2022 30th Mediterranean Conference on Control and Automation (MED)
June 28 - July 1, 2022. Athens, Greece

2022 30th Mediterranean Conference on Control and Automation (MED) | 978-1-6654-0673-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/MED54222.2022.9837293

Fast Planner for MAV Navigation in Unknown Environments Based on Adaptive Search of Safe Look-Ahead Poses

Akash Patel, Björn Lindqvist, Christoforos Kanellakis and George Nikolakopoulos

Abstract— **Autonomous navigation capability is a crucial part for deploying robots in an unknown environment. In this article a reactive local planner for autonomous and safe navigation in subterranean environment is presented. The proposed planning framework navigates the MAV forward in a tunnel such that the MAV gains more information about the environment while avoiding obstacles. The proposed planning architecture works solely based on the information of local surrounding of the MAV thus, making navigation simple yet fast. One of the biggest novelties of the article comes from solving the combined problem of autonomous navigation and obstacle avoidance. The proposed algorithm for selecting the next way point of interest also accounts in the safety margin for traversing to such way point. The approach presented in this article is also different from classical map based global planning algorithms because it favours the next way point away from obstacles in way point selection process and thus providing a safe path for incremental forward navigation. The approach is validated by simulating a MAV equipped with the proposed reactive local planner in order for the MAV to navigate in a subterranean cave environment.**

I. Introduction

In recent times, rapid exploration and mapping of unknown subterranean environments has gained significant interest in the field of robotics. MAVs have potential in being a viable solution in terms of inspection of mines [1], exploration and mapping [2],[3] and infrastructure inspection [4] due to their versatile and fast traversability. Deploying MAVs for exploration of dark, dusty and hostile cave environment is particularly challenging because of at the start of the exploration, no information about the environment is available for navigation. In terms of sensing the surrounding for safe navigation in such environments, only vision based navigation techniques are insufficient [5], therefore adapting to multiple sensors is necessary. The unstructured interior of mines and caves is a major challenge that contribute in uncertainty in measurements. The three main steps describing the autonomous navigation problem are identifying the next way point, safe navigation to the way point and localizing the robot in order to successfully build a map of the environment. In order to safely navigate in an unknown environment it is crucial that the MAV is backed by a sophisticated on board autonomy for Guidance, Navigation and Control (GNC). Having a local planning framework that feeds a safe way point to the robot such that the the exploration becomes fast by avoiding extra safety layers such as dedicated frameworks for obstacle avoidance. In such scenario by navigating to the next way point the MAV can rapidly acquire more information about the surrounding in order to incrementally build a map. The MAVs are also resource constrained vehicles because of the limited time of flight. The proposed framework compliments the idea of efficiently utilising the resources of the vehicle by making the navigation fast. This article presents a local planning and navigation framework which solves the exploration problem based on the information of unknown and free space while utilizing the information of occupied space for solving the obstacle avoidance. The point cloud reconstruction of a martian lava tube using the proposed planner is presented in Figure 1.

The authors are with the Robotics and AI Team, Department of Computer, Electrical and Space Engineering, Luleå University of Technology, Luleå SE-97187, Sweden

Corresponding Author's email: akash.patel@ltu.se

This work has been partially funded by the European Unions Horizon 2020 Research and Innovation Programme under the Grant Agreement No. 869379 illuMINEation and 101003591 NEX-GEN-SIMS.

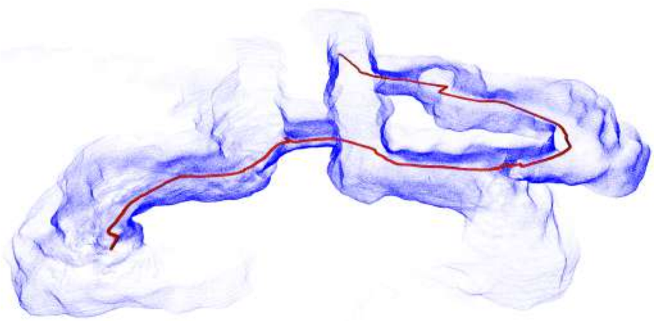


Fig. 1: Reconstruction of explored lava tube using the proposed framework

A. State of the Art

Various planning algorithms have been developed for navigation of aerial platforms in unknown environments, where in general they can be divided in map-based or memory less approaches or their combination. In [6] a hierarchical planning framework that combines map building from fused depth data as well as instantaneous depth data, both organized into separate K-D trees has been proposed. The planner relies on a slower global planner to get a goal location, which is evaluated using motion primitives against the K-D trees with the lowest cost candidate primitive to be selected. In [7] a motion planning method for fast navigation of autonomous MAVs has been developed. The algorithm divides the environment modeling in two parts, i) the deterministically visible area within the onboard sensor range and ii) the probabilistically known area beyond the sensor range from apriori map. The planning method maximizes the likelihood of reaching a goal, where a finite set of candidate trajectories are separated into groups and evaluated for collisions. In [8] a navigation method for MAVs based on disparity image processing has been proposed. More specifically, the disparity image is used for direct collision checking, incorporating C-space expansion of obstacles. The motion planning part verifies obstacle free trajectory, projecting them into the disparity image and comparing their disparity values with the C-space disparity values for collision checking. In [9] a memory-less planner that is partitioning free space in pyramids, using direct depth image measurements has been demonstrated. The use of spatial generation of pyramids of the free spaces, allows for labeling obstacle free trajectories that lie inside the pyramids, while achieving fast generation of large number of candidate trajectories and performs collision checks. In [10] the authors present a reactive navigation system for MAV exploration. The developed algorithm is based on a two layered planning architecture that leverages the global environment map for frontier generation and local instantaneous sensor data for obstacle avoidance based on artificial potential fields. In [11] "FASTER" has been developed, an optimization based planning approach for fast and safe motion in unknown environments. The planner leads to high-speed navigation by allowing to plan in known and unknown configuration space using a convex decomposition in a two trajectory design approach, a fast and a safe trajectory. In [12] a reactive navigation and collision avoidance scheme for MAVs that uses direct 2D pointcloud data from the on board sensor has been proposed. In [13] a method of autonomous navigation in confined spaces using a nmpc is presented. In [14] a collection of sensor based heading regulation methods have been proposed for aerial platform navigation along underground tunnel areas. In this work the heading regulation methods using i) image centroid calculation from either single image depth estimation, or dark area contour extraction, or CNN for dark area extraction and ii) from processing 2D lidar measurements have been described.

B. Contributions

The local planning and navigation architecture of this work is part of the development efforts within the COSTAR team [15], related with the DARPA Sub-T competition [16], while it is applicable for exploration and mapping of cave environments [17]. Based on the above mentioned state-of-the-art, the key contributions to this article are listed in the following manner.

The first contribution of this article stems from proposing an occupancy information based navigation algorithm for the MAV. As it will be presented, the navigation algorithm is developed with the goal of rapidly exploring the unknown areas and thus efficiently utilizing the resources of a MAV. The proposed approach speeds up the exploration by only focusing on the acquired information which is lying in the constrained field of view of the MAV. This enables the algorithm to iterate through less information in order to rapidly provide next way point to the MAV for fast navigation.

The second contribution in this work is combining the obstacle avoidance problem within the same computation that focuses on computing next way point that leads us to unexplored space. This is achieved by associating a cost on selecting the next way point in the trade off between how close the way point is from heading of the MAV and how far from an obstacle the way point could be such that it results in safe path to boundary between unknown and free space. Another contribution includes adapting a nonlinear model predictive control architecture with the proposed navigation framework for building overall autonomy package of guidance, navigation and control. The rest of the article is structured as follows. Section section II presents the proposed occupancy information based navigation framework with focus on generation of safe way points addressing also the obstacle avoidance problem. The section also describes the overall autonomy framework for GNC architecture. In Section section III, simulation results are presented that prove the efficacy of the proposed scheme. Finally Section section IV provides a discussion with concluding remarks of the proposed work.

II. Occupancy Information Based Local Planning

As mentioned in Section I, the proposed method uses occupancy grid maps as a mapping algorithm, which can generate a 2D or 3D probabilistic map. A value of occupancy is assigned to each cell that represents a cell to be either free or occupied in the grid. Suppose $m_{x,y}$ is the occupancy of a cell at (x, y) and $p(m_{x,y} \mid z^t, x^t)$ is the numerical probability, then by using a Bayes filter [18] the odds of the cell to be occupied can be denoted as:

$$\frac{p(m_{x,y} \mid z^t, s^t)}{1 - p(m_{x,y} \mid z^t, s^t)} = \frac{1 - p(m_{x,y})}{p(m_{x,y}} \times \frac{p(m_{x,y} \mid z^{t-1}, s^{t-1})}{1 - p(m_{x,y} \mid z^{t-1}, s^{t-1})}$$

where z^t represent a single measurement taken at the location s^t. In order to construct a 3D occupancy grid an existing framework OctoMap [19] based on octree is used in this work. In this case each voxel is subdivided into eight voxels until a minimum volume is reached. The minimum volume is defined as the resolution of the octree. In the octree if a certain volume of voxel is measured and if it is found to be occupied, then the node containing that voxel is initialized and marked as occupied. We use the the ray casting operation for the nodes between the occupied node and the sensor, in the line of ray, can be initialized and marked as free. This process leaves the uninitialized nodes to be marked unknown until the next update in the octree. As new sensor data arrives, the octree is updated based on each new measurement. Let us denote the estimated value of the probability $P(n \mid z_{1:t})$ of the node n to be occupied for the sensor measurement $z_{1:t}$ by:

$$P(n|z_{1:t}) = [1 + \frac{1 - P(n|z_t)}{P(n|z_t)} \frac{1 - P(n|z_{1:t-1})}{P(n|z_{1:t-1})} \frac{P(n)}{1 - P(n)}]^{-1}$$

where P_n is the prior probability of node n to be occupied. Let us denote the occupancy probability as P^o.

$$p^{voxelstate} = \begin{cases} Free, & \text{if } P^o < P_n \\ Occupied, & \text{if } P^o > P_n \end{cases}$$

As described earlier, the proposed approach explicitly models the occupied space in order to mark it as an obstacle for further incorporation of this information for formulating avoidance behaviour. The proposed navigation algorithm is made up of three essential modules, namely the way point generation, the optimal way point selection incorporating also collision check and finally goal publisher as presented in Figure 2.

The first module converts a laser or depth point cloud into a voxel grid V defined as $V = \{\vec{x}\}$ based in the occupancy information as discussed earlier. The next module generates potential way points based on the occupancy probability function for each voxel corresponding to each sensor measurement. The potential way points are generated at the boundary between known (free) space and unknown space. The potential way points are then fed into the next module, which evaluates each way point based on a cost function and selects an optimal way point. This point is published to the MAV's controller as a temporary goal point to visit and the overall process continues for each sensor scan until there is no unknown area is remaining.

The classical definition of frontiers [20] considers a cell in the voxel grid to be a frontier if at least one of it's neighbour cell is marked as unknown. In the real life scenario, where the area that needs to be mapped is vast, the process of checking each voxel is highly inefficient. Therefore in the proposed algorithm the number of required unknown neighbours can be set by user before the start of the mission that reduces the overall computational complexity by a significant margin. Another improvement compared to classical approach is made by not allowing any neighbouring voxel to be an occupied voxel. Due to which there are no way points are generated in extremely narrow openings (smaller than the size of the MAV). Moreover there will be no potential way points generated in the walls due to uncertainty in sensor measurements.

The fast update in the algorithm results in the process where if a previously generated way point is detected within the sphere of radius r (set by user at the start of operation), then the node belonging to that point is marked as free. This assists in limiting the undesired change in vertical velocity of the MAV if unknown points are detected above or below the MAV. A Velodyne 3D Lidar is used to to get the point cloud and because of the limited vertical field of view of the 3D Lidar, the above mentioned layer of refinement helps in an overall stable yet fast operation. The potential way points are denoted as p^f and a list containing those points is denoted as $\{F\}$, the list containing occupied points is denoted as $\{O\}$, the number of required unknown neighbours are denoted as n, the radius of hypothetical sphere around the MAV is denoted as r, The angle between way point vector $\vec{p^f}$ and MAV's heading $\vec{v}_d$ is denoted as α and the angle between occupied point vector $\vec{p^{occupied}}$ and MAV heading is denoted as β and the field of view of the MAV is denoted as θ. Let us also denote the High priority way point set as $\{\mathbb{H}\}$ and Low priority set as $\{\mathbb{L}\}$.

Algorithm 1: Way Point Generation

```
Input : m_octree : Currentoctree, n, r
Output : {F},{O}
for cell ∈ m_octree.updatedStateCells() do
    if cell.is.Free() then
        if cell.distance() < r then
            i = 0;
            for neighbours ∈ cell.getNeighbour() do
                if neighbour.isOccupied() then
                    i = 0;
                    break;
                else
                    i = i + 1
            if i >= n then
                {F}.add(cell);
    else
        {O}.add(cell);
```

In Figure 3, $(X_{\mathbb{B}}, Y_{\mathbb{B}}, Z_{\mathbb{B}})$ represent the body fixed coordinate frame and $(X_{\mathbb{W}}, Y_{\mathbb{W}}, Z_{\mathbb{W}})$ represent the world coordinate frame. The pseudo code for the potential way point p^f generation from occupancy information is depicted in the Algorithm algorithm 1. The cell which has at least n number of unknown neighbours is considered to be a potential way point and is marked as p^f at the same distance as the sensor range set at the beginning of navigation. In order to reduce the number of cells to be examined in each iteration, only the cells whose state (free or occupied) is changed are examined. The generated way points are added to a list and are examined in each update if they still satisfy the conditions of being a potential way point. In a negative return for such condition the way points are deleted from the list. Based on the α and θ angles with respect to the MAV's heading the way points are added to the high and low priority list as presented in the algorithm algorithm 2.

In order to further speed up the way point selection process, the $p^f \in \{\mathbb{H}\}$ are checked and if they obey the safety margin, then such p^f is considered as optimal way point WP and is sent to the controller directly. For a large environment it is computationally inefficient to go through each point, while evaluating the optimal way point to visit. Therefore, in the proposed method, the evaluation is done high and low priority order. When $\nexists\ p^f \in \{\mathbb{H}\}$, a $p^f \in \{\mathbb{L}\}$ is selected based on the cost function.

The pseudo code for the p^f validation as well as High and Low priority list extraction is presented in Algorithm algorithm 2. $\forall\ p^f \in \{\mathbb{H}\}$ are computed for extracting p^f such that $\alpha \in [-\pi, \pi]$ is minimum. The way points from occupancy information are generated in the world frame $(\mathbb{W})$ but the frontier vector $\vec{p_f}$ is calculated relative to the position of the MAV. As shown in Figure 3, the angle α with respect to $\mathbb{B}$ is defined as:

$$\alpha = cos^{-1}\left(\frac{\vec{p^f} \cdot \vec{v_d}}{|\vec{p^f}| \cdot |\vec{v_d}|}\right)$$

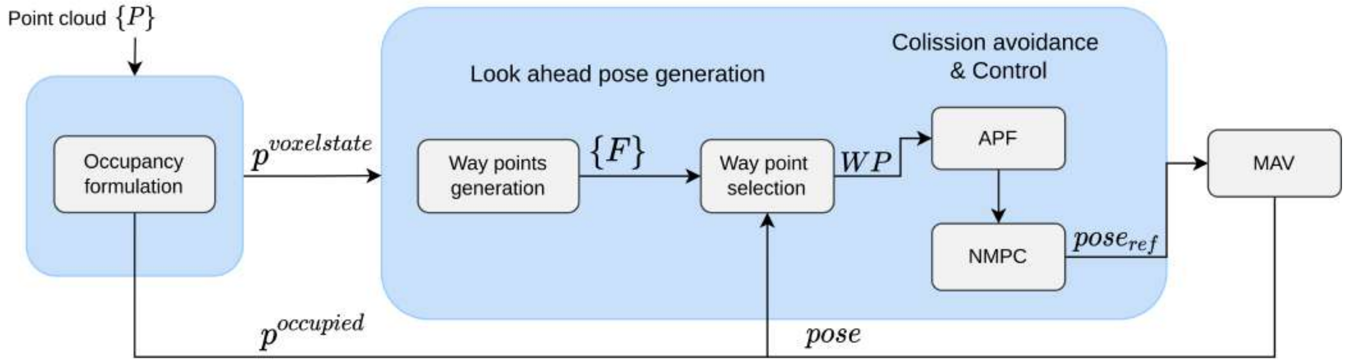


Fig. 2: Navigation framework

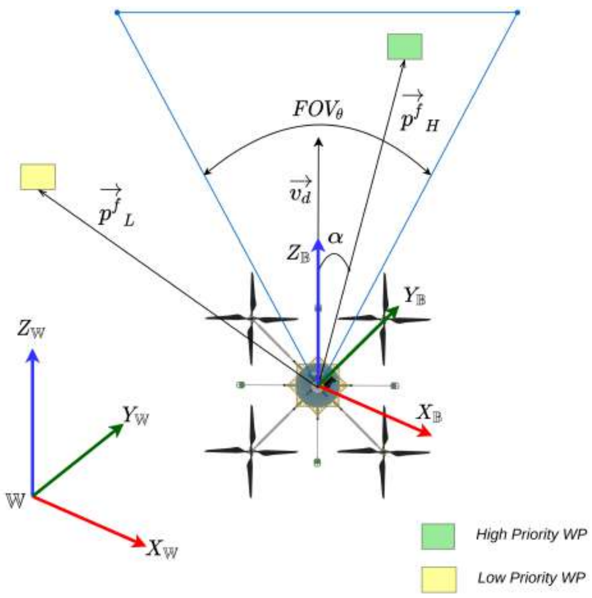


Fig. 3: Way point priority assignment with respect to MAV heading

Algorithm 2: Way Point Selection

Input : $\{F\}$
$n, r, \alpha, \beta, \theta$
Output : $\{\mathbb{H}\}$, $\{\mathbb{L}\}$
for $cell \in \{F\}$ **do**
  if $cell\_distance < r$ **then**
    $i = 0$;
    for $neighbours \in cell.getNeighbour()$ **do**
      if $neighbour.isOccupied()$ **then**
        $i = 0$;
        $break$;
      else
        $i = i + 1$;
    if $i < n$ **then**
      $\{F\}.remove(cell)$;
    for $p^f \in p^{valid}()$ **do**
      if $\alpha < \theta/2$ **then**
        $p^f\_\{\mathbb{H}\}.add(cell)$
      else
        $p^f\_\{\mathbb{L}\}.add(cell)$

1) Obstacle Avoidance: As discussed previously the algorithm algorithm 1 also outputs a list of occupied cells which has occupancy probability P^o higher than 0.5 thus considering the the cluster of such cells as an obstacle. Considering the obstacle vector angle β a cost function is formulated in a following manner.

$$\mathbb{C}(\vec{p}_{obs}, \vec{p}_f, \vec{v}_d) = \overbrace{\frac{1}{\mathcal{W}_o\sqrt{(p^f_x - p^{obs}_x)^2 + (p^f_y - p^{obs}_y)^2}}}^{\text{Avoidance cost}} + \tag{1}$$

$$\overbrace{\mathcal{W}_h||sin^{-1}(\frac{\vec{p}_f \times \vec{v}_d}{|\vec{p}_f||\vec{v}_d|})||}^{\text{Heading cost}} \tag{2}$$

Where, $\mathcal{W}_o$ and $\mathcal{W}_h \in \mathbb{R}$ are defined as weights associated to avoidance and heading cost respectively. When $\exists\ p^{occupied} \in FOV(\theta)$ an avoidance cost as described in Equation 2 is calculated for all $p^f \in FOV(\theta)$ and a way point for which the cost ζ is minimum, is selected as next best way point to visit.

$$\zeta = \underset{\{p^f \in \{\mathbb{H}\}\}}{\arg\min}\ \mathbb{C}(\vec{p}_{obs}, \vec{p}_f, \vec{v}_d) \tag{3}$$

Such selection process allows a reasonable trade off between how far the way point is selected from an obstacle as well as reducing the changes in heading of the MAV while navigating. The clear interpretation of the proposed way point selection is presented in Figure 4. Figure 4 shows the RVIZ view of the MAV with respect to the body fixed frame $\mathbb{B}$. The yellow voxels represent potential way points $p^f \in \{\{\mathbb{H}\}, \{\mathbb{L}\}\}$. The cyan voxels represent the occupied cells or obstacles. Based on the above mentioned selection process as formulated in Equation 2 and Equation 3, the next best way point is presented as Red voxel computed using the proposed approach.

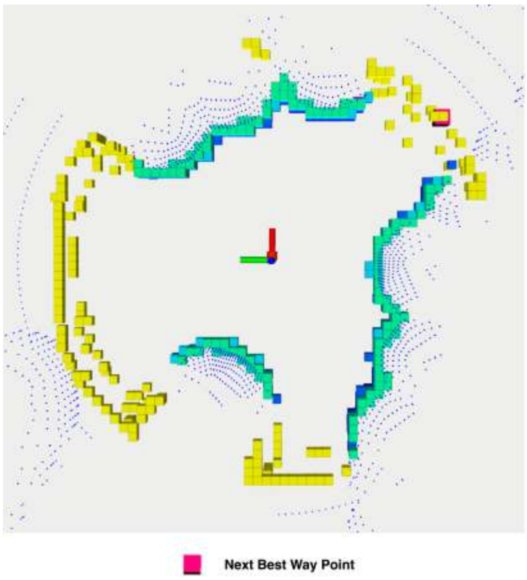


Fig. 4: Validation of safe way point selection process

A. Artificial Potential Fields based Collision avoidance

This work utilizes a purely reactive 3D Artificial Potential Field (APF), described in more detail in [3], relying only on the instantaneous pointcloud stream from the 3D lidar. It is based on letting each point in the lidar pointcloud within a specified volume defined by radius r_F result in a repulsive force, and then summing all such point-forces to get the total repulsive force. Let us denote the local point cloud generated by the 3D lidar as $\{P\}$, where all points are described by a relative position to the lidar as $\rho = [\rho_x, \rho_y, \rho_z]$. Also denote the repulsive force as $F^r = [F^r_x, F^r_y, F^r_z]$. Denote the list of points $\boldsymbol{\rho}_r \in \{P\}$, where $|| \rho^i_r || \leq r_F$ and $i = 0, 1, \ldots, N_{\rho_r}$ (and as such N_{ρ_r} is the number of points to be considered for the repulsive force). The repulsive force is:

$$F^r = \sum_{i=0}^{N_{\rho_r}} L(1 - \frac{|| \rho^i_r ||}{r_F})^2 \frac{-\rho^i_r}{|| \rho^i_r ||}$$

where L is the repulsive constant. The attractive force F^a can be seen as the vector from $\hat{p}^{\mathcal{B}}$ to the next best way point WP as $F^a = WP^{\mathcal{B}} - \hat{p}^{\mathcal{B}}$. Additionally, we impose saturation limits on the magnitude of forces, the rate of change of forces and also normalizing the resulting total force F as to always generate forces of the same magnitude and to enforce stable and non-oscillatory flight behavior. The output is the obstacle-free position reference given to the NMPC as $p^{\mathcal{B}}_{ref} = F + \hat{p}^{\mathcal{B}}$, where $p^{\mathcal{B}}_{ref}$ are the first three elements of x_{ref}, such that $x_{\text{ref}} = [p^{\mathcal{B}}_{ref,x}, p^{\mathcal{B}}_{ref,y}, p^{\mathcal{B}}_{ref,z}, 0, 0, 0, 0, 0]^\top$.

III. Simulated Exploration Mission Results

The proposed planning framework is based on computing next best way points in a safe yet fast manner. Using the Mars Coaxial Quadrotor [21] for simulations we present a realistic scenario where the proposed planner directly benefit in optimally utilizing the resources of such vehicle. In order to validate the proposed framework, the simulations are performed using the Rotors simulator [22] in ROS (Robot Operating System) and Gazebo Environment.

In Figure 5, the rapid exploration of a lava tube structure using the proposed framework is presented. In Figure 5, it is evident that the proposed planning algorithm computes the next best way point WP such that the WP lies away from obstacle (walls in this case) but still lies close enough from the current heading of the MAV. As the MAV moves forward the occupied space is modelled in each iteration in order to build a map of the environment as presented in Figure 5.

In Figure 5 the modelled occupied space is sparse due to the high resolution of Voxels in occupancy formulation for faster update. The used 3D LiDAR also has limited field of view and thus the occupied points in positive Z direction are not considered for modelling. Figure 5 show the nature of rapid yet safe nature of exploration using the proposed approach. The navigation nature of the proposed approach gives comparable results to the state of the art exploration frameworks including dedicated global path planning algorithms. Figure 6 show the reconstruction of explored parts of lava tube using the proposed framework.

IV. Conclusions

In this work we have presented a reactive local planning method for autonomous navigation in subterranean environments. The proposed approach adaptively searches for the next best way point to visit such that the speed of exploration can be maintained. The adaptive search for such points also takes into account the obstacle avoidance problem within the search such that the MAV can maintain it's heading while avoiding potential collisions. Extensive simulation results have been presented that demonstrate the deployment of the proposed planner in relevant cave environment for autonomous obstacle-free flight without apriori knowledge of the environment.

References

[1] C. Kanellakis, S. S. Mansouri, G. Georgoulas, and G. Nikolakopoulos, "Towards autonomous surveying of underground mine using mavs," in *International Conference on Robotics in Alpe-Adria Danube Region*. Springer, 2018, pp. 173–180.

[2] B. Lindqvist, A.-a. Agha-mohammadi, and G. Nikolakopoulos, "Exploration-rrt: A multi-objective path planning and exploration framework for unknown and unstructured environments," *arXiv preprint arXiv:2104.03724*, 2021.

[3] B. Lindqvist, C. Kanellakis, S. S. Mansouri, A. akbar Agha-mohammadi, and G. Nikolakopoulos, "Compra: A compact reactive autonomy framework for subterranean mav based search-and-rescue operations," 2021.

[4] S. S. Mansouri, C. Kanellakis, E. Fresk, D. Kominiak, and G. Nikolakopoulos, "Cooperative uavs as a tool for aerial inspection of the aging infrastructure," in *Field and Service Robotics*. Springer, 2018, pp. 177–189.

[5] T. Özaslan, G. Loianno, J. Keller, C. J. Taylor, V. Kumar, J. M. Wozencraft, and T. Hood, "Autonomous navigation and mapping for inspection of penstocks and tunnels with mavs," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1740–1747, 2017.

[6] M. Ryll, J. Ware, J. Carter, and N. Roy, "Efficient trajectory planning for high speed flight in unknown environments," in *2019 International conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 732–738.

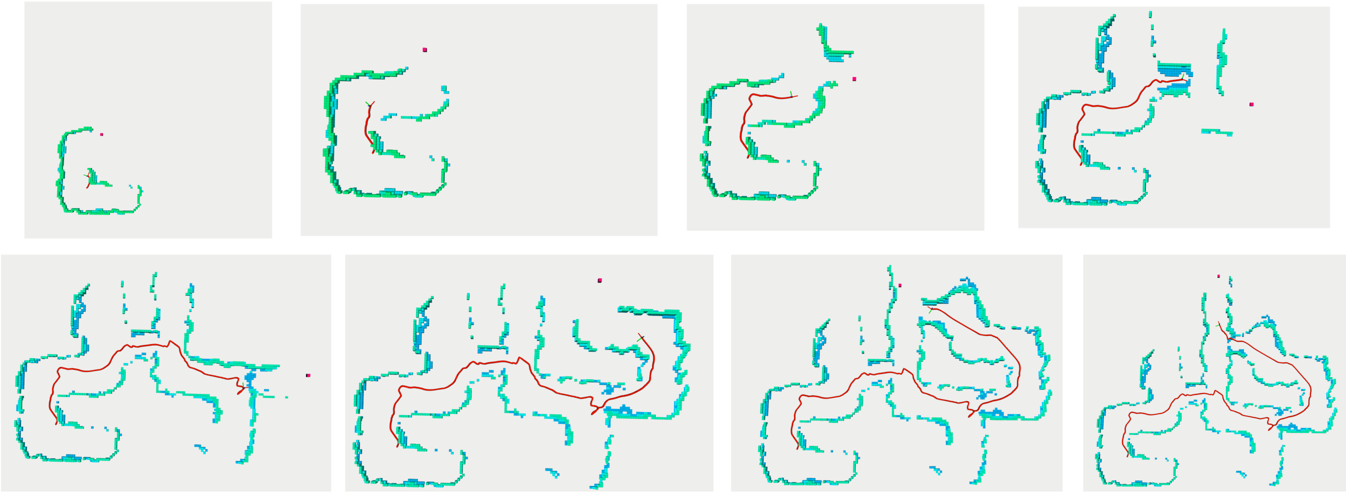
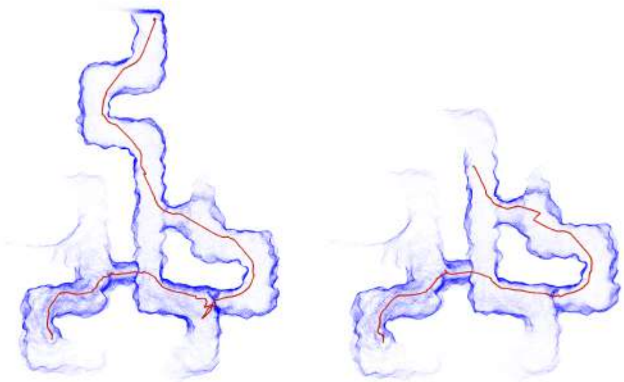


Fig. 5: Incremental lava tube exploration using the proposed planner



(a) (b)

Fig. 6: Reconstruction of the explored environment

[7] J. Zhang, C. Hu, R. G. Chadha, and S. Singh, "Maximum likelihood path planning for fast aerial maneuvers and collision avoidance," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 2805–2812.

[8] L. Matthies, R. Brockers, Y. Kuwata, and S. Weiss, "Stereo vision-based obstacle avoidance for micro air vehicles using disparity space," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 3242–3249.

[9] N. Bucki, J. Lee, and M. W. Mueller, "Rectangular pyramid partitioning using integrated depth sensors (rappids): A fast planner for multicopter navigation," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4626–4633, 2020.

[10] S. Ahmad, A. B. Mills, E. R. Rush, E. W. Frew, and J. S. Humbert, "3d reactive control and frontier-based exploration for unstructured environments," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 2289–2296.

[11] J. Tordesillas, B. T. Lopez, M. Everett, and J. P. How, "Faster: Fast and safe trajectory planner for navigation in unknown environments," *IEEE Transactions on Robotics*, 2021.

[12] B. Lindqvist, S. S. Mansouri, J. Haluška, and G. Nikolakopoulos, "Reactive navigation of an unmanned aerial vehicle with perception-based obstacle avoidance constraints," *IEEE Transactions on Control Systems Technology*, 2021.

[13] E. Small, P. Sopasakis, E. Fresk, P. Patrinos, and G. Nikolakopoulos, "Aerial navigation in obstructed environments with embedded nonlinear model predictive control," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 3556–3563.

[14] C. Kanellakis, S. S. Mansouri, M. Castaño, P. Karvelis, D. Kominiak, and G. Nikolakopoulos, "Where to look: a collection of methods formav heading correction in underground tunnels," *IET Image Processing*, vol. 14, no. 10, 2020.

[15] A. Agha, K. Otsu, B. Morrell, D. D. Fan, R. Thakker, A. Santamaria-Navarro, S.-K. Kim, A. Bouman, X. Lei, J. Edlund *et al.*, "Nebula: Quest for robotic autonomy in challenging environments; team costar at the darpa subterranean challenge," *arXiv preprint arXiv:2103.11470*, 2021.

[16] DARPA. (2020) DARPA Subterranean (SubT) challenge. Accessed: February 2021. [Online]. Available: https://www.darpa.mil/program/darpa-subterranean-challenge

[17] G. Nikolakopoulos and A. Agha, "Pushing the limits of autonomy for enabling the next generation of space robotics exploration missions," *Computer*, vol. 54, no. 11, pp. 100–103, 2021.

[18] Z. Chen *et al.*, "Bayesian filtering: From kalman filters to particle filters, and beyond," *Statistics*, vol. 182, no. 1, pp. 1–69, 2003.

[19] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous robots*, vol. 34, no. 3, pp. 189–206, 2013.

[20] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*. IEEE, 1997, pp. 146–151.

[21] A. Patel, A. Banerjee, B. Lindqvist, C. Kanellakis, and G. Nikolakopoulos, "Design and model predictive control of mars coaxial quadrotor," 2021.

[22] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, *Robot Operating System (ROS): The Complete Reference (Volume 1)*. Cham: Springer International Publishing, 2016, ch. RotorS—A Modular Gazebo MAV Simulator Framework, pp. 595–625. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-26054-9_23