

# Motion Primitives-based Path Planning for Fast and Agile Exploration using Aerial Robots

Mihir Dharmadhikari<sup>1</sup>, Tung Dang<sup>2</sup>, Lukas Solanka<sup>3</sup>, Johannes Loje<sup>3</sup>,  
 Huan Nguyen<sup>2</sup>, Nikhil Khedekar<sup>2</sup>, and Kostas Alexis<sup>2</sup>

**Abstract**— This paper presents a novel path planning strategy for fast and agile exploration using aerial robots. Tailored to the combined need for large-scale exploration of challenging and confined environments, despite the limited endurance of micro aerial vehicles, the proposed planner employs motion primitives to identify admissible paths that search the configuration space, while exploiting the dynamic flight properties of small aerial robots. Utilizing a computationally efficient volumetric representation of the environment, the planner provides fast collision-free and future-safe paths that maximize the expected exploration gain and ensure continuous fast navigation through the unknown environment. The new method is field-verified in a set of deployments relating to subterranean exploration and specifically, in both modern and abandoned underground mines in Northern Nevada utilizing a 0.55m-wide collision-tolerant flying robot exploring with a speed of up to 2m/s and navigating sections with width as small as 0.8m.

## I. INTRODUCTION

Research in autonomous robotic exploration and mapping of unknown environments is expanding into an ever increasing set of applications. Pushing the frontiers of where robots can be utilized as explorers [1–4], first responders [5, 6], and inspectors [7, 8], aerial vehicles, in particular, are currently employed in a multitude of civilian and military applications. However, despite the unprecedented progress in the domain and the variety of exploration path planning methods proposed [9–17], the current state-of-the-art in autonomous exploration, as demonstrated experimentally, is limited to rather low-speed conservative missions as the robots try to guarantee safe navigation and simultaneous optimized selection of subsequent exploration moves given their real-time onboard localization and mapping capabilities. However, low-speed exploration does not allow the exploitation of the full agility of small flying robots and prohibits large-scale exploration due to their limited battery life.

Motivated by the above, this work proposes a Motion primitives-Based exploration path Planner (MBPlanner) that provides fast and agile paths particularly tailored to aerial robotic systems. By relying on a randomized search policy

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.

<sup>1</sup> Mihir Dharmadhikari is an undergraduate student pursuing Electronics and Instrumentation Engineering at Birla Institute of Technology and Science (BITS) Pilani Goa Campus, India f20160616@goa.bits-pilani.ac.in

<sup>2</sup> The authors are with the Autonomous Robots Lab, University of Nevada, Reno, 1664 N. Virginia, 89557, Reno, NV, USA.

<sup>3</sup> The authors are with Flyability SA, Route du Lac 3, 1094 Paudex, Switzerland. The authors have contributed equally.

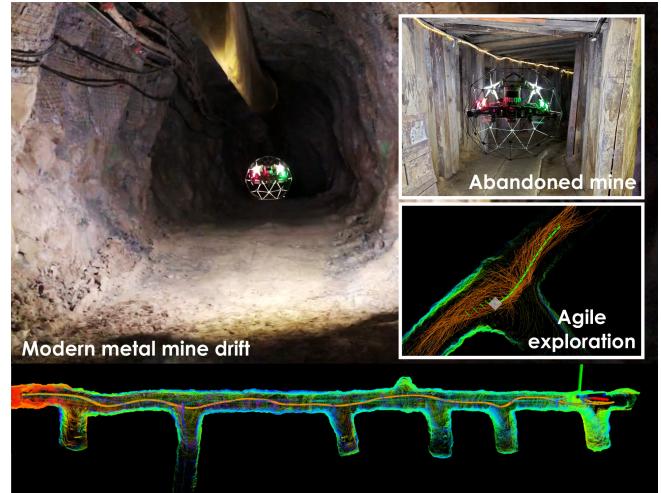


Fig. 1. Instances of fast and agile autonomous motion primitives-based exploration path planning in large-scale and narrow subterranean environments utilizing a collision-tolerant aerial robot.

based on motion primitives, the new method inherently accounts for the dynamics of the flying vehicle and thus allows for agile dynamic trajectories during the autonomous exploration mission. Additionally, in contrast with most existing methods (including previous work of the authors [13, 15–17]) the considered admissible paths do not assume a hovering condition (zero velocity) at the last vertex of each selected path but on the contrary, consider paths ending with any final velocity, as long as a deaccelerating and stopping maneuver can be identified that, if required, can bring the robot to a safe collision-free configuration in the future. As such, the sampled paths are optimized for navigation with agility and thus have the potential to maximize the length of exploration of robotic systems, especially for Micro Aerial Vehicles (MAVs) which are constrained by their limited onboard battery. Each candidate admissible path is then evaluated with respect to its exploration gain given only the bounds of the unknown area, and the best solution, refined for safety, is selected and executed, while the whole process is iteratively repeated. To further support this process of fast and agile exploration, the contributed planning method is combined with an online volumetric representation method [18] that intrinsically provides Euclidean Signed Distance Field calculations and thus facilitates rapid evaluation of the collision-free properties of each candidate edge.

The proposed approach on fast and agile exploration path planning was experimentally evaluated with an emphasis on

subterranean environments which can be simultaneously of very large scale and very narrow, thus demanding efficient exploration and agile maneuvering. More specifically, field test results are presented from a collection of underground mine environments in Northern Nevada. Combining this research with a novel collision-tolerant aerial robot design, we push the frontier of fast and agile exploration in confined settings. In fact, even the very constraint of “last resort safety”, after evaluating all possible deaccelerating paths, is relaxed in a certain experiment thus allowing to understand the limits of autonomous aerial robotic exploration and mapping in tight environments.

The remainder of this paper is organized as follows. Section II outlines related work, followed by the problem statement in Section III. The proposed approach is detailed in Section IV. Evaluation studies are detailed in Section V, followed by conclusions drawn in Section VI.

## II. RELATED WORK

Robotic exploration of unknown environments has attracted the interest of the research community and over the years a number of methods focusing on the exploration path planning problem have been proposed [9–12]. Volumetric exploration of unknown spaces has been historically addressed by frontier-based methods [10], where the objective of the path-planning algorithm is to actively guide the robot towards the frontiers of its sensor range. Similarly, sampling based methods have also been employed where the objective is to sample the “next-best-view” [9, 13] of the environment which maximizes the volumetric observation of the unknown space. More recent efforts in this domain have focused on multi-objective planning [14–16], where the planned paths are optimized for multiple objectives either simultaneously or in a cascaded manner. Emphasizing on large-scale exploration, the work in [17] proposed a graph-based local search combined with a global re-planning architecture to reposition the robot towards frontiers previously discovered along large-scale underground topologies. Likewise, path planning paradigms involving teams of multiple robots for exploration have also been proposed [19, 20]. However, despite the fact that the generic problem of robotic exploration of unknown environments has been approached through a variety of techniques, the current state-of-the-art has not presented methods and systems facilitating agile and fast exploration especially in large-scale and narrow environments, a fact contradicting known progress in fast autonomous flight [21–23]. Motivated by the above and cognizant of the fact that especially for Micro Aerial Vehicles, speed of exploration directly translates to exploration depth given their limited battery, in this work, we emphasize on an agile path planning strategy to facilitate fast exploration in simultaneously large and narrow environments. As an ideal field verification campaign, we demonstrate this work in the context of subterranean exploration and mapping with the robots operating fully autonomously and without any human intervention.

## III. PROBLEM STATEMENT

The problem of autonomous exploration, as considered in this work, is that of incrementally unveiling a bounded volume  $V \subset \mathbb{R}^3$  in a manner that is efficient and cognizant of the dynamic abilities and limitations of a robotic system. This is to determine which parts of the initially unmapped space  $V_{unm} \stackrel{\text{init}}{=} V$  are free  $V_{free} \subset V$  or occupied  $V_{occ} \subset V$ . In this work, the volume is discretized in an occupancy map  $\mathcal{M}$  that is comprised of cubical voxels  $m \in \mathcal{M}$  with edge length  $r$ . Since for most sensors – including cameras – perception stops at surfaces, sometimes hollow spaces or narrow pockets cannot be explored, thus leading to a residual volume:

*Definition 1 (Residual Volume)* Let  $\Xi$  be the simply connected set of collision free configurations and  $\bar{\mathcal{V}}_m \subseteq \Xi$  the set of configurations from which the voxel  $m$  can be perceived. The residual volume is  $V_{res} = \bigcup_{m \in \mathcal{M}} (m \mid \bar{\mathcal{V}}_m = \emptyset)$ .

The exploration problem, for a vehicle with state vector  $\mathbf{x}$  and control input  $\mathbf{u}$ , is then defined as:

*Problem 1 (Volumetric Exploration)* Given a bounded volume  $V^E$  and a dynamic model of the vehicle dynamics  $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$ , find a collision free path  $\sigma$  starting at an initial configuration  $\xi_{init} \in \Xi$  that leads to identifying the free and occupied parts  $V_{free}^E$  and  $V_{occ}^E$ , such that there does not exist any collision free configuration from which any piece of  $V^E \setminus \{V_{free}^E, V_{occ}^E\}$  could be perceived. Thus,  $V_{free}^E \cup V_{occ}^E = V^E \setminus V_{res}^E$ . Feasible paths  $\sigma$  of this problem respect the dynamic model  $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$ , are subject to the limited Field of View (FoV) of the sensor, its effective sensing distance, and robot dynamics constraints.

## IV. PROPOSED APPROACH

The proposed motion primitives-based planner (MBPlanner) is a new approach to the problem of exploration path planning that is tailored to aerial robots exploring complex, possibly narrow but large-scale environments thus requiring agility and high speed. First, the method builds on top of an online volumetric representation paradigm [18] that intrinsically provides Euclidean Signed Distance Field (ESDF) calculations and thus allows for rapid evaluation of collision-free paths. Second, the method exploits the strengths of motion primitives with respect to their ability to search the configuration space of the robot, while simultaneously offering paths that respect the vehicle dynamics and can thus optimize the navigation performance as high-speed exploration is the only way for battery-constrained Micro Aerial Vehicles to explore large-scale environments. To ensure sufficient density in searching the configuration space, every iteration of the method operates within a local window around the current robot location. Third, the method introduces the key feature of sampling paths that do not end in leaf vertices that assume zero velocity but on the contrary, arrive with a speed that can support the continuous need of fast exploration. As the sampled paths do not assume zero final velocity, admissible solutions are not only checked if they are currently collision-free but also if there is a path that

can ensure that the robot remains safe in its future trajectory given the final configuration of a certain candidate path. Finally, admissible solutions are evaluated with respect to the anticipated exploration gain, as further penalized by factors that account for the path “safety” and its alignment with previous exploration direction, and the best path is provided as a reference to the onboard controller, while the overall process is then iteratively repeated. This section details the aforementioned steps and provides insight with respect to the underlying computational cost.

### A. Algorithm Description

Algorithm 1 outlines the main steps during every iteration of the proposed motion primitives-based exploration planner. The planner assumes a rigid body model approximation of the aerial robot dynamics represented by the configuration state  $\xi = [x, y, z, v_x, v_y, v_z, \psi]$ , where  $v_*$  are velocity states and  $\psi$  the heading. Every iteration starts by acquiring the current robot configuration  $\xi_0$ , while it is noted that in order to enable fast exploration, it is not assumed that the robot has arrived at  $\xi_0$ , or will be commanded to a new final configuration with zero end velocity. Subsequently, the method utilizes control space sampling, in particular, sampling sequences of control vector values  $\mathbf{u} = [a_x, a_y, a_z]^T$  consisting of the 3D acceleration to be applied on the modeled rigid body representation of the vehicle. A tree of motion primitives-based paths is then built and the associated collision-free and future-safe branches are extracted. Subsequently, an exploration gain metric is evaluated across each of the admissible branches before the best is selected and provided as a reference to the robot controller to be tracked and followed. The following subsections provide further detail into the key functionalities of this process.

#### Algorithm 1 Motion Primitives Based Exploration Planner

```

1:  $\xi_0 \leftarrow \text{GetCurrentConfiguration}()$ 
2:  $\mathbb{T}_L \leftarrow \text{BuildLocalTree}(\xi_0)$ 
3:  $\Sigma_L \leftarrow \text{GetBranchesFromTree}(\mathbb{T}_L)$ 
4: ComputeVolumetricGain( $\mathbb{T}_L$ )  $\triangleright$  For all vertices
5:  $g_{best} \leftarrow 0$ 
6:  $\sigma_{L,best} \leftarrow \emptyset$ 
7: for all  $\sigma \in \Sigma_L$  do
8:    $g_\sigma \leftarrow \text{ComputePathGain}(\sigma)$ 
9:   if  $g_\sigma > g_{best}$  then
10:     $g_{best} \leftarrow g_\sigma; \sigma_{L,best} \leftarrow \sigma$ 
11:   end if
12: end for
13:  $\Sigma_S \leftarrow \text{GetSimilarPaths}(\sigma_{L,best})$ 
14:  $\sigma_{L,best} \leftarrow \text{GetSafestPath}(\Sigma_S)$ 
15: return  $\sigma_{L,best}$ 
```

### B. Motion Primitives for Fast Exploration

For planning purposes, we model the aerial robot as a rigid body actuated by a constrained acceleration vector  $\mathbf{u} \in \mathbf{U}$ . Let  $\{\tilde{\mathbf{u}}_{k,\mu}^p\}$  denote a motion primitive, which is a sequence of  $\mu$  functionals from an interval of time into the control space

$\mathbf{U}$  from which possible control actions may be sampled. Let the overall interval of time (for the whole  $\{\tilde{\mathbf{u}}_k^p\}$  sequence) start at  $t_k$  and stop at  $t_{k+1}$ , where  $k$  is the current planner iteration and  $t_{k+1}$  a final time that depends on the particular primitive. Then provided the currently explored subset of the map  $\mathbb{M}_E$ , alongside the current state configuration  $\xi_k$ , there exists a set  $\mathbb{U}_k^p(\xi_k)$  of motion primitives that leads to a tree  $\mathbb{T}_L$  of paths that are collision free  $\{\sigma_k^{k+1}\}$  up to  $t_{k+1}$ , and for which there exists a path  $\beta(\sigma_k^{k+1})$  that brings the robot to a safe hovering situation starting from the end configuration of  $\sigma_k^{k+1}$  as discussed in Section IV-C. For each of the  $\mu$  functionals  $\tilde{\mathbf{u}}_{k,\mu}^p$  within the  $\{\tilde{\mathbf{u}}_k^p\}$  sequence, we acquire the associated state transition  $\xi_k^\mu = \mathbf{f}_p(\xi_k, \tilde{\mathbf{u}}_k^p)$  in which  $\xi_k$  is the initial condition (the robot configuration at algorithm iteration  $k$ ), and  $\mathbf{f}_p$  represents the rigid body dynamics. Similarly, and by extension, we acquire the whole state transition from  $\xi_k$  to  $\xi_{k+1}$ . This process is visually depicted in Figure 2. As visualized, this approach allows to search the robot configuration space within  $\mathbb{M}_E$  by sampling a set of paths  $\{\sigma_k^{k+1}\}$  that respect the system dynamics and satisfy input constraints in terms of maxima and minima in the commanded acceleration. Last but not least, it is highlighted that in hope of identifying paths that simultaneously ensure efficient and fast exploration, the process of sampling the control vector  $\mathbf{u}$  is organized in two steps, with the first sampling accelerations with directions around the previous exploring direction, and the second sampling accelerations with uniformly chosen directions in order to enable the search of the full configuration space within  $\mathbb{M}_E$ . In the current implementation of MBPlanner, 25% of the sampled control vectors have biased acceleration directions, while the remaining 75% enables the planner to do comprehensive search of the available volume.

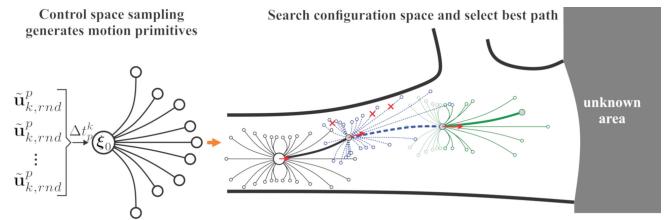


Fig. 2. Visualization of the use of motion primitives to search the robot configuration space within the currently explored map  $\mathbb{M}_E$ . Each sample control action  $\tilde{\mathbf{u}}_{k,\mu}^p$  can take a random value  $\tilde{\mathbf{u}}_{k,rnd}^p$  that reflects the commanded 3D acceleration of the robot constrained only by set minima and maxima. Starting from a certain robot configuration  $\xi_k$  and sampling a sequence of motion primitives  $\{\tilde{\mathbf{u}}_k^p\}$  the method allows to sample paths  $\sigma_k^{k+1}$  which are collision-free and for the end configuration of which there exists a path to bring the robot to a hovering configuration in a safe manner. Collision-free paths that fail to satisfy this latter requirement are not admissible, and marked with a red  $\times$  in this visualization.

### C. Future-safe Paths

As discussed, the planned paths - and thus also the selected best path  $\sigma_{L,best}$  per iteration - often do not arrive at the end configuration with zero final velocity. This, however, raises the question of whether the robot is guaranteed to be safe in the future. In order to ensure this desired constraint,

the MBPlanner further identifies if there exists a candidate path  $\beta(\sigma_k^{k+1})$  that starts from the end configuration of every admissible  $\sigma_k^{k+1}$  and safely brings the robot to a hovering configuration within  $\mathbb{M}_E$ . This safety-ensuring path is identified by first evaluating the effect of a) a maximum deacceleration in a direction opposite to that of the expected motion of the vehicle towards the end configuration of  $\sigma_k^{k+1}$ , and then b) a fixed set of (de)acceleration profiles  $\{\mathbf{a}_{\max}^{break}\}$  with different directions. Figure 3 illustrates the requirement set for an admissible path to not only be collision-free across its length but also be future-safe.

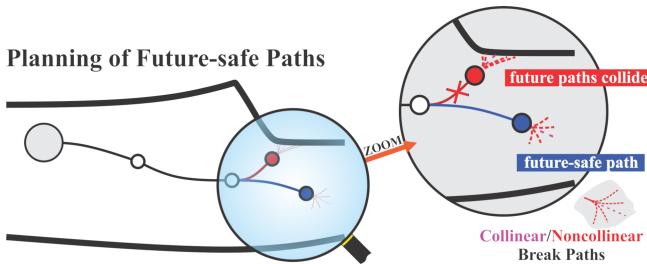


Fig. 3. Admissible paths of MBPlanner are not only collision-free across their length but also future-safe in the sense that the algorithm identifies at least one deaccelerating path that brings the robot back to a hovering condition in a safe manner within the explored map  $\mathbb{M}_E$ .

#### D. Underlying Mapping Representation

Careful consideration of the underlying mapping representation led to the selection of Voxblox [18], a volumetric mapping pipeline based on Truncated Signed Distance Fields (TSDFs). TSDFs are a common implicit surface representation widely used within the computer graphics and vision communities. TSDFs are built fast and simultaneously allow to smooth out sensor noise by aggregating and filtering over multiple observations. At the same time, Voxblox has been demonstrated to dynamically build Euclidean Signed Distance Fields (ESDFs) out of TSDFs fast (e.g. faster than Octomap [24]) and with decreased computational cost, thus facilitating rapid collision checking functionality which is essential for fast exploration. In fact, Voxblox has been shown to be 20 times faster than standard raycasting into a TSDF, and up to 2 times faster than even grouped Octomap insertions when performing merging on the publicly released EuRoC dataset [18, 25]. In Section V we provide further computational evidence that justifies the selection of Voxblox as the appropriate mapping representation framework. In fact, given the limited onboard computational resources of increasingly smaller Micro Aerial Vehicles and the simultaneous goal of fast and agile exploration, the selection of an applicable volumetric mapping method is key into achieving the envisioned performance.

#### E. Exploration Gain

Provided the sampled collision-free and future-safe paths  $\{\sigma_k^{k+1}\}$  the planner computes the associated **VolumetricGain** for every vertex of each of the  $\{\sigma_k^{k+1}\}$  paths. This corresponds to the cumulative unmapped volume that an onboard depth sensor  $\mathbb{S}$  (or a collection of sensors)

would perceive depending on the robot's configuration at each vertex. This volumetric gain is then utilized together with other decay functions related to time cost and direction to compute the exploration gain for each path. In particular, for each path  $\sigma_\ell^i := \sigma_k^{k+1}|_i$  and  $\nu_j^i \in \sigma_k^{k+1}|_i, j = 1 \dots m_i$  is a set of vertices along that path, **ExplorationGain**( $\sigma_\ell^i$ ) is computed as follows:

$$\text{ExplorationGain}(\sigma_\ell^i) = e^{-\gamma_S S(\sigma_\ell^i, \sigma_{exp}) - \gamma_T T(\sigma_\ell^i)} \sum_{j=1}^{m_i} \text{VolumetricGain}(\nu_j^i) \quad (1)$$

where  $T(\sigma_\ell^i)$  is the time cost to traverse the whole path  $\sigma_\ell^i := \sigma_k^{k+1}|_i$ . Combined with a tuning parameter  $\gamma_T$ , this decay function aims to penalize slow trajectories in order to achieve higher exploration rate. Moreover, when the robot stays near the branching points of the environment (e.g. intersections), vertices that lie at the edge of intercept branches are usually assigned with significant volumetric gains towards locally occluded areas, which intuitively prioritize paths that follow back-and-forth maneuvers between those branches to maximize the current exploration rate. This behavior, however, is sometimes undesirable in practice since it proposes unnecessary change in exploration direction leading to high jerk trajectories and also makes the planner more sensitive to small occluded areas. To eliminate such behavior in certain scenarios, the function  $S(\sigma_\ell^i, \sigma_{exp})$  and its tuning factor  $\gamma_S$  are introduced to penalize paths with high dissimilarity to the currently estimated exploration direction. The similarity metric in this case is derived utilizing the Dynamic Time Warping (DTW) method which computes the cumulative Euclidean distance between the planned path and a pseudo straight exploration path  $\sigma_{exp}$  with the same length. The direction of  $\sigma_{exp}$  is simply estimated through a low-pass filter over a temporal window of the robot's pose.

Based on the computed exploration gains as discussed, a path with the highest gain is selected  $\sigma_{high-gain}$  for further consideration. More specifically, to improve the robot's safety in high speed settings and narrow environments, the planner performs an extra check to identify a similar but safer path from  $\sigma_{high-gain}$ . This step is proposed under an assumption that the difference in exploration gain between similar paths is due to inevitable noise from the volumetric gain calculation; for instance, a simplified model of the range sensor and mapping noise. As a result, a set of neighboring paths  $\Sigma_S$  is first identified using the DTW metric. Among  $\Sigma_S$ , the path with the largest safety margin, which is the minimum distance from the path to its intermediate obstacles, is then selected as  $\sigma_{best}$  and finally conducted by the robot, while the whole procedure is iteratively repeated.

#### F. Computational Cost

The proposed planner presents reduced computational complexity, a fact largely assisted by the utilized volumetric representation framework. Assuming  $N_M$  spanned motion primitive sequences of  $N_V^{avg}$  vertices each on average and average edge length  $d_{avg}$ , the two major computationally expensive steps are related to identifying which of the paths are

collision-free and evaluating the volumetric gain across each candidate path. With Voxblox employing voxel hashing, the lookups inside the volumetric representation have a computational cost of  $\mathcal{O}(1)$  [18]. Considering a fixed discretization and map resolution  $r$ , collision checking for the candidate paths thus becomes  $\mathcal{O}(N_M N_V^{avg} d_{avg}/r)$ . Similarly, assuming a depth sensor with range  $d_{max}$ , FoV  $[F_H, F_V]$  and resolution  $[r_H, r_V]$ , the growth rate for the volumetric gain evaluation takes the form  $\mathcal{O}(N_M N_V^{avg} F_H F_V d_{max}/(r_H r_V))$ . The aforementioned growth rate functions correspond to a significant reduction compared to other methods, especially those utilizing more traditional octree-based occupancy map representations such as in [13, 17].

## V. EVALUATION STUDIES

A set of experimental and simulation studies were conducted to evaluate the performance of MBPlanner and in particular, its potential to enable fast and agile exploration even in large-scale and geometrically confined environments. The presented simulation studies involve different types of environments, while the experimental field evaluation relates to the autonomous exploration of challenging subterranean settings and specifically underground mines.

### A. Simulation Studies

In order to evaluate and fine-tune the proposed motion primitives-based planner prior to its experimental verification, two simulation studies inside a) a subway station, and b) an underground mine took place. The simulation studies are conducted using the RotorS Simulator [26], while the local planning window of the planner (the volume within which the motion primitives-based tree is built) is set to  $D_L$  of  $length \times width \times height = 40 \times 40 \times 8m$  and the robot bounding box  $D_R$  is considered equal to  $length \times width \times height = 0.6 \times 0.6 \times 0.6m$ . Both simulation studies were conducted assuming a quadrotor MAV model integrating a LiDAR sensor with  $[F_H, F_V] = [360, 30]^\circ$  and  $d_{max} = 50m$  and the exploration speed was set to 2m/s. This test, depicted in Figure 4, indicates the ability of the proposed method to explore both large-scale environments such as underground mines and buildings with multiple levels (including going up floors through the opening of stairways).

### B. Experimental Evaluation

Towards a realistic evaluation of the outlined solution for fast and agile exploration, we conducted experiments inside two underground mines in Northern Nevada, namely the “Lucerne” new mine drift in Virginia City that was more than 170m in length, alongside the abandoned “Chollar Mine” that involved extremely narrow geometry. A collision tolerant platform developed by Flyability SA is utilized for all the presented experiments. It is a prototype, equipped with a Jetson TX2 onboard processor and an Ouster 3D LiDAR. The Ouster LiDAR has  $[F_H, F_V] = [360, 30]^\circ$  and  $d_{max} = 100m$ , while the map update is only processed for the first 50m for enhanced measurement reliability. This flying robot has a

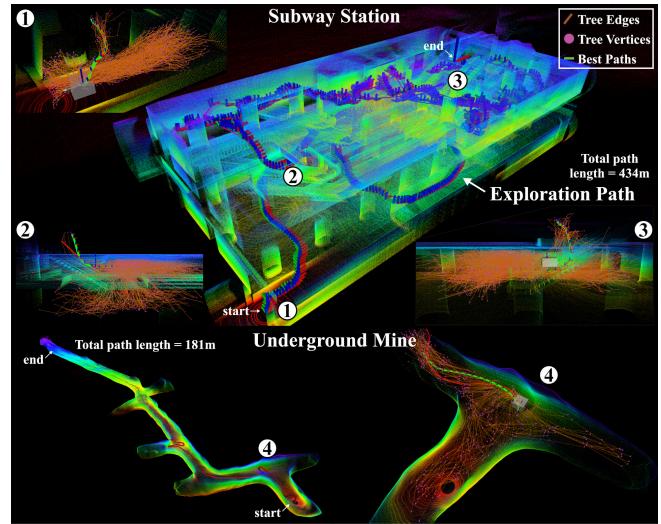


Fig. 4. Simulation-based evaluation of the MBPlanner both in a subway station and in an underground mine. Selected paths: 1) take-off and initial maneuver, 2) ascending to the next level through the opening of a staircase, 3) also ascending to the next level, 4) handling an intersection inside a mine.

diameter of 0.55m and can sustain collisions with obstacles with speeds of up to 2m/s due to its design (Figure 1).

The first field experiment is presented in Figure 5. The aerial robot is deployed outside the mine (at its portal) and is commanded to explore the whole underground drift without any assumption of a prior map and only provided general bounds of the space to be mapped. As presented, the MBPlanner provides efficient paths ensuring the quick exploration of the underground environment. In fact, the average flying speed of the robot in this experiment was 1.8m/s which is very close to its safety limit for collision tolerance and allows efficient large-scale exploration despite the necessarily small endurance of such battery-powered Micro Aerial Vehicles. Figure 6 presents the exploration rate in terms of new volume uncovered per second, while Table I presents the associated average statistics for the performance of MBPlanner in this experiment indicating the small computational cost of the method per iteration  $t_c$  in combination with its high exploration rate  $e_r$ .

TABLE I  
KEY STATISTICS FOR MBPLANNER

	$e_r(m^3/s)$	$t_c(s)$	$d_{avg}(m)$
Lucerne Mine	35.717	0.469	12.593
Chollar Mine	2.323	0.445	5.681

The second field experiment related to the deployment of the collision-tolerant aerial robot to explore the initial section of the particularly narrow (cross-sectional width between 0.8m to 1.2m) abandoned “Chollar Mine” also in Virginia City, Nevada. In this experiment, the very constraint of prohibiting any possible future collision is relaxed and we exploit the collision-tolerance of the utilized vehicle by only checking if a spherical volume with diameter 0.6m comes

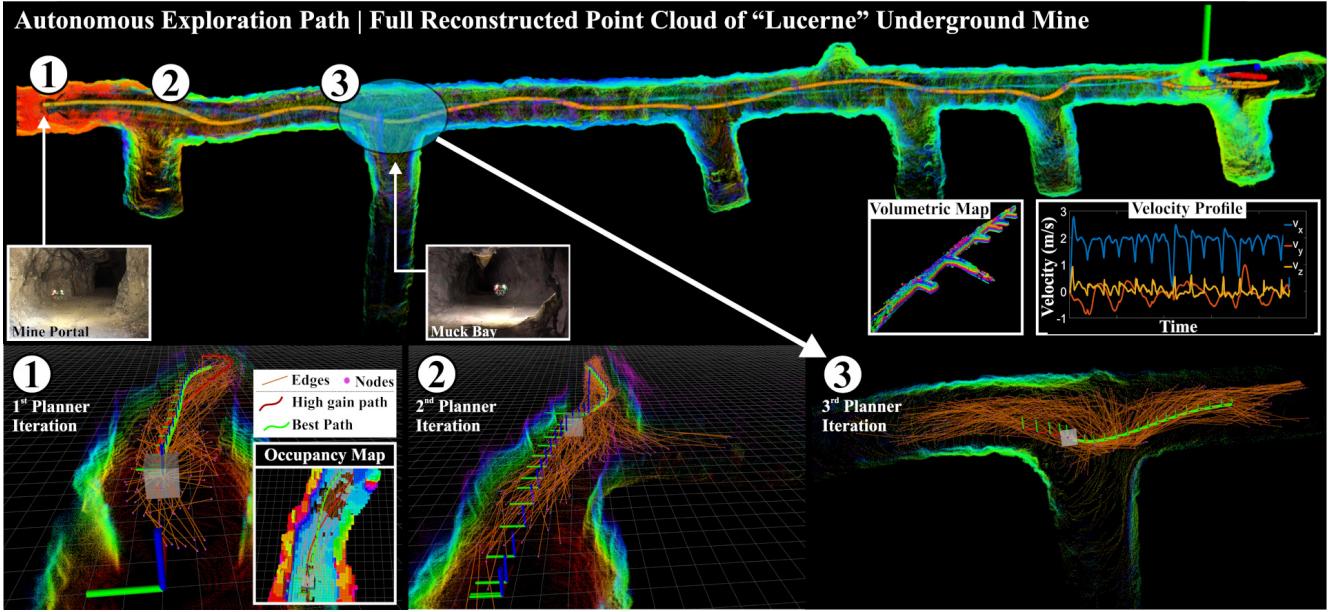


Fig. 5. Autonomous exploration mission inside an underground mine in Nevada, USA. The robot entered the mine from the main portal and then rapidly explored the whole tunnel (165m) within 1 : 30s. The nominal velocity was set to 2m/s. Sub-figure (1) presents an example of the motion primitive based random tree built by the proposed planner, as well as its best exploration path in the first iteration of the mission. Furthermore, the planner is triggered before the robot reaches the last waypoint of the currently conducted path in order to enable smooth transition between planning iterations. An indicative example of this approach is shown in sub-figure (2) which visualizes a continuous path from the second planning step appended to the current path. Finally, the top figure presents the full 3D point cloud of the environment acquired by the onboard LiDAR sensor and also the robot’s velocity plot with an average flying speed of 1.8m/s.

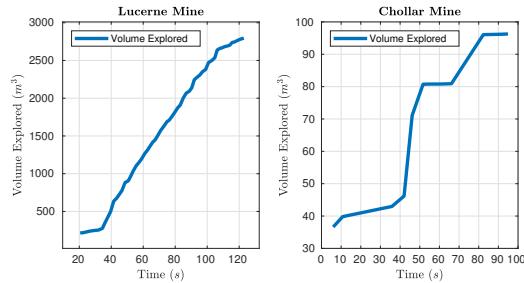


Fig. 6. Volume in  $m^3$  explored as function of time.

in collision across the planner paths. Figure 7 presents the relevant exploration result conducted with an average speed of 0.4m/s, while Figure 6 and Table I present the relevant statistics.

## VI. CONCLUSIONS

In this work, a motion primitives-based path planner for fast and agile exploration using aerial robots operating in large-scale and confined environments was proposed. The algorithm is characterized by its ability to identify paths that exploit the dynamic properties of micro aerial vehicles and thus provide high exploration rate due to appropriate viewpoint selection and fast navigation. The method is field-demonstrated in both active and abandoned underground mines using an autonomous collision-tolerant aerial robot. Future work will involve field testing in diverse settings such as metro stations, cave networks and other environments with sizes that demand fast exploration to mitigate the endurance limitations of battery-powered small aerial robots.

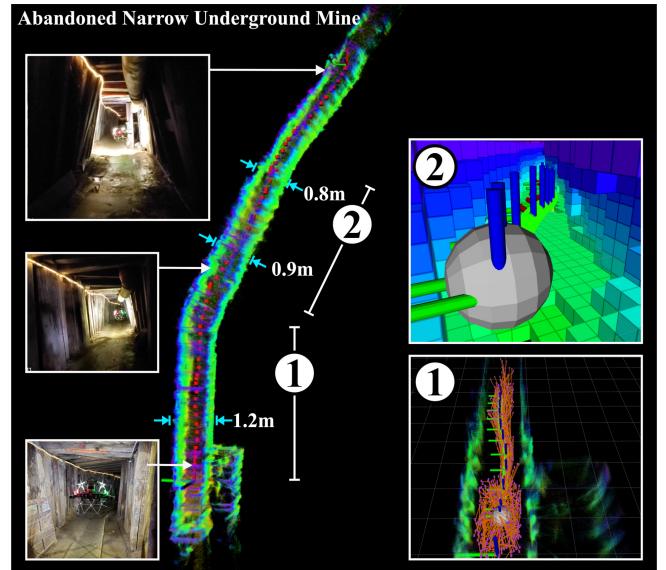


Fig. 7. Autonomous exploration inside the abandoned narrow underground Chollar mine in Virginia City, Nevada, USA. The robot was deployed at the entrance of the mine to explore the initial narrow part of the mine. The total exploration distance was 30m at a maximum speed of 0.5m/s while going through passages as narrow as 80cm. The images on the left show the robot at various instances in the exploration process, while the 3D map constructed using point cloud acquired from the onboard LiDAR of the robot is shown in the center. Sub-figure (1) on the right shows an example of the motion primitive based tree with the selected path and sub-figure (2) shows the robot (modeled as a sphere) in the occupancy map representation highlighting the extremely narrow environment. In this experiment, the collision-free paths are evaluated by modeling the robot as a sphere with 0.6m diameter and allowing future paths to come into collision with small velocities ( $< 2$ m/s) exploiting the collision-tolerance properties of the employed flying robot.

## 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] S. Khattak, C. Papachristos, and K. Alexis, "Keyframe-based direct thermal-inertial odometry," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2019.
- [3] 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*.
- [4] 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.
- [5] 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.
- [6] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. L. Grix, 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.
- [7] 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.
- [8] 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.
- [9] C. Connolly *et al.*, "The determination of next best views," in *IEEE International Conference on Robotics and Automation 1985*. IEEE, 1985.
- [10] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *CIRA'97*. IEEE, 1997, pp. 146–151.
- [11] 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.
- [12] 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.
- [20] 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.
- [13] 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>
- [14] 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.
- [15] 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.
- [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] T. Dang, F. Mascarich, S. Khattak, C. Papachristos, and K. Alexis, "Graph-based path planning for autonomous robotic exploration in subterranean environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, November 2019.
- [18] 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.
- [19] 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.
- [21] G. Loianno, D. Scaramuzza, and V. Kumar, "Special issue on high-speed vision-based autonomous navigation of uavs," *Journal of Field Robotics*, vol. 1, no. 1, pp. 1–3, 2018.
- [22] M. Beul, D. Droseschel, M. Nieuwenhuisen, J. Quenzel, S. Houben, and S. Behnke, "Fast autonomous flight in warehouses for inventory applications," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3121–3128, 2018.
- [23] K. Mohta, M. Watterson, Y. Mulgaonkar, S. Liu, C. Qu, A. Makineni, K. Saulnier, K. Sun, A. Zhu, J. Delmerico *et al.*, "Fast, autonomous flight in gps-denied and cluttered environments," *Journal of Field Robotics*, vol. 35, no. 1, pp. 101–120, 2018.
- [24] 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.
- [25] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.
- [26] 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.