whose occupancy status is unknown.

Equation 1 is rewritten in log-odds form to avoid numerical instabilities and quicken updates $L_{occ}(v|z_{occ,t})$ to the current log-odds occupancy estimate $L_{occ}(v|z_{occ,1:t})$:

$$L_{occ}\left(v|z_{occ,1:t}\right) = \log\left(\frac{P_{occ}\left(v|z_{occ,1:t}\right)}{1-P_{occ}\left(v|z_{occ,1:t}\right)}\right) \tag{2a}$$

$$= L_{occ}\left(v|z_{occ,1:t-1}\right)+L_{occ}\left(v|z_{occ,t}\right), \tag{2b}$$

$$\begin{aligned} &\text{where } L_{occ}\left(v|z_{occ,1:t}\right)\in\left[\mathcal{L}_{occ,min},\mathcal{L}_{occ,max}\right], \\ &\mathcal{L}_{occ,min}\in(-\infty,0)\,,\mathcal{L}_{occ,max}\in(0,\infty), \\ &L_{occ}\left(v|z_{occ,t}\right)=\begin{cases}\mathcal{L}_{occ,hit}>0\,\text{on occupied space}\\ \mathcal{L}_{occ,miss}<0\,\text{on free space.}\end{cases} \end{aligned}$$

The updates are represented by the log-odds form of the inverse sensor model $P_{occ}(v|z_{occ,t})$. For a beam-based range finder (e.g., lidar, sonar), the inverse sensor model can be approximated as a constant in log-odds form (Thrun, 2002). A ray is cast from the sensor to the incident voxel and all voxels that the beam passes through, as determined by the Bresenham algorithm (Amanatides and Woo, 1987), are measured as free with a log-odds update of $\mathcal{L}_{occ,miss}$, while the incident voxel is measured as occupied and updated with $\mathcal{L}_{occ,hit}$. The log-odds occupancy estimate is also bounded between $\mathcal{L}_{occ,min}$ and $\mathcal{L}_{occ,max}$ to improve performance in dynamic environments and increase memory efficiency (via pruning).

2.2 Traversability classification and mapping

We extend Hornung et al. (2013) to include traversability classification and mapping capabilities, thus enabling traversability-aware navigation and encouraging robot endurance in the presence of rough terrain. Specifically, we estimate a continuous *traversability cost*, i.e., the difficulty of traversing the terrain, primarily as it pertains to wheeled or tracked robots.

The traversability cost estimation described here relies on the pointcloud's geometric attributes, including slope and surface curvature. Other geometric characteristics, like roughness (fitness to a plane), have been shown to improve navigation performance (Ye, 2007). The incorporation of non-geometric features, such as friction (Hughes et al., 2019; Rodríguez-Martínez et al., 2019) and collapsibility (Haddeler et al., 2022), may also significantly improve navigation performance over adverse terrain, however, these features are often computationally expensive to calculate and less applicable to travel at low-moderate speeds. Learned terrain features (Guan et al., 2022; Shaban et al., 2022; Seo et al., 2023) offer a cheap and fast approach for real-time traversability estimation, yet the efficacy of these methods remains dependent on costly training and poses challenges in adapting to completely unknown environments. After extensive physical and simulated testing, we chose to focus solely on the slope and surface curvature as this approach proved to be efficient and provided adequate assurance of the robot's endurance.

We use the same 3D pointcloud from a time-of-flight lidar as was used in Section 2.1 to classify traversability in the environment (Figure 3A). The proposed method estimates the instantaneous traversability cost $\tau_{p,t}$ of a point p at time t (Equation 3) as a normalized weighted sum of the slope $(1-|\hat{\boldsymbol{n}}\cdot\hat{\boldsymbol{k}}|)$ and the curvature κ at the point:

$$\tau_{p,t}=\left\|c_{slope}\left(1-|\hat{\boldsymbol{n}}_{\boldsymbol{p},\boldsymbol{t}}\cdot\hat{\boldsymbol{k}}|\right)^{3}+c_{curv}\kappa_{p,t}\right\|\in[0,1]\,, \tag{3}$$

where a traversability cost of $\tau_{p,t}=0$ is easily traversed and $\tau_{p,t}=1$ is untraversable. The slope penalty c_{slope} and curvature penalty c_{curv} were tuned over numerous physical and simulated test deployments to prevent navigation into obstacles or up steep faces, and to encourage routes over smooth terrain, respectively. The slope, curvature, and final traversability cost as applied to the entire pointcloud can be seen in Figures 3B–D, respectively.

In our implementation, we pre-process the pointcloud with a voxel-grid filter to achieve a uniform resolution of 5 cm and then estimate the normal vector and surface curvature of each point via a least-squares plane fit to its twenty-four nearest neighbors as determined by a kd-tree. Applied to a relatively planar patch of points, this approach effectively estimates normal information from a square patch covering 400 square centimeters. This approach is faster but yields similar results compared to pre-processing the full-resolution pointcloud and estimating normal information from neighbors based on distance directly.

Because slope and curvature represent the first and second derivatives of the positions of the points, respectively, they are jointly afflicted by the uncertainty associated with the point positions. We empirically analyze the effect of point position noise on the point traversability cost by generating and calculating the traversability cost terms associated with a noisy, planar pointcloud (Table 1). We generate pointclouds for four sets of parameters with varying amounts of noise in the x, y, and z-axes. Then we calculate the slope and curvature cost terms in Eq. 3 and analyze the average and standard deviation cost of each point in the pointcloud over 20 samples of each parameter set. The practical effect is that extremely noisy pointclouds of planar terrain may appear especially rough or slightly sloped, as illustrated in Table 1. We wish to avoid extremely rough terrain as equally as sloped terrain and we perform instantaneous traversability classification of single pointclouds using a high precision 3D lidar which exhibits sub-centimeter standard deviation in point position (Ouster, 2023), so the effects of noise are negligible in our approach.

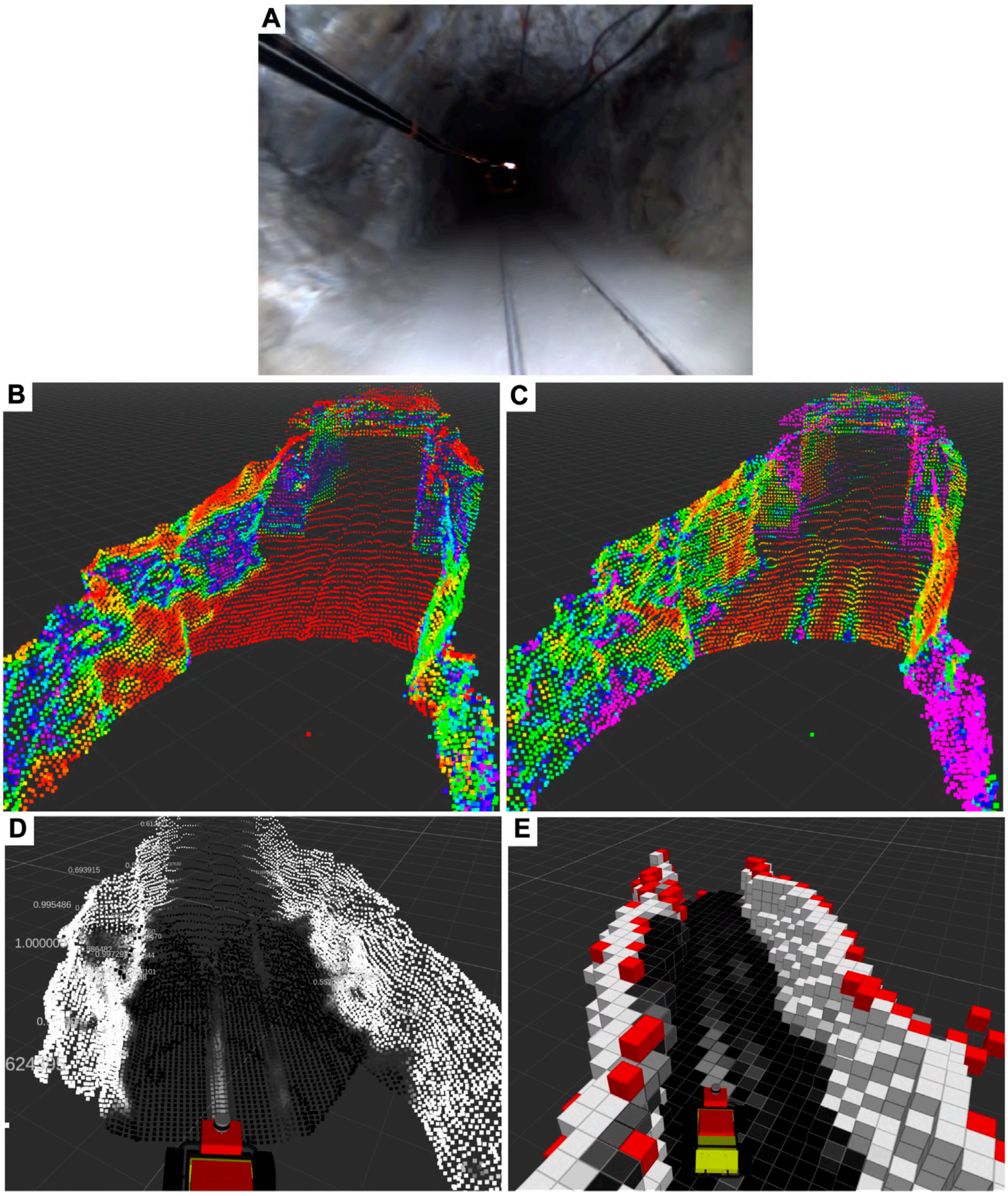

FIGURE 3
A traversability map is generated during a physical test deployment in the Edgar Mine in Idaho Springs, CO, USA **(A)**. Geometric measurements on 3D pointclouds, including normal **(B)** and curvature **(C)** estimation, are combined to produce traversability-labeled pointclouds **(D)** which are then fused into a grid map **(E)**. Note in **(E)** the classification of the walls as non-traversable (white), the rail tracks as semi-traversable (grey), and the surrounding flat ground as traversable (black). Red voxels indicate those which are occupied but have no traversability data.

We estimate the traversability cost $\tau_{v,t}$ of the region within a given voxel v of the grid map from points that correspond to that voxel $p \in \mathbb{V}$. The Figure 2 traversability-labeled pointclouds are spatially and temporally fused into a semantic grid map (Figure 3E) via an exponential moving average:

$$\tau_{v,t} = \tau_{v,t-1} P_{occ,v,t} + \tau_{p\in\mathbb{V},t} \left(1 - P_{occ,v,t}\right) \in [0,1]. \qquad (4)$$

This approach incorporates new measurements more readily when the probability of occupancy is low, as these points contain more information, and this method performed best during testing.

2.3 Stairway classification and mapping

While a geometric assessment of the environment prevents the robot from traversing over unsafe areas, there are situations where we want to exploit semantic information to explicitly traverse over known obstacles such as a stairway. To this end, we utilize the aforementioned unlabeled 3D pointcloud to classify stairways in the environment based on the work in Westfechtel et al., 2016, Westfechtel et al., 2018. We utilize this package in the proposed work for its superior performance in detecting various types of stairways from sparse pointcloud data. This package was modified to operate in the ROS framework.

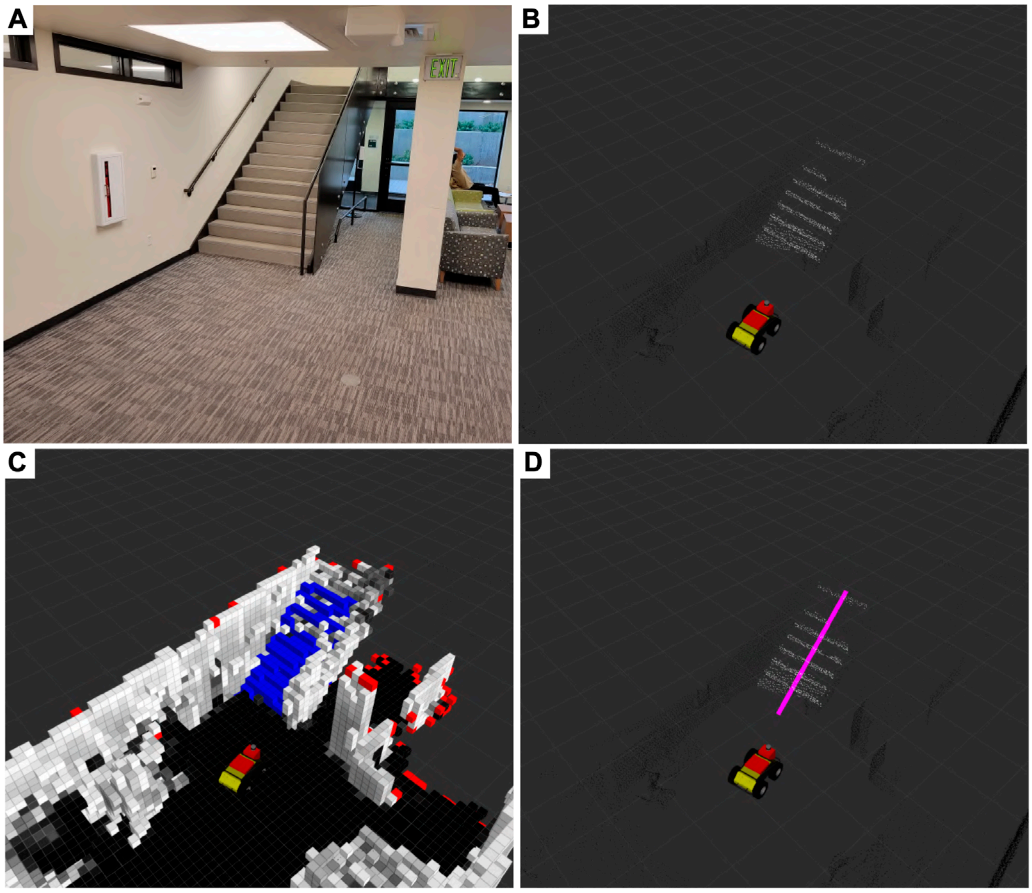

FIGURE 4
Stairways **(A)** are mapped during a physical test deployment in the Engineering Center at the University of Colorado Boulder in Boulder, CO, United States of America. Parts of a 3D pointcloud that belong to a stairway white in **(B)** are segmented via StairwayDetector (Westfechtel et al., 2016; 2018) and then fused into an occupancy map **(C)**. Additionally, we cluster stair voxels (blue) and extract the primary axis pink in **(D)** to inform the terrain-aware planner.

Briefly, classification via this approach consists of four major steps: 1) preanalysis, in which downsampling, filtering, normal/curvature estimation, and floor separation are performed; 2) segmentation via region growing algorithm, which segments the pointcloud into continuous regions; 3) plane extraction, in which the flat planes which constitute the riser, tread, and rail regions of each step are extracted; and 4) recognition, in which the tread and riser regions are connected and analyzed via a graph to determine whether they make up a valid set of stairs.

Similarly to the case of occupancy data in Section 2.1, we use the binary Bayes filter (Equations 5 and 6) to integrate the stair-labeled pointclouds (Figure 4B) into the semantic grid map (Figure 4C). We model the probability $P_{stair}(v|z_{stair,1:t})$ that a voxel v is spatially correlated with a stairway given measurements $z_{stair,1:t}$ as points in a labeled pointcloud produced by a binary stairway classifier up until time t:

$$
\begin{aligned}
&P_{stair}\left(v|z_{stair,1:t}\right)\\
&\quad= \left[1+\frac{1-P_{stair}\left(v|z_{stair,t}\right)}{P_{stair}\left(v|z_{stair,t}\right)}\,\frac{1-P_{stair}\left(v|z_{stair,1:t-1}\right)}{P_{stair}\left(v|z_{stair,1:t-1}\right)}\,\frac{P_{stair}\left(v\right)}{1-P_{stair}\left(v\right)}\right]^{-1}, \quad (5)\\
&\quad\text{where } P_{stair}\left(v|z_{stair,1:t}\right)\in(0,1);
\end{aligned}
$$

$$
\begin{aligned}
L_{stair}\left(v|z_{stair,1:t}\right) &= \log\left(\frac{P_{stair}\left(v|z_{stair,1:t}\right)}{1-P_{stair}\left(v|z_{stair,1:t}\right)}\right) && \text{(6a)}\\
&= L_{stair}\left(v|z_{stair,1:t-1}\right)+L_{stair}\left(v|z_{stair,t}\right), && \text{(6b)}\\
&\text{where } L_{stair}\left(v|z_{stair}\right)\in\left[\mathcal{L}_{stair,min},\mathcal{L}_{stair,max}\right],\\
&\mathcal{L}_{stair,min}\in(-\infty,0)\,,\mathcal{L}_{stair,max}\in(0,\infty),\\
L_{stair}\left(v|z_{stair,t}\right) &= \begin{cases}\mathcal{L}_{stair,hit}>0\,\text{on stair}\\ \mathcal{L}_{stair,miss}<0\,\text{on non}-\text{stair.}\end{cases}
\end{aligned}
$$

The parameters of the algorithm are tuned so as to heavily weigh positive stair-point detections and lightly weigh negative detections. This aggressive strategy was adopted based on observations of sparse true positive detections but very few false positive detections and to encourage comprehensive exploration by stair-capable platforms.

In addition to the semantic map itself, the principal axis of the stairway (seen in Figure 4D) and a Boolean indicator as to whether the robot is nearing the start of a stairway is calculated and published for use by the navigation system. The principal axis of the stairway is calculated by clustering the stair voxels and performing eigenvalue decomposition on the cluster. This provides a straight-line path that the navigation system can use to traverse

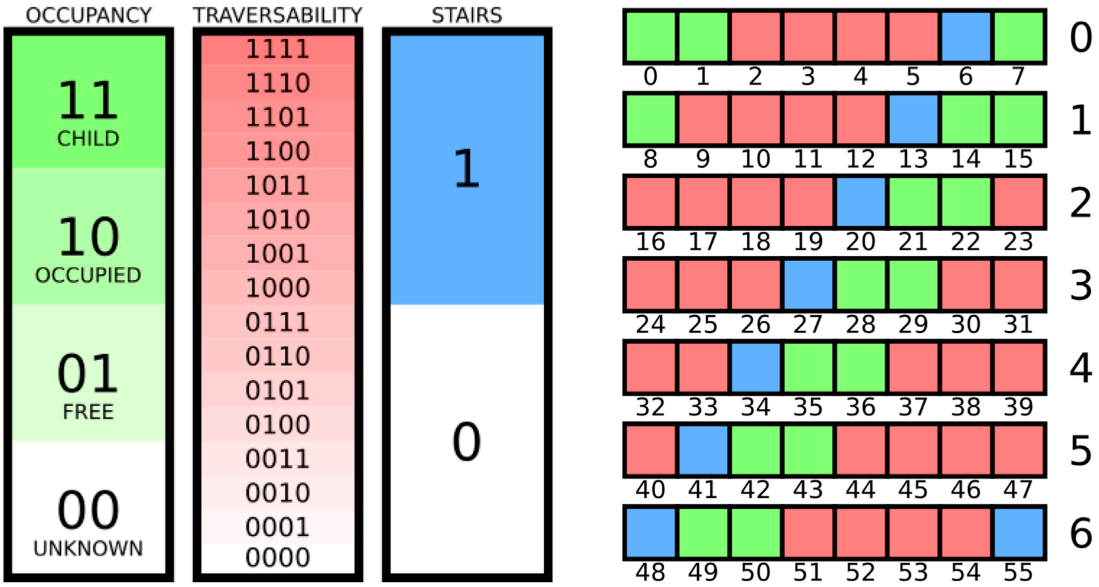

FIGURE 5
Compression scheme for encoding the semantic grid map. Occupancy information requires 2 bits, traversability information requires 4 bits, and stair-presence information requires 1 bit, for a total of 7 bits per voxel (left). All eight sub-cubes of a child node of the octree require 6 bytes (right).

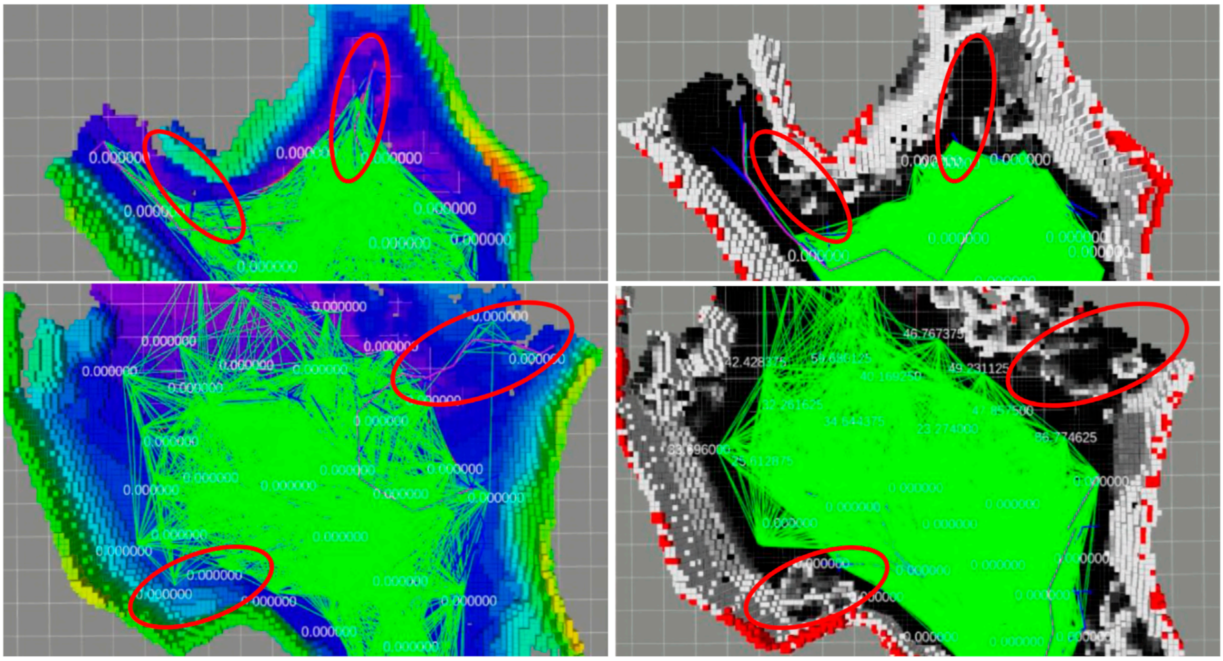

FIGURE 6
Qualitative simulation results of traversability-aware planning in the SubT Simulator's *simple_cave_03*, a simulated cave environment. We compare the span of the planner graph (green) using the proposed traversability mapping method (right; where white voxels indicate non-traversable terrain, grey indicates semi-traversable, and black indicates traversable) to planning based simply on occupancy-only mapping (left; where the grid map is colored by height). Data was collected in two scenarios (top and bottom) from a single deployment. Highlighted (red circles) are instances where the planner's graph (green lines) has extended into regions of rough terrain when planning without traversability information (left) but successfully avoided such regions when utilizing traversability information (right). An example of catastrophic navigation is also shown (rightmost red circle in the bottom-left panel) where the planner has chosen a path (purple line) that leads the robot into rough terrain, at which point the virtual robot becomes physically stuck.

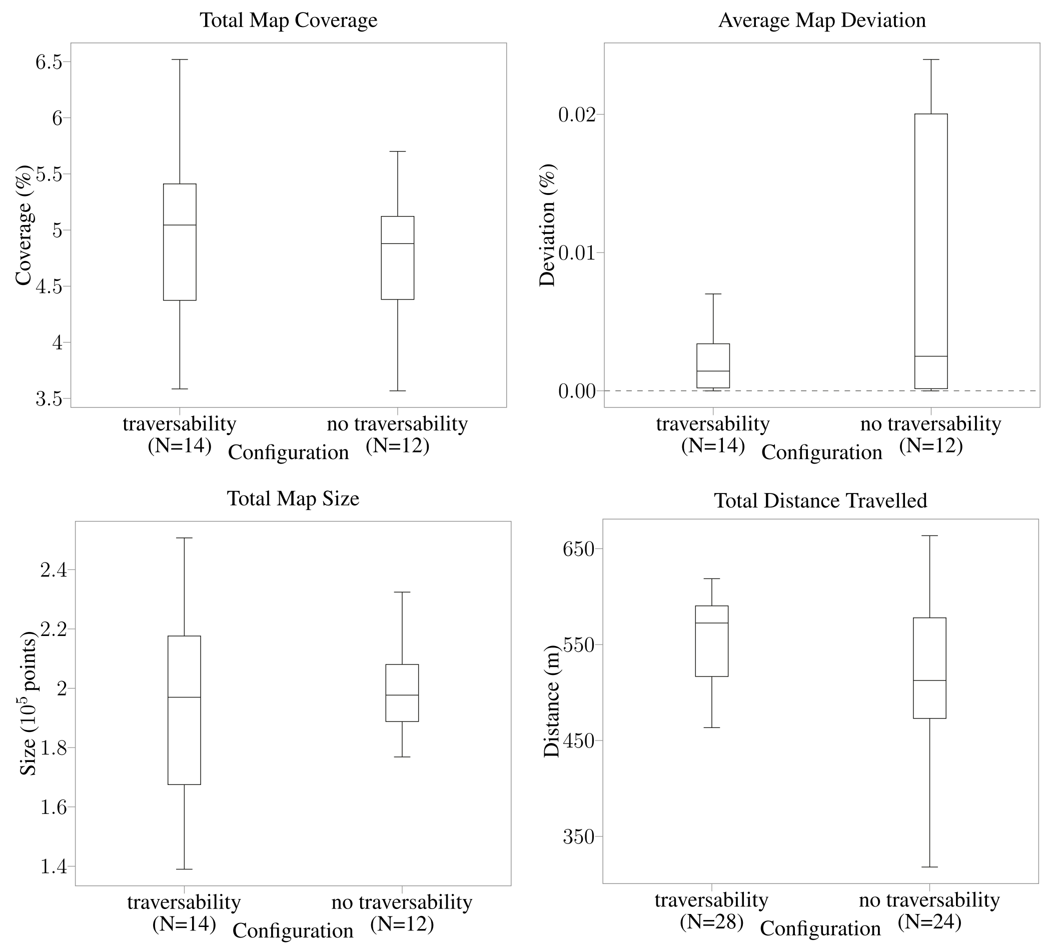

FIGURE 7
Quantitative simulation results for cooperative exploration performance in the SubT Simulator's *simple_cave_01*, a simulated cave environment. We compare exploration performance using the proposed traversability mapping method ("traversability") to occupancy-only mapping ("no traversability"). Data was collected from twenty-six deployments of a fleet of two Husky robots.

the stairway in a quick and orderly manner. The Boolean indicator serves as a signal to the low-level vehicle controller to transition from a non-stair-traversing state to a stair-traversing state and *vice versa*.

2.4 Differential map merging and sharing

Traversing large subterranean environments efficiently with multiple agents can be achieved using a shared semantic representation of the environment. However, regularly transmitting large, probabilistic, high-fidelity maps of such large environments over bandwidth-constrained wireless networks is not practical. Map differences are an intuitive solution that has been shown to facilitate efficient data transfers (Sheng et al., 2004).

Prior to sharing, the system compresses its locally-maintained, fully-probabilistic 3D grid map into a memory-efficient form. Compression involves binning the data associated with each voxel into a reduced set of classifications while preserving expressive information. An illustration of the compression scheme can be seen in Figure 5. The octree data structure also enables maps to be "pruned" on send and "expanded" on receipt to further improve network efficiency.

Occupancy information in the grid map is compressed similarly to that of the original implementation in Hornung et al. (2013):

$$B_{occ}(n) = \begin{cases} \mathcal{B}_{occupied} & \text{if } isLeaf() \text{ and } P_{occ}(v|z_{occ,1:t}) > P_{occ}(v) \\ \mathcal{B}_{free} & \text{if } isLeaf() \text{ and } P_{occ}(v|z_{occ,1:t}) < P_{occ}(v) \\ \mathcal{B}_{unknown} & \text{if } isLeaf() \text{ and } P_{occ}(v|z_{occ,1:t}) = P_{occ}(v), \\ \mathcal{B}_{child} & \text{else} \end{cases} \tag{7}$$

where $B_{occ}(n) \in \mathbb{Z}_4$ is the occupancy classification of node n in the octree, *isLeaf*() is true if the node is a "leaf" node (i.e., a node at the greatest depth of the octree and for which a corresponding voxel v exists), $\mathcal{B}_{occupied}$ is a classification which defines a leaf node as an occupied voxel, $\mathcal{B}_{free}$ is a classification which defines a leaf node as a free voxel, $\mathcal{B}_{unknown}$ is a classification which defines a leaf node as a voxel with unknown occupancy, $\mathcal{B}_{child}$ is a classification which defines the node as a non-leaf node that requires further expansion, $P_{occ}(v|z_{occ,1:t}) \in (0,1)$ is the estimated probability that the voxel is occupied after measurements $z_{occ,1:t}$, and $P_{occ}(v) \in (0,1)$ is the prior probability that the voxel is occupied.

The traversability measurement of each voxel is compressed via discretization to enable expressiveness in the traversability

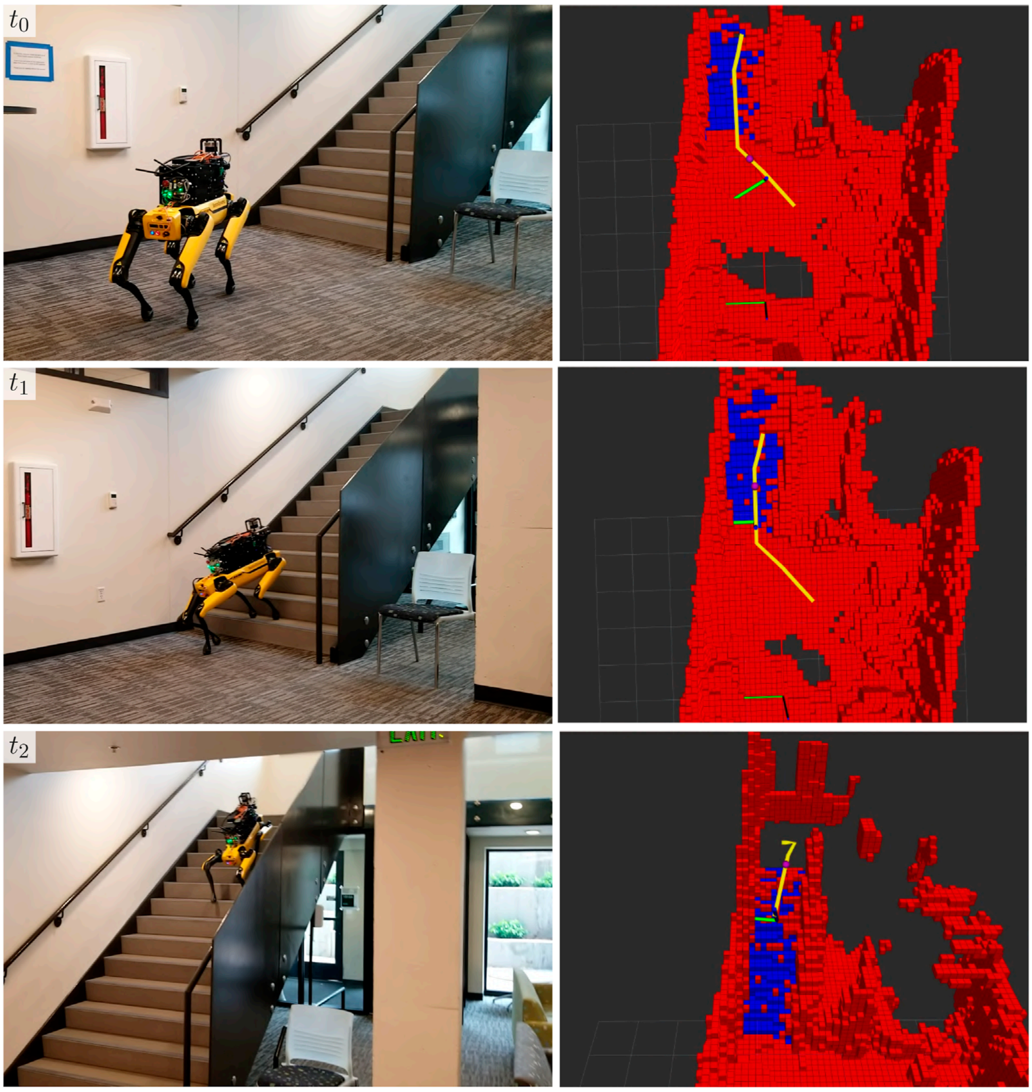

FIGURE 8
A demonstration of stair-aware planning where a Spot robot traverses a stairway (left) in the Engineering Center at the University of Colorado Boulder in Boulder, CO, United States of America. The corresponding semantic grid map and the pose of the robot with respect to it are shown (right), with stair voxels shown in blue.

information in addition to bandwidth efficiency. The measurement consumes m-bits of memory per voxel to represent 2^m traversability classifications, such that

$$B_{trav}(v) = \mathcal{B}_{trav,j} \text{ for } \mathcal{P}_{trav,j-1} < P_{trav}(v|z_{trav,1:t}) \le \mathcal{P}_{trav,j}, \tag{8}$$

where $B_{trav}(v) \in \mathbb{Z}_{2^m}$ is the discretized traversability classification of voxel v and $\mathcal{B}_{trav,j}$ is the jth classification representing probabilities between $\mathcal{P}_{trav,j-1}(v)$ and $\mathcal{P}_{trav,j}(v)$ for $j = 1, \ldots, m$. Throughout this work, $m = 4$.

Stair-presence information in the grid map is compressed via a simple binary encoding:

$$B_{stair}(v) = \begin{cases} \mathcal{B}_{stair} & \text{for } P_{stair}(v|z_{stair,1:t}) > P_{stair}(v) \\ \mathcal{B}_{nonstair} & \text{for } P_{stair}(v|z_{stair,1:t}) \le P_{stair}(v), \end{cases} \tag{9}$$

where $B_{stair}(v) \in \mathbb{Z}_2$ is the occupancy classification of voxel v, $\mathcal{B}_{stair}$ is a classification which defines a voxel as corresponding to a stairway, $\mathcal{B}_{nonstair}$ is a classification which defines a voxel as not corresponding to a stairway, $P_{stair}(v|z_{stair,1:t}) \in (0,1)$ is the estimated probability that the voxel corresponds to a stairway after measurements $z_{stair,1:t}$, and $P_{stair}(v) \in (0,1)$ is the prior probability that the voxel corresponds to a stairway.

Instead of expensively sharing whole compressed grid maps, we developed a differential mapping and merging solution to reduce network bandwidth. In our approach, each agent generates two semantic grid maps: a “self-map” (Eq. 10a) and based solely on the aggregation of its own measurements; and a “merge-map” (Eq. 10c) and based on measurements from the whole fleet. Newly updated voxels of the self-map are extracted over predetermined time intervals and constructed into a “diff-map” (Eq. 10b). The

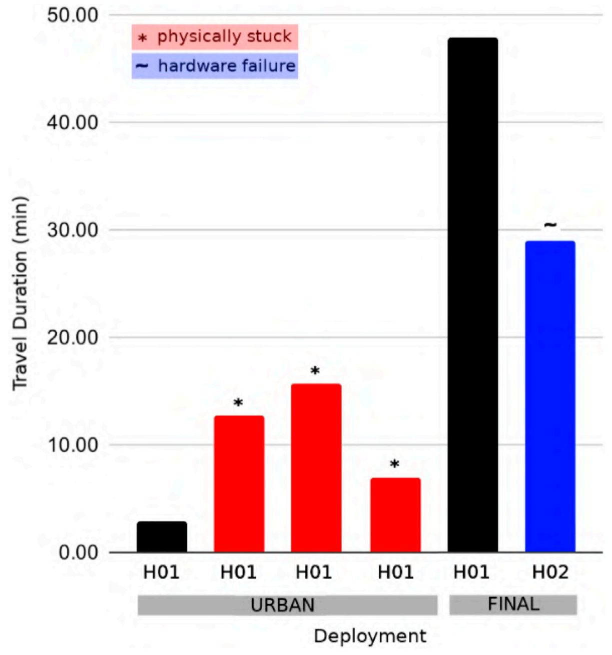

FIGURE 9
Travel duration is illustrated for physical competition deployments of wheeled Husky robots at the SubT Circuit Urban Event and the SubT Final Circuit Event. Deployments that were unable to explore for the total hour due to becoming stuck or experiencing a hardware failure are indicated. Note that the terrain-aware semantic grid mapping method proposed in this work was introduced between the Urban and Final events.

diff-maps are transmitted to other agents as network conditions allow and are incorporated into their merge-maps. Each agent gives priority to its own measurements and incorporates only those voxels from external diff-maps into its merge-map that correspond to "unknown" voxels in its self-map. This strategy ensures that the merge-map remains uncontaminated by misaligned or corrupted diff-maps sent by compromised agents, thereby eliminating the necessity for cooperative localization while still enabling cooperative exploration. Formally.

$$S_{i,t} = \mathcal{F}(v) \;\; \forall \; v \in \mathcal{V} \text{ if } P_{occ}\left(v|z_{occ,1:t}\right) \neq P_{occ}(v) \tag{10a}$$

$$D_{i,t} = S_{i,t} - \cap\left(S_{i,t-1}, S_{i,t}\right) \tag{10b}$$

$$M_{j,t} = \cup\left(S_{j,t}, D_{i,t} - \cap\left(S_{j,t}, D_{i,t}\right)\right), \tag{10c}$$

Where $S_{i,t}$ is the self-map of agent i at time t, $\mathcal{F}$ is a nonlinear aggregation function, $\mathcal{V}$ is the set of all observed voxels, $P_{occ}(v|z_{occ,1:t})$ is the probability that voxel v is occupied given measurements $z_{occ,1:t}$, $P_{occ}(v)$ is the prior probability that voxel v is occupied, $D_{i,t}$ is the diff-map of agent i at time t, and $M_{j,t}$ is the merge-map of agent j at time t (assuming the diff-map is instantaneously transmitted from agent i to agent j).

Transmission of diff-maps between agents is performed over a mesh network of radio beacons with a custom transport layer which facilitates fast reconnect times and prioritization controls (Biggie and McGuire, 2022).

2.5 Terrain-aware, cooperative path planning

A sampling-based path planner (Ahmad and Humbert, 2022) utilizes the mapping method described thus far to quickly and safely explore large, unstructured environments amidst rough terrain and stairs. The planner maintains a graph of nodes over the environment and searches the graph via A* for paths that contribute towards the exploration mission. Exploration tasks are divided into global exploration and local exploration. During global exploration, preferred paths include those with high volumetric gain in the grid map. Local exploration prioritizes sampling the immediate environment for new candidate graph nodes which are evaluated for connectivity to the main graph using candidate edges. The candidate edge is discretized into a series of candidate poses, and points spanning the vehicle footprint at each candidate pose are projected in the direction of gravity. Incident voxels are considered the "ground" and are tested for traversability and stair validity. A pose is:

- *traversability-valid* if the pose spans traversable voxels, i.e., the average traversability cost of the ground voxels is below an average traversability threshold (10% in this work) and the maximum traversability cost is below a maximum traversability threshold (20% in this work); and
- *stair-valid* if the pose spans voxels belonging to a stairway, i.e., a threshold (30% in this work) of the ground voxels have been classified as belonging to a stairway.

Threshold values were carefully adjusted over extensive physical and simulated test deployments to ensure that the observed expansion of the planner graph (see Figures 6, 12) aligned with our objective of maintaining a conservative risk profile. If each pose along the candidate edge passes all relevant validity checks, then the candidate node and edge are added, thus expanding the planner graph.

To facilitate planning with both wheeled and quadrupedal robotic platforms, we use the following rules. The stair-validity test is ignored if the vehicle is not stair-capable. The traversability-validity test is ignored if the pose is stair-valid and the vehicle is configured to be stair-capable, due to the observation that when a voxel within the grid map is identified as part of a stairway (Section 2.3), it will often be perceived as untraversable (Section 2.2) as well.

If a stairway is mapped on a stair-capable robot, it will often result in the path planner directing the robot to traverse the stairway due to its associated volumetric gain and thus increased reward during global exploration. When a stairway is mapped for a stair-capable robot, the path planner is likely to optimize the robot's trajectory to include traversing the stairway. This strategic decision is driven by the perceived volumetric gain associated with scaling the stairs, which contributes to an increased reward within the context of global exploration. In the case of the Boston Dynamics Spot robot, if the planner attempts to lead the Spot up a stairway, then prior to the start of the stairway, the low-level vehicle controller will activate Spot's stair-optimized gait and switch from tracking the plan to tracking the straight path up the stairs described in Section 2.3.

To encourage cooperative exploration, we developed two "deconfliction" schemes. The first is *map deconfliction*, in which the vehicle performs global exploration using the merged map that

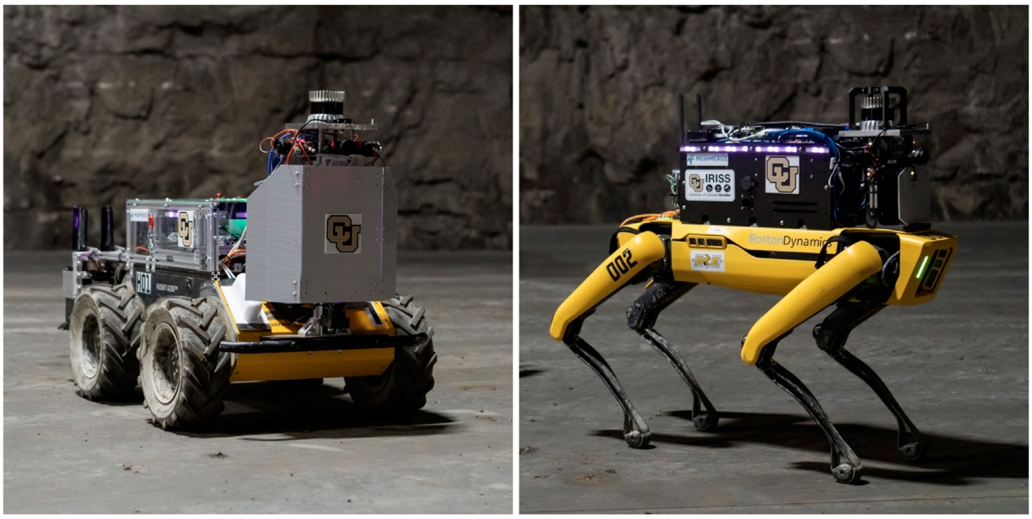

FIGURE 10
The Clearpath Husky (left) and Boston Dynamics Spot (right) robots used in physical testing and competition in the SubT Final Event.

consists of its own measurements as well as those that were measured by and received from other agents in the fleet. This allows the planner to waste no time generating paths to regions the robot has not seen previously but can result in less variety in trajectories. The second scheme is *pose deconfliction*, in which the planner performs global exploration using the historical trajectories of other agents in the fleet and is biased towards regions of the map that the fleet has not yet explored. This results in a greater variety of trajectories but can cause the vehicle to explore more slowly. The deconfliction schemes are not mutually exclusive and may also be run in parallel.

3 Results

We evaluate the proposed semantic grid mapping method in both simulated and physical experiments where robots autonomously explore previously unknown, mobility-challenged environments. In Section 3.1, we report on the traversability-mapping capabilities of the proposed method in a high-fidelity simulation environment. In Section 3.2, we physically demonstrate the utility of the stair-mapping capabilities of the proposed method. In Section 3.3, we examine the performance of the proposed mapping method as part of Team MARBLE's semi-autonomous exploration system in the DARPA Subterranean Challenge (DARPA, 2022) where the system placed third.

3.1 Traversability simulation

Simulated environments provide an opportunity to quickly and repeatedly evaluate methods in a controlled environment. To this end, we utilize the DARPA SubT Simulator (OSRF, 2022; Chung et al., 2023) which was built to facilitate the development of and competition between robotic search-and-rescue software stacks. This physics-based simulator based on the Ignition Gazebo framework is open-sourced OSRF (2022) and includes simulated sensor noise, realistic terramechanics, and bandwidth-constrained RF communications, as well as several worlds which emulate subterranean environments, such as subway tunnels and caves.

We qualitatively evaluate and illustrate in Figure 6 the effect of the traversability-mapping capability described in Section 2.2 on the propagation of the corresponding planner graph described in Section 2.5. Data was collected by running our terrain-aware navigation stack in two scenarios in the *cave_simple_03* virtual world of the SubT Simulator: with traversability-mapping capabilities enabled, and with occupancy-only mapping capabilities. Graph extension is used as a qualitative metric in each scenario to compare planning performance with traversability information to that which only utilizes occupancy information. Several instances are highlighted in which the planner graph incorrectly extended into regions of rough terrain as a result of only utilizing occupancy information, but then avoided extending into such regions when traversability was considered.

We also evaluate the effect of the traversability-mapping capability quantitatively via twenty-six simulated deployments of a fleet consisting of two Husky robots in the SubT Simulator (Figure 7). Robots were simply tasked with autonomously exploring the *cave_simple_01* virtual world of the SubT Simulator. We operated the proposed mapping system on both robots simultaneously during 1-h deployments and varied whether the traversability-mapping capabilities were enabled or disabled prior to starting. On simulation start, the first Husky was programmed to immediately start exploring; the second Husky was programmed to start exploring 1 min later. In all deployments, robots operated with occupancy- and stairway-mapping capabilities enabled; in 54% of deployments, robots also operated with traversability-mapping capabilities enabled ("traversability"). *Pose deconfliction* and *map deconfliction* (as explained in Section 2.5) were enabled and disabled, respectively, for all deployments. This is the same cooperation strategy employed during the SubT Final Event (as described in

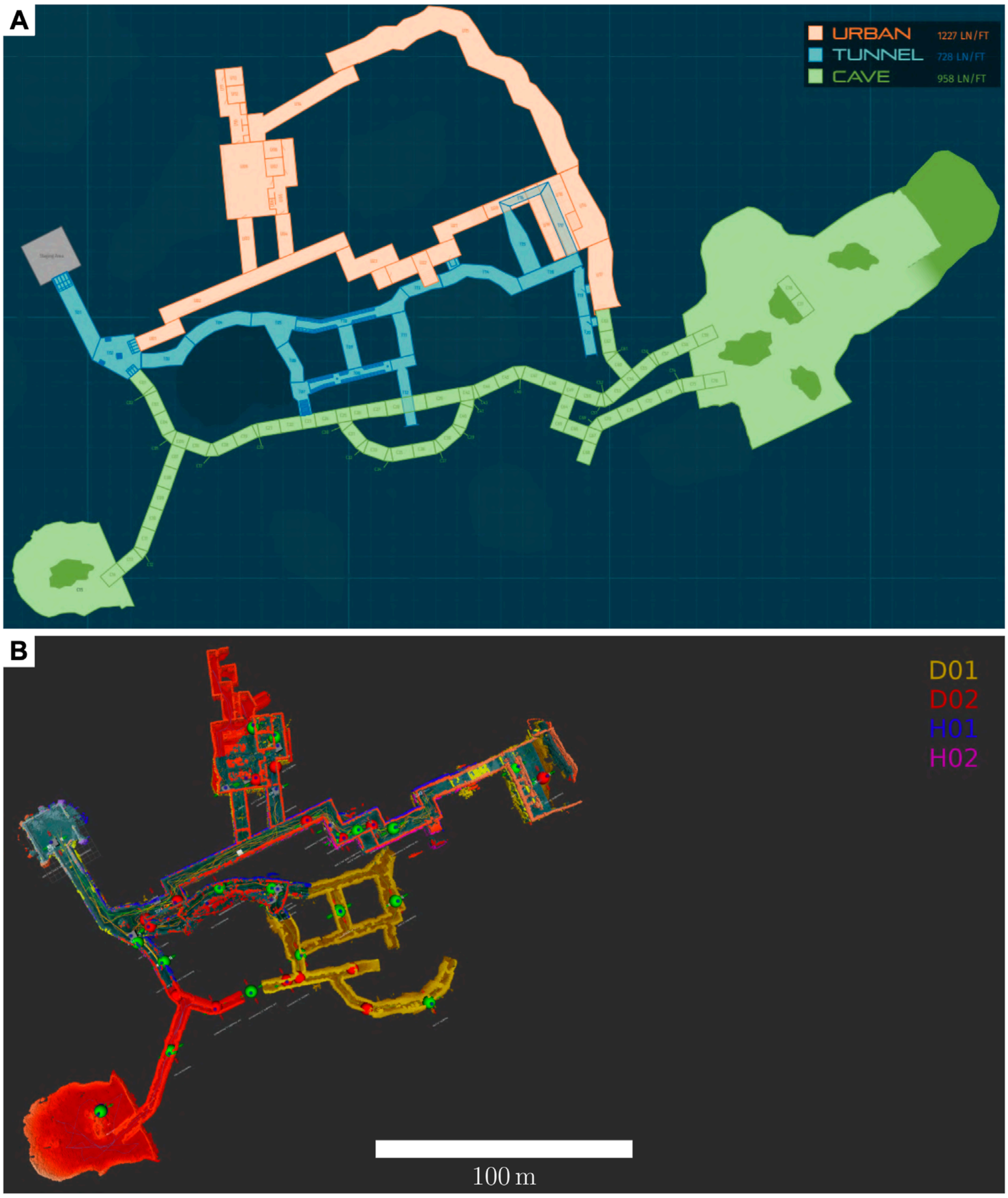

FIGURE 11
Merged map **(B)** from four robot agents during their deployment in the Subterranean Challenge Final Event course **(A)**. The map colors in A indicate the three types of subterranean environments which were emulated: tunnel, urban, and cave. In B, yellow and red maps indicate those from quadruped Spot robots, and blue and pink maps indicate those from wheeled Husky robots. All robots were deployed from the square room on the left side of the map, shown in grey in **(A)**. This map has been corrected for localization errors to illustrate the collective mapping capabilities of MARBLE Mapping.

Section 3.3) and allows for a meaningful comparison between simulated and physical experiments. Simulated deployments were performed on a PC running Ubuntu 18.04 with an AMD Ryzen 9 5900X 12-Core processor and an NVIDIA GeForce RTX 3070.

Performance is evaluated via several metrics:

1) *Total map coverage*, the percentage of ground-truth points near (within a threshold distance) an input map point at the end of the run;
2) *Average map deviation*, the average percentage of input map points far (greater than a threshold distance) from the nearest ground-truth point over the run;
3) *Total map size*, the number of points in the input map at the end of the run; and
4) *Total distance traveled* the total path length of each vehicle's trajectory at the end of the run.

Metrics one to three are provided by the SubT Map Analysis package (Schang et al., 2021) and apply to the performance of the whole

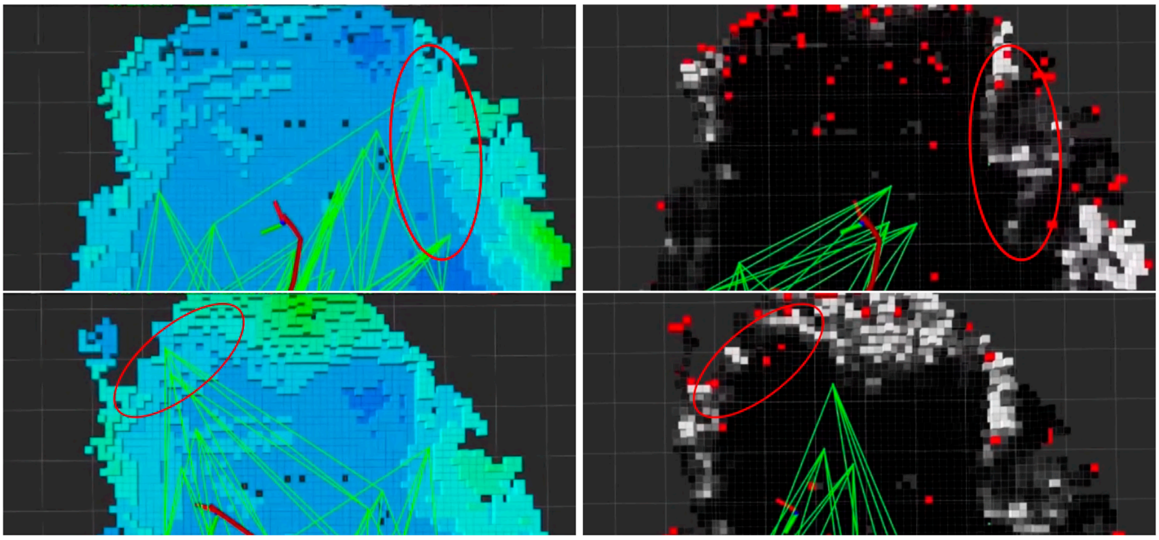

FIGURE 12
Results of traversability-aware planning (right; where white indicates non-traversable, grey indicates semi-traversable, and black indicates traversable) compared to planning based simply on occupancy (left; where the grid map is colored by height). Data was collected from a single deployment for two scenarios (top and bottom) in a physical subterranean environment. Highlighted (red circles) are instances where the planner's graph (green lines) has extended into regions of rough terrain when planning without traversability information but successfully avoided such regions when utilizing traversability information.

fleet. Metric 4 applies to the performance of each robot, hence the doubling in the number of trials.

3.2 Stair demonstration

We evaluate the stair-mapping and planning capabilities in real-world scenarios using the Boston Dynamics Spot platform. Due to limitations in simulated controllers for quadruped robots in the SubT Simulator, we were unable to quantitatively evaluate stair traversal by a quadruped. However, we observe that across hundreds of hours of physical testing leading up to the SubT Final Event, the Spot was successfully able to plan up all encountered stairways. A qualitative demonstration of the Spot planning up a stairway at the University of Colorado Boulder Engineering Center is shown in Figure 8. Note that our implementation of the proposed method is limited to only planning new routes over ascending stairways, due to constraints on the field-of-view of the lidar sensor.

3.3 DARPA subterranean (SubT) challenge

DARPA organized the SubT Challenge, a series of four competition events held between August 2018 and September 2021, to motivate the development of new robot technologies. The challenge involved teams from academic and industrial backgrounds who were tasked with designing multi-agent robotic exploration systems dedicated to search and rescue operations (DARPA, 2022). These teams competed in a variety of subterranean environments, including subterranean mines, urban areas like subway tunnels, and cave-like structures. The primary objective was to successfully navigate and search these complex terrains in order to locate artifacts, which were pre-defined objects used as indicators of human presence.

We compare the travel duration of robots from the Urban Circuit Event and the Final Circuit Event of the SubT Challenge to evaluate the performance of the traversability mapping system. In the urban event, we approximated the traversability of the terrain based solely on the occupancy grid map and utilized a Frontier-based navigation system (Ohradzansky et al., 2020; Ohradzansky et al., 2021). We developed the traversability mapping component of the mapping framework proposed in this work and deployed it in the final event to improve the travel duration and thus exploration capabilities of the robots in the fleet. We observe that the proposed mapping framework approximately doubled the total exploration time across wheeled robots in realistic scenarios filled with complex terrain (Figure 9). Furthermore, none of the robots running the traversability mapping component became physically stuck during the final event.

Team MARBLE fielded two types of robots in the SubT Challenge Final Circuit Event (Figure 10): two differential drive ClearPath Husky A200 robots (denoted with the name prefix "H"), and two quadrupedal Boston Dynamics Spot robots (denoted with the name prefix "D"). The Husky robots utilized a 64-core AMD Ryzen Threadripper 3990X equipped with 128 GB of RAM, 4 TB of SSD storage, and dual NVIDIA GTX 1650 GPUs to accelerate object detection inference times. The Spot robots utilized an AMD Ryzen 5800U with 64 GB of RAM and 2 TB of storage. This main computer was paired with a Jetson Xavier AGX to process the camera streams and perform object detection. Both robots also featured Ouster OS-1 64-beam lidars for localization and mapping, a LORD Microstrain 3DM-GX5-15 IMU for localization, and several pieces of commercial and custom electrical hardware for battery management and power distribution.

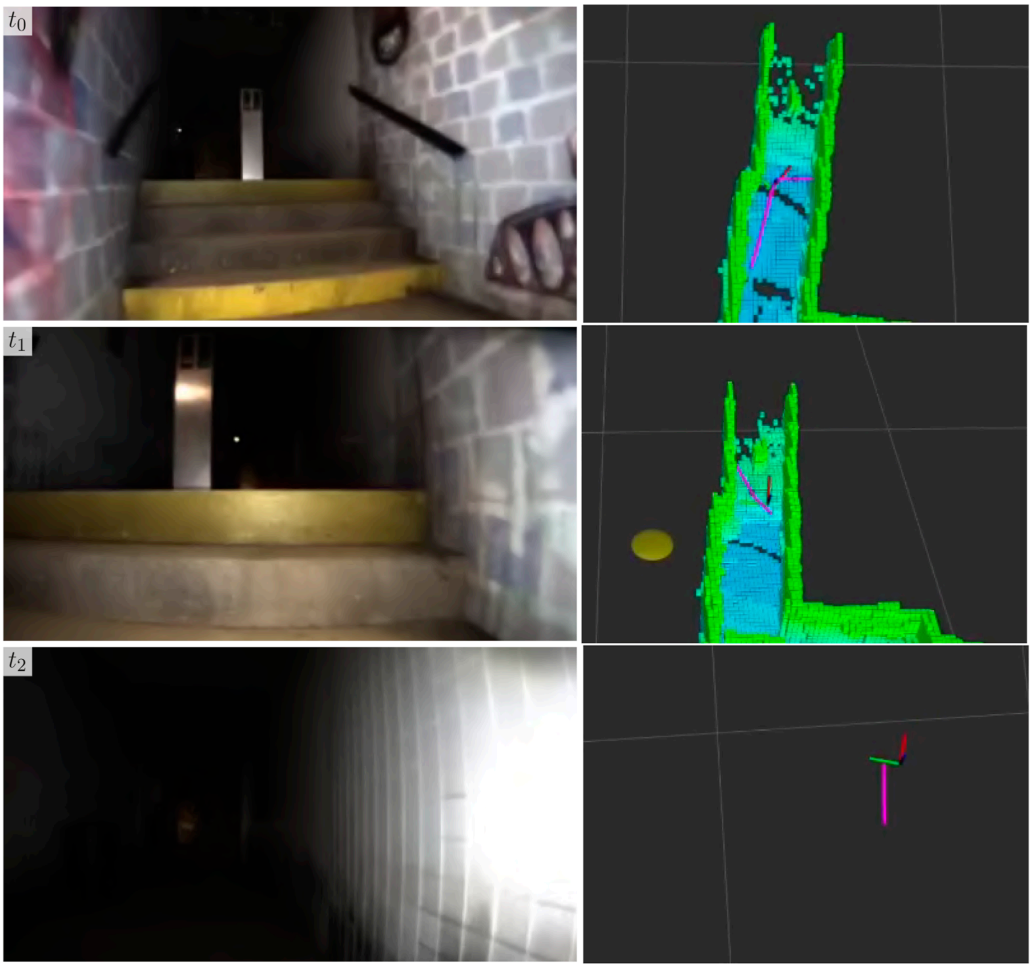

FIGURE 13
Example of stair traversal by a Boston Dynamics Spot robot during the SubT Final Circuit Event competition. The forward-facing camera of the robot (left) and the corresponding robot pose (right) is shown. In the top two rows, the occupancy map and current path are shown. In the bottom row, the occupancy map has been hidden to visualize the straight-line path produced by the stair-mapping module for a low-level vehicle controller, as described in Section 2.3.

In the implementation of the terrain-aware navigation stack used in the final event, only the Husky robots utilized the traversability information in the map, as the rough terrain avoidance capabilities built-in to the Spot robot were deemed capable of ensuring robot survival. This decision stemmed from the adoption of a strategy that discouraged any inhibition of the Spot robots' ability to explore. Additionally, only the Spot robots were deemed stair-capable and thus used the stair-presence information to explore and traverse stairs. Finally, to conserve network bandwidth, only *pose deconfliction* as described in Section 2.5 was used.

During the final event, *diff maps* from all robots are transmitted to the base station and a merged map was constructed to provide situational awareness and enable flexible, semi-autonomous control to a human operator. The merged map (Figure 11) illustrates the extent to which the fleet explored the environment.

We illustrate the qualitative performance of the traversability-mapping capabilities in physical environments (Figure 12) using the approach introduced in Section 3.1. Planner graph extension is used as a metric to compare planning performance with traversability information to that which only utilizes occupancy information. Data was collected by running Team MARBLE's terrain-aware navigation stack during a post-competition deployment at the SubT Final Circuit Event.

Qualitative results of the stairway classification and mapping method developed in Section 2.3 at the final event are shown in Figure 13, in which a Spot robot traverses a short flight of stairs in the urban portion of the environment during a competition deployment. The planner directs the robot up the stairs, however, because it accurately detects the stairs and extracts the straight-line path up the stairs, it follows the straight-line path rather than the original plan which would have led the robot up the stairs at an angle and jeopardized the deployment. As a result of traversing this stairway, the robot was able to detect one more artifact, and potentially two more had the object detection stack worked as expected. Since it had already seen the stairway, the robot was also able to easily plan back down the stairway and continue exploring other areas of the map.

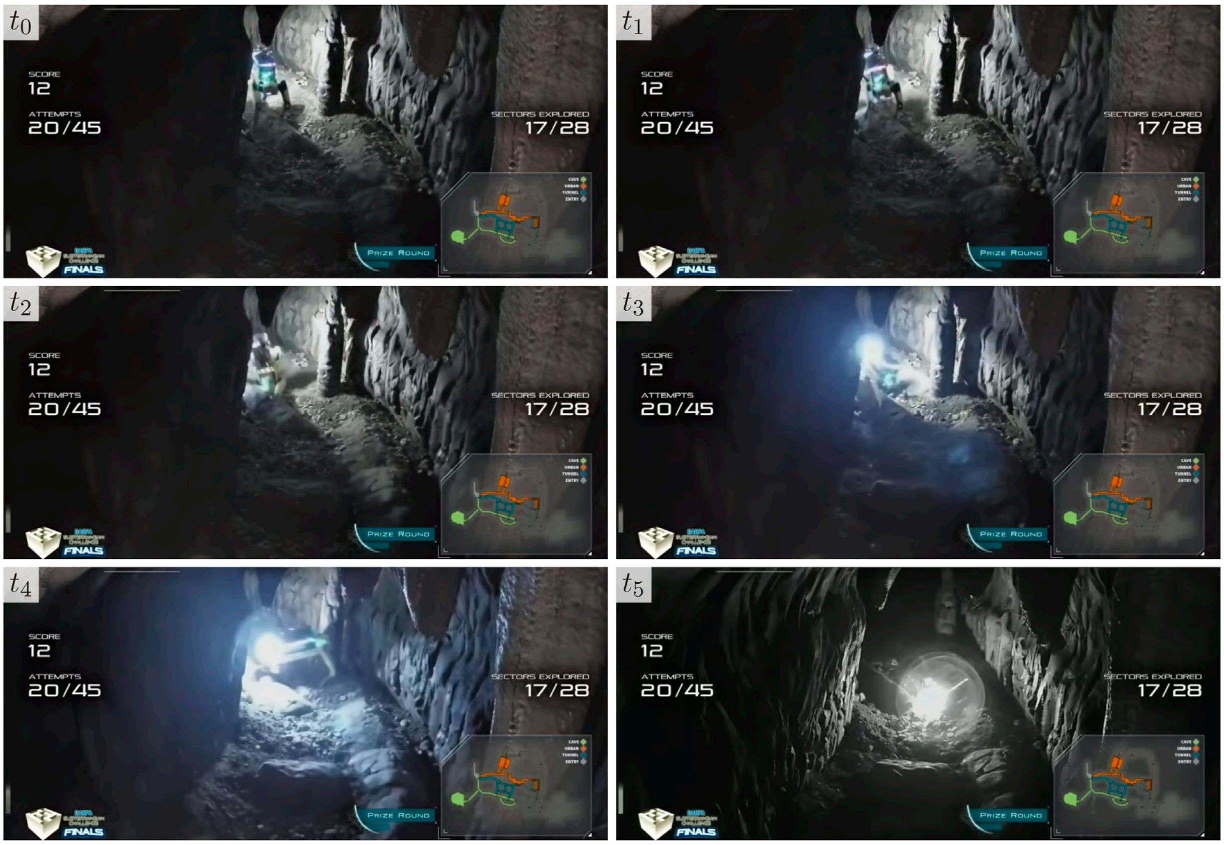

FIGURE 14
Instance from the SubT Final Event where a Spot robot became unstable while attempting to traverse rough terrain. The robot was not running the traversability mapping component proposed in this work. Photo credit: DARPA.

4 Discussion

4.1 Performance of traversability-mapping

The traversability-mapping component of the proposed terrain-mapping solution performed excellently in its function of facilitating safe and conservative planning in order to enable continuous exploration. This is evidenced by the improvement in travel duration upon introduction of the proposed terrain-mapping solution as shown in Figure 9. As can be seen, the robots which utilized the traversability-aware navigation stack in the final event traveled for much longer, and without any instances where they got physically stuck. This is largely a result of the traversability mapping capabilities of the terrain-aware mapping method presented in this chapter, however, the roles of operational experience and other technical improvements in maximizing exploration duration cannot be ignored. The qualitative effects are apparent when comparing the planning graph for traversability mapping to occupancy-only mapping (Figures 6, 12). An example of catastrophic navigation is shown in Figure 6, where the occupancy-only planner directs the robot to enter rough terrain, at which point the robot becomes physically stuck and the simulated deployment must terminate. The quantitative effects are also significant, as shown in Figure 7. We see that traversability-aware mapping results in more distance traveled due to enduring exploration and less average map deviation due to reduced localization error caused by traversing rough terrain. While traversability-mapping also enabled the greatest map coverage and largest map size, the aggregate improvement is less significant, likely due to the deconfliction strategy discussed in Section 2.5.

4.2 Performance of stair-mapping

The stair-mapping component of the proposed terrain-mapping solution performed well in its function of facilitating more expressive exploration. This is evidenced by the additional artifact that the robot scored as a result of climbing a short flight of stairs during the final event competition deployment. The effect of this artifact is particularly fortunate considering that Team MARBLE scored just one point higher than the next-highest scorer DARPA 2022.

4.3 Limitations

Despite the excellent performance, the proposed terrain-mapping system has some limitations.

Perhaps the most crucial limitation is the challenge of observing and subsequently mapping and traversing a descending stairway, due to the lidar sensor lacking a field of view that encompasses the ground directly in front of the robot. This limitation resulted in the vehicle failing to explore a large region of the environment in the SubT Final Event.

Another limitation is the inability to reliably map very thin obstacles, such as dangling cables or wire fencing which feature prominently in true search and rescue scenarios. This challenge is inherent to many grid mapping approaches and requires more sophisticated classification techniques to address.

Finally, as described in Section 3.3, the Spot robots operated in the SubT Final Event without the proposed traversability-mapping component to maximize exploration. While this may have improved the Spot robot exploration performance, it also resulted in one instance where the robot attempted and failed to traverse rough terrain, resulting in a fall and no subsequent travel for the remainder of the run (Figure 14).

4.4 Conclusions and future work

Future improvements may address these aforementioned limitations and enable even safer and more comprehensive exploration. One such improvement would be to expand the field of view of the stairway classification and mapping sensors to enable the traversal of descending stairways. Future work may also include the “roughness” (i.e., the quality of a plane-fitting operation) in the traversability cost calculation. This would improve the perception of thin obstacles and the traversability classification of rough terrain, however, such operations can be computationally expensive. Finally, the proposed mapping framework’s modularity facilitates its compatibility with various multi-agent and navigation stacks. Future work may utilize this feature to allow an independent evaluation of its performance.

Nonetheless, the proposed terrain-mapping solution demonstrates an effective method to enable safe and enduring exploration in large, unstructured, subterranean environments with bandwidth-constrained communication. This is evidenced by statistical analysis of performance in simulation and Team MARBLE’s third place finish in the DARPA Subterranean Challenge Final Event.

Data availability statement

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession numbers can be found below: arpg.colorado.edu.

Author contributions

MM conducted most of the experiments, set up the software and robots, and was a primary developer and operator. HB supported MM in these tasks and also primarily contributed to the networking stack. CH provided project oversight and technical direction in each of these areas. All authors contributed to the article and approved the submitted version.

Funding

This work was funded by DARPA Cooperative Agreement HR0011-18-2-0043, NSF #1764092, NSF #1830686, and USDA-NIFA 2021-67021-33450. Results presented in this paper were obtained using the Chameleon testbed supported by the National Science Foundation.

Acknowledgments

A special thanks to Team MARBLE, especially Shakeeb Ahmad and Michael Ohradzanksy for their efforts on the navigation stack and Dan Riley for his effort on the cooperative mapping stack that were featured in this work. Finally, the authors would like to acknowledge the DARPA Subterranean Challenge staff for their hard work in designing, building, and hosting the SubT Challenge.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher’s note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Ahmad, S., and Humbert, J. S. (2022). Efficient sampling-based planning for subterranean exploration. In *2022 IEEE/RSJ international conference on intelligent robots and systems (IROS)* IEEE, 7114–7121.

Amanatides, J., and Woo, A. (1987). A fast voxel traversal algorithm for ray tracing. *Eurographics* 87, 3–10. doi:10.2312/egtp.19871000

Biggie, H., and McGuire, S. (2022). “Heterogeneous ground-air autonomous vehicle networking in austere environments: practical implementation of a mesh network in the darpa subterranean challenge,” in *2022 18th international conference on distributed computing in sensor systems (DCOSS)* (IEEE), 261–268.

Biggie, H., Rush, E. R., Riley, D. G., Ahmad, S., Ohradzansky, M. T., Harlow, K., et al. (2023). Flexible supervised autonomy for exploration in subterranean environments. *J. Field Robotics* 3, 125–189. doi:10.55417/fr.2023004

Chung, T. H., Orekhov, V., and Maio, A. (2023). Into the robotic depths: analysis and insights from the darpa subterranean challenge. *Annu. Rev. Control, Robotics, Aut. Syst.* 6, 477–502. doi:10.1146/annurev-control-062722-100728

DARPA (2022). *DARPA subterranean challenge*. Available at: https://www.darpa.mil/program/darpa-subterranean-challenge.

Fan, D. D., Otsu, K., Kubo, Y., Dixit, A., Burdick, J., and Agha-Mohammadi, A.-A. (2021). *Step: Stochastic traversability evaluation and planning for safe off-road navigation. arXiv preprint arXiv:2103.02828*.

Fankhauser, P., Bloesch, M., Gehring, C., Hutter, M., and Siegwart, R. (2014). "Robot-centric elevation mapping with uncertainty estimates," in *Mobile service robotics* (World Scientific), 433–440.

Fankhauser, P., and Hutter, M. (2016). "A universal grid map library: implementation and use case for rough terrain navigation," in *Robot operating system (ROS)* (Springer), 99–120.

Guan, T., Kothandaraman, D., Chandra, R., Sathyamoorthy, A. J., Weerakoon, K., and Manocha, D. (2022). Ga-nav: efficient terrain segmentation for robot navigation in unstructured outdoor environments. *IEEE Robotics Automation Lett.* 7, 8138–8145. doi:10.1109/lra.2022.3187278

Haddeler, G., Chuah, M. Y. M., You, Y., Chan, J., Adiwahono, A. H., Yau, W. Y., et al. (2022). Traversability analysis with vision and terrain probing for safe legged robot navigation. *Front. Robotics AI* 9, 887910. doi:10.3389/frobt.2022.887910

Harms, H., Rehder, E., Schwarze, T., and Lauer, M. (2015). "Detection of ascending stairs using stereo vision," in *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (IEEE), 2496–2502.

Hebert, M., Caillas, C., Krotkov, E., Kweon, I.-S., and Kanade, T. (1989). "Terrain mapping for a roving planetary explorer," in *IEEE international conference on robotics and automation (IEEE)*, 997–1002.

Hines, T., Stepanas, K., Talbot, F., Sa, I., Lewis, J., Hernandez, E., et al. (2021). Virtual surfaces and attitude aware planning and behaviours for negative obstacle navigation. *IEEE Robotics Automation Lett.* 6, 4048–4055. doi:10.1109/lra.2021.3065302

Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., and Burgard, W. (2013). Octomap: an efficient probabilistic 3d mapping framework based on octrees. *Aut. robots* 34, 189–206. doi:10.1007/s10514-012-9321-0

Hughes, D., Heckman, C., and Correll, N. (2019). Materials that make robots smart. *Int. J. Robotics Res.* 38, 1338–1351. doi:10.1177/0278364919856099

Labbé, M., and Michaud, F. (2019). Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *J. Field Robotics* 36, 416–446. doi:10.1002/rob.21831

Lluvia, I., Lazkano, E., and Ansuategi, A. (2021). Active mapping and robot exploration: A survey. *Sensors* 21, 2445. doi:10.3390/s21072445

Moravec, H., and Elfes, A. (1985). "High resolution maps from wide angle sonar," in *Proceedings. 1985 IEEE international conference on robotics and automation (IEEE)*, 2, 116–121.

Ohradzansky, M. T., Mills, A. B., Rush, E. R., Riley, D. G., Frew, E. W., and Humbert, J. S. (2020). "Reactive control and metric-topological planning for exploration," in *2020 IEEE international conference on robotics and automation (ICRA)* (IEEE), 4073–4079.

Ohradzansky, M. T., Rush, E. R., Riley, D. G., Mills, A. B., Ahmad, S., McGuire, S., et al. (2021). *Multi-agent autonomy: Advancements and challenges in subterranean exploration. arXiv preprint arXiv:2110.04390*.

Oleynikova, H., Taylor, Z., Fehr, M., Siegwart, R., and Nieto, J. (2017). "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), 1366–1373.

OSRF (2022). *DARPA SubT simulator by OSRF*. Available at: https://github.com/osrf/subt.

Ouster (2023). *OS1*. Available at: https://ouster.com/products/scanning-lidar/os1-sensor/.

Payeur, P., Hébert, P., Laurendeau, D., and Gosselin, C. M. (1997). Probabilistic octree modeling of a 3d dynamic environment. *Proc. Int. Conf. Robotics Automation (IEEE)* 2, 1289–1296. doi:10.1109/ROBOT.1997.614315

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., et al. (2009). "Ros: an open-source robot operating system," in *ICRA workshop on open source software (Kobe, Japan)*, 3, 5.

Rodríguez-Martínez, D., Van Winnendael, M., and Yoshida, K. (2019). High-speed mobility on planetary surfaces: A technical review. *J. Field Robotics* 36, 1436–1455. doi:10.1002/rob.21912

Sánchez-Rojas, J. A., Arias-Aguilar, J. A., Takemura, H., and Petrilli-Barceló, A. E. (2021). Staircase detection, characterization and approach pipeline for search and rescue robots. *Appl. Sci.* 11, 10736. doi:10.3390/app112210736

Schang, A., Rogers, J., and Maio, A. (2021). *Map analysis*. (Version 1.0). Available at: https://github.com/subtchallenge/map_analysis.

Seo, J., Kim, T., Ahn, S., and Kwak, K. (2023). *Metaverse: Meta-learning traversability cost map for off-road navigation. arXiv preprint arXiv:2307.13991*.

Shaban, A., Meng, X., Lee, J., Boots, B., and Fox, D. (2022). "Semantic terrain classification for off-road autonomous driving," in *Conference on robot learning* (PMLR), 619–629.

Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., and Rus, D. (2020). "Lio-sam: tightly-coupled lidar inertial odometry via smoothing and mapping," in *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (IEEE), 5135–5142.

Shan, T., Wang, J., Englot, B., and Doherty, K. (2018). "Bayesian generalized kernel inference for terrain traversability mapping," in *Conference on robot learning* (PMLR), 829–838.

Sheng, W., Yang, Q., Ci, S., and Xi, N. (2004). "Multi-robot area exploration with limited-range communications," in *2004 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (Japan: Sendai), 2, 1414–1419.

Thrun, S. (2002). Probabilistic robotics. *Commun. ACM* 45, 52–57. doi:10.1145/504729.504754

Westfechtel, T., Ohno, K., Mertsching, B., Hamada, R., Nickchen, D., Kojima, S., et al. (2018). Robust stairway-detection and localization method for mobile robots using a graph-based model and competing initializations. *Int. J. Robotics Res.* 37, 1463–1483. doi:10.1177/0278364918798039

Westfechtel, T., Ohno, K., Mertsching, B., Nickchen, D., Kojima, S., and Tadokoro, S. (2016). "3d graph based stairway detection and localization for mobile robots," in *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (IEEE), 473–479.

Ye, C. (2007). Navigating a mobile robot by a traversability field histogram. *IEEE Trans. Syst. Man, Cybern. Part B Cybern.* 37, 361–372. doi:10.1109/tsmcb.2006.883870