Graph-based Path Planning for Autonomous Robotic Exploration in Subterranean Environments

Tung Dang, Frank Mascarich, Shehryar Khattak, Christos Papachristos, and Kostas Alexis

***Abstract*— This paper presents a novel strategy for autonomous graph-based exploration path planning in subterranean environments. Attuned to the fact that subterranean settings, such as underground mines, are often large-scale networks of narrow tunnel-like and multi-branched topologies, the proposed planner is structured around a bifurcated local- and global-planner architecture. The local planner employs a rapidly-exploring random graph to reliably and efficiently identify collision-free paths that optimize an exploration gain within a local subspace. Accounting for the robot endurance limitations and the possibility that the local planner reaches a dead-end (e.g. a mine heading), the global planner is engaged when a return-to-home path must be derived or when the robot should be re-positioned towards an edge of the exploration space. The proposed planner is field evaluated in a collection of deployments inside both active and abandoned underground mines in the U.S. and in Switzerland.**

I. INTRODUCTION

Autonomous robotic exploration and mapping is expanding into an ever increasing set of applications in both the civilian and military domains alike. Aerial robots for example are currently utilized in a multitude of surveillance [1], industrial inspection [2, 3], search and rescue [4, 5], and commercial applications [6]. Nevertheless, a variety of environments remain particularly challenging for robotic entry and resilient autonomy. In particular, in this work we focus on the problem of autonomous exploration and mapping in subterranean settings. Subterranean worlds are characterized by a set of properties that render them difficult for autonomous aerial robots, including the fact that they are often extremely long and large-scale, particularly narrow and confined, self-similar, sensing-degraded, and communications-deprived, while they also typically feature rough and dynamic terrain. Despite these major challenges, the benefits of robotic autonomy underground can be very important across application domains and industries. Robots for mine rescue in emergency conditions inside underground mines, inspection of subterranean metropolitan infrastructure (e.g. subways, sewers), exploratory missions within caves and lava tubes on both Earth and in space, are indicative domains of relevant application.

In this work we build on top of a set of team contributions that address the problem of autonomous localization and mapping for aerial robots inside sensing-degraded environments [7–10] and focus on the problem of autonomous exploration path planning in a manner cognizant to the particularities of subterranean settings. We propose a new Graph-Based exploration path Planner (GBPlanner) that employs a bifurcated *local* and *global* planning architecture tailored to the underground application domain. The local planner operates within a bounded volume around the current robot location and employs a rapidly-exploring random graph sampling kernel to evaluate robot paths that maximize an information gain related to exploring further unmapped volume. The best path is selected such that the information gain is maximized and is then executed by the robot. When the local planner reports inability to identify paths that offer important information gain (e.g. when the robot is at a dead end such as a mine heading) or when the battery is running low, the global planning step is engaged and is responsible to either efficiently plan a path to the robot's homing location or towards a previously perceived edge of the exploration space (e.g. an unvisited branch). This bifurcated sampling-based scheme was identified to be capable of ensuring efficient and robust exploration for the large-scale, narrow, tunnel-like and multi-branched underground environments.

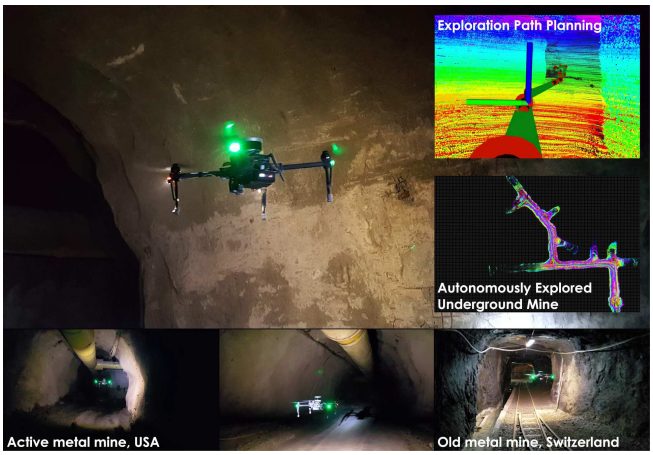

Fig. 1. Instances of autonomous graph-based exploration path planning in subterranean environments. The presented results are based on field tests in both active and inactive metal mines in the U.S. and in Switzerland. An indicative mission is shown at `https://youtu.be/VWgEVTeABdE`

The proposed approach on efficient exploration path planning in subterranean environments was evaluated in a multitude of field experiments using aerial robots including tests in both active and inactive mines in the U.S. and in Switzerland. More specifically, field test results are presented from missions that relate to a) autonomous long-term exploration in the development level of an active gold mine in the U.S.,

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Agreement No. HR00111820045. The presented content and ideas are solely those of the authors.

The authors are with the Autonomous Robots Lab, University of Nevada, Reno, 1664 N. Virginia, 89557, Reno, NV, USA `tung.dang@nevada.unr.edu`

Manuscript 1749 submitted to 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Received March 1, 2019.

and b) the exploration and auto-homing inside an abandoned, particularly narrow, underground iron mine in Switzerland.

The remainder of this paper is structured as follows: Section II presents related work, followed by the problem statement in III. The proposed approach is detailed in Section IV, while evaluation studies are presented in Sections V and VI. Finally, conclusions are drawn in Section VII.

II. RELATED WORK

A rich body of work has focused on the general problem of exploration path planning [11–21]. Early work includes the sampling of "next–best–views" [19], and frontiers–based exploration [20]. Recent efforts proposed receding horizon multi-objective planning [16, 17, 22, 23], multi-robot methods [24, 25], and emphasized extensive field work [18, 26]. However, in the majority of these efforts the exploration problem considered does not reflect the particularities of subterranean settings. Typically, the existing planning methods are mostly designed to deal with either outdoor areas or relatively small structured facilities. Neither of these cases reflect the challenges of the very large-scale, tunnel-like, multi-branching, narrow, and broadly complex underground 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 [27, 28] present methods for topological exploration of subterranean 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 [29–32] emphasized motion planning in geometrically-constrained environments although not specifically targeting subterranean domains. From a systems perspective, the works in [33, 34] overview technological innovations in ground and submersible subterranean robotics. 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 through a bifurcated approach of efficient and solutions-complete local planning strategy which is optimized for volumetric gain, combined with a global planning layer that is responsible for ensuring the autonomous homing operation of the robot and its re-positioning close to regions of unknown space after the robot has reached a dead-end (e.g. a mine heading). The system is extensively field tested in a multitude of underground mine deployments using aerial robots, while its algorithmic design makes it also applicable to other exploration problems involving large-scale and geometrically constrained environments with tunnel-like corridors and branching points.

III. PROBLEM STATEMENT

The exploration path planning problem, as considered in this work, is that of autonomously exploring and mapping an initially unknown space that broadly belongs to the class of subterranean environments. Such settings (e.g. underground mines, tunnels, and cave networks) present key challenges and inner structure that must be accounted for. Namely, that a) they are often of *very* large-scale, and b) their traversable space is constrained from the surrounding geometry in a tunnel-like topology consisting of tube-like structures and branching points. The first observation prohibits the planning problem from taking place in a computationally efficient manner at the full scale of the environment bounds. However, the latter observation provides a noticeable characteristic that the planning algorithm may exploit; focusing the exploration on iteratively updated local sub-spaces generates a potentially large exploration reward without the penalty of exhaustively searching through the whole space. This insight allows us to re-cast the exploration path planning problem into two bifurcated stages, namely that of the *local planner* responsible for efficient exploration within a spatial window around the current robot pose, and that of the *global planner* which is responsible for a) autonomous homing, and b) re-positioning of the robot to the edge of the exploration space when the local planner reports completion of its local search potential.

Let $\mathbb{M}$ be a 3D occupancy map of the environment that is incrementally built from observations of an onboard depth sensor $\mathbb{S}$. 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. Furthermore, let $d_{\max}$ be the maximum effective range 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 $\xi_t = [x_t, y_t, z_t, \psi_t]$. Furthermore, 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 $V_{*,explored} = V_* \backslash V_{*,res}$. Therefore, we define the concepts of "local completion" and "global completion" for the local and global planners respectively.

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 $V_{D_G,explored} = V_{D_G} \setminus V_{D_G,res}$.

The local and global planner problems are as follows.

***Problem** 1 (Local 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 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 $\mathbb{S}$. When "local completion" is reported by this planner, the global planner is to be engaged.

***Problem** 2 (Global Planner)* Given 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 unmapped

Manuscript 1749 submitted to 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Received March 1, 2019.

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}.

IV. PROPOSED APPROACH

This section outlines the proposed graph-based planner.

A. Architectural Overview

Subterranean settings are often comprised of very large-scale combinations of narrow tunnel networks with multiple branching points and long geometrically-constrained drifts. Excellent examples include those of underground mines, metropolitan subway infrastructure, and cave networks. In response to these facts, an appropriate planning strategy must be able to provide fast and efficient solutions for volumetric exploration despite the large-scale and geometrically-constrained character of such environments, while simultaneously offering a) safe return-to-home functionality, and b) solution resourcefulness when the exploration process reaches a dead-end (e.g. a mine heading). Such a solution is proposed and structured in the local and global planner architecture detailed below and outlined in Figure 2.

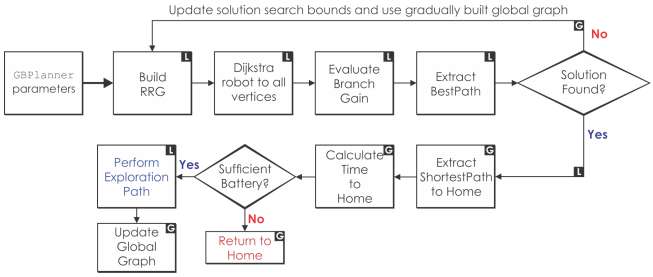

Fig. 2. Architecture of the proposed planner. The letter "L" indicates blocks that are designed as part of the local planner (although they are re-utilized by the global layer). The blocks marked with the letter "G" are related to the global planning layer (e.g. related to the auto-homing operation).

Both the local and global planning layers are built based on sampling-based rapidly-exploring random graph search, a choice that reflects notable features of underground environments (e.g. mines) including the fact that they are undirected and potentially include cycles.

B. Local Planner

In response to the needs for long-term exploration in complex, large-scale and multi-branched subterranean environments, the local planner is the core of the proposed method and ensures that the robot develops an efficient exploratory behavior tailored to the geometries of underground settings by utilizing the rapidly-exploring random graph algorithm [35]. First, an undirected graph $\mathbb{G}_L$ is incrementally built inside the local space $\mathbb{M}_L$ starting from the current robot configuration ξ_0 (set as the root vertex). Given the updated occupancy map $\mathbb{M}$, a bounded "local" volume V_{D_L} with dimensions D_L centered around ξ_0 and the respective map subset $\mathbb{M}_L$, and a bounding box D_R representing the physical size of the robot (alongside the respective map subset $\mathbb{M}_R(\xi_*)$ for a robot state ξ_*), the planner randomly samples a configuration ξ_{rand} inside V_{D_L} and checks if it is collision-free ($\mathbb{M}_R(\xi_{rand}) \in \mathbb{M}_{free}$). If this newly sampled vertex passes the collision check, the planner attempts to connect a straight path between this vertex and the closest vertex of the graph using nearest neighbor search [36]. If this connection is collision-free, the new vertex and the straight edge are added to the vertex set $\mathbb{V}$ and edge set $\mathbb{E}$ of the graph respectively. The planner continues to attempt to connect more collision-free paths between this vertex to nearby vertices within a defined radius δ to form a denser graph until the maximum number of vertices $N_{\mathbb{V},\max}$ or edges $N_{\mathbb{E},\max}$ is exceeded. Subsequently, the Dijkstra's algorithm [37] is utilized to find all the shortest paths Σ_L in the graph starting from the root vertex ξ_0 (root to all destinations). This aims to avoid unnecessary zig-zag motions which are undesirable in narrow space scenarios like mine tunnels or cave networks. Then, the planner computes the $\mathbf{VolumetricGain}$ for each vertex, which is the cumulative unmapped volume that an onboard depth sensor $\mathbb{S}$ would perceive depending on the robot's configuration at each vertex. This volumetric gain, combined with other weight functions related to distance and direction, is used to calculate the exploration gain for each shortest path. More specifically, 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 $\mathbf{ExplorationGain}(\sigma_i)$ is calculated as follows:

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

where $\mathcal{S}(\sigma_i, \sigma_{exp})$, $\mathcal{D}(\nu_1^i, \nu_j^i)$ are weight functions with tunable factors $\gamma_{\mathcal{S}}, \gamma_{\mathcal{D}} > 0$, and $\mathcal{D}(\nu_1^i, \nu_j^i)$ 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 penalize longer paths in order to favor a higher exploration rate. Moreover, $\mathcal{S}(\sigma_i, \sigma_{exp})$ is a similarity distance metric between the planned path as compared to a pseudo straight path σ_{exp} with the same length along the currently estimated exploration direction. This decay factor is essential to prevent the robot from a sudden, but not persistently derived, change in its exploration direction which might happen when the robot enters an intersection area with multiple branches or muckbays and the planner proposes small back-and-forth paths between them to chase higher rewards based on the local volumetric calculation. The exploration direction is simply estimated using a low-pass filter over a time window of robot pose configurations. Given this exploring direction vector, denoted as ϕ_{exp}, we then generate a pseudo straight path σ_{exp} using the same length with σ_i. Subsequently, the Dynamic Time Warping method (DTW) [38], commonly used in signal processing to estimate the similarity between time series signals, is utilized to compute the distance. Given the weighted calculation of gain, and the derivation of Dijkstra solutions, the Dijkstra path $\sigma_{L,best}$ that maximizes for exploration gain is selected

and conducted by the robot, while the whole procedure is then iteratively repeated. The pseudo code for the local planner is provided in Algorithm 1, and Algorithm 2. It is noted that as formal inference of local completion requires explicit knowledge of the residual space $V_{D_L,res}$, in practice this condition is detected when no path with exploration gain above a small threshold $g_\epsilon > 0$ is derived.

Algorithm 1 Local Planner

$\xi_0 \leftarrow \mathbf{GetCurrentConfiguration}()$
$\mathbb{G}_L \leftarrow \mathbf{BuildLocalGraph}(\xi_0)$
$\Sigma_L \leftarrow \mathbf{GetDijkstraShortestPaths}(\mathbb{G}_\mathbf{L}, \xi_\mathbf{0})$
$\mathbf{ComputeVolumetricGain}(\mathbb{G}_\mathbf{L})$ ▷ For all vertices
$g_{best} \leftarrow 0$
$\sigma_{L,best} \leftarrow \emptyset$
for all $\sigma \in \Sigma_L$ **do**
  $g_\sigma \leftarrow \mathbf{ExplorationGain}(\sigma)$
  if $g_\sigma > g_{best}$ **then**
    $g_{best} \leftarrow g_\sigma$; $\sigma_{L,best} \leftarrow \sigma$
  end if
end for
return $\sigma_{L,best}$

Algorithm 2 Build a Local Graph

function BUILDLOCALGRAPH(ξ_0)
  $\mathbb{V} \leftarrow \{\xi_0\}$; $\mathbb{E} \leftarrow \emptyset$
  $\mathbb{G}_L = (\mathbb{V}, \mathbb{E})$
  $D_L \leftarrow \mathbf{SetLocalBound}(\xi_0)$ ▷ 3D bounded space
  while $N_\mathbb{V} \leq N_{\mathbb{V},\max}$ **and** $N_\mathbb{E} \leq N_{\mathbb{E},\max}$ **do**
    $\xi_{rand} \leftarrow \mathbf{SampleFree}(D_L)$
    $\xi_{nearest} \leftarrow \mathbf{NearestVertex}(\mathbb{G}_L, \xi_{rand})$
    if $\mathbf{CollisionFree}(\xi_{rand}, \xi_{nearest})$ **then**
      $\mathbb{V} \leftarrow \mathbb{V} \cup \{\xi_{rand}\}$; $\mathbb{E} \leftarrow \mathbb{E} \cup \{\xi_{rand}, \xi_{nearest}\}$
      $\Xi_{near} \leftarrow \mathbf{NearestVertices}(\mathbb{G}_L, \xi_{rand}, \delta)$
      for all $\xi_{near} \in \Xi_{near}$ **do**
        if $\mathbf{CollisionFree}(\xi_{rand}, \xi_{near})$ **then**
          $\mathbb{E} \leftarrow \mathbb{E} \cup \{\xi_{rand}, \xi_{near}\}$
        end if
      end for
    end if
  end while
  return $\mathbb{G}_L = (\mathbb{V}, \mathbb{E})$
end function

C. Global Planner

The global planner contributes two important roles in the exploration mission, namely those of autonomous homing and re-positioning across paths offering exploration gain when local completion is detected. Towards efficient and endurance-aware autonomous homing, we propose a method to gradually build a lightweight global graph $\mathbb{G}_G$ by taking advantage of the local sub-graphs $\{\mathbb{G}_L\}$ built at every iteration of the local planner. In particular, after every local planning iteration, the global planner performs two steps to expand its graph. The first one is adding the current best path from the local graph $\sigma_{L,best}$ to $\mathbb{G}_G$ with the primary goal of supporting the return-to-home feature. To add any path to the global graph, each vertex belonging to the path is considered as a new sample and is sequentially added to $\mathbb{G}_G$ using a procedure similar to the one listed in Algorithm 2. This is to ensure that the global planner can always find the shortest path inside the explored space for homing instead of doing naive backtracking by just rewinding its previous waypoints which can cost significant time once the robot repeatedly passes over the same region. In the second step, the global planner attempts to add other shortest paths from the local graph with high volumetric gain.

Ultimately, for the auto-homing feature, the global planner applies Dijsktra's algorithm to search for the shortest path inside $\mathbb{G}_G$ to the predefined home vertex. A return-to-home calculation takes place at every iteration of the planner and the time-to-home T_H is compared to the current remaining flight time T_R. When $|T_R - T_H| \leq \epsilon_H$ (where $\epsilon_H > 0$ a constant), or the full environment is explored, the robot is commanded to return to home. Furthermore, when the local planner reports completion, a variation of the Local Planner in Algorithm 1 is engaged with D_L set to the full bounds D_G of the space to be explored (as opposed to local bounds around the robot pose), and the currently built global graph $\mathbb{G}_G$ is used as a prior on top of which the **BuildLocalGraph** method initializes. This global planning step allows the robot to identify remaining free space across all the boundaries (edges) of the space and thus offers solution resourcefulness when the local planner has reached a topology dead-end. Furthermore, initialization of the search based on the gradually built $\mathbb{G}_G$ mitigates the complexity of the graph search problem within the full exploration space inside large-scale environments. The global planning strategy therefore enables resilient exploration of large-scale and complex subterranean settings and return-to-home functionality.

D. Analysis of Complexity

To analyze the complexity of the proposed algorithm, which is essential to understand and tune the computation time in practice, we perform a detailed analysis of each sub-function within GBPlanner. First, regarding the local planner algorithm, the complexity of the $\mathbf{BuildLocalGraph}$ function is analyzed as follows. Given a 3D space with dimensions D_L, the probability to get a sample in the free space is $\rho = V_{D_L,free}/V_{D_L}$, where $V_{D_L,free}$ is the volume of total free space inside volume V_{D_L} of the local space. Moreover, the occupancy map used in this work is based on the octree data structure [39] which in turn leads to a $\mathcal{O}(\log(n))$ 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 resolution r, the status checking for each voxel will take $\mathcal{O}(\log(V_{D_G}/r^3))$. Subsequently, checking for collision-free navigation for the whole robot's body, V_{D_R} being its volume, takes another cost factor V_{D_R}/r^3. Therefore, the complexity for the function **SampleFree** is $\mathcal{O}(V_{D_L}/V_{D_L,free} \times V_{D_R}/r^3 \times \log(V_{D_G}/r^3))$. In an analogous manner, for the functions **CollisionFree** and

TABLE I
COMPLEXITY ANALYSIS OF THE LOCAL PLANNER

Functions	Complexity
SampleFree	$\mathcal{O}(V_{D_L}/V_{D_L,free} \times V_{D_R}/r^3 \times \log(V_{D_G}/r^3))$
CollisionFree	$\mathcal{O}(V_{D_R}/r^3 \times d_{avg}/r \times \log(V_{D_G}/r^3))$
AddConnections	$\mathcal{O}(N_{near}V_{D_R}/r^3 \times d_{avg}/r \times \log(V_{D_G}/r^3))$
BuildLocalGraph $\mathcal{O}(N_{\mathbb{V}}V_{D_L}/V_{D_L,free} \times V_{D_R}/r^3 \times \log(V_{D_G}/r^3)+$ $N_{\mathbb{V}}N_{near}V_{D_R}/r^3 \times d_{avg}/r \times \log(V_{D_G}/r^3))$	
Dijkstra	$\mathcal{O}(N_{\mathbb{V}} \log(N_{\mathbb{V}}))$
VolumetricGain	$\mathcal{O}(N_{\mathbb{V}}F_H F_V d_{max}/(r_H r_V r) \times \log(V_{D_G}/r^3))$
ExtractBestPath	$\mathcal{O}(N_{\mathbb{V}})$

AddConnections, the complexity analysis takes similar form as shown in Table I, where d_{avg} is the average length of edges in the graph, and N_{near} is the number of nearby neighbors to check for extra connections in order to build a denser graph. For the **NearestVertex** function, we are using a kd-tree [36], thus the search time is $\mathcal{O}(\log(N_{\mathbb{V}}))$. In addition, the underlying data structure used for the graph is based on an adjacency list which costs constant time $\mathcal{O}(1)$ to add a new vertex and $\mathcal{O}(\log(N_{\mathbb{E}}/N_{\mathbb{V}}))$ for adding an edge. Furthermore, the ray casting model for a range sensor with Field of View (FoV) $F_H \times F_V$ and resolution $r_H \times r_V$ is utilized to compute the volumetric gain for each sampled vertex. The overall complexity is shown in Table I. For the global planner, since we make use of the already-built local graph; the complexity is similar to that Algorithm 2.

V. SIMULATION BASED EVALUATION

In order to evaluate and tune the proposed graph-based exploration planner prior to its experimental verification, a simulation study within a large-scale underground mine-like environment took place. Using the RotorS Simulator [40], a study was conducted with the local planner set to operate within a sliding window with dimensions D_L of *length*×*width*×*height* = $40 \times 40 \times 4$m and a robot bounding box D_R of *length*×*width*×*height* = $1.5 \times 1.5 \times 0.5$m. The exploration speed was set to 1m/s. This test, depicted in Figure 3, unconstrained from the endurance limitations of real robots, allowed to tune the parameters of the GBPlanner and to verify the operation of the local planner, as well as the auto-homing and re-positioning features.

VI. EXPERIMENTAL EVALUATION

The proposed Graph-Based exploration Planner was then experimentally verified with respect to its ability to enable fast, efficient and robust exploration autonomy inside real-life complex subterranean settings. In particular, two different underground metal mines were utilized as field test sites. The first was the Turquoise Ridge Joint Venture (TRJV) underground mine in Winnemucca, Nevada, a modern and active gold mine with multiple levels underground (e.g. at $1,300$ft, $1,500$ft, and $1,700$ft depth) and very long mine drifts. The second was the Gonzen mine at Sargans, Switzerland, an old iron mine abandoned in 1966. Each of these mines present unique characteristics and challenges. The TRJV mine represents a very large-scale environment with

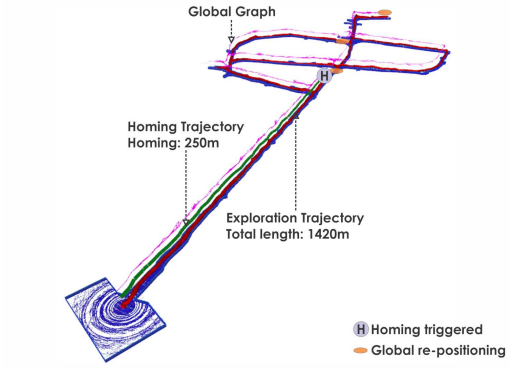

Fig. 3. Simulation-based evaluation of the graph-based exploration path planner in a km-scale environment prior to its experimental verification.

modern, *relatively* spacious ($\sim$ 4m wide) but very long drifts and headings, and a highly organized structure with sensing conditions that are often extremely degraded due to dust. The Gonzen mine incorporates more narrow mine drifts (often less than 3m wide) and relatively more irregular branching points. Both environments represent indicative cases for subterranean robotics with the presented tests being conducted in the framework of preparations for the ongoing DARPA Subterranean Challenge [41].

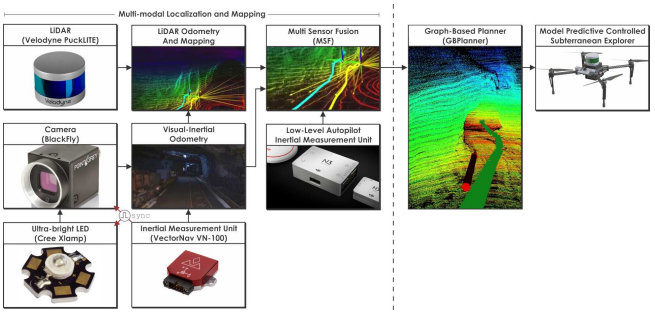

Fig. 4. System overview of our aerial subterranean robotic scout.

All the presented experiments were performed with an autonomous aerial robot developed around a DJI Matrice M100 and integrating a multi-modal sensor fusion solution combined with loosely-coupled LiDAR Odometry And Mapping, as well as visual-inertial localization. The system relies on Model Predictive Control (MPC) for its automated operation and the GBPlanner subscribes to the data provided by the localization and mapping solution and provides references to the onboard MPC. Details for the overall system solution can be found in [7, 9, 10, 42], while a system overview is illustrated in Figure 4. Notably, the depth sensor $\mathbb{S}$ integrated is a Velodyne PuckLITE which provides a horizontal and vertical field of view of $F_H = 360°$, $F_V = 30°$. It has 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 tests was set to *length*×*width*×*height* = $1.4 \times 1.4 \times 0.5$m.

In a first field experiment to evaluate the proposed planner operation, the "development" level of the TRJV mine

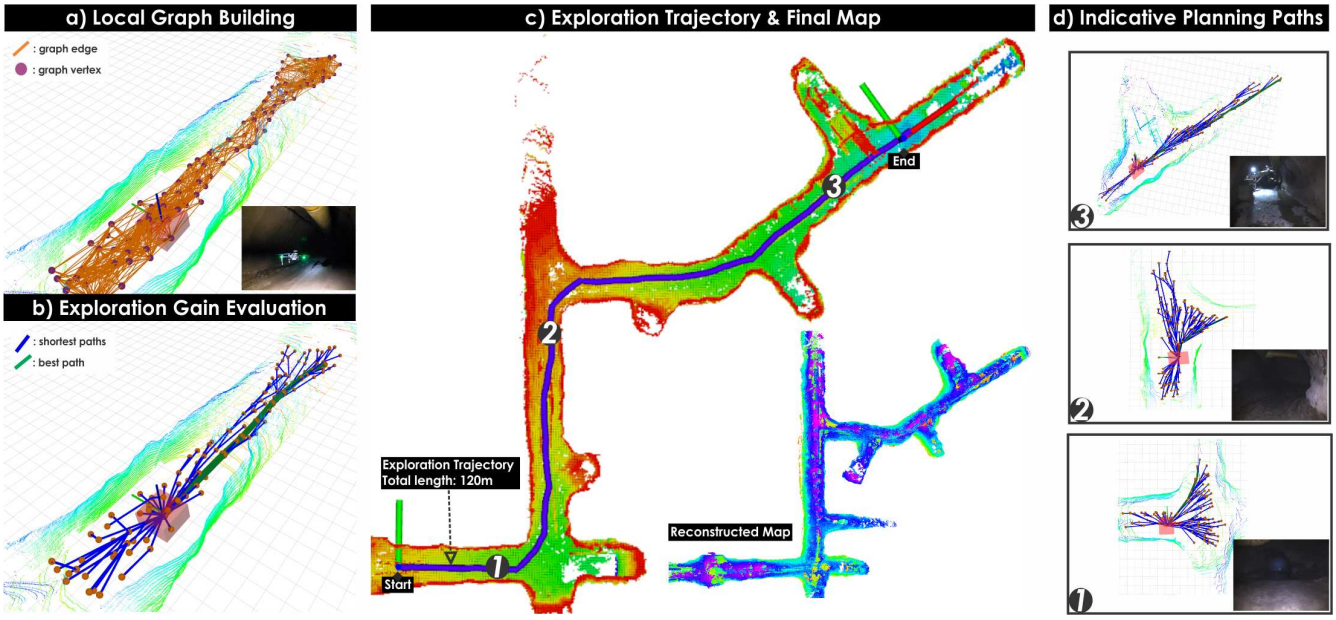

Fig. 5. Graph-based exploration path planner evaluation inside the TRJV gold mine in Winnemucca Nevada. The robot initiates its completely autonomous mission at the lower-left corner and completes the exploration at the upper right subset of the map. The GBPlaner derives efficient exploration paths which, as shown, lead to a total trajectory that intuitively follows the mine structure. In further detail, subfigure (a) presents the local graph built by the local planner, while subfigure (b) presents the Dijkstra solutions and the selected best exploration path. Similarly, subfigure (d) presents details of the exploration process at the points "1", "2", and "3" in subfigure (c). The reconstructed point cloud is further shown in top-down view. A video of the complete mission can be found at `https://youtu.be/VWgEVTeABdE`

(1, 300ft underground) was explored. The robot was only provided with general bounds D_G of the total volume to be explored, while the sliding window of the local planner V_L was set to a bounding box D_L of *length*×*width*×*height* = $40 \times 40 \times 4$m. Setting a nominal flight speed of 0.5m/s, the planner was engaged right after the take-off maneuver and through iterative steps of the local planner, the robot explored two drifts of the mine as presented in Figure 5. The entire mission took place in fully autonomous mode. As observed in the depicted result, the exploration path primarily follows the general structure of the mine which is also intuitively the efficient solution given the 360° horizontal field of view of the onboard LiDAR unit. When the planner was required to cope with points in the mine topology where there are small sub-spaces (small muckbays and equipment storing regions) that offer branching points but with small potential for volumetric exploration, the planned paths benefit from directionality bias and maintain high exploration rate focusing on the main drift. A video of this mission can be found at `https://youtu.be/VWgEVTeABdE`.

In the second field experiment, a section of the Gonzen mine is assigned to the planner for exploration. Similar to the previous experiment, the robot is only provided the general bounds of exploration, and the sliding window of the local planner is again set to *length*×*width*×*height* = $40 \times 40 \times 4$m. As this experiment had the explicit goal of evaluating the autonomous homing function and simultaneously pushing the exploration frontier of the robot in narrow settings, the exploration speed was set to 0.75m/s but the return-to-home speed was set to 1m/s. Figure 6 depicts the relevant results.

Finally, Figure 7 presents more illustrative images regarding the reconstruction quality of these underground mapping missions. As shown important structural details can be identified, while the onboard vision system further supports understanding of the subterranean environment. These results demonstrate the ability of the proposed planner to enable high performance autonomous exploration behaviors inside complex real-life subterranean environments. Following a tradition over our previous work [16, 17, 23], the GBPlanner will be open-sourced to the community.

VII. CONCLUSIONS

In this work, a graph-based path planner tailored to the task of subterranean exploration was proposed and demonstrated. The planner efficiently guides the robot within large-scale, multi-branched and tunnel-like narrow environments. The algorithm employs a two-stage local and global planner architecture with the local planner enabling exploration efficiency and the global planner offering return-to-home and topological re-positioning functionality. Field-hardened in underground mines using aerial robots, future work will involve its field evaluation and verification in a multitude of subterranean settings that further include cave networks and subway infrastructure using both aerial and ground platforms.

REFERENCES

[1] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas, "Cooperative air and ground surveillance," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 16–25, 2006.

[2] A. Bircher, M. Kamel, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel and R. Siegwart, "Three-dimensional coverage path planning via viewpoint resampling and tour optimization for aerial robots," *Autonomous Robots*, pp. 1–25, 2015.

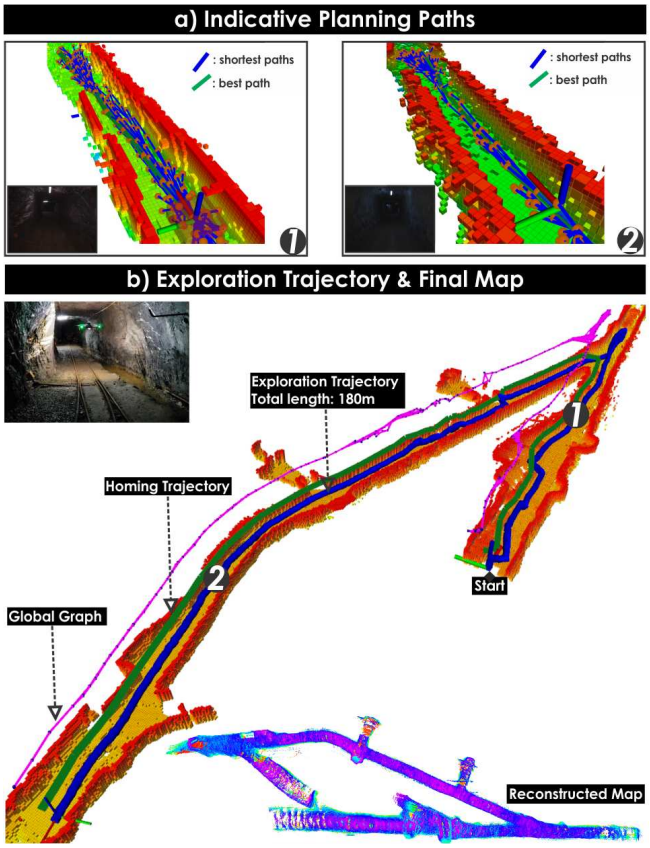

Fig. 6. Graph-based exploration path planner evaluation inside the Gonzen mine, Switzerland. The GBPlaner derives locally efficient exploration paths which, as shown, lead to a total trajectory that intuitively follows the mine structure. As shown in subfigure (b), this mission presents both the exploration path (derived in a sequence of planning steps) and the return-to-home trajectory. Instances of the operation of the local planner are further shown in the top subfigure, while the total reconstructed map is also depicted.

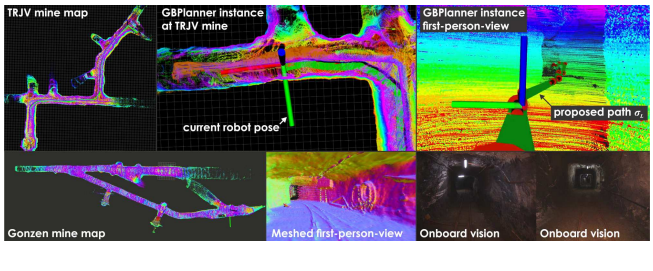

Fig. 7. Indicative mapping results from the presented underground exploration missions in the TRJV gold mine and the Gonzen iron mine. As shown, important structural details and an overall understanding of the subterranean geometry are provided, alongside the onboard collected camera data. The presented mesh representation is derived using the open-source Voxblox mapping package [43].

[3] A. Bircher, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel and R. Siegwart, "Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 6423–6430. [Online]. Available: https://github.com/ethz-asl/StructuralInspectionPlanner

[4] H. Balta, J. Bedkowski, S. Govindaraj, K. Majek, P. Musialik, D. Serrano, K. Alexis, R. Siegwart, and G. De Cubber, "Integrated data management for a fleet of search-and-rescue robots," *Journal of Field Robotics*, vol. 34, no. 3, pp. 539–582, 2017.

[5] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. L. Grixa, F. Ruess, M. Suppa, and D. Burschka, "Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue," *IEEE robotics & automation magazine*, vol. 19, no. 3, pp. 46–56, 2012.

[6] B. Rao, A. G. Gopi, and R. Maione, "The societal impact of commercial drones," *Technology in Society*, vol. 45, pp. 83–90, 2016.

[7] S. Khattak, C. Papachristos, and K. Alexis, "Keyframe-based direct thermal–inertial odometry," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2019.

[8] Shehryar Khattak, Christos Papachristos, and Kostas Alexis, "Visual-thermal landmarks and inertial fusion for navigation in degraded visual environments," in *IEEE Aerospace Conference (AeroConf)*, March 2019.

[9] C. Papachristos, S. Khattak, F. Mascarich, and K. Alexis, "Autonomous navigation and mapping in underground mines using aerial robots," in *2019 IEEE Aerospace Conference*, March 2019, p. *to appear*.

[10] C. Papachristos, and K. Alexis, "Thermal-inertial localization for autonomous navigation of aerial robots through obscurants," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2018.

[11] R. Marchant and F. Ramos, "Bayesian optimisation for informative continuous path planning," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6136–6143.

[12] A. Jones, M. Schwager, and C. Belta, "A receding horizon algorithm for informative path planning with temporal logic constraints," in *IEEE International Conference on Robotics and Automation*. IEEE, 2013.

[13] D. Berenson, P. Abbeel, and K. Goldberg, "A robot path planning framework that learns from experience," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*.

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

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

[16] C. Papachristos, S. Khattak, and K. Alexis, "Uncertainty–aware receding horizon exploration and mapping using aerial robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2017.

[17] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova and R. Siegwart, "Receding horizon "next-best-view" planner for 3d exploration," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2016. [Online]. Available: https://github.com/ethz-asl/nbvplanner

[18] L. Yoder and S. Scherer, "Autonomous exploration for infrastructure modeling with a micro aerial vehicle," in *Field and Service Robotics*. Springer, 2016, pp. 427–440.

[19] C. Connolly *et al.*, "The determination of next best views," in *IEEE International Conference on Robotics and Automation 1985*.

[20] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *CIRA'97*. IEEE, 1997, pp. 146–151.

[21] S. Arora and S. Scherer, "Randomized algorithm for informative path planning with budget constraints," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 4997–5004.

[22] C. Papachristos, M. Kamel, M. Popović, S. Khattak, A. Bircher, H. Oleynikova, T. Dang, F. Mascarich, K. Alexis, and R. Siegwart, "Autonomous exploration and inspection path planning for aerial robots using the robot operating system," in *Robot Operating System (ROS)*. Springer, 2019, pp. 67–111.

[23] T. Dang, C. Papachristos, and K. Alexis, "Visual saliency-aware receding horizon autonomous exploration with application to aerial robotics," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2018.

[24] J. Delmerico, E. Mueggler, J. Nitsch, and D. Scaramuzza, "Active autonomous aerial exploration for ground robot path planning," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 664–671, 2017.

[25] T. Andre and C. Bettstetter, "Collaboration in multi-robot exploration: to meet or not to meet?" *Journal of Intelligent & Robotic Systems*, vol. 82, no. 2, pp. 325–337, 2016.

[26] F. Mascarich, S. Khattak, C. Papachristos, and K. Alexis, "A multi-modal mapping unit for autonomous exploration and mapping of

underground tunnels," in *IEEE Aerospace Conference (AeroConf)*, March 2018.

[27] D. Silver, D. Ferguson, A. Morris, and S. Thayer, "Topological exploration of subterranean environments," *Journal of Field Robotics*, vol. 23, no. 6-7, pp. 395–415, 2006.

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

[29] D. Calisi, A. Farinelli, L. Iocchi, and D. Nardi, "Autonomous navigation and exploration in a rescue environment," in *IEEE International Safety, Security and Rescue Rototics, Workshop, 2005.* IEEE, 2005, pp. 54–59.

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

[31] M. Rufli, D. Ferguson, and R. Siegwart, "Smooth path planning in constrained environments," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 3780–3785.

[32] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *Robotics Research*. Springer, 2016, pp. 649–666.

[33] A. Morris, D. Ferguson, Z. Omohundro, D. Bradley, D. Silver, C. Baker, S. Thayer, C. Whittaker, and W. Whittaker, "Recent developments in subterranean robotics," *Journal of Field Robotics*, vol. 23, no. 1, pp. 35–57, 2006.

[34] P. Novák, J. Babjak, T. Kot, P. Olivka, and W. Moczulski, "Exploration mobile robot for coal mines," in *International Workshop on Modelling and Simulation for Autonomous Systems*. Springer, 2015, pp. 209–215.

[35] S. Karaman and E. Frazzoli, "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*. IEEE, 2009, pp. 2222–2229.

[36] A. W. Moore, "An intoductory tutorial on kd-trees," 1991.

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

[38] A. G. Bachrach, "Trajectory bundle estimation for perception-driven planning," Ph.D. dissertation, Massachusetts Institute of Technology, 2013.

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

[40] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, "Rotors-a modular gazebo mav simulator framework," in *Robot Operating System (ROS)*. Springer, 2016, pp. 595–625.

[41] Defense Advanced Research Projects Agency (DARPA), "DARPA Subterranean Challenge." [Online]. Available: https://subtchallenge.com/

[42] M. Kamel, T. Stastny, K. Alexis, and R. Siegwart, "Model predictive control for trajectory tracking of unmanned aerial vehicles using ros," *Springer Book on Robot Operating System (ROS)*.

[43] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.