Explore Locally, Plan Globally: A Path Planning Framework for Autonomous Robotic Exploration in Subterranean Environments

Conference Paper · December 2019

DOI: 10.1109/ICAR46387.2019.8981594

4 authors:

Tung Dang
University of Nevada, Reno

Shehryar Khattak
California Institute of Technology

Frank Mascarich
University of Nevada, Reno

Kostas Alexis
Norwegian University of Science and Technology

Explore Locally, Plan Globally: A Path Planning Framework for Autonomous Robotic Exploration in Subterranean Environments

Tung Dang, Shehryar Khattak, Frank Mascarich, Kostas Alexis

***Abstract*— This paper presents a path planning strategy for the autonomous exploration of subterranean environments. Tailored to the specific challenges and particularities of underground settings, and especially the fact that they often are extremely large in scale, tunnel-like, narrow and multi-branched, the proposed method employs a bifurcated local- and global-planning design to enable exploration efficiency and path planning solution resourcefulness. The local planner builds on top of minimum-length random trees and efficiently identifies collision-free paths that optimize for exploration within a local subspace, while simultaneously ensuring enhanced obstacle clearance and thus safety. 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 utilizes an incrementally built graph to search within the full range of explored space and is engaged when the robot should be re-positioned towards a frontier of the exploration space or when a return-to-home path must be derived. The proposed approach is field evaluated in a set of deployments in an exploratory underground mine drift in Northern Nevada.**

Fig. 1. Instance captured during an autonomous exploration mission inside an underground mine, as well as a 3D point cloud of the mine reconstructed onboard and in real-time. An indicative mission is shown at `https://youtu.be/-wE3MwNhF_4`.

I. INTRODUCTION

Autonomous robotic applications utilizing aerial robots have witnessed an expansive growth both in terms of capacity and diversity over the past decade. Aerial robots are now being deployed in a variety of critical application domains, such as disaster response [1], search and rescue operations [2], surveillance [3], industrial inspection [4] and exploration and mapping of unknown spaces [5]. One of the major desired qualities for these robotic deployments is the robot's capability to perform its assigned tasks autonomously, without need for human operator intervention, as it not only minimizes the risk to human life but also lowers the cost of operation, while ensuring repeatability of tasks. A key requirement towards enabling full robotic autonomy is to equip aerial robots with the ability to autonomously navigate through challenging environments in the absence of prior maps and external guidance. A particular case of these challenging, hard to access, difficult to navigate and previously unknown environments are subterranean tunnels, underground mines, and cave networks which have recently become a focus of the robotics community especially due to the DARPA Subterranean Challenge. These dark, dirty and dangerous settings are difficult to autonomously navigate and explore as they are often extremely long and large in scale, narrow and constrained, and present conditions of sensor degradation such as the presence of air-borne obscurants, as shown in Figure 1.

To address the needs and challenges of autonomous subterranean exploration, in this work a novel exploration path planning algorithm is presented and is tailored to the specific challenges of subterranean settings. In particular, to enable both efficient exploration along very long underground drifts and the capacity to properly handle multi-branching topologies, the presented algorithm employs a two–stage path planning architecture consisting of a rapid tree-based local exploration layer and a graph-based global planning layer. The local exploration layer utilizes a random tree of robot configurations and enables rapid exploration within an area around the current vehicle location by sampling paths that maximize local volumetric exploration. Furthermore, the local planner also ensures robot safety by remaining cognizant of the distance between the robot and its immediate surroundings and obstacles. The global planning layer utilizes a graph-based search and ensures continuous exploration of the environment by redirecting the robot to the frontiers of volumetric exploration in the global context. This step is triggered when the local planner is unable to continue exploration and thus the re-positioning of the robot to a frontier of the exploration space is needed. In addition, the global planner is also responsible of ensuring the robot's safe return-to-home by remaining aware of the battery limi-

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`

tations and continually calculating an optimized homing path to the initial take-off position. The proposed autonomous path planning architecture is thoroughly evaluated both in simulation, as well as through a set of real-world robotic field deployments in an underground mine.

The remainder of the paper is organized as follows: Section II overviews the related work. Section III formulates the autonomous robotic exploration problem, with Section IV detailing the proposed solution. A simulation-based evaluation study is provided in Section V, followed by experimental results of robotic deployments in an underground mine in Section VI. Finally, conclusions are drawn in Section VII.

II. RELATED WORK

Robotic exploration of unknown environments remains a key area of interest in the robotics community and over the years a number of methods focusing on the exploration path planning problem have been proposed [6–9]. Volumetric exploration of unknown spaces has been historically addressed by frontier-based methods [7], where the objective of the path-planning algorithm is to actively guide the robot towards the frontiers of its sensor perception range. Similarly, sampling based methods have also been employed where the objective is to sample the "next–best–view" [6] of the environment which maximizes the volumetric observation of the unknown space. More recent efforts in this domain have focused on multi-objective planning [10–12], where the planned paths are optimized for multiple objectives either simultaneously or in a cascade manner. Likewise, path planning paradigms involving teams of multiple robots for exploration have also been proposed [13, 14]. However, despite the fact that the generic problem of robotic exploration of unknown environments has been approached through a variety of techniques, only a small segment of the robotics community has focused on the specific and unique challenges of robotic exploration of subterranean environments. In this context, the pioneering works in [15], proposed topological exploration of underground mines based on the detection of intersections and exploration of edges using a ground based platform. Similarly, while the works presented in [16–18] do not propose methods specific to the exploration of subterranean environments, they do address the problem of robot path planning in narrow and constrained settings.

Motivated by the progress in the literature and challenges of subterranean environments, in this work a path planning algorithm tailored to the needs and constraints of underground environments is presented. Cognizant to the fact that subterranean settings are often of very large–scale, multi–branched and simultaneously narrow and confined, the proposed algorithm employs a bifurcated design that simultaneously enables safety-aware rapid local exploration, as well as readily–available global exploration planning and safe return-to-home capacity.

III. PROBLEM STATEMENT

The task of exploring subterranean environments, as considered in this paper, is a specific instance of the generic volumetric exploration problem in the robotics research which aims to build a complete map of a previously unknown but bounded space. Let $\mathbb{M}$ be a 3D occupancy map of the environment which is incrementally built from an on-board range sensor $\mathbb{S}$, as well as robot poses estimated online from a localization system $\mathbb{O}$. Moreover, let d_{max} and $[f_h \times f_v]$ be the maximum effective range and field of view of $\mathbb{S}$ respectively. The occupancy map $\mathbb{M}$ is organized in cubical voxels and each of them is classified as either free, occupied, or unknown. Furthermore, let us denote the simplified robot state for planning purposes as the combination of the vehicle's 3D position and heading $\xi_t = [x_t, y_t, z_t, \psi_t]^T$. Notably, the subterranean environments are often very large in scale while presenting very narrow geometries, span across multiple directions and are typically accessed from a small number of entry points (e.g. portals, cave openings); thus given the lack of prior knowledge, the boundary of the exploration space for planning purposes in such environments is assumed to be particularly large (virtually - though not strictly - "unbounded"). Furthermore, given that observations from most range sensors stop at surfaces, it follows that there will exist free space in hollow, narrow, and inaccessible parts of the environment which cannot be mapped. This leads to a residual volume portion $\mathbb{M}_{res}$ that remains unknown. Therefore, exploration is considered complete if the planner reports explored volume $V_{explored} = V_{full} \setminus V_{res}$, where V_{full} the volume within the bounds of the unknown space to be explored.

Problem 1 *(Volumetric Exploration for Subterranean Environments)* Given the model of the range sensor, the robot's dynamics constraints, and the limited flight time, find a feasible path to navigate the robot either a) towards unmapped areas to maximize its explored occupancy map or b) to return to a predefined home location when the map is complete or the battery power reaches its critical level. Cognizant to the fact that the volume to be explored is likely to exceed the capabilities of most robotic systems, in this work we emphasize performing volumetric exploration by optimizing the robot viewpoint selection process at any local step of its path planning process and given the current knowledge of known and unknown space.

IV. PROPOSED APPROACH

A. Exploration Planner Architecture

Subterranean environments usually consist of long and confined corridors connected by intersections which impose multiple challenges for exploration. In particular, the exploration algorithm must be scalable to deal with very large environments (e.g. km in length) and be responsive to perform rapid exploration under time constraints (e.g. due to limited battery). To tackle such challenges, we propose a greedy search strategy with two layers, namely a) a *local exploration layer* which focuses on exploiting the local space surrounding the robot, and b) a *global exploration layer* which performs planning within the global space.

Searching in a local space with fixed dimensions facilitates a faster response, while simultaneously ensuring that the search complexity remains independent of the scale of the environment. In the event of the local exploration layer reaching a dead-end, or any other case which prohibits the derivation of a path that allows effective exploration, the global planner is invoked to either re-position the robot towards a frontier of the exploration space or to return the robot to its homing location. For the indicative case of underground mine exploration, the local planner typically ensures that the robot keeps exploring efficiently across mine drifts, while the global planner re-positions the system to previously visited and multi-branching intersections where there is good expectation that further exploration can be efficiently continued. Overall, for planning purpose, the local exploration layer utilizes the RRT* [19] algorithm, while the global planner relies on a graph-based search method. In addition, optimized paths from the planner are executed by the on-board controller of the robot. The diagram for the proposed path planning solution is described in the Figure 2.

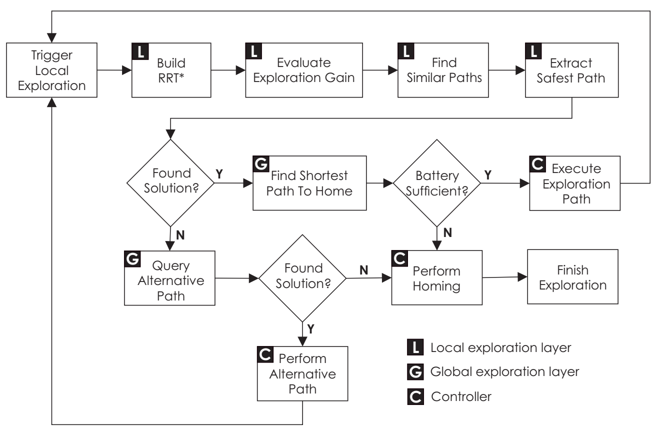

Fig. 2. Architecture of the proposed exploration planner. Blocks with letter "L" are part of the local exploration layer (although they could be re-utilized in the global layer), while "G" blocks belong to the global exploration layer.

B. Local Exploration Layer

The local exploration layer, which is considered as the front-end of the proposed exploration architecture, is designed to provide a collision-free and safety-aware path for the robot to be able to acquire the map of the environment rapidly and in a safe manner. At its core, it makes use of the sampling-based RRT* algorithm to first build a random tree in the local space surrounding the current position of the robot. The RRT* algorithm is selected given its capability to provide collision-free and minimum-length paths which are essential properties in order to achieve rapid exploration. The planner then evaluates the derived collision-free paths and identifies the branch with the highest exploration gain given the currently acquired occupancy map. Given this derivation, another refinement and selection step is performed to possibly choose a similar but safer path for exploration.

Let $\mathbb{T}_L$ be a local random tree which is built using the RRT* algorithm and Σ_L be a set of branches or paths from the root to all leaf nodes $\mathbb{T}_L$. Given a branch $\sigma_i \in \Sigma_L, i = 1...n$ and $\nu^i_j, j = 1...m_i$ are nodes along that branch, we consider an exploration gain that is implemented similarly to that of our previous contributions [5, 11]:

$$\mathbf{ExplorationGain}(\sigma_i) = \\ e^{-\gamma_{\mathcal{S}}\mathcal{S}(\sigma_i,\sigma_{ref})} \sum_{j=1}^{m_i} \mathbf{VolumetricGain}(\nu^i_j) e^{-\gamma_{\mathcal{D}}\mathcal{D}(\nu^i_1,\nu^i_j)} \quad (1)$$

where **VolumetricGain** is the total volume of unknown voxels expected to be perceived from the on-board range sensor if the robot was to be positioned at that node ν^i_j. Moreover, $\mathcal{D}(\nu^i_1, \nu^i_j)$ is the Euclidean distance from each node to the root node of the tree to penalize longer but not proportionally informative paths with a tunable factor $\gamma_{\mathcal{D}}$. In addition, let $\mathcal{S}(\sigma_i, \sigma_{ref})$ be the similarity between a path σ_i and a pseudo-straight path σ_{ref} that has the same length but is aligned with the currently estimated exploration direction ϕ_{ref}. ϕ_{ref} is simply estimated using a low-pass filter over a temporal buffer from the latest robot's poses, while the similarity between paths is calculated using the Dynamic Time Warping method (DTW) for time series signals [20]. $\mathcal{S}$ together with its tunable factor $\gamma_{\mathcal{S}}$ are empirically introduced as a direction bias to encourage the robot to maintain its exploration direction across the main underground drifts and corridors and prevent it from performing sudden back-and-forth maneuvers among branches at intersections due to temporarily high exploration gain. Additionally, this prevents the robot from entering shallow bays (e.g. muck bays, storage places, first-aid stations in underground mines) which are sufficiently mapped with a simple pass in front of the bay. Thus, after building a tree using the RRT* algorithm, the exploration gain using Equation 1 is computed for all the paths in Σ_L. Subsequently, the path with the highest gain is considered as a tentatively "good" solution σ_{good} - in essence the best derived path judging only in terms of exploration gain. However, given uncertainties in localization, mapping and control, navigation in the often constrained and narrow tunnels of subterranean environments requires additional safety considerations to ensure reliable autonomy. Towards that goal, the planner takes into account the distance from each node to its closest obstacle in the occupancy map (*distance map*: **dm**) and finds the best exploration path that not only maintains high exploration gain but also maximizes the robot's clearance from obstacles. The safety score $s_{safe}(\sigma_i)$ for each path is takes the form:

$$s_{avg}(\sigma_i) = \frac{1}{T}\sum_{j=1}^{m_i} \mathbf{dm}(\nu^i_j)\Delta t \quad (2)$$

$$s_{std}(\sigma_i) = \sqrt{\frac{1}{T}\sum_{j=1}^{m_i} [\mathbf{dm}(\nu^i_j) - s_{avg}(\sigma_i)]^2 \Delta t} \quad (3)$$

$$s_{safe}(\sigma_i) = s_{avg}(\sigma_i) e^{-\lambda_s s_{std}(\sigma_i)} \quad (4)$$

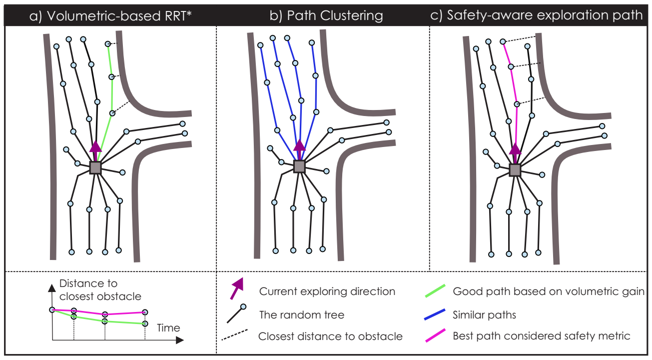

Fig. 3. Illustration of the essential steps in the proposed local exploration layer. From left to right, sub-figure a) shows an example of the best informative path (green) based on the exploration gain. Subsequently, sub-figure b) depicts the clustering of neighboring paths (blue) using the similarity DTW metric. Finally, a safety metric is considered to select a similar but safer path for exploration (pink) as shown in sub-figure c).

More specifically, given that σ_{good} is purely based on the exploration gain, the planner clusters all the paths Σ_{good} that are similar to σ_{good} using the similarity metric based on the DTW method. The path in the cluster with the best safety score is then chosen as the best exploration path σ_{best} and is subsequently executed by the controller. The whole procedure is repeated until no exploration path can be found or the platform's low-battery threshold is reached. In such cases, the global exploration layer is invoked to find an alternate path. The algorithm is illustrated in the Figure 3, while its pseudo code is provided in Algorithm 1.

Algorithm 1 Local Exploration Layer

$\xi_0 \leftarrow \mathbf{GetCurrentConfiguration}()$
$\mathbb{T}_L \leftarrow \mathbf{BuildLocalTree}(\xi_0)$ ▷ Using RRT*
$\mathbf{ComputeVolumetricGain}(\mathbb{T}_L)$
$g_{max} \leftarrow 0$
$\sigma_{good} \leftarrow \emptyset$
for all $\sigma \in \mathbb{T}_L$ **do**
  $g_\sigma \leftarrow \mathbf{ExplorationGain}(\sigma)$
  if $g_\sigma > g_{max}$ **then**
    $g_{max} \leftarrow g_\sigma$; $\sigma_{good} \leftarrow \sigma$
  end if
end for
$\Sigma_{good} \leftarrow \mathbf{FindSimilarPaths}(\mathbb{T}_L, \sigma_{good})$
$\sigma_{best} \leftarrow \mathbf{FindSafestPath}(\Sigma_{good})$
return σ_{best}

C. Global Exploration Layer

The global exploration planner plans within the globally-known explored map of the environment and has two key responsibilities. First, it is responsible to provide an alternate path towards the un-visited areas of the map in order to continue the exploration mission when the local planner cannot find any solution either because the local space has been fully explored or it is too narrow for the robot to proceed further safely. Second, it is responsible for enabling the critical return-to-home functionality. To deal with the very large scale of subterranean environments involving branching and cyclic paths, we propose to incrementally build and continually maintain a lightweight un-directed graph from the iteratively-derived local exploration paths. For this purpose the global exploration layer performs two steps to augment and expand its graph. The first step is to add the current best path σ_{best} to the global graph $\mathbb{G}_G$. In order to extend the global graph with a path σ from the local tree $\mathbb{T}_L$, the planner first introduces all nodes and edges of the current path as new vertices and edges in the graph. It then searches for nearest neighbors for each newly added vertex and forms more collision-free connections. This step ensures that the global planner can always find a feasible and shorter homing path when compared to a naive backtracking method. In the second step, the global planner attempts to add additional paths from the local tree that lead to vertices with high volumetric gain which indirectly encodes the locations and directions of unexplored branches into the global graph. To reduce the number of potential paths to be added, we first perform a simple clustering method based on the DTW similarity metric and then add only the "principal" paths from each cluster to the global graph. The principal paths are selected using the safety score as described in the local exploration layer. More importantly, leaf vertices from the principal paths are marked as potential "*frontiers*"; periodically, the volumetric gain of each frontier vertex is re-evaluated to maintain a short list of preferred frontiers which are considered targets to be visited in the global planner.

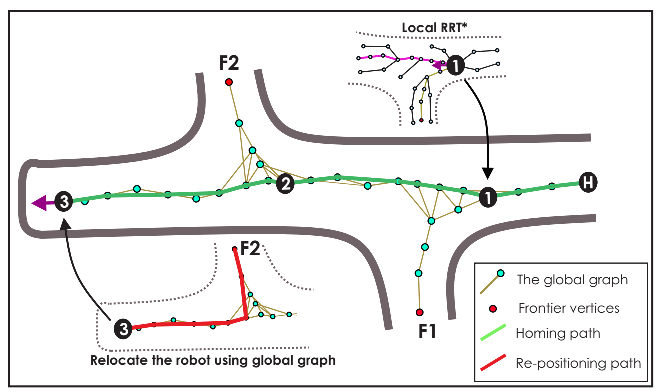

Fig. 4. A visualized instance of the proposed global exploration layer. The green path indicates an exploration path provided by the local planner. (F1) and (F2) are two potential frontier locations leading towards the unexplored branches maintained in the global graph. Once the robot reaches a dead-end at location (3), the global planner is engaged to relocate the robot to another unexplored branch (e.g. frontier F2) to continue the exploration.

For the purpose of the homing operation, the global planner utilizes the Dijsktra's algorithm to search for the shortest path from the current robot location to a predefined home location. Furthermore, to serve the second objective which is that of guiding the robot to an unexplored area to continue the exploration mission once the local planner reports inability to derive a useful path, the global planner first runs Dijsktra's algorithm to derive the shortest paths from the current vertex to all vertices that are marked as frontiers, and then selects a path based on the information gain metric described below.

Let us denote $\nu_{G,cur}$ as a vertex in the global graph representing the current robot location and $\mathcal{F} = \{\nu_{G,i}^{\mathcal{F}}\}$ be a set of available frontiers in the global graph. The problem of choosing the best path which re-positions the robot towards locations and directions from which efficient exploration can be continued is particularly complex. This is due to the fact that this decision takes place over an increasingly large search space (as the robot exploration continues) and is based on the amount of identified unexplored regions across certain directions. For example, when the robot is inside an underground mine and has traveled across multiple intersections, then all those intersections that present additional and unvisited branches are candidate "frontiers", and predicting which candidate will be proven best is very difficult. Thus, we propose an approach based on two basic principles. First, feasible paths must primarily take into account the time endurance of the robot. In particular, any feasible path must allow the robot to have sufficient remaining battery life to be able to first visit the frontier and then at least perform a homing step from that location in the worst-case scenario. Second, the best path should favor high volumetric gain areas that require a short time to arrive from the current location. To factorize these intuitive goals, the exploration gain for the global planer takes the form presented in Equation 5.

$$\begin{aligned}&\mathbf{GlobalExplorationGain}_G(\nu_{G,i}^{\mathcal{F}}) = \\ &\mathcal{T}(\nu_{G,cur}, \nu_{G,i}^{\mathcal{F}})\mathbf{VolumetricGain}(\nu_{G,i}^{\mathcal{F}})e^{-\epsilon_{\mathcal{D}}\mathcal{D}(\nu_{G,cur},\nu_{G,i}^{\mathcal{F}})}\end{aligned} \tag{5}$$

where $\mathcal{T}(\nu_{G,cur}, \nu_{G,i}^{\mathcal{F}})$ is the estimated remaining time to explore if the planner is to choose this frontier $\nu_{G,i}^{\mathcal{F}}$. This parameter is approximately calculated using the remaining endurance time ($\mathbf{RET}$) of the robot and then subtracting the estimated time of arrival ($\mathbf{ETA}$) to traverse from the current vertex to the designated vertex and from there to the home location (Equation 6). The above takes the form

$$\begin{aligned}&\mathcal{T}(\nu_{G,cur}, \nu_{G,i}^{\mathcal{F}}) = \\ &\mathbf{RET} - \mathbf{ETA}(\nu_{G,cur}, \nu_{G,i}^{\mathcal{F}}) - \mathbf{ETA}(\nu_{G,i}^{\mathcal{F}}, \nu_{G,home})\end{aligned} \tag{6}$$

where $\mathcal{D}(\nu_{G,cur}, \nu_{G,i}^{\mathcal{F}})$ is the shortest path length from the current location to the frontier, while the tunable 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 depicted in the Figure 4, while its pseudo code is provided in Algorithm 2 and Algorithm 3.

Algorithm 2 Global Planner

$\nu_0 \leftarrow \mathbf{GetCurrentVertex}()$
$\Sigma_{G0} \leftarrow \mathbf{GetDijkstraShortestPaths}(\mathbb{G}_\mathbf{G}, \nu_\mathbf{0})$ ▷ Shortest paths from current vertex to others.
$\Sigma_{GH} \leftarrow \mathbf{GetDijkstraShortestPaths}(\mathbb{G}_\mathbf{G}, \nu_\mathbf{home})$ ▷ Shortest paths from home vertex to others.
$\Sigma_{Feasible} \leftarrow \mathbf{CheckIfFeasible}(\mathcal{F})$
for all $\sigma \in \Sigma_{Feasible}$ **do**
    $g_\sigma \leftarrow \mathbf{ExplorationGain}_G(\sigma)$
    if $g_\sigma > g_{best}$ **then**
        $g_{best} \leftarrow g_\sigma$; $\sigma_{best} \leftarrow \sigma$
    end if
end for
return σ_{G_G}

Algorithm 3 Build a Global Graph by Adding Paths from the Local Tree

function BUILDGLOBALGRAPH($\mathbb{G}_G, \mathbb{T}_L$)
    $\sigma_{best} \leftarrow \mathbf{GetBestPath}(\mathbb{T}_L)$
    $\mathbf{AddToGraph}(\mathbb{G}_G, \sigma_{best})$
    $\Sigma_{principal} \leftarrow \mathbf{GetClusteredPaths}(\mathbb{T}_L)$
    for all $\sigma \in \Sigma_{principal}$ **do**
        $\mathbf{AddToGraph}(\mathbb{G}_G, \sigma)$
    end for
    return $\mathbb{G}_G$
end function

V. SIMULATION BASED EVALUATION

A simulation study was specifically designed to tune and verify the performance of the proposed approach prior to real experiments. The simulation environment (1.4km long in total), which resembles common underground mines with multiple long and narrow corridors alongside intersections and cycles, was developed in ROS-Gazebo and the RotorS simulator [21]. In addition, the local exploration layer was set to plan within a sliding local space with dimensions $length \times width \times height = 40 \times 40 \times 1m$. The endurance time limit of the robot was removed during the simulation study to allow for the exploration of the large scale environment and to verify the various behaviours of the proposed exploration pipeline, including the local exploration layer, as well as the relocation to frontiers and automated homing using the global exploration layer. During the experiment, the proposed exploration pipeline was able to completely explore and map the whole environment as shown in Figure 5. In particular, the local planner was utilized most of the time to guide the robot through the long and narrow parts of the environments, while the global planner was triggered two times to re-position the robot to unexplored frontiers. Furthermore, as clearly shown in the Figure 5, after completing the exploration of the environment the global planner also provided the shortest return-to-home path for the robot, utilizing the built global graph.

VI. EXPERIMENTAL EVALUATION

To thoroughly evaluate the performance of the proposed path planning algorithm and to demonstrate its real world application, a set of robotic field deployments were conducted at the "Lucerne" underground mine in Northern Nevada. This underground mine is an inactive exploratory mine and the

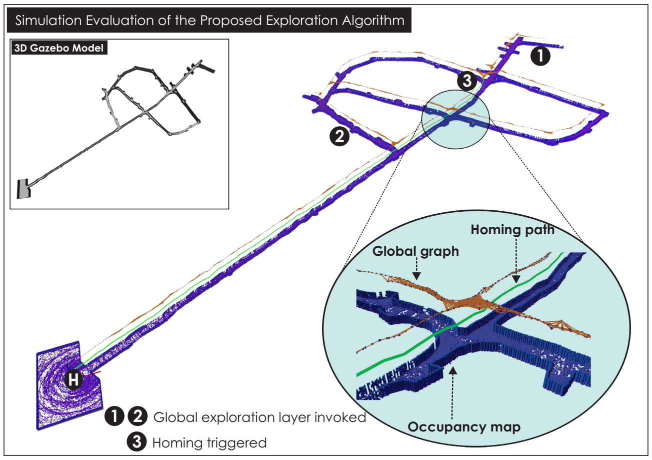

Fig. 5. Simulation evaluation of the proposed exploration approach. The top left subfigure shows the designed 3D model of the environment to be used in ROS-Gazebo. The environment was completely explored using the proposed approach as shown in the main figure. In addition, the bottom right subfigure presents an instance of the built occupancy map, the global graph as well as the safe return-to-home path.

mine structure consists of a main, long and uneven tunnel with shorter closed off sub-tunnels branching off the main tunnel. In addition, this mine provides a surface portal access allowing for the aerial robot to be deployed from outside the underground mine in order to autonomously explore it and to safely return back to the take-off position, hence demonstrating a practical example of fully autonomous robotic exploration of unknown subterranean structures. Details of the deployed aerial robot and the field evaluation studies are provided below.

A. System Overview

For the experimental evaluation of the proposed algorithm, an aerial robot based on the DJI Matrice M100 quadrotor was utilized. The robot integrated a Velodyne PuckLITE LiDAR unit for providing depth measurements with a maximum range of 100m, and horizontal and vertical fields-of-view of $F_H = 360°$ and $F_V = 30°$, respectively. The depth measurements are utilized for the estimation of robot odometry by employing the LiDAR Odometry and Mapping [22] solution, as well as for building the volumetric representation of the environment [23]. The LiDAR odometry estimates are further fused with inertial measurements provided by the autopilot of the robot and utilized by a Model Predictive Controller [24], which is responsible for the autonomous guidance of the robot along the paths generated by the proposed path planning algorithm. All localization, mapping, trajectory tracking control and path planning tasks are executed in real-time using an Intel NUC-i7 (NUC7i7BNH) computer integrated on-board the robot.

B. Autonomous Exploration of an Underground Mine

To evaluate the application of the proposed path planning algorithm to autonomously explore an underground mine, an experiment was conducted with the aerial robot being deployed from outside the mine portal entrance. During this experiment the robot autonomously entered the main mine drift and continued the volumetric exploration of the subterranean structure, until the end of its battery allowance, before safely returning to its initial take-off position. To plan local exploratory paths, a sliding local bounding volume of dimensions *length*×*width*×*height* = $40 \times 40 \times 1$ $meters$ was set. Using the online built volumetric map representation of

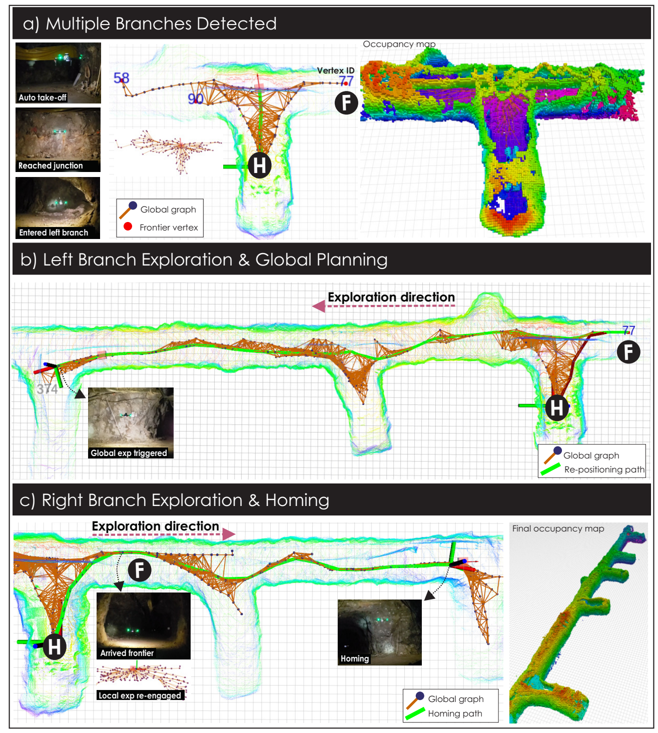

Fig. 7. Autonomous exploration in a multi-branched underground setting. The robot started at the base of a T-junction and detected two potential exploration directions when approaching the junction which were encoded into the global graph as frontier vertices. Once the left branch was completely explored, the global planner was triggered automatically to relocate the robot to a frontier positioned in the right branch. The local planner was then re-engaged to explore the right branch. Finally, based on the time endurance, the return-to-home procedure was automatically activated.

the environment, shown in the image inset (a) of Figure 6, paths are sampled within this local bounding volume. Among the feasible sampled paths, the path that maximizes the volumetric exploration of the environment is selected initially. This path is then compared with its neighboring similar paths to select the safest path for the robot to follow, as shown in image inset (b) of Figure 6 with green and pink paths highlighting the “good” and the “best” paths respectively. Following the selected best paths, at a nominal forward velocity of 0.5m/s, the aerial robot explored the underground mine before returning to its initial take-off position at a velocity of 0.75m/s. As shown in the main image in Figure 6, the path followed by the robot coincides with the general structure of the main drift of the underground mine. Similarly, due to its directional bias, the planning algorithm was able to maintain a high exploration rate along the direction of the main mine drift instead of randomly entering smaller sub-tunnels, which offered potentially small

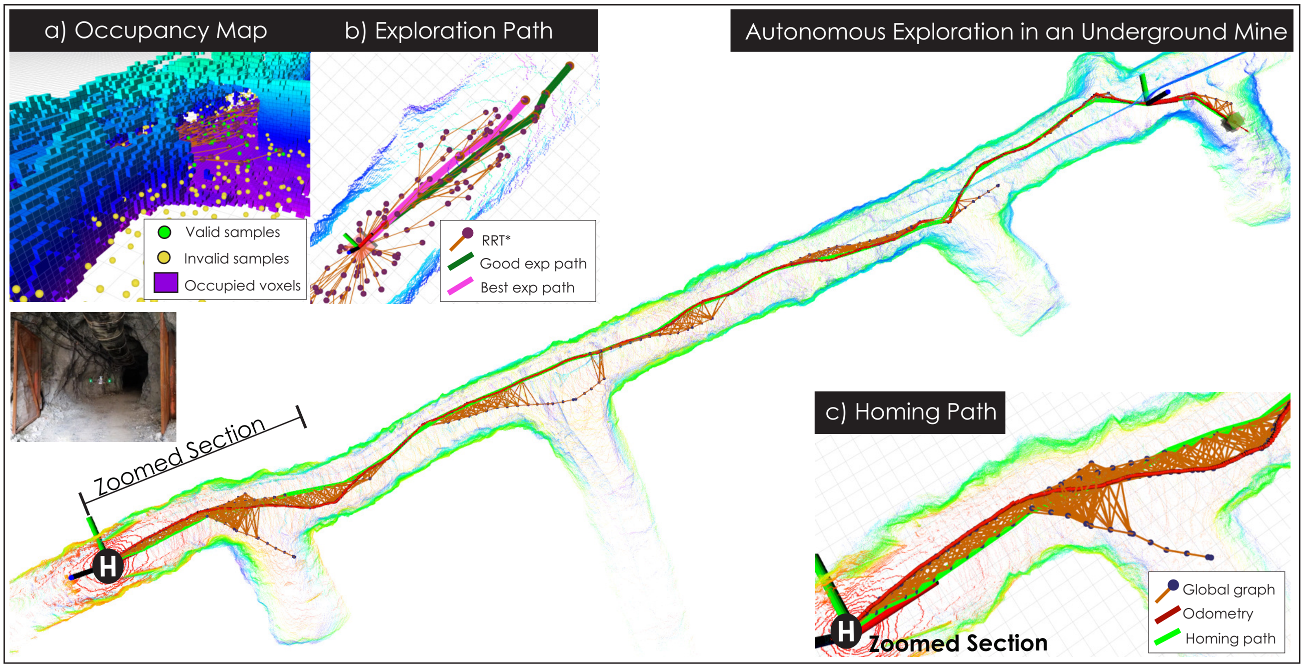

Fig. 6. Autonomous exploration of an underground mine in Northern Nevada. The robot started from the mine portal then progressively explored inside the mine. The explored distance is approximately 150m in length and the whole mission took 8 minutes in total. The homing procedure was automatically triggered and the robot came back to the entrance safely. Subfigure a) shows the occupancy map, while the subfigure b) shows an instance of results from the proposed local exploration layer. The main figure and subfigure c) present the global graph as well as the homing path.

gains for volumetric exploration. The zoomed in portion of the trajectory, shown in image inset (c) of Figure 6, demonstrates the robot trajectory when returning to the initial take-off position, and highlights that since the homing path is the re-calculated shortest path back it slightly differs from the forward path taken during the autonomous exploration phase of the mission.

C. Exploration of a Multi-branched Subterranean Structure

To demonstrate the proposed path planning algorithm's ability to effectively navigate and explore multi-branched subterranean structures, a flight experiment was conducted with the aerial robot starting its mission from the inside one of the sub-tunnels of the underground mine. By starting from inside the sub-tunnel, the main mine drift presented two branching paths for the robot to choose from in order to explore as demonstrated by the volumetric and pointcloud maps of the underground mine shown in image inset (a) of Figure 7. Once the robot reached the first junction it selected the left branch to progressively explore based on the calculation of expected volumetric exploration gain, while marking the right branch as a frontier to explore later, as shown in the image inset (b) of Figure 7. In this mission the local bounding volume was kept the same with the previous experiment, however, to demonstrate the multi-branch exploration behaviour a global bounding volume of dimensions $length \times width \times height = 15 \times \pm 70 \times \pm 5$ meters was set in order to force the robot to return to an exploration frontier after completing the exploration of one of the branches. As shown in the image inset (c) of Figure 7, after exhausting the exploration gain of the left branch, the robot returned directly to the previously marked frontier in the beginning of the right branch and re-started the local exploration procedure until it exhausted its available battery budget, at which point it returned to the initial take-off position and completed the assigned mission. The above experiments demonstrate the efficacy of the proposed planning architecture for the autonomous exploration of subterranean environments.

VII. CONCLUSIONS

In this work a new path planning approach for the exploration of subterranean structures using aerial robots was presented. The proposed approach is tailored to the risks, challenges and topology of underground mines and tunnels. The bifurcated design of the proposed method enables rapid and safe volumetric exploration of local sub-spaces through the local planning layer, while the global planning layer ensures the continuous exploration of large in scale but confined in size environments, alongside the robot's safety by deriving a battery-aware safe return-to-home path. A thorough evaluation of the path planning algorithm, using simulation studies and real-world underground field deployments, verifies its effectiveness and applicability for the exploration of dark, dirty and dangerous subterranean settings.

REFERENCES

[1] D. Thakur, G. Loianno, W. Liu, and V. Kumar, "Nuclear environments inspection with micro aerial vehicles: Algorithms and experiments," *arXiv:1903.06111*, 2019.

[2] R. Siegwart *et al.*, "Legged and flying robots for disaster response," in *World Engineering Conference and Convention (WECC)*. ETH-Zürich, 2015.

[3] 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.
[4] M. Burri, J. Nikolic, C. Hürzeler, G. Caprari, and R. Siegwart, "Aerial service robots for visual inspection of thermal power plant boiler systems," in *International Conference on Applied Robotics for the Power Industry (CARPI)*. IEEE, 2012.
[5] A. Bircher *et al.*, "Receding horizon" next-best-view" planner for 3d exploration," in *IEEE international conference on robotics and automation (ICRA)*, 2016.
[6] C. Connolly, "The determination of next best views," in *Proceedings IEEE International Conference on Robotics and Automation.*, 1985.
[7] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings IEEE CIRA'97.*, 1997.
[8] M. Popovic, G. Hitz, J. Nieto, I. Sa, R. Siegwart, and E. Galceran, "Online informative path planning for active classification using uavs," *arXiv:1609.08446*, 2016.
[9] M. Nieuwenhuisen and S. Behnke, "Search-based 3d planning and trajectory optimization for safe micro aerial vehicle flight under sensor visibility constraints," *arXiv:1903.05165*, 2019.
[10] C. Papachristos *et al.*, "Autonomous exploration and inspection path planning for aerial robots using the robot operating system," in *Robot Operating System (ROS)*. Springer, 2019, pp. 67–111.
[11] 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.
[12] 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.
[13] J. Delmerico *et al.*, "Active autonomous aerial exploration for ground robot path planning," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 664–671, 2017.
[14] 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.
[15] 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.
[16] M. Rufli, D. Ferguson, and R. Siegwart, "Smooth path planning in constrained environments," in *IEEE International Conference on Robotics and Automation*, 2009.
[17] B. MacAllister *et al.*, "Path planning for non-circular micro aerial vehicles in constrained environments," in *IEEE International Conference on Robotics and Automation*, 2013.
[18] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *Springer Robotics Research*, 2016.
[19] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.
[20] A. G. Bachrach, "Trajectory bundle estimation for perception-driven planning," Ph.D. dissertation, Massachusetts Institute of Technology, 2013.
[21] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, "Rotors—a modular gazebo mav simulator framework," in *Robot Operating System (ROS)*. Springer, 2016.
[22] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and Systems*, vol. 2, 2014, p. 9.
[23] A. Hornung *et al.*, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.
[24] M. Kamel, T. Stastny, K. Alexis, and R. Siegwart, "Model predictive control for trajectory tracking of unmanned aerial vehicles using robot operating system," in *Robot Operating System (ROS)*. Springer, 2017, pp. 3–39.