ETH zürich Research Collection

Journal Article

Graph-based Subterranean Exploration Path Planning using Aerial and Legged Robots

Author(s):
Dang, Tung; Tranzatto, Marco; Khattak, Shehryar Masaud Khan; Mascarich, Frank; Alexis, Kostas; Hutter, Marco

Publication Date:
2020-12

Permanent Link:
https://doi.org/10.3929/ethz-b-000450626 →

Originally published in:
Journal of Field Robotics 37(S 8), http://doi.org/10.1002/rob.21993 →

Rights / License:
In Copyright - Non-Commercial Use Permitted →

This page was generated automatically upon download from the ETH Zurich Research Collection. For more information please consult the Terms of use.

ETH Library

Graph-based Subterranean Exploration Path Planning using Aerial and Legged Robots

Tung Dang
Dept of Computer Science and Engineering
University of Nevada, Reno
tung.dang@nevada.unr.edu

Marco Tranzatto
Dept of Mechanical and Process Engineering
ETH Zurich
marcot@ethz.ch

Shehryar Khattak
Dept of Computer Science and Engineering
University of Nevada, Reno
shehryar@nevada.unr.edu

Frank Mascarich
Dept of Computer Science and Engineering
University of Nevada, Reno
fmascarich@nevada.unr.edu

Kostas Alexis
Dept of Computer Science and Engineering
University of Nevada, Reno
kalexis@unr.edu

Marco Hutter
Dept of Mechanical and Process Engineering
ETH Zurich
mahutter@ethz.ch

Abstract

Autonomous exploration of subterranean environments remains a major challenge for robotic systems. In response, this paper contributes a novel graph-based subterranean exploration path planning method that is attuned to key topological properties of subterranean settings, such as large-scale tunnel-like networks and complex multi-branched topologies. Designed both for aerial and legged robots, the proposed method is structured around a bifurcated local- and global-planner architecture. The local planner utilizes a rapidly-exploring random graph to reliably and efficiently identify paths that optimize an exploration gain within a local subspace, while simultaneously avoiding obstacles, respecting applicable traversability constraints and honoring dynamic limitations of the robots. Reflecting the fact that multi-branched and tunnel-like networks of underground environments can often lead to dead-ends and accounting for the robot endurance, the global planning layer works in conjunction with the local planner to incrementally build a sparse global graph and is engaged when the system must be re-positioned to a previously identified frontier of the exploration space, or commanded to return-to-home. The designed planner is detailed with respect to its computational complexity and compared against state-of-the-art approaches. Emphasizing field experimentation, the method is evaluated within multiple real-life deployments using aerial robots and the ANYmal legged system inside both long-wall and room-and-pillar underground mines in the U.S. and Switzerland, as well as inside an underground bunker. The presented results further include missions conducted within the DARPA Subterranean Challenge, a relevant competition on underground exploration.

1 Introduction

The collective progress in robotics has enabled the potential of autonomous robotic exploration and mapping missions to expand into an ever increasing set of applications in both the civilian and military domains alike. Aerial and ground robots are currently employed in a multitude of search and rescue (Balta et al., 2017; Tomic et al., 2012; Michael et al., 2012; Delmerico et al., 2019), industrial inspection (Bircher et al., 2016a; Bircher et al., 2015; Caprari et al., 2012; Balaguer et al., 2000; Sawada et al., 1991; Gehring et al., 2019; Hutter et al., 2018; Kolvenbach et al., 2019), surveillance (Grocholsky et al., 2006), and commercial applications (Rao et al., 2016). Despite the significant progress in both systems and methods, a variety of critical environments remain particularly challenging for robotic access and resilient autonomy. This especially relates to subterranean settings, the autonomous exploration of which remains particularly difficult and calls for new contributions. This fact is reflected by the coordinated efforts of the community organized around the DARPA Subterranean Challenge. Subterranean worlds present a set of properties that render robotic autonomy, whether ground or aerial, difficult, including the fact that they are often extremely long and large-scale, particularly narrow and confined, multi-branched and multi-level, self-similar, sensing-degraded, and communications-deprived. Also, they often feature rough and dynamic terrain including mud, rubble and stairs. Despite these challenges, the importance of robotic autonomy underground and its associated benefits across diverse application domains calls for accelerating the relevant research. Robots for mine rescue, inspection of subterranean metropolitan infrastructure (e.g., subways and sewers), as well as exploratory and scientific missions inside caves and lava tubes on both Earth and in space are all exemplary domains.

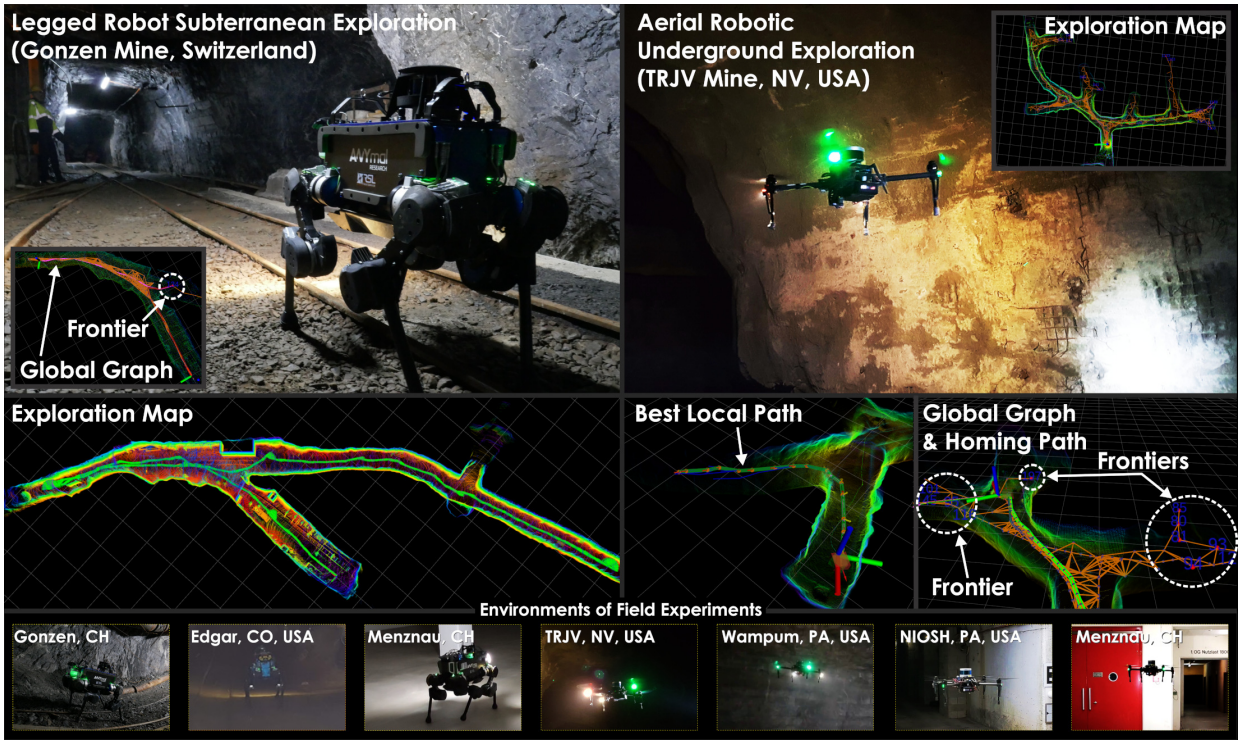

Figure 1: Instances of autonomous graph-based exploration path planning in subterranean environments using both legged and aerial robots. The presented results are based on field tests in multiple, diverse and both active and inactive mines in the U.S. and in Switzerland, as well as an underground bunker.

In this work we investigate potential approaches to address the challenges of autonomous subterranean exploration with both flying and legged robots as depicted in Figure 1. In particular, this paper emphasizes the exploration path planning problem in a manner cognizant to the particularities of subterranean settings and puts further focus on extensive field evaluation in multiple underground mines and bunkers. We propose a new comprehensive Graph-based exploration path Planner (GBPlanner) that builds on a bifurcated *local*

and *global* planning architecture tailored to the underground domain. The local planner operates efficiently within a bounded volume around the current robot location. It employs a rapidly-exploring random graph sampling kernel to evaluate robot paths that maximize an information gain related to exploring more unmapped volume, while simultaneously ensuring collision-avoidance (for either flying or walking systems) and respecting applicable traversability constraints (walking robots). The best path is selected such that the information gain is maximized and is subsequently refined to enhance its "safety" properties. However, within complex and large scale subterranean settings the local planner may report inability to identify admissible paths that offer sufficient information gain (e.g., when the robot reaches a dead-end such as a mine heading). In such situations, the global planning step is engaged and ensures, by searching through a global yet sparse and incrementally built graph, the identification of a path towards a previously perceived edge of the exploration space (e.g., an unvisited mine branch) and thus efficiently re-positions the robot to continue its mission. Furthermore, the global planner also continuously updates a return-to-home path which is triggered when the robot's remaining endurance approaches its limits. This bifurcated sampling-based scheme was identified to be capable of ensuring efficient and robust exploration for the large-scale, narrow, long, often tunnel-like and multi-branched underground environments. It also accounts for possible robot traversability constraints and providing a notion of directionality which is frequently applicable in subterranean settings. The proposed exploration path planning solution for subterranean environments was evaluated using both legged and flying robots tested first in simulation and then within multiple field experiments inside a) both active and inactive long and narrow "long-wall" gold and iron mines in the U.S. and in Switzerland, b) "room-and-pillar" coal mines, and c) an underground bunker. The results also include field deployments in the framework of the DARPA Subterranean Challenge Tunnel Circuit and are accompanied with open-source code and associated datasets that are openly released to the community. In terms of extending our previous efforts in (Dang et al., 2019b; Dang et al., 2019a), this work makes the following additional contributions:

- A revised local graph search that maximizes the robot safety through a new path refinement step.
- A novel global planning methodology that refines the global graph to enhance its topologies and offer improved paths to re-position the robot to frontiers of the exploration space or return-to-home.
- The explicit detection and annotation of frontiers that are then used by the global planning stage to re-position the robot such that it can continue its exploration efficiently.
- The extension of the planner for legged systems by incorporating traversability analysis.
- Detailed complexity analysis for all the steps of both the local and global planning stages.

At the same time this paper emphasizes comprehensive evaluation with a focus on field experiments. To best evaluate our contribution and enable research reproducibility, this paper further contributes:

- An extensive set of new experimental studies involving both flying and walking robots in different types and topologies of underground mines, and a subterranean bunker.
- Comparative studies of the proposed method against approaches of the current state-of-the-art indicating that GBPlanner outperforms established current methods.
- The open-source release of the contributed method on graph-based exploration path planning (ARL, 2020), as well as an extensive dataset (ARL and RSL, 2019).

Finally, we make an effort to make this publication to be as standalone as possible, thus allowing the interested reader to fully understand the method, its performance and possible limitations.

The remainder of this paper is structured as follows: Section 2 presents related work, followed by the problem statement in Section 3. The proposed approach is detailed in Section 4, while evaluation studies are presented in Sections 5 and 6. Finally, conclusions are drawn in Section 7.

2 Related Work

A rich body of work has focused on the general problem of exploration path planning (Marchant and Ramos, 2014; Jones et al., 2013; Berenson et al., 2012; Stoyanov et al., 2010; Popovic et al., 2016; Papachristos et al., 2017; Bircher et al., 2016b; Yoder and Scherer, 2016; Connolly, 1985; Yamauchi, 1997; Arora and Scherer, 2017; Corah and Michael, 2019; Hollinger and Sukhatme, 2014). Early work includes the sampling of "next–best–views" (Connolly, 1985), and frontiers–based exploration (Yamauchi, 1997). More recent efforts have proposed receding horizon multi-objective planning (Bircher et al., 2016b; Papachristos et al., 2019; Dang et al., 2018; Papachristos et al., 2017), multi-robot methods (Delmerico et al., 2017; Andre and Bettstetter, 2016), and emphasized extensive field work (Yoder and Scherer, 2016; Mascarich et al., 2018). Existing planning methods, however, are mostly designed to deal with either outdoor areas or relatively small structured facilities and do not reflect the challenges of the very large-scale, tunnel-like, multi-branching, narrow, and broadly complex subterranean environments (e.g. mines, cave networks, subway infrastructure).

In response to these facts, a niche community has investigated custom solutions to the problem of subterranean exploration. The works in (Silver et al., 2006; Baker et al., 2004) present methods for topological exploration of underground environments based on edge exploration and the detection of intersections. It has been verified on ground platforms with the Groundhog system presenting pioneering levels of exploration capacity. The works in (Calisi et al., 2005; MacAllister et al., 2013; Rufli et al., 2009; Richter et al., 2016) emphasized motion planning in geometrically-constrained environments although not specific to subterranean domains. The contribution in (Li et al., 2019) relates to underground exploration and proposes a navigation solution but lacks field testing. The methods in (Mansouri et al., 2019a; Kanellakis et al., 2019; Mansouri et al., 2019b) outline works on underground aerial robotic exploration via open space attraction, junction and contours detection. The work in (Jacobson et al., 2020) emphasizes the localization challenges in subterranean settings. From a systems perspective, the works in (Morris et al., 2006; Novák et al., 2015) overview innovations in ground and submersible subterranean robotics, while the paper in (Tardioli et al., 2019) provides an overview on the use of ground robots underground. Furthermore, the methods in (Bakambu and Polotski, 2007; Lösch et al., 2018) outline navigation solutions for robotic rovers underground.

Accelerated by the kick-off of the DARPA Subterranean Challenge, the domain of underground exploration path planning is currently experiencing rapid growth. Examples of this relate to the works in (Miller et al., 2020) using legged systems, a dataset on relevant artifact classification (Shivakumar et al., 2019), a method on combining contact and inertial data for confined and degraded flying robot navigation (Lew et al., 2019), the contribution in (Lajoie et al., 2019) on robot localization, the lighter-than-air design in (Huang et al., 2019), as well as our team contributions in (Khattak et al., 2019; Dang et al., 2019b; Papachristos et al., 2019; Dang et al., 2019a; Bjelonic et al., 2019). Motivated by this challenge, the contribution proposed in this paper aims to offer a general in nature, but also optimized in design, path planning methodology for autonomous subterranean exploration. These goals are achieved through a bifurcated approach of efficient and resourceful local planning strategy which is optimized for volumetric gain, combined with a global planning layer that is responsible for ensuring autonomous homing and robot re-positioning after the robot has reached a dead-end (e.g. a mine heading). The method is applicable to diverse robotic configurations and it is demonstrated through both flying and walking systems.

3 Problem Statement

Exploration path planning in subterranean environments, as considered in this work, is an instance of the volumetric exploration problem and as such aims to build a complete map of previously unknown space autonomously. Generally, a subterranean environment is a closed, bounded and connected point set with possible exceptions being access points from above ground (e.g., mine portals or subway entrances) and consists of a large in scale complex network of branches, junctions, multiple levels and openings (e.g., rooms or caves). It may involve a collection of obstacles within it and may present terrain posing severe traversability constraints. As such, these kinds of environments (e.g., underground mines, tunnels, subway infrastructure,

and cave networks) with their distinct characteristics in length and topology introduce multiple challenges that must be accounted for in the planner.

First of all, the scale of the environment prohibits planning from taking place in a computationally efficient manner at the full configuration space of the environment bounds. Nonetheless, the particular geometric properties common among subterranean settings provide a notable feature that the planner may exploit; iteratively exploring local sub-spaces generates large exploration rewards without the penalty of exhaustively searching through the whole space. At the same time as the robot proceeds by local exploration, distinct frontiers of the explored space and critical points such as underground mine branching points and junctions appear. This insight allows us to re-cast the exploration path planning problem into two bifurcated stages, namely that of the *local exploration planner* responsible for efficient exploration within a sliding spatial window around the current robot pose, and that of the *global exploration planner* which is responsible for a) relocating the robot towards previously unexplored regions when the local planner reports its inability to progress, or b) returning to a predefined home location when the remaining endurance is insufficient to continue. Given this two-layer approach, we define the concepts of "local completion" and "global completion".

Let $\mathbb{M}$ be a 3D occupancy map of the environment which is incrementally built from measurements of an onboard depth sensor $\mathbb{S}$, as well as robot poses derived from a localization system $\mathbb{O}$. The map consists of voxels m of three categories, namely $m \in \mathbb{M}_{free}$, $m \in \mathbb{M}_{occupied}$, or $m \in \mathbb{M}_{unknown}$ representing free, occupied, and unknown space respectively, while certain (generally disconnected) subsets of the map may correspond to "no-go" zones $\mathbb{M}_{NG}$ representing possible traversability constraints or other imposed limits. Furthermore, let $d_{\max}$ be the effective range, and $[F_H, F_V]$ be the field-of-view in horizontal and vertical directions of the depth sensor $\mathbb{S}$. In addition, let the robot's configuration at time t be defined as the combination of 3D position and heading $\xi_t = [x_t, y_t, z_t, \psi_t]$. Importantly, since for most range sensors' perception stops at surfaces, sometimes hollow spaces or narrow pockets cannot be fully explored thus leading to a residual map $\mathbb{M}_{*,res} \subset \mathbb{M}_{unknown}$ with volume $V_{*,res}$ which is infeasible to explore given the robot's constraints. As a result, given a volume V_*, the potential volume to be explored is $V_{*,explored} = V_* \setminus V_{*,res}$.

Definition 1 (Local Completion) *Given a map $\mathbb{M}$, within a local sub-space $\mathbb{M}_L$ of dimensions D_L centered around the current robot configuration, the planner reports "local completion" if $V_{D_L,explored} = V_{D_L} \setminus V_{D_L,res}$.*

Definition 2 (Global Completion) *Given the full occupancy map $\mathbb{M}$ of the environment with dimensions D_G and volume V_{D_G}, the planner considers "global completion" if $V_{D_G,explored} = V_{D_G} \setminus V_{D_G,res}$.*

In practice, it is not possible and unrealistic to identify V_{res}, but completion can be approximated by the lack of a collision-free path inside a planning volume which leads to a space with potentially unknown volume larger than a threshold V_δ. The local and global planner problems are formulated as follows.

Problem 1 (Local Exploration Planner) *Given an occupancy map $\mathbb{M}$ and a local subset of it $\mathbb{M}_L$ centered around the current robot configuration ξ_0, find a collision-free and traversability-aware (when applicable) path $\sigma_L = \{\xi_i\}$ to guide the robot towards unmapped areas and maximize an exploration gain defined as the volume which is expected to be mapped when the robot traverses along the path σ_L with a sensor $\mathbb{S}$. A path is admissible if it is collision-free in the sense of not colliding with 3D obstacles in the map and not going through "no-go" zones that may encode traversability constraints. When "local completion" is reported by this planner, the global planner is to be engaged.*

Problem 2 (Global Exploration Planner) *Given the explored and unknown subsets of an occupancy map $\mathbb{M}$ of the environment and the current robot configuration ξ_0, find a collision-free path σ_G leading the robot towards the frontiers of the unmapped areas. Feasible paths of this planning problem must take into account the remaining endurance of the robot. When the environment is explored completely ("global completion") or the battery limits are approaching, find a collision-free path σ_H to return the robot to its home location ξ_{home}.*

4 Proposed Approach

Subterranean environments usually consist of long and confined corridors connected by intersections, which impose major challenges for robotic exploration. More specifically, the exploration algorithm must be scalable to large environments (e.g., km in length), while responsive enough to facilitate rapid exploration under time constraints (e.g., limited battery). Tailored to these needs, a bifurcated search strategy is proposed, namely a) a local exploration planner which focuses on local exploration surrounding the current robot's pose, and b) a global planner which performs planning globally inside the currently explored space. In particular, the local planner searches within a sliding local space of fixed dimension to enable faster computation, while simultaneously ensuring that the algorithmic complexity remains independent of the environment's scale. However, due to the particularities of underground settings in terms of scale, complexity and topology, the local planner might reach a dead-end or other scenario that prohibits the derivation of an effective exploration path. The global planner is then queried to relocate the robot towards unexplored areas in order to continue its mission. The global planner is also utilized to provide a safe and timely return-to-home path given the robot's time budget. Both the local and global planner employ and exploit sampling-based rapidly-exploring random graph algorithms to search for an optimal exploration path, a choice that reflects noticeable characteristics of underground environments (e.g., mines), which may be topologically modeled as graphs that are undirected and potentially include cycles. The global graph is built incrementally based on selective branches from local sub-graphs. Hence, it maintains a very lightweight graph which is fast to process. The overall flowchart of the exploration solution is presented in Figure 2.

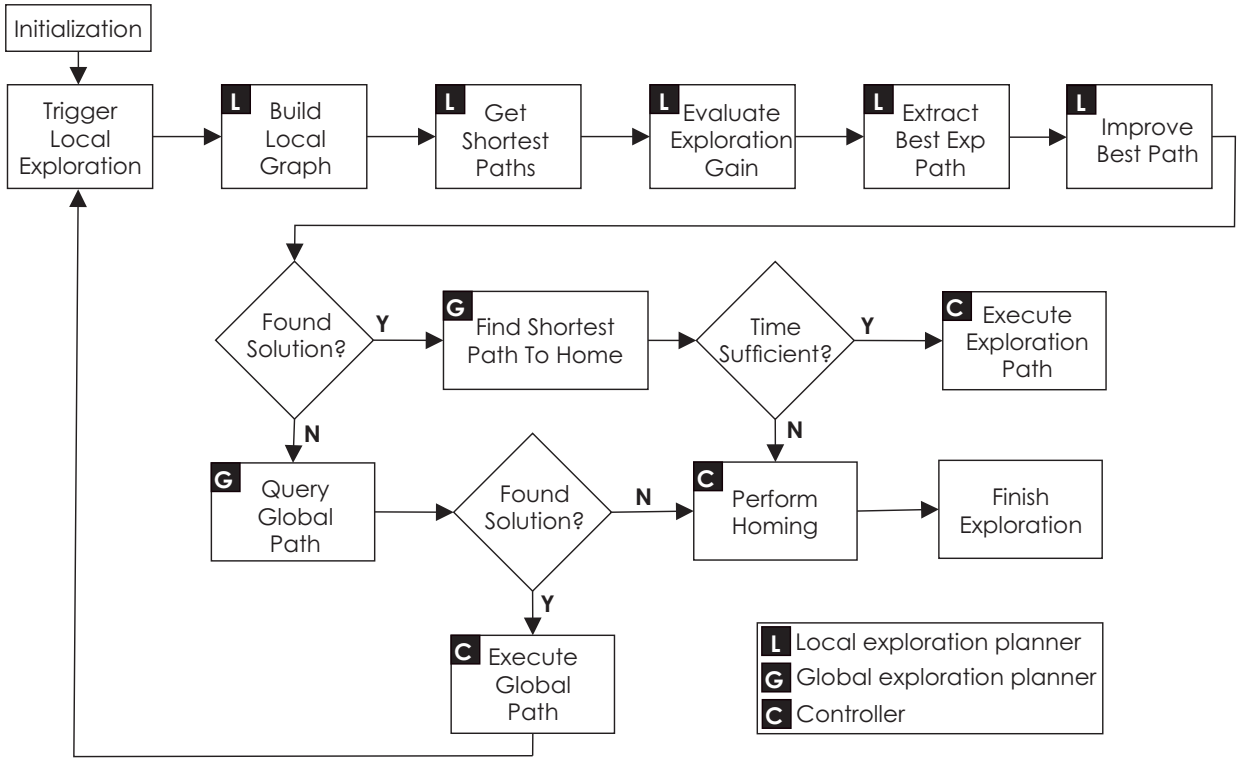

Figure 2: Architecture of the proposed graph-based exploration planner.

4.1 Local Exploration Planner

The local exploration planner, which is the front-end of the proposed exploration architecture, is designed to provide a rapid exploratory behavior tailored to the geometries of underground settings by building upon the rapidly-exploring random graph algorithm. Given the occupancy map $\mathbb{M}$, the current robot's configuration

ξ_0, a local space with dimensions D_L centered around the current robot's position, "no-go" $\mathbb{M}_{NG}$ zones capturing further traversability constraints (when applicable), and a bounding box D_R related to the robot's physical size, the planner randomly samples a collision-free configuration $\mathbb{M}_R(\xi_{rand}) \in \mathbb{M}_{free}$ inside a local volume V_{D_L}. From this feasible sample, the planner then finds its closest vertex $\xi_{nearest}$ in the graph using nearest neighbor search (Moore, 1991). If a straight path connecting the two $[\xi_{rand}, \xi_{nearest}]$ passes the collision check, the new sample and the straight path are both added to the graph as a new vertex and new edge respectively. Next, the planner attempts to connect denser collision-free edges from this newly added vertex to its neighboring vertices within a defined radius δ. This graph building step to create a local graph $\mathbb{G}_L$ is continued until it exceeds a set maximum number of vertices $N_{\mathbb{V},\max}$ or edges $N_{\mathbb{E},\max}$.

Given the built local graph $\mathbb{G}_L$, Dijkstra's algorithm (Cormen et al., 2009) is utilized to find the set of shortest paths Σ_L from the root vertex ξ_0 to all other vertices in the local graph. This step aims to improve the exploration rate by using only minimum-length paths in the graph and also helps to avoid nuisance zig-zag motions that are undesirable in narrow spaces, such as underground tunnels. Zig-zag patterns along a dominant direction have marginal or no benefits in exploration in relatively long and narrow topologies but reduce the exploration rate due to their additional length. Considering sufficient sampling density and as zig-zag motions would increase the path length, using Dijkstra tends to eliminate them. The planner then computes the **VolumetricGain** for each vertex, which is the expected cumulative unmapped volume that an onboard range sensor $\mathbb{S}$ would perceive given the robot's configuration at each vertex. The selection of this information metric is motivated by its ability to give rise to efficient behaviors at a relatively small computational cost, while information-theoretic alternatives could be considered as valid alternatives (Charrow et al., 2015; Tabib et al., 2020). The proposed volumetric gain calculation relies on ray casting, as visualized in Figure 3, and it is a flexible method which can be extended to multiple depth sensors. For instance, it is straightforward to add another term from an upward depth sensor to map the ceiling in wider spaces like cave networks. The volumetric gain, together with other weight functions related to distance and direction, are utilized to compute the **ExplorationGain** for each shortest path in Σ_L. In particular, given a path $\sigma_i \in \Sigma_L, i = 1...n$ with a set of vertices along the path $\nu_j^i \in \sigma_i, j = 1...m_i$, the $\textbf{ExplorationGain}(\sigma_i)$ is calculated as follows:

$$\textbf{ExplorationGain}(\sigma_i) = e^{-\gamma_{\mathcal{S}}\mathcal{S}(\sigma_i,\sigma_{exp})} \sum_{j=1}^{m_i} \textbf{VolumetricGain}(\nu_j^i) e^{-\gamma_{\mathcal{D}}\mathcal{D}(\nu_1^i,\nu_j^i)} \tag{1}$$

where $\mathcal{S}(\sigma_i, \sigma_{exp})$, $\mathcal{D}(\nu_1^i, \nu_j^i)$ are weight functions with tunable factors $\gamma_{\mathcal{S}}, \gamma_{\mathcal{D}} > 0$. Furthermore, $\mathcal{D}(\nu_1^i, \nu_j^i)$ is the cumulative Euclidean distance from a vertex ν_j^i to the root ν_1^i along the path σ_i. This aims to favor combinations of relatively short trajectories that are associated with high gains in order to achieve a higher exploration rate.

It is noted that when the robot approaches branching points of the environment, vertices near edges of the intercepting branch could get very high volumetric gains towards locally occluded regions, which encourages branch-switching back-and-forth paths across the junction in hope of maximizing the exploration rate. A similar situation could happen with small occluded areas which often create false expectations of high gains. Such behavior, however, is often undesirable in practice since it leads to sudden-and-unnecessary change in exploration direction of the robot. To penalize this behavior in certain cases, a similarity function $\mathcal{S}(\sigma_i, \sigma_{exp})$ is introduced to disfavor paths that greatly diverge from the currently estimated exploration direction. The similarity metric, in this case, is developed using the Dynamic Time Warping (DTW) method (Bachrach, 2013), which calculates the cumulative Euclidean distance between the planned path σ_i and a pseudo straight exploration path σ_{exp} with the same length. The direction of σ_{exp} is averaged over a temporal window of the robot's pose. A path that maximizes the **ExplorationGain** is then selected and refined before being conducted by the robot. After each planner run, the whole procedure is iteratively repeated. The whole process is visualized in Figure 4 and 5, while its pseudo-code is provided in Algorithm 1 and Algorithm 2. Note that formal inference of local completion requires explicit knowledge of the residual space $V_{D_L,res}$. In practice, however, this condition is detected when no path with exploration gain above a small threshold $g_\epsilon > 0$ is discovered.

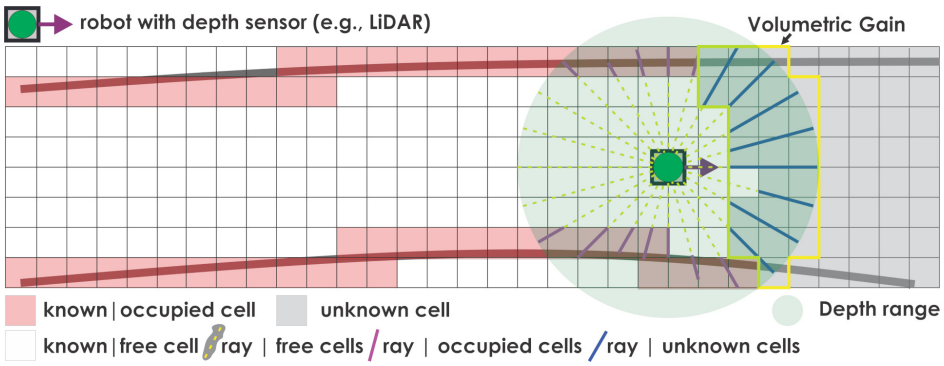

Figure 3: The volumetric gain calculation relies on ray casting. From the current robot pose and given a certain 3D sensor model (here a 2D representation is presented for simplicity), the algorithm identifies how many voxels in the unknown space can be traversed by any ray casted within the sensor frustum model.

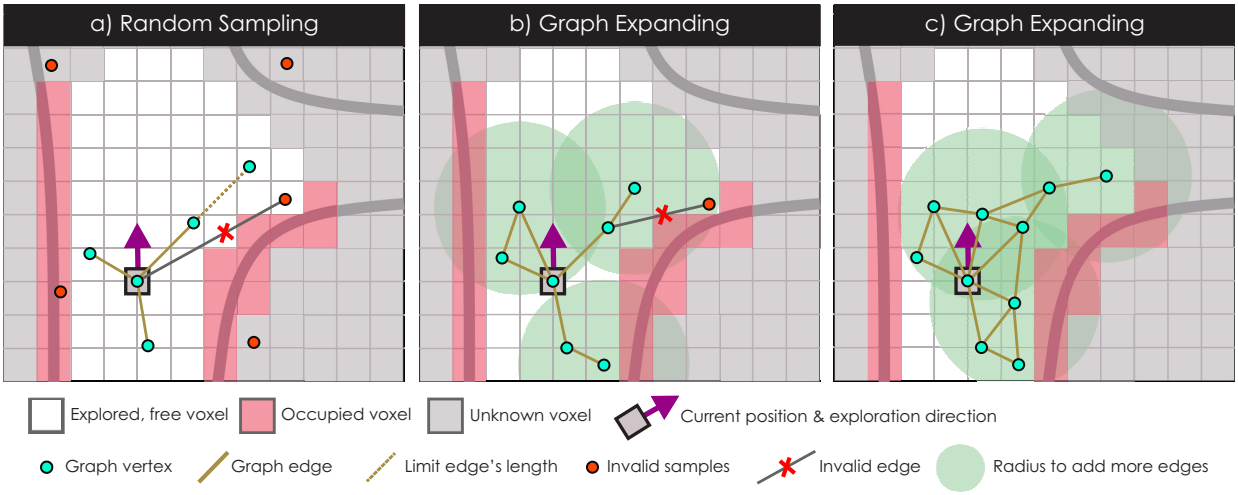

Figure 4: Illustration of random sampling and local graph building. Sub-figure a) presents some initial collision-free vertices and edges being added to the graph. Sub-figures b-c) introduce further sampling iterations to add more vertices and densify the graph with more edges within a radius surrounding each newly added vertex.

Algorithm 1 Local Planner

$\xi_0 \leftarrow$ **GetCurrentConfiguration**()
$\mathbb{G}_L \leftarrow$ **BuildLocalGraph**(ξ_0)
$\Sigma_L \leftarrow$ **GetDijkstraShortestPaths**($\mathbb{G}_\mathbf{L}, \xi_\mathbf{0}$) ▷ From current vertex to others
ComputeVolumetricGain($\mathbb{G}_\mathbf{L}$) ▷ For all vertices
$g_{best} \leftarrow 0$
$\sigma_{L,best} \leftarrow \emptyset$
for all $\sigma \in \Sigma_L$ **do**
    $g_\sigma \leftarrow$ **ExplorationGain**(σ) ▷ Compute exploration gain for each path
    if $g_\sigma > g_{best}$ **then**
        $g_{best} \leftarrow g_\sigma$; $\sigma_{L,best} \leftarrow \sigma$ ▷ Keep the best path
    end if
end for
$\sigma_{L,best} \leftarrow$ **ImprovePath**($\sigma_{L,best}$)
return $\sigma_{L,best}$

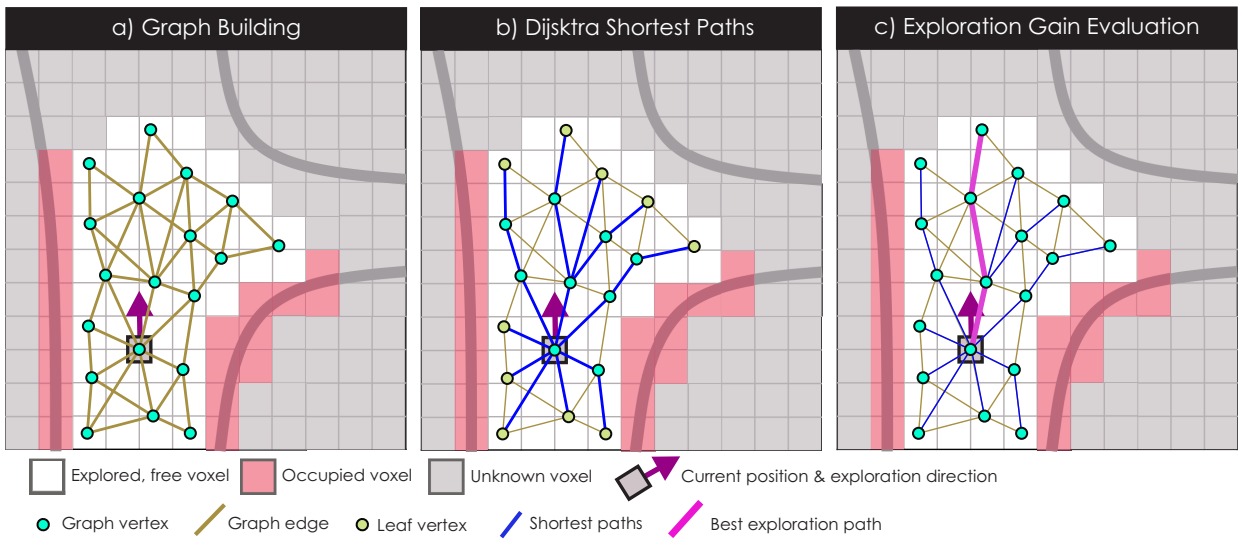

Figure 5: Visualization of an instance of the proposed local exploration planning layer. In a) the first step of the local planner which is building a graph locally based on a random-sampling configuration search method. Subsequently, in b), Dijkstra's algorithm is applied to the graph in order to search for shortest paths from the root to all sampled vertices. In c), the best exploration path is identified utilizing the exploration gain computed for each shortest path.

Algorithm 2 Build a Local Graph

function BUILDLOCALGRAPH(ξ_0)
  $\mathbb{V} \leftarrow \{\xi_0\}$; $\mathbb{E} \leftarrow \emptyset$
  $\mathbb{G}_L = (\mathbb{V}, \mathbb{E})$
  $D_L \leftarrow$ **SetLocalBound**(ξ_0) ▷ 3D bounded space
  while $N_{\mathbb{V}} \leq N_{\mathbb{V},\max}$ **and** $N_{\mathbb{E}} \leq N_{\mathbb{E},\max}$ **do**
    $\xi_{rand} \leftarrow$ **SampleFree**(D_L) ▷ Sample a random vertex inside the 3D bounded space
    $\xi_{nearest} \leftarrow$ **NearestVertex**($\mathbb{G}_L, \xi_{rand}$)
    if CollisionFree($\xi_{rand}, \xi_{nearest}$) **then**
      $\mathbb{V} \leftarrow \mathbb{V} \cup \{\xi_{rand}\}$; $\mathbb{E} \leftarrow \mathbb{E} \cup \{\xi_{rand}, \xi_{nearest}\}$ ▷ Expand graph with the newly sampled vertex
      $\Xi_{near} \leftarrow$ **NearestVertices**($\mathbb{G}_L, \xi_{rand}, \delta$)
      for all $\xi_{near} \in \Xi_{near}$ **do**
        if CollisionFree(ξ_{rand}, ξ_{near}) **then**
          $\mathbb{E} \leftarrow \mathbb{E} \cup \{\xi_{rand}, \xi_{near}\}$ ▷ Add extra collision-free edges to densify the graph
        end if
      end for
    end if
  end while
  return $\mathbb{G}_L = (\mathbb{V}, \mathbb{E})$
end function

4.2 Global Exploration Planner

The global planner performs a search on the currently-explored global space of the environment and provides two key functionalities that ensure the scalability of exploration autonomy in large underground settings. First, it searches for an alternate path towards unexplored areas when the local planner reports an inability to propose any effective exploration path either due to local completion or presence of a passage that is too narrow to proceed. Second, it is responsible for the critical homing procedure once the remaining

time approaches its allowable budget (e.g., battery lifetime, or user-limited time budget). To tackle these challenges given the large scale of subterranean environments involving multiple intersections and cyclic routes, we propose a method to incrementally build and continually maintain a lightweight undirected graph from the iteratively-derived local exploration graphs.

Specifically, the global planner conducts two steps to augment and expand its graph from every sampled local graph. In the first step, the global graph adds the best path $\sigma_{L,best}$ by introducing new vertices and edges along such paths. It also densifies the graph by adding extra collision-free edges from each newly added vertex to their existing neighbors. This aims to ensure that the global planner can always find a feasible and short path instead of relying purely on a naive backtracking approach. Subsequently, the global planner attempts to add additional paths from the list of shortest paths of the local graph which contain vertices with high volumetric gain. By making use of those unvisited but high-gain vertices, the global graph can maintain a short but selective list of potential vertices that implicitly encode the locations of "frontiers" and directions toward unexplored branches of the environment.

To further reduce the number of potential paths from the local graph, the DTW similarity metric is used to cluster all the high gain paths and only the "principal" paths (the longest ones) from each cluster are added to the global graph. Leaf vertices from these principal paths are marked as potential "*frontiers*". Periodically, the volumetric gain of each frontier vertex is re-evaluated to maintain a shorter list of preferred frontiers. Overall, this process enables the critical global frontier re-positioning functionality that facilitates efficient exploration even inside large-scale and multi-branching topologies. It is triggered occasionally to re-position the robot when the local planner cannot derive any informative path. To perform this task, the global planner runs Dijkstra's algorithm two times in order to a) find the shortest paths from the current location towards all potential frontiers, and b) find the shortest paths from all frontiers towards the home location. Similarly, the global planner continuously identifies a return-to-home path based on running Dijkstra's algorithm on the global graph. It commands this path when the the time to return to home approaches the remaining endurance. Subsequently, we provide more detailed analysis with respect to the process of identifying the best path for future exploration.

Let $\nu_{G,cur}$ be a vertex in the global graph representing the current robot configuration, and $\mathcal{F} = \{\nu^{\mathcal{F}}_{G,i}\}$ be a set of updated potential frontiers in the global graph. The problem of selecting a path to maximize the anticipated exploration gain and re-position the robot towards frontiers of the unexplored space in this case is particularly complex. This is mainly due to the fact that this decision takes place over an increasingly larger search space (as the robot proceeds in its exploration). Furthermore, potential unexplored branches are identified based on incomplete volumetric information gathered from the limited field-of-view of the sensor. For instance, when the robot is inside an underground mine and has passed several intersections then multiple unvisited branches from those intersections could become "frontiers", and predicting which candidate will be proven best is difficult. Thus, the global planner proposed in this work is developed based on two basic principles. First, it takes into account the time budget when considering any feasible path, which in turn allows the robot to be able to visit a frontier initially and then at least have sufficient endurance to carry out the homing procedure (worst-case scenario). Secondly, the planner should favor high gain areas requiring a short time to arrive. The exploration gain for the global planer is presented in Equation 2.

$$\mathbf{GlobalExplorationGain}_G(\nu^{\mathcal{F}}_{G,i}) = \mathcal{T}(\nu_{G,cur}, \nu^{\mathcal{F}}_{G,i})\mathbf{VolumetricGain}(\nu^{\mathcal{F}}_{G,i})e^{-\epsilon_{\mathcal{D}}\mathcal{D}(\nu_{G,cur},\nu^{\mathcal{F}}_{G,i})} \tag{2}$$

where $\mathcal{T}(\nu_{G,cur}, \nu^{\mathcal{F}}_{G,i})$ is the estimated remaining exploration time if the planner is to choose the frontier $\nu^{\mathcal{F}}_{G,i}$. This parameter is approximately calculated using the Remaining Endurance Time (**RET**) of the robot and then subtracting the Estimated Time of Arrival (**ETA**) to traverse from the current vertex to the designated vertex and from there to the home location (Equation 3). The above takes the form

$$\mathcal{T}(\nu_{G,cur}, \nu^{\mathcal{F}}_{G,i}) = \mathbf{RET} - \mathbf{ETA}(\nu_{G,cur}, \nu^{\mathcal{F}}_{G,i}) - \mathbf{ETA}(\nu^{\mathcal{F}}_{G,i}, \nu_{G,home}) \tag{3}$$

where $\mathcal{D}(\nu_{G,cur}, \nu^{\mathcal{F}}_{G,i})$ is the shortest path length from the current location to the frontier, while the tunable

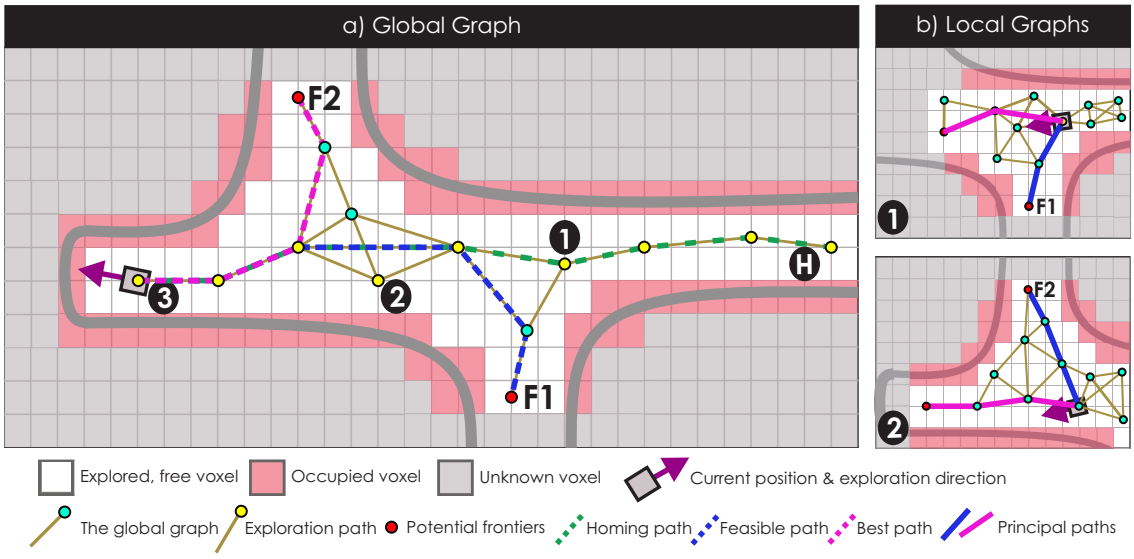

Figure 6: A visualized instance of the proposed global exploration planning layer. Sub-figure b) shows two instances of local graphs at position (**1**) and (**2**) and their corresponding principal paths to be added to the global graph. The global planner first adds these paths directly to its graph. The leaf vertices of these paths are marked as potential frontiers and more collision-free vertices are added around them to densify the global graph. Sub-figure a) depicts a case in which the robot keeps exploring using the local planner until reaching a dead-end at the location (**3**). The global planner is then triggered, and three possible paths are presented in the figure as red, green, and blue dashed lines, respectively. Depending on the remaining endurance and the volumetric gain of each path, the global planner selects the best path using the exploration gain in Equation 2. The homing–green path is for the worst-case scenario when little battery life remains, while the red path towards frontier **F2** is the best case, and the blue path towards frontier **F1** is in between.

parameter $\epsilon_{\mathcal{D}}$ is used to penalize long paths. Conceptually, this can be considered as the "tentative volumetric gain" that the robot could acquire for the time duration $\mathcal{T}()$ if choosing this frontier to explore. The algorithm is further visualized in the Figure 6, while its pseudo code is provided in Algorithm 3 and Algorithm 4.

Algorithm 3 Global Planner

$\nu_0 \leftarrow$ **GetCurrentVertex**()
$\Sigma_{G0} \leftarrow$ **GetDijkstraShortestPaths**($\mathbb{G}_{\mathbf{G}}, \nu_{\mathbf{0}}$) ▷ Shortest paths from current vertex to others
$\Sigma_{GH} \leftarrow$ **GetDijkstraShortestPaths**($\mathbb{G}_{\mathbf{G}}, \nu_{\mathbf{home}}$) ▷ Shortest paths from home vertex to others
$\Sigma_{Frontier} \leftarrow$ **GetPotentialFrontier**($\mathcal{F}$) ▷ List of potential frontiers maintained in the global graph
$g_{best} \leftarrow 0$
$\sigma_{G,best} \leftarrow \emptyset$
for all $\sigma \in \Sigma_{Frontier}$ **do**
    $g_\sigma \leftarrow$ **GlobalExplorationGain**$_G(\sigma)$ ▷ Compute gain for each path leading to a potential frontier
    if $g_\sigma > g_{best}$ **then**
        $g_{best} \leftarrow g_\sigma$; $\sigma_{G,best} \leftarrow \sigma$ ▷ Keep the best one
    end if
end for
$\sigma_{G,best} \leftarrow$ **ImprovePath**($\sigma_{G,best}$)
return $\sigma_{G,best}$

Algorithm 4 Build a Global Graph by Adding Paths from Local Graph

function BUILDGLOBALGRAPH($\mathbb{G}_G, \mathbb{G}_L, \xi_0$)
  $\sigma_{L,best} \leftarrow$ **GetBestPath**($\mathbb{G}_L$) ▷ Calculated in Algorithm 1
  AddPathToGraph($\mathbb{G}_G, \sigma_{L,best}$)
  $\Sigma_L \leftarrow$ **GetDijkstraShortestPaths**($\mathbb{G}_\mathbf{L}, \xi_\mathbf{0}$) ▷ Calculated in Algorithm 1
  $\Sigma_{principal} \leftarrow$ **DoClustering**(Σ_L) ▷ Do path clustering and get principal paths from the clusters
  for all $\sigma \in \Sigma_{principal}$ **do**
    AddPathToGraph($\mathbb{G}_G, \sigma$) ▷ Add each principal path to the global graph
  end for
  return $\mathbb{G}_G$
end function

4.3 Traversability-aware Planning

The proposed exploration pipeline is versatile and can be used for both aerial and legged platforms. However, for the legged robot, the sampling space is constrained to the xy plane relative to the current height of the robot to reflect the fact that it traverses locally on 2.5D and that an appropriate locomotion controller is responsible for the full-body system control. Furthermore, due to inevitable uncertainties from the mapping framework, the occupancy map (Hornung et al., 2013; Oleynikova et al., 2017) is used primarily for collision checking of the robot's body, while an elevation map (Fankhauser et al., 2018; Fankhauser et al., 2014) is employed to create a 2.5D representation of the terrain close to the robot, that then can be used to perform traversability analysis on a much finer resolution when it comes to its legs.

First of all, depth information from a front-facing depth camera is converted into an elevation map, which is a probabilistic terrain estimation as a grid-based map including upper and lower confidence bound. The obtained map is limited around the robot and reflects the pose uncertainty that is aggregated through the motion of the robot (robot-centric mapping). This map is then used to perform traversability analysis. To perform this task, the work in (Wermelinger et al., 2016) is employed. The method utilizes filters to extract terrain characteristics such as slope, roughness, and steps. These features are then combined through weighted summation to derive a traversability metric, while both a total and individual thresholds for each feature are applied allowing to classify the paths as traversable or untraversable for the legged robot and its footprint. Naturally, this requires an augmentation in the algorithmic design of the proposed planner as opposed to merely performing collision-checking. In principle, the traversability analysis can address which paths are admissible by the robot but the derived traversability map can only cover areas in much closer proximity as compared to that of the occupancy map. This is because the traversability map requires dense measurements (a 2cm grid size is used for the presented experiments) to ensure reliable checks for all four legs of the robot. However, planning only within short distances around the robot is bound to lead to relatively inefficient exploration behaviors. To overcome this issue but still honor traversability constraints, an engineering technique is employed according to which the areas outside the elevation map are presumably traversable thus allowing to plan longer high-gain paths. An additional traversability test is recursively invoked to verify in an online manner every waypoint along the trajectory, based on the direction and magnitude of the robot's velocity. If the robot approaches an untraversable region, the planner is re-triggered to provide an alternate path. For representation and implementation efficiency, the proposed planner interfaces the traversability analysis and represents non-traversable regions as "no-go" zones.

4.4 Path Improvement

Given the best exploration path from the planner σ_{best}, local or global, a refinement step is considered to adjust the path to further improve the safety of the robot. Primarily this step aims to modify all vertices along the path to be farther away from obstacles. Such a modification is allowed under an assumption that the associated change in the exploration gain is usually negligible. This assumption has been verified both in simulation and in practice and strongly holds especially due to the tunnel-like topologies of underground

environments. The observed difference in gain is not larger than inevitable errors introduced due to noise in the volumetric gain calculation process. The refinement process is detailed below and visualized in Figure 7.

First, the path is iteratively pruned to remove short edges (e.g., < 0.5m) created due to the random sampling process. Each vertex is then adjusted within its feasible polygons using a method motivated from previous work (Liu et al., 2017; Deits and Tedrake, 2015). More specifically, for each edge $[\nu_i, \nu_{i+1}] \in \sigma_{best}$, a set of hyperplanes is estimated by gradually inflating an ellipsoid centering along that edge; vertex ν_{i+1} is then moved to the center of a polygon formed by computed hyperplanes and a plane perpendicular to the edge at that vertex. The modification is only permitted if the adjusted path is still collision-free. This step is repeatedly performed for all vertices of the path. As a result, the modified path is similar but safer than the given path because all its vertices are pushed away from their surrounding obstacles.

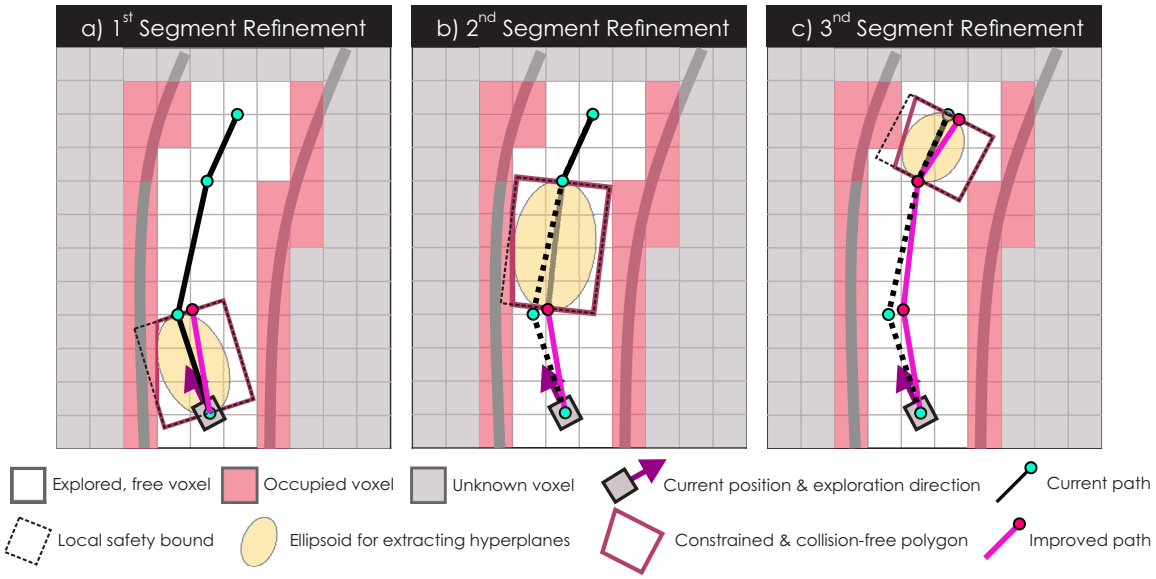

Figure 7: An approach to improve the robot's safety by adjusting the path farther from its intermediate obstacles. There are three iterations needed to adjust all three segments represented in sub-figures a-c.

4.5 Complexity Analysis

To allow a further understanding of the computational complexity of the proposed approach, a detailed analysis of each method of the planner is presented. Overall, most of the complexity terms of GBPlanner can be attributed to two operations, namely a) the process of collision checking while building the graph, and b) voxel status query during volumetric gain calculations. Both these processes require an adequate mapping framework whose selection is thus critical. Therefore, during the development phase of the planner, two state-of-the-art mapping frameworks were actively utilized and evaluated, namely Octomap (Hornung et al., 2013) and Voxblox (Oleynikova et al., 2017). Octomap is widely adopted by the robotics community, while Voxblox provides particularly fast query of voxel status due to its underlying hashing data structure (Oleynikova et al., 2017). This in turn is critical to reduce the overall computational load. GBPlanner is developed to interface with either Octomap or Voxblox depending on user choices.

More specifically, from an algorithmic standpoint, Octomap makes use of an octree data structure for memory efficiency which in turn leads to $\mathcal{O}(\log(n))$ cost to check the status of a voxel, where n is the number of nodes in the octree. Hence, given the volume V_{D_G} of the map with voxel resolution r, the status checking for each voxel will take $\mathcal{O}_{\mathbb{M},\text{Octomap}} = \mathcal{O}(\log(V_{D_G}/r^3))$ which is increasingly expensive as the map gets larger. On the other hand, Voxblox incrementally and dynamically builds Euclidean Signed Distance Fields (ESDFs) out of Truncated Signed Distance Fields (TSDFs) with voxel hashing, which costs constant time $\mathcal{O}_{\mathbb{M},\text{Voxblox}} = \mathcal{O}(1)$

Table 1: Complexity Analysis of the Local Planner

Functions	**Complexity**
SampleFree	$\mathcal{O}(V_{D_L}/V_{D_L,free} \times V_{D_R}/r^3 \times \mathcal{O}_\mathbb{M})$
CollisionFree	$\mathcal{O}(V_{D_R}/r^3 \times d_{avg}/r \times \mathcal{O}_\mathbb{M})$
AddConnections	$\mathcal{O}(N_{near}V_{D_R}/r^3 \times d_{avg}/r \times \mathcal{O}_\mathbb{M})$
BuildLocalGraph $\mathcal{O}(N_\mathbb{V}V_{D_L}/V_{D_L,free} \times V_{D_R}/r^3 \times \mathcal{O}_\mathbb{M} + N_\mathbb{V}N_{near}V_{D_R}/r^3 \times d_{avg}/r \times \mathcal{O}_\mathbb{M})$	
Dijkstra	$\mathcal{O}(N_\mathbb{V}\log(N_\mathbb{V}))$
VolumetricGain	$\mathcal{O}(N_\mathbb{V}F_HF_Vd_{max}/(r_Hr_Vr) \times \mathcal{O}_\mathbb{M})$
ExtractBestPath	$\mathcal{O}(N_\mathbb{V})$
ImprovePath	$\mathcal{O}(N_{\mathbb{E},\sigma_{best}} \times V_{D_S,\text{avg}}/r^3 \times \mathcal{O}_\mathbb{M})$

to lookup a voxel in the map. Within the experiments presented in Section 6 the planner employed either Octomap or Voxblox. For the rest of this analysis let us simplify and denote $\mathcal{O}_\mathbb{M}$ as the complexity of voxel status checking depending on the underlying mapping framework.

Regarding the local planner, the function **BuildLocalGraph** is examined first. Let D_L denote the dimensions of the local planning volume. Using a uniform sampling kernel the probability to get a feasible sample is $\rho = V_{D_L,free}/V_{D_L}$, where $V_{D_L,free}$ is the volume of free space inside the total volume of the local space V_{D_L}. Moreover, collision checking for the whole robot's body, modeled as a cuboid with volume V_{D_R}, costs V_{D_R}/r^3. Thus, the overall complexity of the function **SampleFree** is $\mathcal{O}(V_{D_L}/V_{D_L,free} \times V_{D_R}/r^3 \times \mathcal{O}_\mathbb{M})$. Similarly, the complexity terms for the functions **CollisionFree** and **AddConnections** are listed in Table 1, where d_{avg} is the average length of edges in the graph, and N_{near} is the average number of neighboring vertices in the graph requiring collision checks to form extra edges. In addition, we utilize the kd-tree (Moore, 1991) data structure with $k = 3$ in order to perform 3D position-based neighbor search. Therefore, the complexity of the function **NearestVertex** is $\mathcal{O}(\log(N_\mathbb{V}))$, and the complexity for function **NearestVertices** is bounded as $\mathcal{O}(N_\mathbb{V}^{1-1/k} + N_{near})$ using range search inside the kd-tree. Furthermore, as the local random graph in this work is typically rather sparse, the adjacency list is utilized as its underlying data structure which in turn costs $\mathcal{O}(1)$ and $\mathcal{O}(\log(N_\mathbb{E}/N_\mathbb{V}))$ for adding a vertex and an edge respectively. The computational cost for the volumetric gain calculation for each vertex is also depicted in Table 1, where $r_H \times r_V$ is the resolution of the range sensor. Lastly, for the **ImprovePath** function let $N_{\mathbb{V},\sigma_{best}}$ and $N_{\mathbb{E},\sigma_{best}}$ denote the number of vertices and edges along the best exploration path respectively. Let $V_{D_S,\text{avg}}$ be the average safety local volume enclosing an edge, then the cost to extract all obstacles inside this volume is $\mathcal{O}(V_{D_S,\text{avg}}/r^3 \times \mathcal{O}_\mathbb{M})$. Converting the local occupied voxels into hyperplanes takes $\mathcal{O}(N_{\mathbb{E},\sigma_{best}} \times n_{max})$ (n_{max} being the maximum number of iterations) which is negligible (in practice $n_{max} < 50$). Then, the complexity term of **ImprovePath** takes the form $\mathcal{O}(N_{\mathbb{E},\sigma_{best}} \times V_{D_S,\text{avg}}/r^3 \times \mathcal{O}_\mathbb{M})$.

For the global planner, since it makes use of results from the already-built local graph and utilizes analogous functions, the complexity overall is similar to that of Algorithm 2 and is shown in Table 2. The function **AddPathToGraph** will add collision-free paths, extracted from the local graph, directly to the global graph, then performs collision checking at every newly added vertex to form extra edges and thus make the global graph denser. Its complexity is shown in Table 2, where N_{new} is the total number of newly added vertices. Furthermore, the method **DoClustering** scans through all the branches utilizing the DTW metric to group them into clusters which costs only $\mathcal{O}(N_{branch})$, where N_{branch} is the number of Dijkstra shortest paths. Longest paths in each cluster are considered as principal paths and then added to the global graph.

5 Simulation Studies

A set of simulation studies were conducted to not only allow to tune and verify the performance of the proposed path planning approach prior to real-life field deployment but also to enable comparison against

Table 2: Complexity Analysis of the Global Planner

Functions	**Complexity**
AddToGraph	$\mathcal{O}(N_{new} \times N_{near} \times V_{D_R}/r^3 \times d_{avg}/r \times \mathcal{O}_{\mathbf{M}})$
Dijkstra	$\mathcal{O}(N_{\mathbb{V}} \log(N_{\mathbb{V}}))$
DoClustering	$\mathcal{O}(N_{branch})$
BuildGlobalGraph $\mathcal{O}(N_{new} \times N_{near} \times V_{D_R}/r^3 \times d_{avg}/r \times \mathcal{O}_{\mathbf{M}}) + \mathcal{O}(N_{\mathbb{V}} \log(N_{\mathbb{V}})) + \mathcal{O}(N_{branch})$	
VolumetricGain	$\mathcal{O}(N_{Frontier} F_H F_V d_{max}/(r_H r_V r) \times \mathcal{O}_{\mathbb{M}})$
ExtractBestPath	$\mathcal{O}(N_{\mathbb{V}})$
ImprovePath	$\mathcal{O}(N_{\mathbb{E},\sigma_{best}} \times V_{D_S,\text{avg}}/r^3 \times \mathcal{O}_{\mathbb{M}})$

other state-of-the-art methods in exploration path planning. More specifically, two simulated environment configurations were utilized - the first representing an underground room-and-pillar mine and the second an underground long-wall mine. The room-and-pillar mine environment is multiple km-long and presents an array of multi-way intersections that challenge the planner's behavior. The latter involves multiple long and narrow corridors, alongside intersections and cycles. In all simulated tests the performance of GBPlanner is presented, alongside comparison against the receding horizon Next-Best-View Planner (NBVP) (Bircher et al., 2016b) and the Frontiers exploration algorithm (FrontierPlanner) (Yamauchi, 1997) implemented for 3D environments and combined with an optimal sampling-based motion planner for collision-free navigation to the frontiers (S. Karaman and E. Frazzoli, 2011; Karaman and Frazzoli, 2009).

The room-and-pillar simulated environment consists of two sections, left and right, with the first being relatively more spacious regarding the width of its corridors and the second rather more constrained. Hence, for a fair comparison between three planners, two exploration scenarios are employed corresponding to the left and the right sections of the mine. A quadrotor model, similar to the real robotic system described in section 6.1.1, is developed in ROS-Gazebo utilizing the RotorS simulator (Furrer et al., 2016). Each planner is run 15 times independently starting from the same location and executed for 15 minutes per simulated mission. In addition, the average flying speed for exploration in this case is set to be $2m/s$. Relevant simulation results presenting the performance of a) GBPlanner, b) NBVP, and c) FrontierPlanner against the left and right subsets of this environment are shown in Figures 8 and 9 respectively. Furthermore, statistical comparison data with respect to the exploration rate of each method are depicted in Figure 10. As it can be seen, GBPlanner outperforms the other approaches. More specifically, NBVP manages to provide reasonable solutions but is challenged by the narrow geometry of the environment, while the Frontier-based planner fails to provide comparable results. The enhanced performance of GBPlanner is largely attributed to the value of the bifurcated local-global planning architecture with the local mode allowing dense sampling around the robot and thus efficient exploration despite the imposed geometric constraints, alongside with efficient re-positioning to the frontiers calculated at the global scale. On the contrary, NBVP for example performs planning in the whole explored space which enforces a rather more sparse configuration due to the fact that otherwise the computational cost will increase rapidly as detailed in (Bircher et al., 2016b). It is noted that to make this comparison fair, both GBPlanner and NBVP are tuned to present similar computational cost per iteration which in turn necessitated similar amount of vertices to be sampled for each planner iteration. Voxblox is used as the volumetric mapping framework in these results. Furthermore, it is also mentioned that FrontierPlanner provides the reference exploration point and a collision-free path is then derived using the motion planning algorithm in (S. Karaman and E. Frazzoli, 2011).

During the second simulation study, the model of the ANYmal legged platform was tested inside a subset of a long-wall underground mine environment involving long corridors, branches, and cycles. The GBPlanner operated in this environment by mostly engaging the local planning stage guiding the robot through the long and narrow parts of the environment. Furthermore, it occasionally queried the global planner after reaching a dead-end to reposition the robot to another unexplored area quickly. A comparative study against NBVP and the FrontierPlanner is also conducted. In particular, each planning algorithm is simulated 15 times starting from the same location and provided a time budget of 1h, while the walking speed of ANYmal is

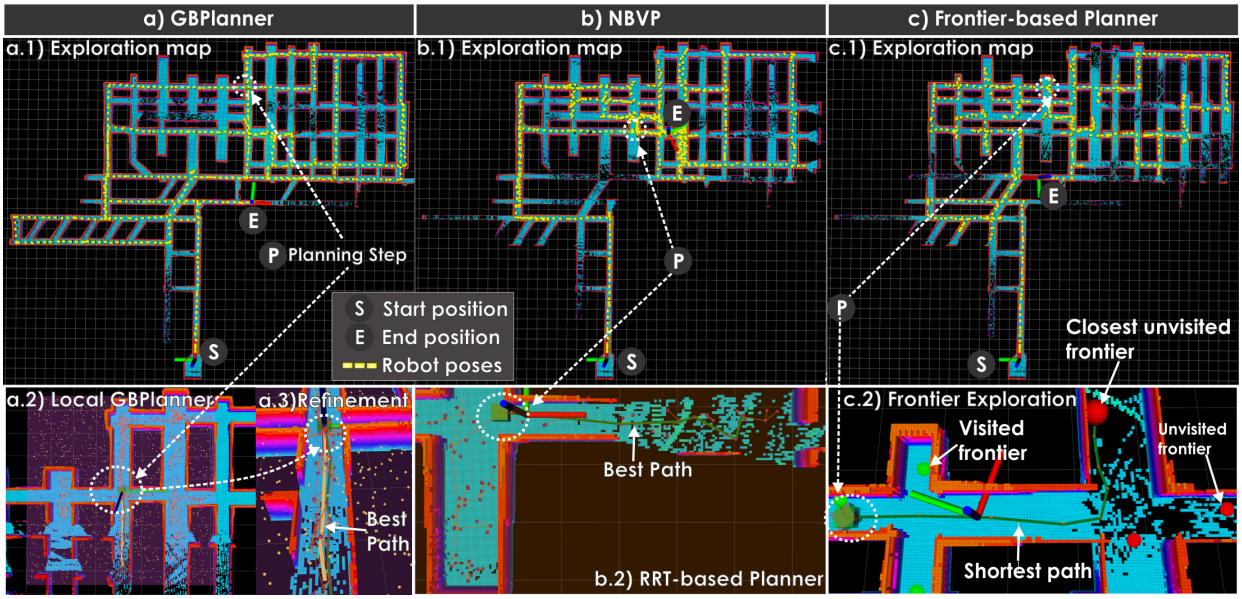

Figure 8: Indicative results for the autonomous exploration in the left section of the simulated room-and-pillar mine. Overall, all three planners achieve good exploration performance in this environment largely thanks to its spacious corridors, as well as the existence of multiple nearby solutions at any location inside the environment. As shown in sub-figure a), the mission with GBPlanner was almost complete, while solutions from NBVP in sub-figure b) tend to stay in spacious corridors since its samples are spanned over the whole map. Furthermore, paths derived from the NBVP often present zig-zag shape which is undesirable in practice. The FrontierPlanner achieves adequate performance in this case but with a lower rate as compared to GBPlanner and NBVP. This is primarily because FrontierPlanner lacks a set of specific optimizations that GBPlanner performs, including the volumetric gain formulation and the directionality bias.

set to $0.3m/s$. An indicative set of exploration results is shown in Figure 11, associated with the statistical analysis in Figure 12. As shown in Figure 12, GBPlanner continues to surpass the others primarily due to its bifurcated approach. In addition, while the FrontierPlanner provides encouraging results due to the geometric simplicity of this environment (spacious, essentially 2D) , NBVP mostly fails to obtain a comparable result.

These simulation studies and the presented comparative analysis of GBPlanner, which is open-sourced (ARL, 2020), against methods of the current state-of-the-art served to verify the overall performance of the method, understand and evaluate key design choices, such as the role of the global and local planning modes, and allow to tune the method prior to real-life use. The next section presents an extensive set of field experiments in diverse subterranean settings that allow to further evaluate the planner performance and its applicability to both flying and legged robotic systems. The presented field experiments further include a set of challenges not present in the simulation studies, including true 3D worlds (with changes in inclination/declination), traversability constraints, tight endurance limitations, necessity to perform auto-homing, importance of having safety-improved paths, noisy sensor readings and more. Neither NBVP nor the FrontierPlanner present specific functionalities relevant to these challenges - at least in their current implementations.

6 Experimental Evaluation

The proposed approach on subterranean exploration path planning was rigorously evaluated through a set of field experiments conducted in different and diverse underground environments across different geographical regions and based on multiple flying and walking robotic platforms. In terms of environments this included two long-wall underground mines in the U.S. (located in Nevada and Colorado), one abandoned long-wall

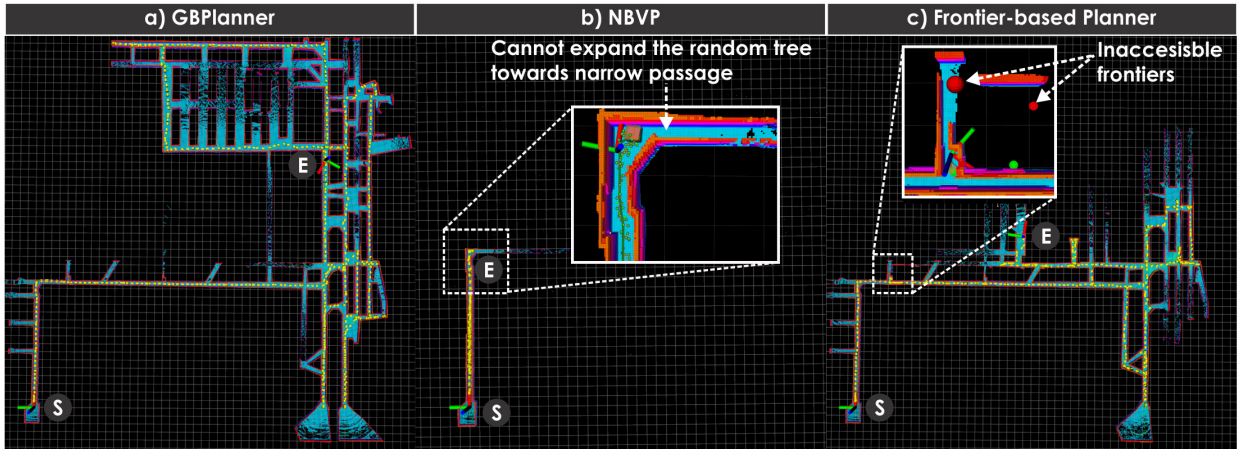

Figure 9: Indicative results for the autonomous exploration in the right section of the simulated room-and-pillar mine. GBPlanner performs best in this scenario, while the FrontierPlanner shows slower exploration and the NBVP gets trapped inside the first part of the mine. Due to the inefficient sampling approach over the full boundaries of the environment, a random tree from NBVP is generally biased towards spacious areas, which leads to its inability to find feasible paths through narrow passages. As shown in b), the robot commanded by NBVP keeps flying back and forth inside the first corridor and cannot proceed further due to a very tight passage after the first turn ($1.8m$ in width). GBPlanner does not suffer from this issue thanks to its local sampling policy. For FrontierPlanner, it depends on the relative position of the selected frontier to the current robot position. In general, it could take significant effort to derive a motion planning solution for the FrontierPlanner if the frontier is far away from the robot position. Furthermore, as shown in c), inaccessible frontiers from narrow areas could degrade the performance of the FrontierPlanner quickly since the planner needs to spend more planning iterations to identify such cases and then re-plan to another frontier. It is noted that with respect to the limitation presented by NBVP, a major increase in the sampling density could possibly resolve it but at a very significant computational cost.

underground mine in Switzerland, two room-and-pillar underground mines in the U.S. (both in Pittsburgh), and one underground bunker in Switzerland. Each experiment enabled to evaluate different aspects of the path planning functionality as summarized in Table 3. The TRJV mine in Northern Nevada is an active gold mine presenting a modern and very large-scale environment with long corridors and multiple branches. On the other hand, the Gonzen mine in Switzerland is an old abandoned iron mine consisting of narrow drifts ($<$ 3m) with rail-tracks on the ground. Moreover, the Wampum underground facility in Pittsburgh represents an interesting underground mine structure called "room-and-pillar" with wide corridors and intersections ($>$ 5m) distributed as a chessboard pattern. Apart from those test sites, the proposed planner was also utilized during two DARPA Subterranean Challenge events that took place in Colorado and Pittsburgh respectively. The Edgar Mine in Colorado incorporates passages with different sizes varying from 1.4m to a few meters. The NIOSH mine in Pittsburgh is also an experimental site comprising of long drifts at its portals leading to room-and-pillar structure further down inside the mine. The Edgar mine was utilized during the STIX event for team integration testing purposes, while the NIOSH mine was used for the first competition of the SubT Challenge. Finally, the Menznau underground bunker represents a more structured underground facility that was used in preparation for the Urban Circuit of the DARPA Subterranean Challenge.

6.1 Aerial and Legged Robots used in this Study

With respect to robotic platforms, two aerial robots ("Alpha" and "Charlie") (Dang et al., 2019b; Khattak et al., 2020b) were utilized, alongside the ANYmal quadrupedal legged platform (Hutter et al., 2016).

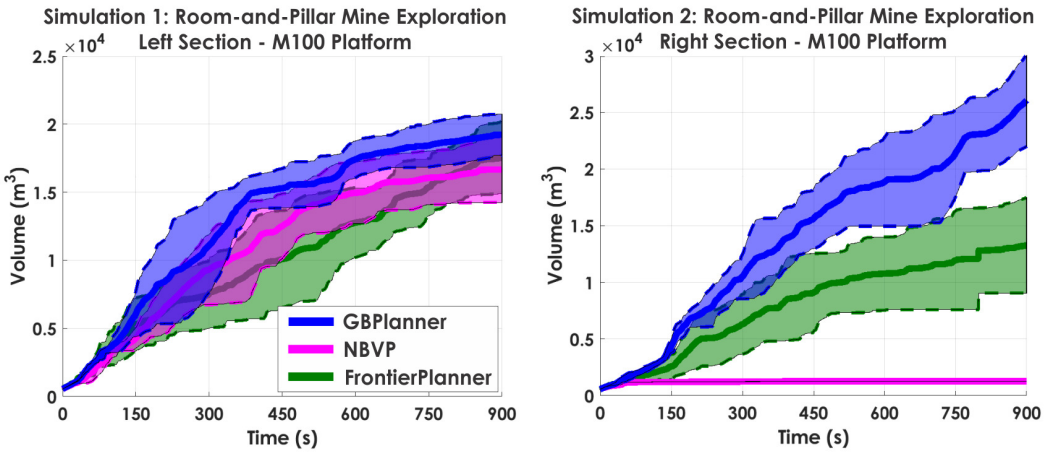

Figure 10: Exploration progress from the three considered planners (GBPlanner, NBVP, and FrontierPlanner) in a simulated room-and-pillar mine with two scenarios that took place in the left and the right section of the environment using an aerial robot model (M100). The left and the right sub-figures show the changes in the explored volume during 15 independent runs inside the left and the right section of the mine respectively. The solid lines present the average over the whole 15 runs associated with shaded areas which are the lower and upper bound from those runs. Overall, GBPlanner outperforms the other two methods and works reliably in both cases. NBVP is incapable of providing planning solutions for exploring narrow spaces due to its fixed–global sampling space setting, thus leading to an insufficient number of samples to find feasible paths through narrow corridors within reasonable computational bounds. Furthermore, the FrontierPlanner achieves slower exploration rate and is sensitive to inaccessible frontiers that arise in tight spaces.

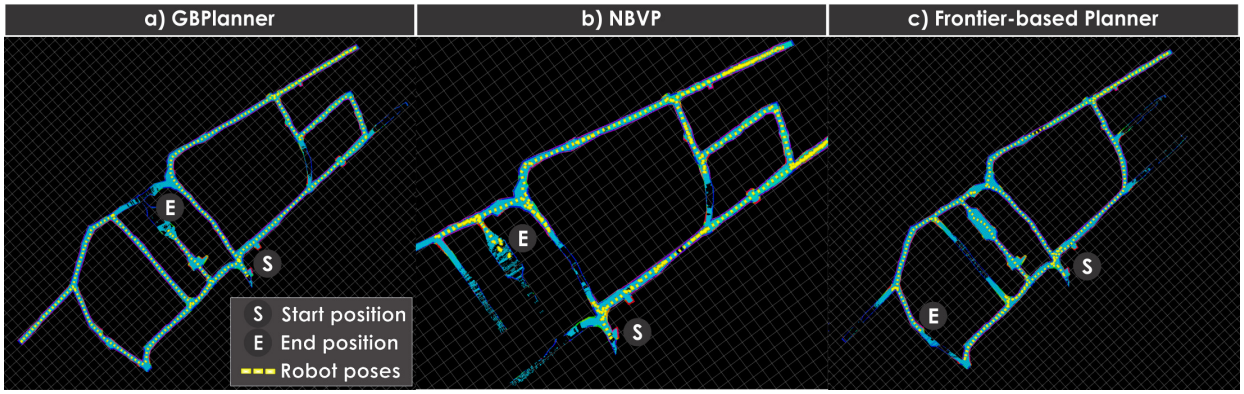

Figure 11: Indicative results for autonomous exploration inside a simulated long-wall mine. The GBPlanner achieves the best result thanks to its bifurcated approach utilizing both local planner for rapid exploration and global planner for fast relocation. The FrontierPlanner shows a comparable result with a slightly lower exploration rate, while the NBVP demonstrates very slow exploration progress primarily due to its branch switching behavior when approaching intersections.

6.1.1 Subterranean Aerial Robotic Scouts

The utilized aerial robotic scouts called “Alpha” and “Charlie” are both based around a DJI Matrice M100 and integrate a multi-modal sensor fusion solution combining LiDAR (Velodyne PuckLITE or Ouster OS-1), visual (FLIR Blackfly) and inertial (VectorNav VN-100) estimation, while Charlie further integrates a thermal camera (FLIR Tau2 LWIR) but the platforms are otherwise identical. Both flying systems rely on Model Predictive Control (MPC) for their automated operation and GBPlanner subscribes to the data

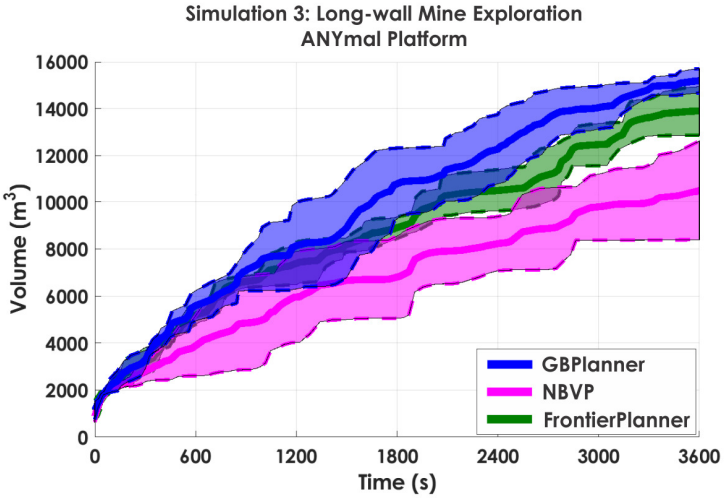

Figure 12: Exploration progress from three planners (GBPlanner, NBVP, and FrontierPlanner) in a simulated long-wall mine with the ANYmal platform. The solid lines present the average explored volume over 15 missions, enclosing by a lower and upper bound in shaded areas. In this simulated environment, both GBPlanner and FrontierPlanner almost completed the full exploration for the whole environment thanks to its geometric simplicity, yet the GBPlanner still demonstrates enhanced performance. The NBVP shows the lowest exploration rate since it continually switches between branches when traversing near intersections.

provided by the localization and mapping system (Khattak et al., 2020a) and provides references to the onboard controller. All the processing takes place onboard and in real-time based on an Intel NUC-i7 (NUC7i7BNH) computer that further interfaces the attitude and thrust control unit from DJI. Both LiDAR sensors integrated, the Velodyne PuckLITE on the Alpha robot and the Ouster OS-1 on the Charlie robot, provide a horizontal and vertical field of view of $F_H = 360°$, $F_V = 30°$ respectively. They both have a maximum range of 100m, while a map update takes place for the first 50m of ranging. Based on the robot size and safety considerations, the bounding box D_R for all flying robot tests was set to $length \times width \times height = 1.4 \times 1.4 \times 0.5$m. A system overview is illustrated in Figure 13.

6.1.2 ANYmal Legged Robot

ANYmal is a quadrupedal robot designed for autonomous operation in challenging environments (Bellicoso et al., 2018). The legs of this versatile machine are driven by twelve equal series elastic actuator units

Test Site	Location	Robotic Platforms	Date	Planner Modes
TRJV Mine	NV, U.S.	Charlie Aerial Robot	1/2019	Local & Global & Homing
Gonzen Mine	Sargans, CH	ANYmal	10/2019	Local & Global & Homing
Wampum Underground	PA, U.S.	Alpha Aerial Robot	8/2019	Local & Homing
Edgar Mine	CO, U.S.	ANYmal	4/2019	Local & Homing
NIOSH Mine	PA, U.S.	Alpha Aerial Robot	8/2019	Local & Homing
Menznau Bunker	Menznau, CH	ANYmal & Alpha	12/2019	Local & Homing

Table 3: Field testing environments considered in this study. Our experiments took place inside mines of different geologies and configurations, namely active gold long-wall mines (TRJV), two previously abandoned and now research mines one of which one long-wall (Edgar) and the other a room-and-pillar coal mine (NIOSH), an old and particularly large room-and-pillar coal mine (Wampum), and an abandoned long-wall iron mine (Gonzen). Furthermore, an additional test in an underground bunker (Menznau) was conducted.

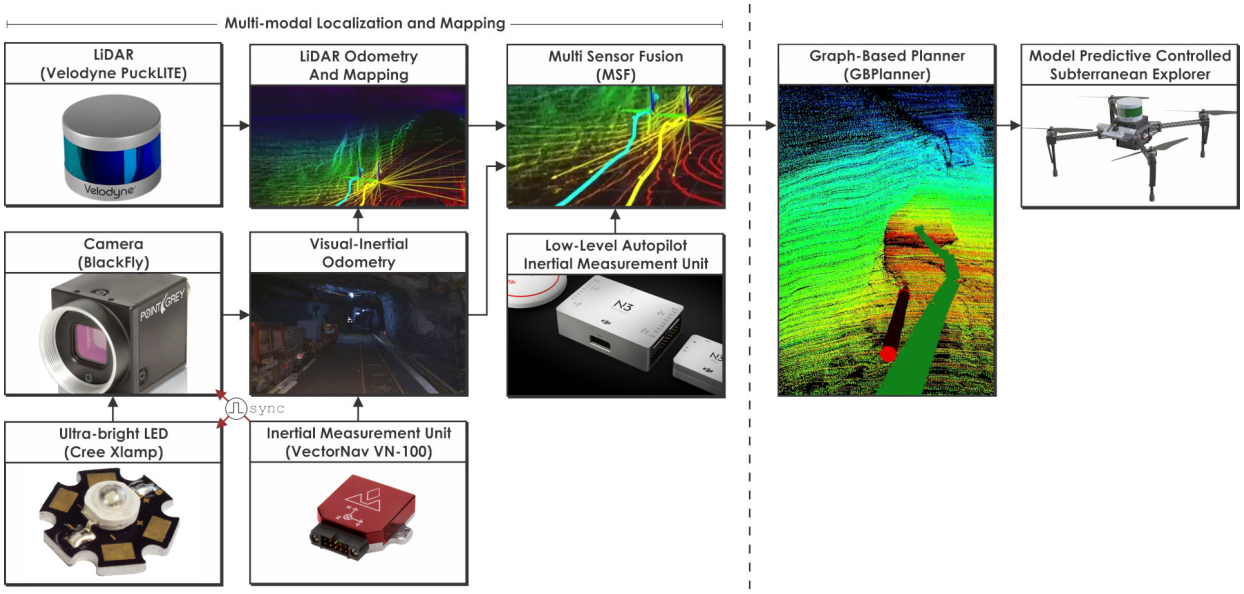

Figure 13: System overview of our aerial subterranean robotic scout "Alpha". The "Charlie" robot is similar with the difference of utilizing an Ouster OS-1 LiDAR as opposed to the Velodyne PuckLITE and further integrating a FLIR Tau2 LWIR thermal camera.

mounted at the joints designed to achieve a large range of motion, allowing to overcome obstacles and stairs. In addition, the robot is water and dust resistant and thanks to its rollover bar, Kevlar belly plate, and shock absorbers it is also resilient to falls. ANYmal can carry a variety of sensors as payload, based on the scenario where it is deployed. During the experiments presented in this work, the robot features a LiDAR (Velodyne PuckLITE) sensor, a depth camera (Intel RealSense D435), and an inertial measurement unit. Leg kinematics and inertial measurements are fused together to generate a so-called "leg odometry" by the onboard state estimator (Bloesch et al., 2013). The robot encloses in its main body three high performance computers (each featuring Intel i7 CPU) and one graphics processing unit (Jetson Xavier). Each on-board computer is responsible for one key aspect of the mission: locomotion, navigation, and mission execution. All the processing takes place onboard and in real-time. Lidar pointclouds, depth camera output, and leg odometry are then used by a multi-modal sensor fusion algorithm to provide a unified mapping and localization solution to the planner and a path follower module. Similar to the aerial robot scouts, the used Velodyne PuckLITE provides a horizontal and vertical field of view of $F_H = 360°$, $F_V = 30°$. Based on the robot size and safety considerations, the bounding box D_R for ANYmal was set to $length \times width \times height = 1.2 \times 0.85 \times 0.7$m. A system overview is illustrated in Figure 14.

6.2 Aerial Robotic Exploration of a Long-Wall Underground Mine

The first presented experiment relates to the autonomous exploration of a working level of the active TRJV underground long-wall gold mine in Northern Nevada. In this experiment, we present a first comprehensive evaluation of all the GBPlanner functionalities (for flying robots) with the goal to allow a clear understanding of its overall functionality. As presented in Figure 15 the autonomous exploration of this underground mine required that the GBPlanner utilizes all its modes of operation, namely local, global and auto-homing. The system starts from the robot take-off position and as all the space around it is initially unknown it engages the local planner. Since the environment is a long-wall mine this naturally leads to the robot following a certain branch for a significant distance up to the point of reaching an exploration "dead-end" (in this case a machine-shop inside the underground mine). As the algorithm not only explores locally but simultaneously builds a sparse global graph on which it identifies the frontier vertices with the highest expected exploration gain, the method was able to command an optimized and collision-free re-positioning path towards the

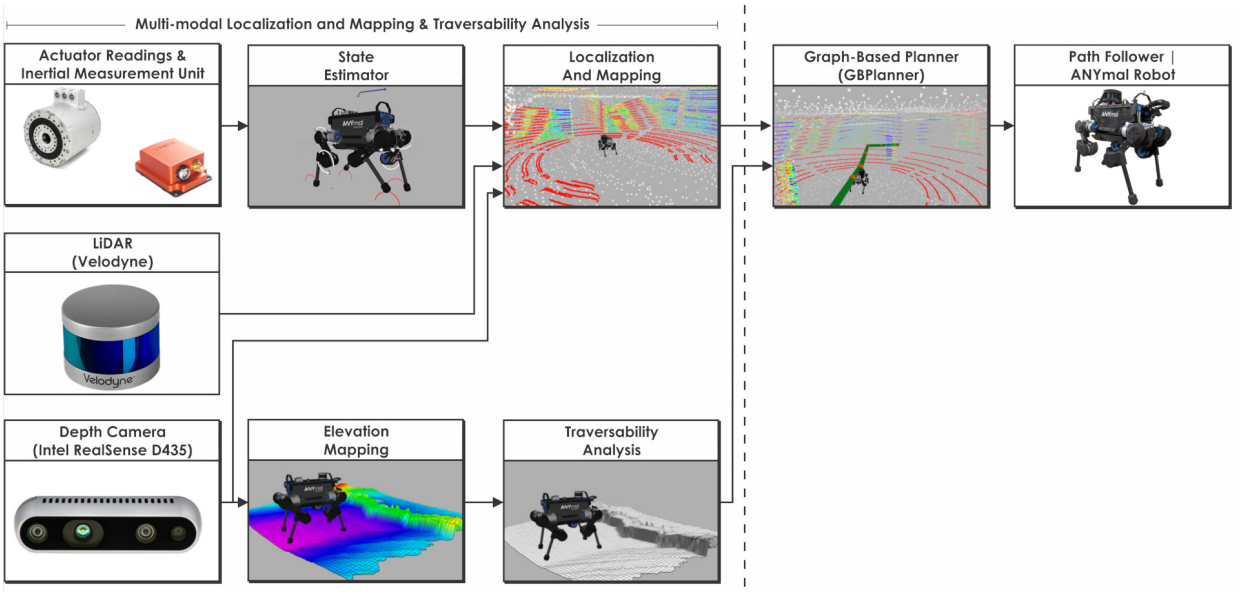

Figure 14: System overview of our ANYmal legged subterranean robot.

best exploration frontier. When the robot reached the frontier, the local mode was re-engaged and enabled efficient exploration in this region until it was identified that the system must perform auto-homing due to the limits of its battery life.

6.3 Walking Robot Exploration of an Underground Long-Wall Mine

An analogous case study was conducted using the ANYmal robot in order to evaluate the GBPlanner performance in such long-wall underground mines using legged systems. The specific test took place in Sargans, Switzerland at the Gonzen abandoned iron mine and the results are shown in Figure 16. A first difference with the previous test relates to the use of a legged system with the planner now further accounting for traversability constraints. In this test the robot initiates its mission and the local planner is engaged. As the system has to respect traversability constraints but the traversability map is only short-range the resolving architecture to continuously re-evaluate traversability and if needed command a short back-track motion to re-plan is demonstrated. Furthermore, similar to the previous experiment with a flying robot, GBPlanner continuously builds its local graph on which frontiers are also calculated. Thus, when ANYmal reached a local dead-end then a re-positioning, optimized and collision-free, path to the identified frontier is commanded. Notably, when the robot is re-positioned to the frontier, a fall-back homing path is also derived. In fact a re-positioning path is only commanded if the endurance is sufficient to also allow homing. Once the robot reached the commanded frontier, it continues its traversability-aware local exploration up to a point that a return-to-home path is commanded.

6.4 Aerial Robotic Exploration of an Underground Room-and-Pillar Mine

Underground environments and particularly mines present a variety of geometric configurations presenting different challenges for robots deployed to explore them autonomously. Beyond long-wall underground mines, another common configuration is the “room-and-pillar”. Room-and-pillar is a mining system in which the mined material is extracted across a horizontal plane, creating horizontal arrays of rooms and pillars. To evaluate the GBPlanner functionality in such a scenario, a test was conducted in the Wampum Underground facility which contains an abandoned and re-purposed underground coal mine with room-and-pillar structure. During this test the Alpha aerial robotic scout was tested and as shown in Figure 17 managed to explore multiple pillars of the environment despite its large-size (in cases, an intersection was more than 12m-wide).

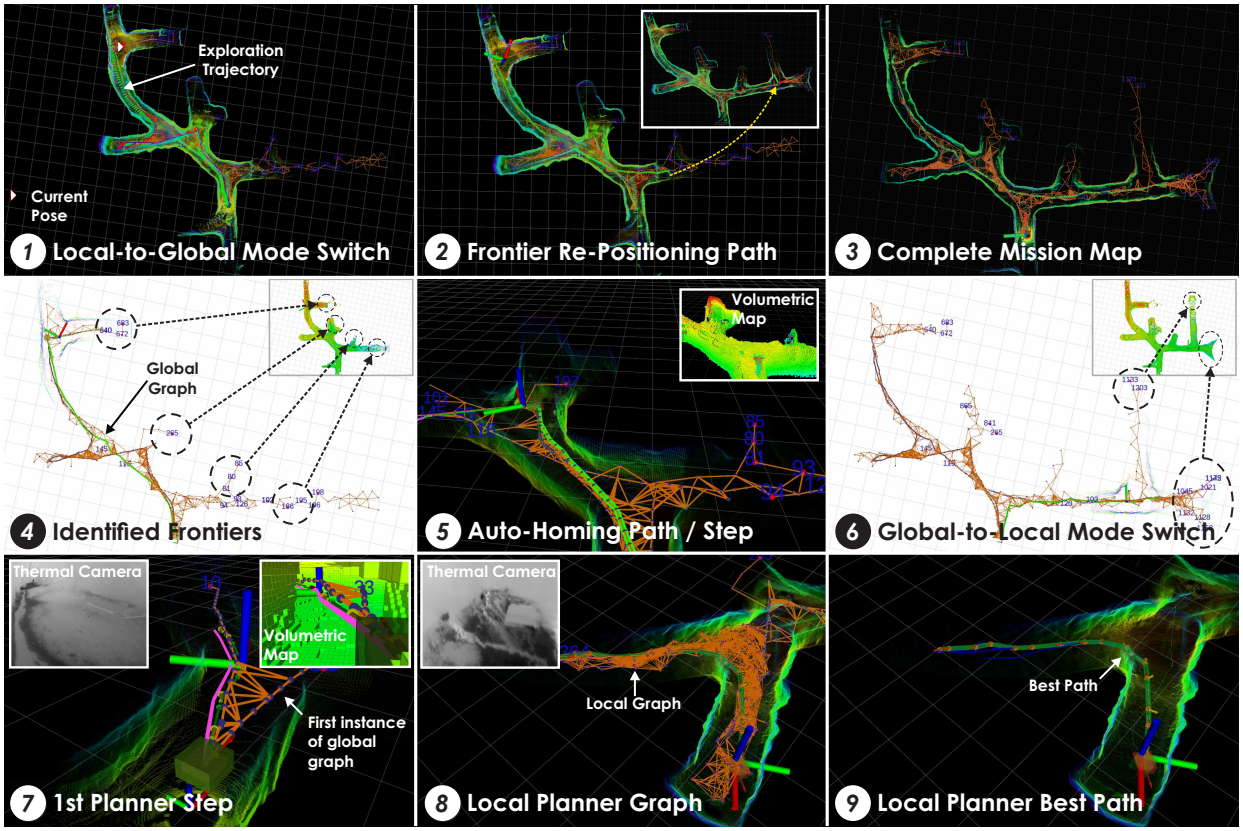

Figure 15: Autonomous exploration inside TRJV underground gold mine in Northern Nevada. This experiment demonstrates both the local and global planning functionalities of the proposed method. Initially the robot explores with the local mode engaged, while simultaneously building a global graph on which it calculates the gain of frontier vertices. When the local planner presents inability to explore further efficiently (due to a dead-end), then global planner is engaged and the frontier with maximum expectation for subsequent fruitful exploration is selected and an optimized collision-free path to its position is derived. When the robot reaches the frontier, then the local exploration mode is re-engaged and the process continues to map more of the unknown space. Eventually, the system identifies that its battery approaches its limit and timely calculates and commands a return-to-home (auto-homing) path. The approximate exploration path in this experiment was $230m$ long. A video of this mission can be found at `https://youtu.be/SNMsKAnCQkw`.

During this test the local planner is continuously engaged up to the point that the robot's battery life drops to a level that a return-to-home path must be commanded for the system to safely come back. Among others, this test contributed into evaluating the ability of GBPlanner to negotiate multi-way intersections.

6.5 Deployment in the DARPA Subterranean Challenge

The proposed exploration path planning framework was further tested within the framework of the DARPA Subterranean Challenge "Tunnel Circuit" - the first leg of this DARPA competition relating to the exploration of underground mines - as well as the preparatory "STIX" integration event. During these tests there was only a single human-operator who was responsible to command several robots, legged and flying, to enter the underground mine and then explore on their own. We present two field results, namely a) the exploration of the first segment of the "NIOSH underground mine" in Pittsburgh during the "Tunnel Circuit" competition days using the Aerial Scout flying robot, as well as b) the exploration of a particularly narrow section of the Edgar Underground Mine using the ANYmal legged robot. In all these tests it is important to note that a) there is a single human operator of all the robotic team, and b) initially the robots have to align their

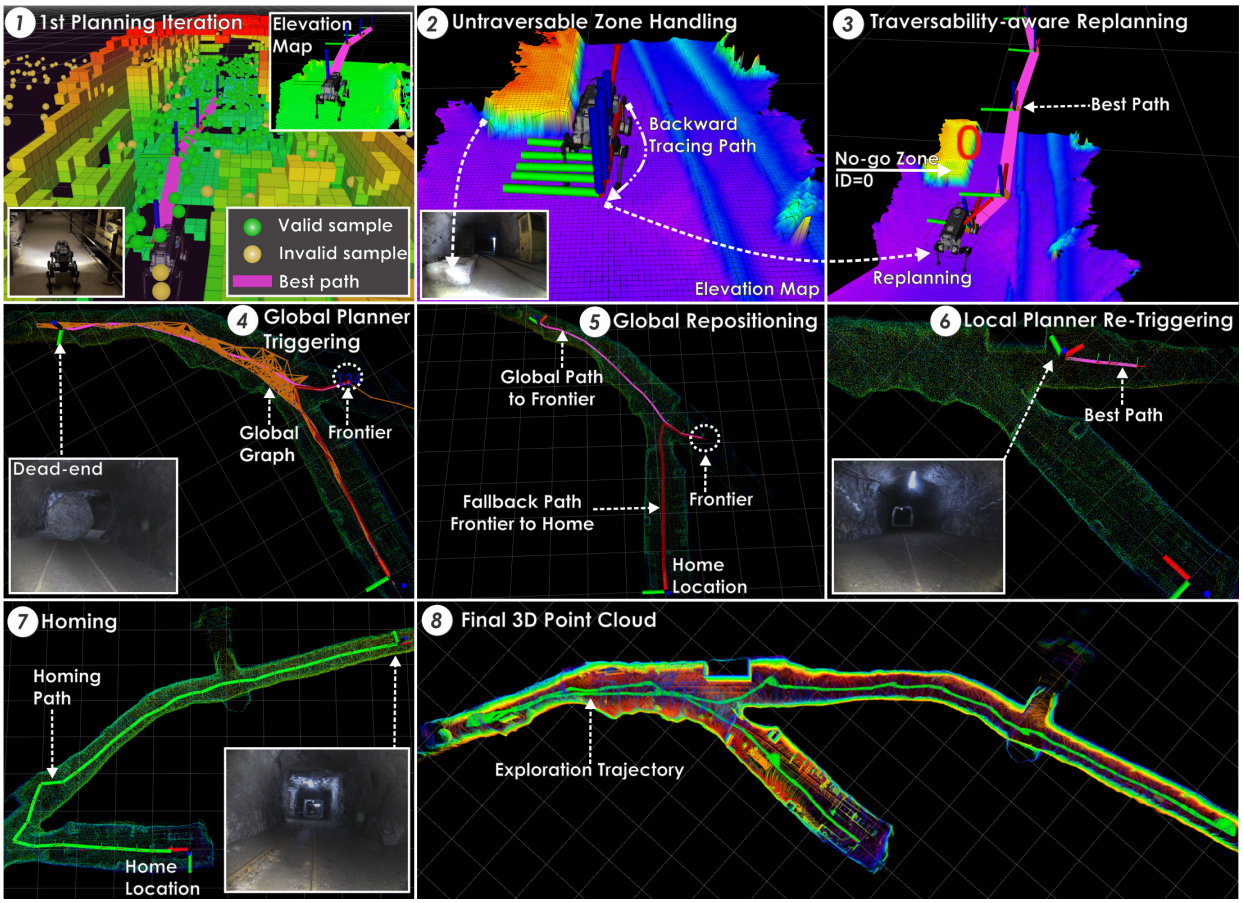

Figure 16: Autonomous exploration mission inside the Gonzen mine in Switzerland. The experiment demonstrates the full use of both local and global planner modes from the proposed planning solution. Utilizing both an occupancy map and a traversability map, the local planner recursively provides safe paths for ANYmal to explore the environment. In the presented sub-figures 1–3, both maps are colorized by the z-value or the height of each voxel. In the event of detecting untraversable zones along the exploration path, the robot is first commanded to backtrack a short distance (0.5m), the untraversable area is marked as a "no-go" zone and the local planner is then re-triggered to provide an alternative trajectory. Once the robot reaches a dead-end, the global planner is employed to relocate the robot towards another potential frontier. The homing procedure is eventually engaged when the remaining time approaches its maximum budget. The approximate exploration path in this mission was $300m$ long. A video of this mission can be found at `https://youtu.be/W9lgdmDg6UM`.

coordinate frame with a DARPA reference frame defined, among other options, through three AprilTags set-up by DARPA. The latter serves to allow scoring and assessing map quality by DARPA.

6.5.1 ANYmal Robot at the DARPA STIX Integration Exercise

The DARPA SubTerranean Integration eXercise (STIX) took place at the Edgar underground research mine in Colorado. This mine is an abandoned long-wall mine that used to produce silver, gold, lead and copper. As part of this pre-competition activity robotic systems were deployed at two portals of the mine, namely the "Army Portal" and "Miami Portal" and then proceeded to explore autonomously. Figure 18 presents an autonomous exploration mission conducted by the ANYmal robot utilizing GBPlanner and deployed at the Miami portal. The mission was subject to specific time constraints (60min for all the deployed robots)

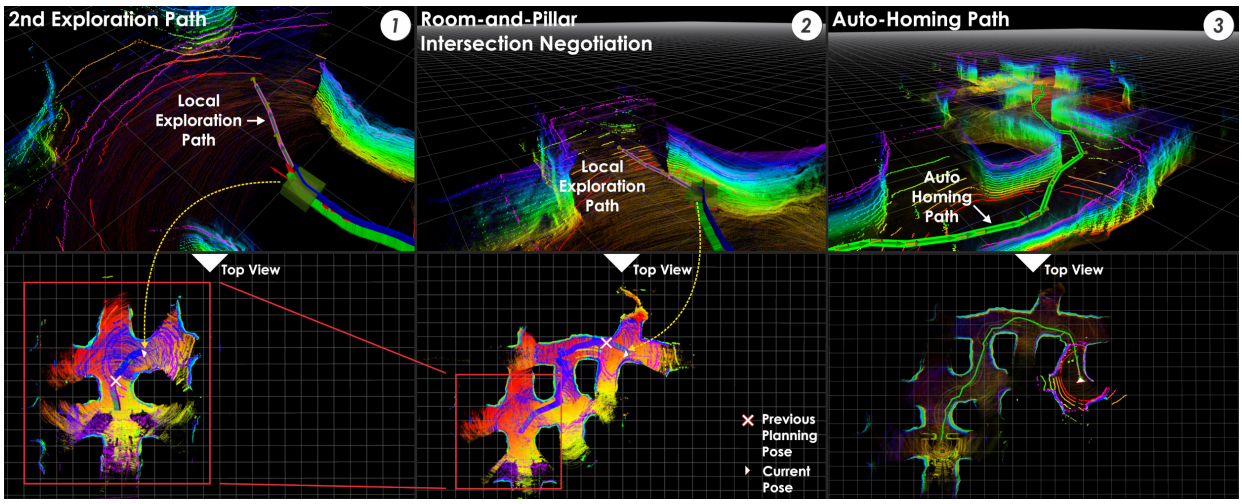

Figure 17: Autonomous exploration inside the "Wampum Underground" facility which corresponds to an abandoned room-and-pillar underground coal mine. Due to the room-and-pillar structure the environment involves a very large set of intersections that have to be negotiated by the planner. In this experiment the local planner is engaged continuously and only when the battery of the robot approaches its limits, the global planning layer is automatically engaged to offer an auto-homing path. In the first subplot we present the second planning step, the second subplot presents a case where the planner negotiates an intersection, and the third presents the return-to-home path. A relevant video result accompanies this field deployment. The approximate exploration path in this test was 180m long. A video of this experiment can be found at `https://youtu.be/cxfz8cqcuhA`.

and no communication link was available other than that self-established by the robots. In this mission, the planner guides the robot based on its local planning mode and finally commands a return-to-home path. Being a test early in our development process, in this experiment Octomap is used for volumetric mapping.

6.5.2 Alpha Aerial Robot at the DARPA Subterranean Challenge "Tunnel Circuit"

The DARPA Subterranean Challenge Tunnel Circuit took place in the facilities of the National Institute for Occupational Safety and Health (NIOSH) in Pittsburgh which involve an abandoned, now re-purposed for research, room-and-pillar underground coal mine. Each participating team, and as part of this also our team "CERBERUS", had to deploy all its robots from outside the mine and only a single human-operator could command the sequence with which robots enter the subterranean environment and which areas they should explore. No communication link was also provided other than the one self-deployed by the robots. During the 4-th scored run of our team we deployed the Alpha aerial robotic scout to explore the long entrance corridor of the NIOSH underground mine. The single human-operator only provided approximate bounds of the exploration space and the system autonomously entered the underground mine, explored as per the limits of its battery life and automatically returned back to its starting position. The specific environment is quite narrow (often less than 3m corridor width). The relevant result is depicted in Figure 19. As shown, the robotic system started from outside the mine portal and proceeded to explore automatically. Initially the robot had to detect the portal gate and align with a DARPA-provided coordinate frame defined by three AprilTags, then pass the gate autonomously and subsequently engage GBPlanner to perform its mission. The local planning mode was initially engaged and allowed the robot to explore this first part of the mine. As the initial exploration bounds provided by the human operator were conservative with respect to the robot exploration rate, the operator subsequently re-defined the bounds and the system explored up to the end of this first straight mine tunnel section. Eventually, as the system endurance approached its limits the planner automatically commanded a return-to-home path and the robot safely exited the mine and landed in the competition "staging" area.

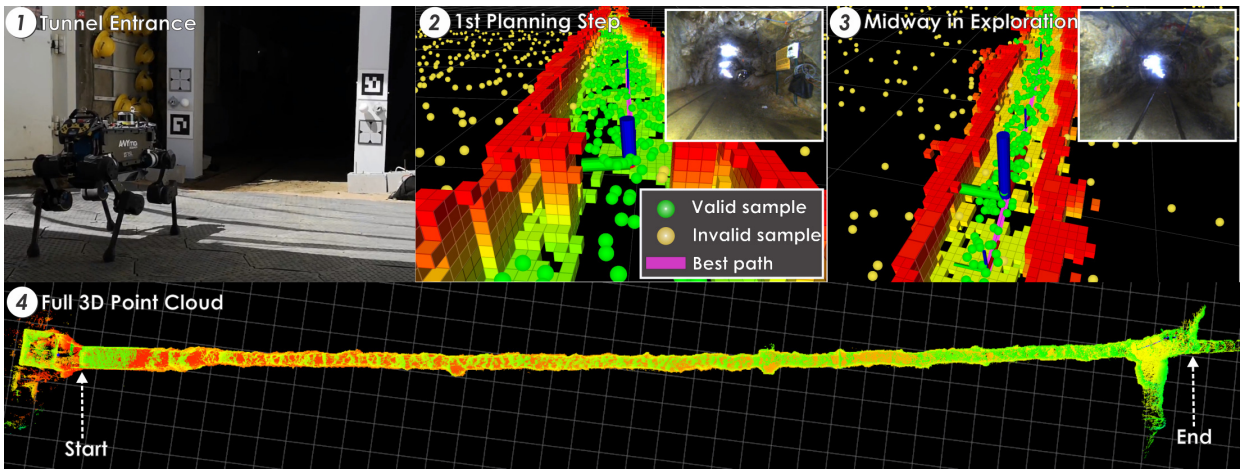

Figure 18: Autonomous exploration inside the Edgar mine in Colorado. This mission is a part of the system integration exercise "STIX", which was organized by DARPA to prepare for the first competition of the SubT Challenge. After manually performing the initial alignment between the robot's coordinate frame with the DARPA reference frame and entering the mine, the local planner was engaged and utilized to navigate the robot to explore the mine progressively. This mine is very long and narrow, while its terrain is uneven and covered by a long rail-track which is a common feature inside old mines. The robot was forced to stop inside the mine after one hour of total team deployment time as this is a hard time limit for any scored runs in the DARPA SubT Challenge. The approximate exploration path in this mission was 200m long. A video of this mission can be found at `https://youtu.be/JFKeB8Q7B2s`.

6.6 Exploration of an Underground Bunker

Subterranean settings can be extremely diverse with underground mines, caves, and urban subway infrastructure being only indicative examples. As part of our team preparation for the "Urban Circuit" of the DARPA Subterranean Challenge we further evaluated the proposed planner inside an underground bunker in Menznau, Switzerland. The Menznau bunker involves a network of narrow corridors and rooms offering the evaluation of the planner ability to traverse constrained passages and efficiently explore wide spaces. In this test both the Alpha flying robot and ANYmal were deployed and explored sections of this facility.

6.6.1 Aerial Robot inside the Menznau Underground Bunker

The exploration of the Menznau Bunker took place by combining the flying and legged robots. The "Alpha" flying robot was commanded to initiate the mission and the system proceeded to take-off, pass the entrance gate to the area to be explored and then GBPlanner was automatically engaged to enable efficient exploration. The system proceeded with the local planner engaged exploring multiple bunker rooms and traversing several narrow corridors. The system was simultaneously building its sparse global graph which at the end enabled the commanding of an efficient and collision-free auto-homing path. The respective result is shown in Figure 20 where the performance of the exploration process can be visually assessed. Of great importance is the improvement of the return-to-home path - as opposed to a naive backtracking approach - since the system optimized the path to the home destination and visibly reduced the distance to be traversed.

6.6.2 ANYmal Robot inside the Menznau Underground Bunker

The ANYmal robot was commanded to explore the same region to simultaneously evaluate a multi-robot mapping functionality that our team was developing. As such, the order of procedures employed in this experiment are analogous to the one before, namely a) initializing mission, b) calibrating against the external

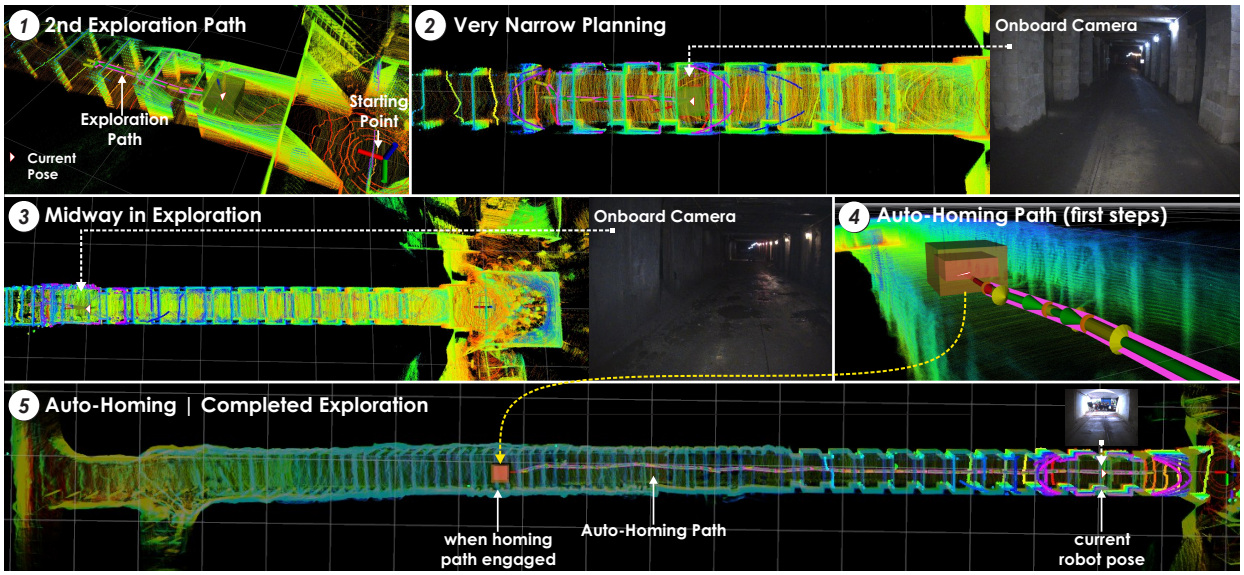

Figure 19: Autonomous exploration inside the NIOSH Underground Mine in Pittsburgh as part of the DARPA Subterranean Challenge "Tunnel Circuit" competition activities. The single human operator of all the CERBERUS robot team had designated the entrance corridor to be explored by the "Alpha" flying robot by providing approximate bounds of the exploration space. The complete mission was fully autonomous including a) take-off, b) alignment with DARPA frame (based on provided AprilTags), c) pass-through the mine entrance gate, d) engaging the exploration planner in local planning mode, and finally e) automated return-to-home. This environment is particularly narrow and thus the property of the planner to exploit local graph search is key to its capacity to find solutions. The approximate exploration path in this mission was 180m long. A video of this mission can be found at `https://youtu.be/mw0qy05Fo6Q`.

frame, c) passing the gate, d) enabling the graph-based exploration and guiding the robot through local planning steps, and finally e) commanding a safe return-to-home path. The conducted mission, including the exploration from both the flying and the legged robots was timed to fit within a 30min window. Figure 21 presents the relevant exploration mission by the ANYmal robot. As can be observed, especially due to the narrow cross-section of ANYmal, the planner efficiently traverses very narrow settings and corridors.

6.7 Discussion

The presented results allow for a comprehensive evaluation of the performance of the proposed graph-based exploration path planner in challenging and diverse subterranean environments. Each planning functionality, including the local exploration, the global functionality to re-position the robot towards frontiers and enable safe auto-homing, handling of collision-avoidance, respecting traversability constraints, negotiating multi-way intersections and navigating narrow corridors was evaluated. GBPlanner demonstrated its capacity to offer a unified high-performance solution applicable to different robots and in a large variety of underground environments. To help facilitate further research in the domain, alongside this paper we release a) an open-source implementation of the presented algorithm (ARL, 2020), as well as b) a large open dataset with planning, localization and mapping, point cloud and when applicable camera data of all the abovementioned experiments to allow research re-producibility and extension (ARL and RSL, 2019). Notably, across the diverse field tests presented, the planner parameters were largely invariant. Apart from the different bounding box reflecting different robot sizes and the size of the search volume of the local planner (set to smaller for more narrow environments and larger otherwise) the rest of the parameters were identical across the tests.

As a further point it is worth briefly outlining some important challenges and limitations identified in the process of these field experiments. First, in most of these underground environments, especially inside dry

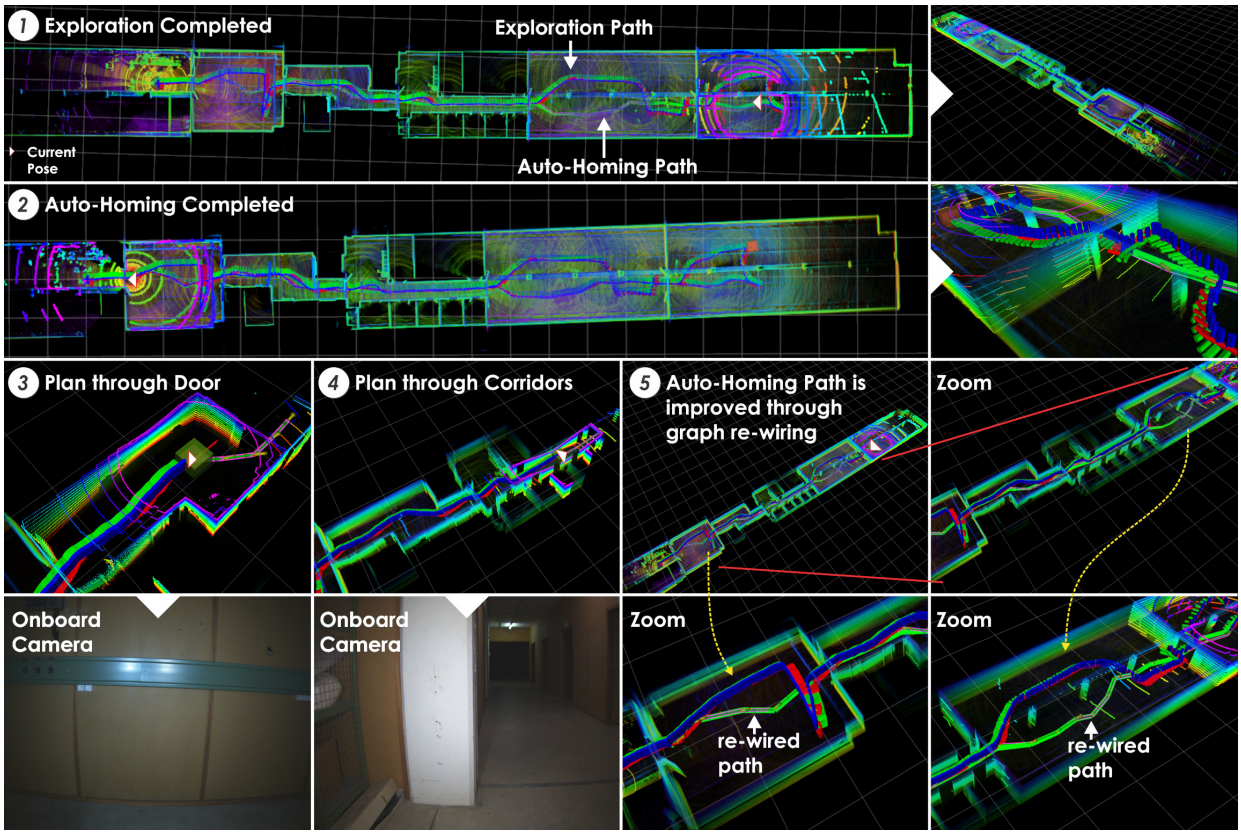

Figure 20: Autonomous exploration inside an underground bunker in Switzerland using a flying robot. This experiment presents the local exploration functionality of the planner in such an environment and the auto-homing provided by the global planner. The system explores through both wide and constrained environments, passes through multiple doors and traverses narrow corridors up to the point that the algorithm identifies the need to command an auto-homing path due to the limits of the system's battery life. The approximate exploration path in this experiment was 180m long. A video of an experiment inside the underground bunker is available at `https://youtu.be/0HBfc9Fq0K0`. The recording of the experiment in the video stops before the return-to-home path is executed.

mines, the aerial robot created dense clouds of dust which challenged the onboard localization process and the reconstructed volumetric map. Although, through a set of contributions our team has focused on enabling resilient localization in such conditions (Khattak et al., 2020b). On the other hand, both the Octomap and Voxblox volumetric mapping frameworks occasionally presented the error of populating free voxels as occupied due to LiDAR measurements that stopped on the dust cloud, while in other cases LiDAR data had gaps due to refraction. The first problem was addressed but at a slight penalty of needing more measurements to map "thin" objects (e.g., a metal bar of small diameter). Simultaneously, as the robot explores areas that contain cycles the incorporation of a loop-closure architecture, not extensively implemented in the experiments presented above, is a key functionality. Broadly, further work in the direction of multi-modal localization and mapping is needed for complete and resilient autonomy in subterranean environments.

An additional insight gained from our results relates to the importance of agile exploration given the size of underground environments, often spanning several kilometers, and the limited endurance of small robotic systems. The presented results do contain techniques to optimize this behavior, including planning the next path before the robot completes its current reference, incorporation of the future reference in the model predictive control loops but improvements on that front can be beneficial. Last but not least, we want to highlight that for ground systems long-range traversability analysis can be key towards faster exploration as

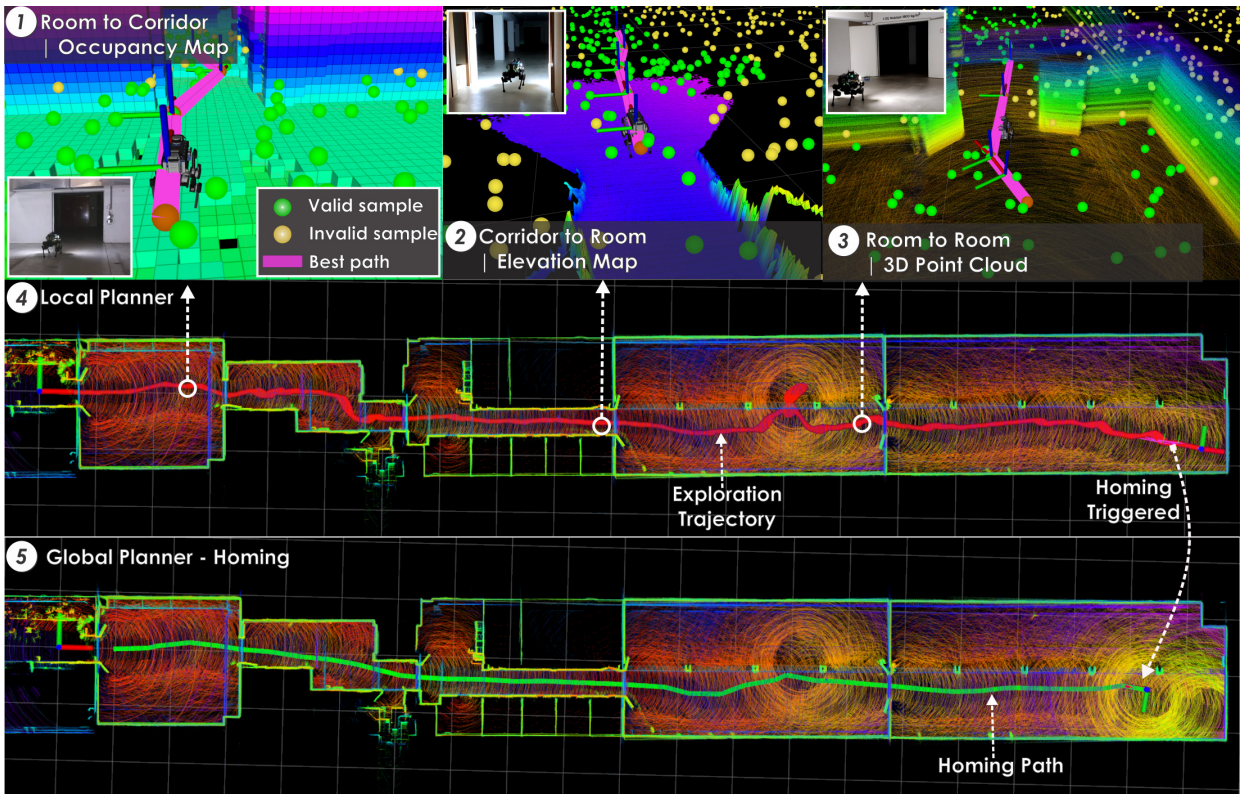

Figure 21: Autonomous exploration inside an underground bunker in Switzerland. This experiment features both local and global (-homing) functionalities of the proposed planner. Utilizing trajectories provided by the local planner, the robot navigates safely through a previously unmapped underground bunker with multiple doors, narrow passages, as well as relatively large rooms. The homing procedure is finally engaged at time appropriate for the provided time budget constraint allocated for this mission. The approximate exploration path in this mission was 200m long. A video of the complete experiment can be found at `https://youtu.be/r48WBG55_dY`.

otherwise the onboard planner has to resort to either being efficient but occasionally unrealistically optimistic (possibly planning towards non-traversable regions) or safe but conservative (planning only very short paths ahead).

7 Conclusions and Future Work

This paper detailed a new graph-based exploration path planning strategy tailored to underground environments and applicable to both walking and flying robot configurations. Through a bifurcated local- and global-planning architecture the method is efficient and presents high performance in subterranean environments that are at the same time large-scale, narrow, and contain multiple branches. The local stage of the planner facilitates efficient exploration, while also allowing to identify paths that traverse very narrow corridors and respect applicable traversability constraints. As the robot explores on its own it incrementally builds a sparse global graph which in turn is used to re-position itself towards frontiers of the exploration space or to provide a safe return-to-home path. The planner is evaluated first in a collection of simulation studies and compared with other methods of the state-of-the-art. Subsequently, a wide collection of field experiments is presented involving different types of environment geometries and thus allowing comprehensive evaluation. Overall, the method presented high performance against diverse environments and robots and was field-proven with respect to its resilience.

Future research will focus on addressing further hard problems in subterranean exploration. First, this relates to multi-storey exploration in environments with staircases, vertical stopes and other mobility challenges connecting different levels of the environment. Second, we will strive to augment our exploration autonomy solution to exploit multi-robot teams intelligently. Currently, GBPlanner was fielded as part of multi-robot deployments in the DARPA SubT Challenge Tunnel and Urban Circuit but fundamentally every robot was operating on its own given an allocation of a bounded volume of interest (with the map within being initially unknown). Our research will seek to implement efficient map sharing and multi-robot exploration behaviors using GBPlanner as the autonomy kernel of this approach.

Acknowledgments

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Agreement No. HR00111820045, by the Swiss National Science Foundation through the National Centre of Competence in Research Robotics (NCCR Robotics), and the European Union's Horizon 2020 research and innovation programme under grant agreement No. 780883. The presented content and ideas are solely those of the authors.

References

Andre, T. and Bettstetter, C. (2016). Collaboration in multi-robot exploration: to meet or not to meet? *Journal of Intelligent & Robotic Systems*, 82(2):325–337.

ARL (2020). Open-Source Release of the Graph-based Exploration Path Planner. `https://github.com/unr-arl/gbplanner_ros`.

ARL and RSL (2019). Autonomous Robots Lab Website: GBPlanner, Online Supplemental Material. `https://www.autonomousrobotslab.com/gbplanner.html`.

Arora, S. and Scherer, S. (2017). Randomized algorithm for informative path planning with budget constraints. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4997–5004. IEEE.

Bachrach, A. G. (2013). *Trajectory bundle estimation For perception-driven planning*. PhD thesis, Massachusetts Institute of Technology.

Bakambu, J. N. and Polotski, V. (2007). Autonomous system for navigation and surveying in underground mines. *Journal of Field Robotics*, 24(10):829–847.

Baker, C., Morris, A., Ferguson, D., Thayer, S., Whittaker, C., Omohundro, Z., Reverte, C., Whittaker, W., Hahnel, D., and Thrun, S. (2004). A campaign in autonomous mine mapping. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 2. IEEE.

Balaguer, C., Giménez, A., Pastor, J. M., Padron, V., and Abderrahim, M. (2000). A climbing autonomous robot for inspection applications in 3d complex environments. *Robotica*, 18(3):287–297.

Balta, H., Bedkowski, J., Govindaraj, S., Majek, K., Musialik, P., Serrano, D., Alexis, K., Siegwart, R., and De Cubber, G. (2017). Integrated data management for a fleet of search-and-rescue robots. *Journal of Field Robotics*, 34(3):539–582.

Bellicoso, C. D., Bjelonic, M., Wellhausen, L., Holtmann, K., Günther, F., Tranzatto, M., Fankhauser, P., and Hutter, M. (2018). Advances in real-world applications for legged robots. *Journal of Field Robotics*, 35(8):1311–1326.

Berenson, D., Abbeel, P., and Goldberg, K. (2012). A robot path planning framework that learns from experience. In *2012 IEEE International Conference on Robotics and Automation*, pages 3671–3678. IEEE.

Bircher, A., Alexis, K., Burri, M., Oettershagen, P., Omari, S., Mantel, T., and Siegwart, R. (2015). Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6423–6430. IEEE.

Bircher, A., Kamel, M., Alexis, K., Burri, M., Oettershagen, P., Omari, S., Mantel, T., and Siegwart, R. (2016a). Three-dimensional coverage path planning via viewpoint resampling and tour optimization for aerial robots. *Autonomous Robots*, 40(6):1059–1078.

Bircher, A., Kamel, M., Alexis, K., Oleynikova, H., and Siegwart, R. (2016b). Receding horizon" next-best-view" planner for 3d exploration. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 1462–1468. IEEE.

Bjelonic, M., Sankar, P. K., Bellicoso, C. D., Vallery, H., and Hutter, M. (2019). Rolling in the deep–hybrid locomotion for wheeled-legged robots using online trajectory optimization. *arXiv preprint arXiv:1909.07193*.

Bloesch, M., Hutter, M., Hoepflinger, M. A., Leutenegger, S., Gehring, C., Remy, C. D., and Siegwart, R. (2013). State estimation for legged robots-consistent fusion of leg kinematics and imu. *Robotics*, 17:17–24.

Calisi, D., Farinelli, A., Iocchi, L., and Nardi, D. (2005). Autonomous navigation and exploration in a rescue environment. In *IEEE International Safety, Security and Rescue Rototics, Workshop, 2005.*, pages 54–59. IEEE.

Caprari, G., Breitenmoser, A., Fischer, W., Hürzeler, C., Tâche, F., Siegwart, R., Nguyen, O., Moser, R., Schoeneich, P., and Mondada, F. (2012). Highly compact robots for inspection of power plants. *Journal of Field Robotics*, 29(1):47–68.

Charrow, B., Kahn, G., Patil, S., Liu, S., Goldberg, K., Abbeel, P., Michael, N., and Kumar, V. (2015). Information-theoretic planning with trajectory optimization for dense 3d mapping. In *Robotics: Science and Systems*, volume 11.

Connolly, C. (1985). The determination of next best views. In *Proceedings. 1985 IEEE international conference on robotics and automation*, volume 2, pages 432–435. IEEE.

Corah, M. and Michael, N. (2019). Distributed matroid-constrained submodular maximization for multi-robot exploration: Theory and practice. *Autonomous Robots*, 43(2):485–501.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to algorithms*. MIT press.

Dang, T., Mascarich, F., Khattak, S., Nguyen, H., Khedekar, N., Papachristos, C., and Alexis, K. (2019a). Field-hardened robotic autonomy for subterranean exploration. In *Conference on Field and Service Robotics (FSR)*.

Dang, T., Mascarich, F., Khattak, S., Papachristos, C., and Alexis, K. (2019b). Graph-based path planning for autonomous robotic exploration in subterranean environments. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3105–3112. IEEE.

Dang, T., Papachristos, C., and Alexis, K. (2018). Visual saliency-aware receding horizon autonomous exploration with application to aerial robotics. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2526–2533. IEEE.

Deits, R. and Tedrake, R. (2015). Computing large convex regions of obstacle-free space through semidefinite programming. In *Algorithmic foundations of robotics XI*, pages 109–124. Springer.

Delmerico, J., Mintchev, S., Giusti, A., Gromov, B., Melo, K., Horvat, T., Cadena, C., Hutter, M., Ijspeert, A., Floreano, D., Gambardella, L. M., Siegwart, R., and Scaramuzza, D. (2019). The current state and future outlook of rescue robotics. *Journal of Field Robotics*, 36(7):1171–1191.

Delmerico, J., Mueggler, E., Nitsch, J., and Scaramuzza, D. (2017). Active autonomous aerial exploration for ground robot path planning. *IEEE Robotics and Automation Letters*, 2(2):664–671.

Fankhauser, P., Bloesch, M., Gehring, C., Hutter, M., and Siegwart, R. (2014). Robot-centric elevation mapping with uncertainty estimates. In *International Conference on Climbing and Walking Robots (CLAWAR)*.

Fankhauser, P., Bloesch, M., and Hutter, M. (2018). Probabilistic terrain mapping for mobile robots with uncertain localization. *IEEE Robotics and Automation Letters (RA-L)*, 3(4):3019–3026.

Furrer, F., Burri, M., Achtelik, M., and Siegwart, R. (2016). Rotors-a modular gazebo mav simulator framework. In *Robot Operating System (ROS)*, pages 595–625. Springer.

Gehring, C., Fankhauser, P., Isler, L., Diethelm, R., Bachmann, S., Potz, M., Gerstenberg, L., and Hutter, M. (2019). Anymal in the field: Solving industrial inspection of an offshore hvdc platform with a quadrupedal robot. In *12th Conference on Field and Service Robotics (FSR 2019)*.

Grocholsky, B., Keller, J., Kumar, V., and Pappas, G. (2006). Cooperative air and ground surveillance. *IEEE Robotics & Automation Magazine*, 13(3):16–25.

Hollinger, G. A. and Sukhatme, G. S. (2014). Sampling-based robotic information gathering algorithms. *The International Journal of Robotics Research*, 33(9):1271–1287.

Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., and Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*.

Huang, Y.-W., Lu, C.-L., Chen, K.-L., Ser, P.-S., Huang, J.-T., Shen, Y.-C., Chen, P.-W., Chang, P.-K., Lee, S.-C., and Wang, H.-C. (2019). Duckiefloat: a collision-tolerant resource-constrained blimp for long-term autonomy in subterranean environments. *arXiv preprint arXiv:1910.14275*.

Hutter, M., Diethelm, R., Bachmann, S., Fankhauser, P., Gehring, C., Tsounis, V., Lauber, A., Guenther, F., Bjelonic, M., Isler, L., et al. (2018). Towards a generic solution for inspection of industrial sites. In *Field and Service Robotics*, pages 575–589. Springer.

Hutter, M., Gehring, C., Jud, D., Lauber, A., Bellicoso, C. D., Tsounis, V., Hwangbo, J., Bodie, K., Fankhauser, P., Bloesch, M., Diethelm, R., Bachmann, S., Melzer, A., and Hoepflinger, M. (2016). ANYmal - a highly mobile and dynamic quadrupedal robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 38–44. IEEE.

Jacobson, A., Zeng, F., Smith, D., Boswell, N., Peynot, T., and Milford, M. (2020). What localizes beneath: A metric multisensor localization and mapping system for autonomous underground mining vehicles. *Journal of Field Robotics*.

Jones, A., Schwager, M., and Belta, C. (2013). A receding horizon algorithm for informative path planning with temporal logic constraints. In *IEEE International Conference on Robotics and Automation*. IEEE.

Kanellakis, C., Karvelis, P., and Nikolakopoulos, G. (2019). Open space attraction based navigation in dark tunnels for mavs. In *International Conference on Computer Vision Systems*, pages 110–119. Springer.

Karaman, S. and Frazzoli, E. (2009). Sampling-based motion planning with deterministic μ-calculus specifications. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 2222–2229. IEEE.

Khattak, S., Mascarich, F., Dang, T., and Papachristos, Christos Alexis, K. (2019). Robust thermal-inertial localization for aerial robots: A case for direct methods. In *2019 IEEE International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1061–1068.

Khattak, S., Nguyen, H., Mascarich, F., Dang, T., and Alexis, K. (2020a). Complementary multi-modal sensor fusion for resilient robot pose estimation in subterranean environments. In *2020 IEEE International Conference on Unmanned Aircraft Systems (ICUAS)*.

Khattak, S., Papachristos, C., and Alexis, K. (2020b). Keyframe-based thermal–inertial odometry. *Journal of Field Robotics*, 37(4):552–579.

Kolvenbach, H., Valsecchi, G., Grandia, R., Ruiz, A., Jenelten, F., and Hutter, M. (2019). Tactile Inspection of Concrete Deterioration in Sewers with Legged Robots. In *Field and Service Robots (FSR)*, page 14, Tokyo, Japan.

Lajoie, P.-Y., Ramtoula, B., Chang, Y., Carlone, L., and Beltrame, G. (2019). Door-slam: Distributed, online, and outlier resilient slam for robotic teams. *arXiv preprint arXiv:1909.12198*.

Lew, T., Emmei, T., Fan, D. D., Bartlett, T., Santamaria-Navarro, A., Thakker, R., and Agha-mohammadi, A.-a. (2019). Contact inertial odometry: Collisions are your friend. *arXiv preprint arXiv:1909.00079*.

Li, H., Savkin, A. V., and Vucetic, B. (2019). Autonomous area exploration and mapping in underground mine environments by unmanned aerial vehicles. *Robotica*, pages 1–15.

Liu, S., Watterson, M., Mohta, K., Sun, K., Bhattacharya, S., Taylor, C. J., and Kumar, V. (2017). Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments. *IEEE Robotics and Automation Letters*, 2(3):1688–1695.

Lösch, R., Grehl, S., Donner, M., Buhl, C., and Jung, B. (2018). Design of an autonomous robot for mapping, navigation, and manipulation in underground mines. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1407–1412. IEEE.

MacAllister, B., Butzke, J., Kushleyev, A., Pandey, H., and Likhachev, M. (2013). Path planning for non-circular micro aerial vehicles in constrained environments. In *2013 IEEE International Conference on Robotics and Automation*, pages 3933–3940. IEEE.

Mansouri, S. S., Castaño, M., Kanellakis, C., and Nikolakopoulos, G. (2019a). Autonomous mav navigation in underground mines using darkness contours detection. In *International Conference on Computer Vision Systems*, pages 164–174. Springer.

Mansouri, S. S., Karvelis, P., Kanellakis, C., Koval, A., and Nikolakopoulos, G. (2019b). Visual subterranean junction recognition for mavs based on convolutional neural networks. In *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*, volume 1, pages 192–197. IEEE.

Marchant, R. and Ramos, F. (2014). Bayesian optimisation for informative continuous path planning. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 6136–6143. IEEE.

Mascarich, F., Khattak, S., Papachristos, C., and Alexis, K. (2018). A multi-modal mapping unit for autonomous exploration and mapping of underground tunnels. In *2018 IEEE aerospace conference*, pages 1–7. IEEE.

Michael, N., Shen, S., Mohta, K., Mulgaonkar, Y., Kumar, V., Nagatani, K., Okada, Y., Kiribayashi, S., Otake, K., Yoshida, K., et al. (2012). Collaborative mapping of an earthquake-damaged building via ground and aerial robots. *Journal of Field Robotics*, 29(5):832–841.

Miller, I. D., Cladera, F., Cowley, A., Shivakumar, S. S., Lee, E. S., Jarin-Lipschitz, L., Bhat, A., Rodrigues, N., Zhou, A., Cohen, A., et al. (2020). Mine tunnel exploration using multiple quadrupedal robots. *IEEE Robotics and Automation Letters*, 5(2):2840–2847.

Moore, A. W. (1991). An intoductory tutorial on kd-trees.

Morris, A., Ferguson, D., Omohundro, Z., Bradley, D., Silver, D., Baker, C., Thayer, S., Whittaker, C., and Whittaker, W. (2006). Recent developments in subterranean robotics. *Journal of Field Robotics*, 23(1):35–57.

Novák, P., Babjak, J., Kot, T., Olivka, P., and Moczulski, W. (2015). Exploration mobile robot for coal mines. In *International Workshop on Modelling and Simulation for Autonomous Systems*, pages 209–215. Springer.

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 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Papachristos, C., Khattak, S., and Alexis, K. (2017). Uncertainty-aware receding horizon exploration and mapping using aerial robots. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 4568–4575. IEEE.

Papachristos, C., Khattak, S., Mascarich, F., Dang, T., and Alexis, K. (2019). Autonomous aerial robotic exploration of subterranean environments relying on morphology–aware path planning. In *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 299–305. IEEE.

Popovic, M., Hitz, G., Nieto, J., Sa, I., Siegwart, R., and Galceran, E. (2016). Online informative path planning for active classification using uavs. *arXiv preprint arXiv:1609.08446*.

Rao, B., Gopi, A. G., and Maione, R. (2016). The societal impact of commercial drones. *Technology in Society*, 45:83–90.

Richter, C., Bry, A., and Roy, N. (2016). Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments. In *Robotics Research*, pages 649–666. Springer.

Rufli, M., Ferguson, D., and Siegwart, R. (2009). Smooth path planning in constrained environments. In *2009 IEEE International Conference on Robotics and Automation*, pages 3780–3785. IEEE.

S. Karaman and E. Frazzoli (2011). Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7):846–894.

Sawada, J., Kusumoto, K., Maikawa, Y., Munakata, T., and Ishikawa, Y. (1991). A mobile robot for inspection of power transmission lines. *IEEE Transactions on Power Delivery*, 6(1):309–315.

Shivakumar, S. S., Rodrigues, N., Zhou, A., Miller, I. D., Kumar, V., and Taylor, C. J. (2019). Pst900: Rgb-thermal calibration, dataset and segmentation network. *arXiv preprint arXiv:1909.10980*.

Silver, D., Ferguson, D., Morris, A., and Thayer, S. (2006). Topological exploration of subterranean environments. *Journal of Field Robotics*, 23(6-7):395–415.

Stoyanov, T., Magnusson, M., Andreasson, H., and Lilienthal, A. J. (2010). Path planning in 3d environments using the normal distributions transform. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3263–3268. IEEE.

Tabib, W., Goel, K., Yao, J., Boirum, C., and Michael, N. (2020). Autonomous cave surveying with an aerial robot. *arXiv preprint arXiv:2003.13883*.

Tardioli, D., Riazuelo, L., Sicignano, D., Rizzo, C., Lera, F., Villarroel, J. L., and Montano, L. (2019). Ground robotics in tunnels: Keys and lessons learned after 10 years of research and experiments. *Journal of Field Robotics*, 36(6):1074–1101.

Tomic, T., Schmid, K., Lutz, P., Domel, A., Kassecker, M., Mair, E., Grixa, I. L., Ruess, F., Suppa, M., and Burschka, D. (2012). Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue. *IEEE robotics & automation magazine*, 19(3):46–56.

Wermelinger, M., Fankhauser, P., Diethelm, R., Krüsi, P., Siegwart, R., and Hutter, M. (2016). Navigation planning for legged robots in challenging terrain. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1184–1189. IEEE.

Yamauchi, B. (1997). A frontier-based approach for autonomous exploration. In *CIRA'97*, pages 146–151. IEEE.

Yoder, L. and Scherer, S. (2016). Autonomous exploration for infrastructure modeling with a micro aerial vehicle. In *Field and Service Robotics*, pages 427–440. Springer.