Autonomous 3D Reconstruction with UAVs

Elim Yi Lam Kwan (ylk25)

***Abstract*—Unmanned Aerial Vehicles (UAVs) are widely used for documenting and monitoring topography. Moreover, conventional transmission-based remote controller are range-limiting and it becomes a challenge to explore large, unknown, and unstructured environments with them. Numerous studies have been carried out to automate the exploration task and one of the most common approaches is to use Frontier Exploration. However, it is subject to the local minima problem and in the field of robotics, this is often mitigated by introducing stochasticity to the system. Therefore, in this project, our objective is to investigate the impact of randomness on frontier guided exploration algorithms, and compared it with respect to the random walk (baseline) and non-autonomous approach. We have achieved 11.13% percentage increase in mesh coverage comparing to the baseline using Frontier Exploration with Randomness in a single target environment. We also concluded that stochasticity is useful for a less congested environment, whereas more complicated structures would require more guidance during exploration. Finally, we hope that our work could provide insights on developing solutions and frameworks for navigation challenge with exploration objective. Code and videos are available at the following link [1]**

I. Introduction

Unmanned Aerial Vehicle (UAV) photogrammetry has a wide range of applications, from archaeology [1] to civil engineering [2], and to earth science [3]. They are mainly used for acquiring topographic data, which are keys to many studies. UAVs have several favourable features, for instance, low flight height allows high-resolution data to be collected and lengthy experiments that span across phenological stages could be conducted [4]. This is credited to the reduction in aerial remote sensing cost. Moreover, the challenge of exploring large, unknown, and unstructured environments remains because transmission-based remote controllers may have limited range, which makes them unsuitable for scanning a large area. An automated solution is required and the most straightforward approach is to program the quadrotor to followed predefined trajectories. This solution requires the least onboard processing power, yet, the downside is it does not generalise well to unstructured environments. Hence, in this project, we are interested in investigating automated adaptive solutions for exploration tasks, which includes autonomous navigation and next best view planning.

Next-best-view algorithms refer to planning our next step such that they optimise our information gain regarding the world. One of the most popular algorithms is Frontier Exploration [5]. The idea behind is that the agent will actively transverse to the boundary of known and unknown space to explore the map. Moreover, it is known that frontier exploration algorithms may occasionally suffer from local minima problem, as it tends to explore the closest frontier point only [6]. Therefore, in this project, we would like to examine the frontier guided approach and investigate the impact of stochasticity on the algorithms under different environment. Moreover, we hypothesis that, similar to other control systems, in a more complicated environment, more guidance is still required (i.e. limited stochasticity).

The major achievements in this project are:

- Generated a full pipeline for UAV 3D Reconstruction, from sensor coordinate frame transfer, autonomous navigation to next-best-view planning
- As a proof of concept, we also quantified the gain of introducing frontier prioritisation in quadrotor exploration, which outperformed our random walk baseline by 11.13%
- Identified the performance differences between guided and unguided solutions, autonomous and non-autonomous solutions

II. Related Work

In this section, we will analyse related work on 3D reconstructions using UAVs, environment parameterisation, algorithms for autonomous navigation and next-best-view planning.

A. 3D Reconstruction

In 3D reconstruction, the spatial and geometric relationships of the target can be determined with multiple viewpoint images or depth map. If multiple sensors are available, stereo vision techniques [7] can be applied. It compares information about the scene from 2 viewpoints and compute depth information by examining the relative positions of objects in the two panels. Alternatively, one can also utilise the mobility of the drones to capture scenes from various viewpoints and extract spatial information using structure-from-motion algorithms [8]. The advantages are it only requires a low-cost RGB camera, moreover, the complexity of the algorithms and low execution speed refrain it from being applied commercially. Depth data can also be obtained directly from range sensors, which calculate distance from obstacle using the time it takes for the outgoing pulse to return to the source. Common examples are Light Imaging Detection and Ranging (LIDAR) sensors and RAdio Detection And Ranging (RADAR) sensors. Their major difference is LIDAR sensor uses laser light pulses, whereas RADAR sensor uses radio wave. Since laser light waves are shorter than radio waves, LIDAR is capable of generating a more precise representation of the world. The trade-off is RIDAR perform better in a harsh operating environment and is less expensive. Moreover, since the project focused on simulation and lab-controlled environment, LIDAR was adopted for our application.

[1] https://github.com/elimkwan/ROS-Structure-From-Motion

B. Environment Parameterisation

LIDAR sensors allow us to sample the continuous space with laser beams and the scans can be translated to data points in X, Y and Z coordinates with corresponding probabilities of occupancy. These sets of points are referred to as point cloud. To enable navigation and planning, maps are generated. In general, users will use grid or topology maps depending on the applications. For instance, in 2D planning, occupancy information could be stored in Manhattan distance 8-connected grids. In this project, we enhanced the grid to 3D and upgraded the grid structure such that each location is associated with a dictionary string. This forms our 3D Occupancy Grid. However, more efficient data structures which leverage the topological approach to represent 3D environment exist. OctoMap [9] is an example of it, it utilised octrees structure to form a multi-resolution 3D Occupancy Grid. In this project, we have utilised it to visualise the results of our exploration algorithms. Due to the time constraints, it was not integrated into our planning algorithms but is recommended for future work.

C. Autonomous Navigation

The basis of autonomous navigation is collision avoidance. General approaches include artificial potential field method, sampling-based method and graph-search method. Maps could be transformed into potential field [10], where obstacle and goal have the highest and lowest potential on the map respectively. However, this method is prone to local minima. Alternatively, a sampling-based approach such as Rapidly exploring Random Trees (RRT) [11] could be used. We first sampled a random position and connected it to the nearest existing tree node given that it satisfied the collision check. The tree would tend to expand to unexplored areas due to random sampling and would stop once the goal location is found. Moreover, RRT is better suited for continuous space planning. Since our defined occupancy grid is quite sparse, the conventionally graph-search method would be more applicable. Hence, a standard AStar pathfinding algorithm was used in this project.

D. Next-Best-View Planning

The naive implementation would be to use the random walk algorithm, which instruct the agent to transverse to the random regions that belong to the free space. This model has been applied as our baseline. An improvement of it would be to use frontiers to guide exploration. Frontier are regions on the boundary between open space and unexplored space. B. Yamauchi [5] proposed that the agent should transverses to the closest frontier whenever possible, such that unexplored space could be scan. Under this scheme, the robot can constantly increase its knowledge of the world. Our scheme is similar to the work of B. Yamauchi, but instead of applying a depth-first search, we used AStar to extract the collision-free path for transversing to the frontier. The reason behind this is it is better suited for the structure of our discretised 3D occupancy grid. In the future, we can also explore the impact of using various pathfinding methodologies. Other more sophisticated method includes optimising the path with partial scene geometry [12], but due to the time constraint of the project, it was not being investigated.

III. Methodology

In this section, an overview of the framework and implementation details of different modules in the pipeline will be discussed.

A. Overview

Fig. 1: Overview of the Framework

The overall perception-action loop consists of several modules as summarised in Figure 1. Firstly, the laser scan data were transformed into point cloud representation. Moreover, data were transfer from 'laser0_frame' to the 'world' coordinate frame, which all the planning and mapping were based on. Afterwards, a 3D occupancy grid was constructed based on the occupancy information from the point cloud and laser scan. Then, the next-best-view algorithms were used to decide on where to travel next and a motion plan could be generated by the path planning algorithms. Finally, Proportional Controller, also known as P-Controller, was used for velocity scaling and considerations for the holonomicity of the system will be discussed in Section III-E.

B. Perception

Transforming laser scan data to point cloud representation is the starting point of the project. We emulated the effect of an onboard LIDAR sensor with laser beams simulation, appropriate ranges (-2.3561 rad and 2.3561 rad) were set. LIDAR sensor fired laser beams at different angles as shown in Figure 2, The time of flight of the reflected pulses will reveal the robot's distances from the obstacles. Moreover, with the information of the start, end and increment angle of the scan, points can be translated to Cartesian coordinates (x, y, z). For an arbitrary $Point_i$:

$$\begin{aligned}
\theta &= \pi - (angle_{start} + angle_{increment} \times i) \\
x &= \text{range}_i \times \cos(\theta) \\
y &= \text{range}_i \times \sin(\theta) \\
z &= 0
\end{aligned}$$

Note that z is always 0 in the laser coordinate frame because scans are in 2D.

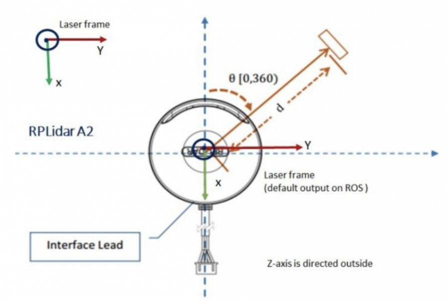


Fig. 2: Laser Scan to Point Cloud Conversion [13]

Secondly, coordinate frame transformation is also important in the field of robotics. These locations obtained above are relative to the LIDAR sensor. To extract the world coordinates (x_w, y_w, z_w) of these readings (x_l, y_l, z_l), we can project these points on a target frame and perspective projections can be represented by matrices:

$$\begin{bmatrix} r_{00} & r_{01} & r_{02} & d_0 \\ r_{10} & r_{11} & r_{12} & d_1 \\ r_{20} & r_{21} & r_{22} & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_l \\ y_l \\ z_l \\ 1 \end{bmatrix} = \begin{bmatrix} x_w & y_w & z_w & 1 \end{bmatrix} \quad (1)$$

where $r_{i,j}$ and $d_{i,j}$ corresponds to the rotational and displacement transformation.

We utilised the TF Listener in ROS to keep track of the homogeneous transformation matrix from the $laser0\_frame$ to the $world\_frame$, then applied the matrix to the Cartesian coordinates calculated. The TF library helps maintain the relationship between coordinate frames in a tree structure buffered in time and its detailed structure is in Figure 3.

Finally, a 3D occupancy grid is generated based on the coordinates of the occupied and free points. If the size of the grid is less than the agent, the configured space can be calculated with the Minkowski Sum. It diluted the occupied region to ensure that as long as the robot motion control reference point is outside of the region, collision will not occur. Alternatively, we defined a multi-resolution occupancy grid of 2 levels. The top layer is a low-resolution grid, with the size of the grid equals to that of the quadrotor (approximately $0.5m \times 0.5m \times 0.5m$), this forms the 'Regions'. The bottom layer is a high-resolution grid, which keeps tracks of the points being explored, and these grids will be referred to as 'Cells'. Planning would occur at the top layer, whereas the low level layer is only used for computing the occupancy status of the Regions. If more than a certain number of occupied or free

Region	Cells
Occupancy_Status is_Frontier	Occupancy_Status

TABLE I: Features of Regions (top layer grid) and Cells (bottom layer grid) in the 3D Occupancy Grid

Cells exist in a Region, the Region's occupancy status will be altered to either 'Occupied' or 'Free', else it will remain 'Unknown'. The threshold was decided based on empirical experiments. In the future, one should consider enhancing the binary occupancy data with the Bayesian statistical inference, where the occupancy probability could be updated using prior and evidence from the sensor. For frontier detection, we leveraged computer vision techniques and determined the boundary of the known and unknown space through Canny Edge Detector [14], and this was conducted at the top level.

C. Planning - Next-Best-View

In this project, we have investigated 4 exploration algorithms, each of them are associated with different levels of flexibility and complexity:

- **Predefined Path:** As a contrast to other fully autonomous exploration algorithms.
- **Random Walk:** As the baseline of the project.
- **Frontier Guided:** Inspired by the research of B. Yamauchi [5].
- **Frontier Guided with Randomness:** Similarly to the Frontier Guided Algorithm but with additional noise to avoid local minima.

Predefined Path: In this scheme, we have specified a trajectory for the quadrotor to transverse (a simple circle that goes around the target object). This scheme was used to compare the impact of autonomous verse non-autonomous exploration. It will also enable us to explore the trade-off between planners' flexibility and complexity.

Random Walk: Naive implementation for autonomous exploration is the random walk approach. In this scheme, the quadrotor will simply transverse to any available Region, which is chosen randomly. This has been used as the baseline of the project.

Frontier Guided: Our implementation of Frontier Guided Exploration is based on the concepts introduced in the research of B. Yamauchi. We also encoded a distance preference in the frontier selection process, whereas in the original publication, the quadrotor transverse to the closest frontier whenever possible. The distance preference allowed us to select frontier points that are not too close to the agent's current position. We hope that this will improve the rate of convergence of the exploration algorithm.

Frontier Guided with Randomness: This scheme is an improvised version of the previous one. During the experiment, we noticed that the quadrotor encountered local minima quite often. Hence, Gaussian noise was added to the decision making algorithms to avoid the quadrotor being stuck at these local minima. Hence, it is fusion between Random Walk approach and Frontier Guided exploration.

D. Planning - Collision Avoidance

For the reasons explained in Section II-C, we have chosen to deploy the AStar algorithm for extracting the shortest collision-free path between the current location and the target frontier. Moreover, since the motion plan is in 3D instead of 2D, we have to translate our grid into a graph structure. Each

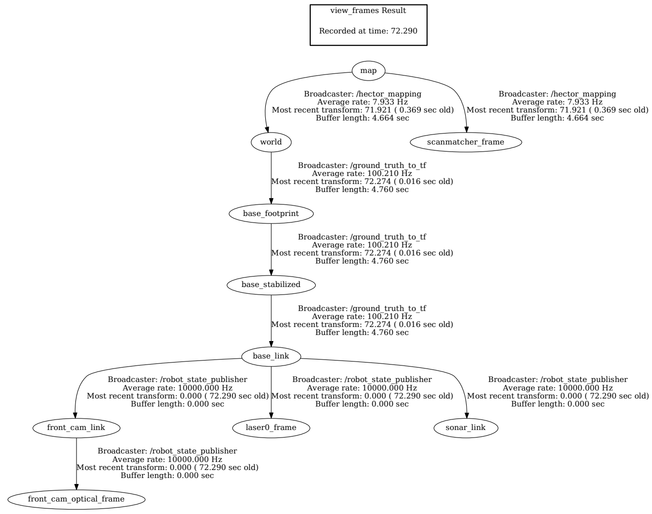


Fig. 3: TF Tree

Region acted as a node in the graph and is connected to 26 Regions, except for those that lie on the edges. A topology-graph-based AStar search [15] was used in this project.

E. Action

Furthermore, delay in the action-perception loop exists and would adversely impact the robot movements. The robot was expected to follow the trajectories defined by the path points when the plan was not being updated. However, time is required to compute the plan and the robot had already moved, this introduces uncertainty to the location of the robot. Therefore, we have also introduced some path following strategies.

The ground truth location of the quadrotor is first extracted from the simulator, then, the path points that is closest to the current location would be filtered out. Assume $Path\_point[i]$ is the closest to the current location, the localisation error e would be:

$$e = current\_location - Path\_point[i] \quad (2)$$

Proportional Controller was then introduced to scale the velocity of the quadrotor based on the localisation error e. Ideally, the higher the error, the slower the robot. Vice Versa. Hence, we defined a velocity scaling factor s with the Sigmoid Function:

$$s = \frac{2}{1 + e^{\frac{1}{0.8}(e-0.9)}} \quad (3)$$

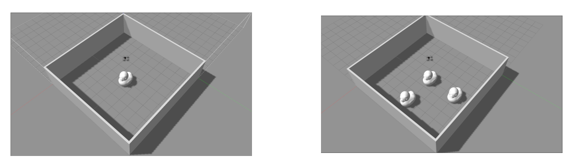
The output velocity would then be moderated as:

$$v_{output} = (Path\_point[i+1] - current\_location) \times s \quad (4)$$

Another precaution procedures to reduce the localisation error was to ensure the action updates are small. We refined the motion plan by splitting path points into smaller segments (in 0.05m intervals).

Finally, we should also take into account the holonomicity of the system. The quadcopter is non-holonomic as it only consists of 4 control inputs through its motors but it has 6 degrees of freedom. For simplicity, we only considered navigation in the spatial dimension, moreover, this led to occasion failures due to violations of physics laws. Therefore, in the future, we should also plan for orientation and utilise feedback linearisation, where the 6 degrees of freedoms (x, y, z, pitch, row, yaw) could be represented as functions of the control inputs of the 4 motors.

F. Experiment Setup



(a) Single Target (b) Multiple Target

Fig. 4: Different Environment

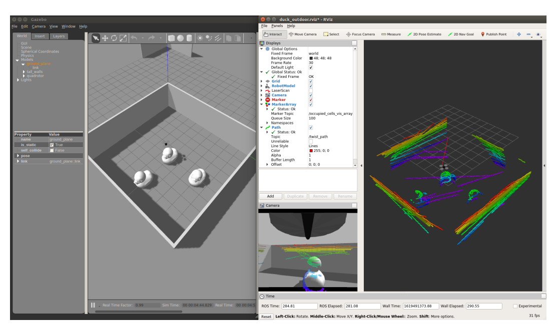


Fig. 5: Experiment Setup

Various technical stacks were utilised in this research. Our work was developed with ROS Kinetic on a GPU-supported

Ubuntu Docker Environment. Hector_quadrotor and Octomap libraries were utilised for simulating the drones in Gazebo and displaying the mapped mesh structure in RViz. Message-passing among ROS nodes could be viewed on the rqt_graph as shown in Appendix A. The general experiment environment is as shown in Figure 5.

Furthermore, the exploration algorithms were tested under two environment: Single Target and Multiple Target as shown in Figure 4. This is because we would like to investigate the algorithms' sensitivity to the environment and to determined whether a more congested environment would impact their performance.

IV. Results

Our frontier guided with randomness algorithm outperformed the baseline random walk implementation by 11.13% in the single-target scenario. In the multi-targets case, our frontier guided algorithm also brought a 3.77% improvements to the mesh coverage compared with the baseline. Details on the performance matrices will be explained in Section IV-A.

A. Performance Matrices

The performance of the schemes will be evaluated based on the attained mesh coverage within a certain duration. Mesh coverage is defined as:

$$\text{Mesh Coverage} = \frac{\text{Number of Explored Cells}}{\text{Total Number of Cells in Occupancy Grid}} \tag{5}$$

The duration was set as 9.5 minutes, which was based on the maximum convergence time observed in the experiments. Moreover, we will also use the visualisation of the occupancy grid to understand more about the functionality and limiting factors of the schemes.

V. Performance

	Predefined	Random (base)	Frontier	Frontier+Random
Single Target	16.63	15.00	14.96	16.67
Multiple Target	16.50	15.38	16.29	15.96

TABLE II: Mesh Coverage Attainable by various Exploration Algorithms in 9.5 minutes (in Percentages)

Through the experiments, we were able to observe some intriguing trends. A summary of their performance was displayed at Table II and Figure 7, which detailed the mesh coverage generated by various exploration algorithms.

A. Comparing Autonomous and Non-Autonomous Solutions

First of all, it can be easily noticed that the predefined path approach seems to provide the best mesh coverage in all cases. Also, from Figure 6, it shows that predefined path solution is the quickest to converge. This indicated that our experiment failed to expose the limitations of non-autonomous solutions because the environment is quite simple. Although non-autonomous solutions have lower planning complexity, they often failed to generalise to a more complex environment. The RViz image in Figure 7e confirmed our hypothesis. It

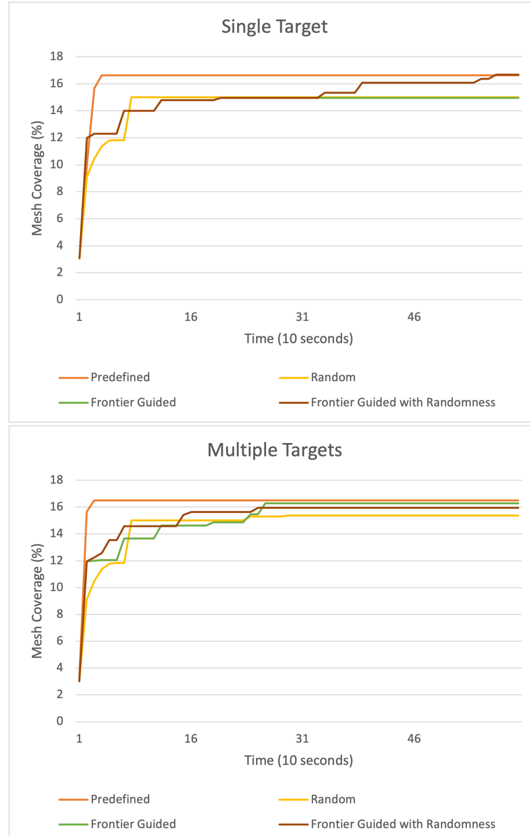


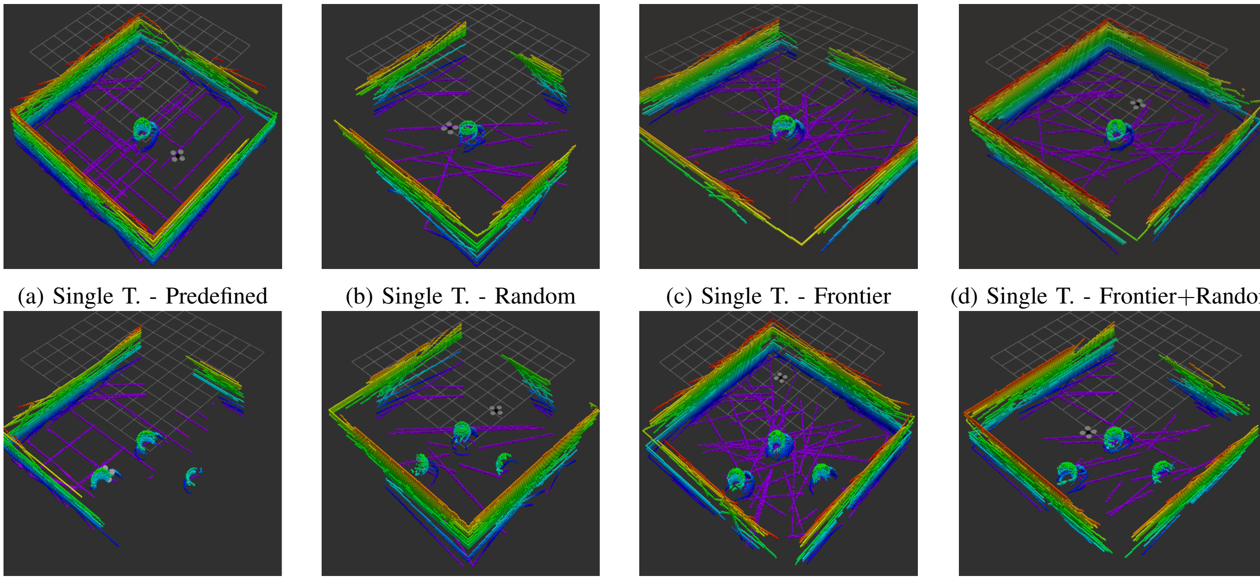
Fig. 6: Convergence Rate of the Exploration Algorithms under Different Environment

shows that the algorithm failed to scan half of the room. Hence, there is a trade-off between algorithm complexity and reconstruction performance.

B. Impact of Randomness on Frontier Guided Algorithm

Secondly, we would like to point out the impact of introducing randomness to frontier guided algorithm. On one hand, our data show that less guidance and more randomness works better for a single target environment; on the other hand, it also shows that more guidance and less randomness works better for a multi-target environment. This confirmed our hypothesis on the benefits of guided exploration - planning effort increases with the complexity of the target environment.

Thirdly, although additional randomness improved the mesh coverage in the single target case (Table II), its ability to navigate and explore unknown spaces remains in doubts. Comparing the visualisation of the occupancy grids in Figure 7b and 7c, it shows that the random walk algorithm often missed out part of the walls. While with the latter approach, the UAV tended to navigate through the whole space, even though it may generate a more sparse representation of the regions. Their differences highlighted the benefit of the frontier exploration approach. Since the UAV would be driven to the frontier, it is guaranteed that its knowledge of the world would increase. While for the random walk approach, its search



(a) Single T. - Predefined (b) Single T. - Random (c) Single T. - Frontier (d) Single T. - Frontier+Random

(e) Multiple T. - Predefined (f) Multiple T. - Random (g) Multiple T. - Frontier (h) Multiple T. - Frontier+Random

Fig. 7: Mesh Coverage of various Exploration Algorithms in Single and Multiple Target Environment

efficiency is based on the concept that the uniform distribution would tend to sample points from unexplored space, which is only guarded by the law of large numbers.

C. *Impact of different Environment*

Last but not least, the rate of convergence is much faster in the case of multiple targets relative to a single target scenario. This can be explained as there is less free space for the agent to explore in the multi-target case. Its knowledge regarding the world saturated earlier. For identical reasons, it would be unfair for us to compare the mesh coverage matrices of the 2 scenarios directly because the multiple targets case will always consist of more unscannable unknown space wrapped within the targets.

D. *Limitations*

Finally, we would also like to acknowledge the shortcomings of our approach and recommend them as future work.

First and foremost, the general reconstruction performance of the algorithms has been quite low due to the sparse occupancy grid. As described in the previous sections, path planning was conducted at the Region levels, instead of the Cell levels to lower the complexity of navigation. Although the collision avoidance framework has shown good performance (the quadrotor was able to navigate on its own for more than 20 minutes without collision), mesh coverage has been sacrificed as a result. Moreover, it was organised in such a way because collision avoidance is the prerequisite for exploration. In the future, when time allowed, we should consider enhancing the granularity of planning to improve precision.

Secondly, the lack of planning in orientation and motion primitive has resulted in inefficiency in exploration. Taking the case in Figure 7c as an example, the front part of the duck was not mapped despite the quadrotor had visited the nearby area (indicated by the purple path). This is because the LIDAR sensor is directional and data would not be registered if it is not within the scan range. This causes inefficiency in exploration and future work should consider devising motion plans that include pitch, row and yaw.

VI. Conclusion

In conclusion, our project showcased a full 3D reconstruction pipeline that attempted to address the challenge of exploring large, unknown, and unstructured environments. We demonstrated the gain of frontier exploration, which helps improve the mesh coverage by 11.13% relative to the random walk baseline in a single target case. Our experiment results also confirmed our hypothesis that stochasticity would bring benefits to a simple environment, but more guidance would be required for a complex environment. A thorough analysis of the performance of the algorithms with visualisations and videos were provided. We hope that our work would be inspirational for solving navigation challenge with exploration objective.

References

[1] P. Fallavollita et al. "UAS for archaeology. New perspectives on Aerial documentation". In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XL-1/W2 (Aug. 2013), pp. 131–135. DOI: 10.5194/isprsarchives-XL-1-W2-131-2013.

[2] Wang Jizhou, Lin Zongjian, and Li Chengming. "Reconstruction of buildings from a single UAV image". In: (Jan. 2004).

[3] Christopher Hackney and Alexander Clayton. “2.1.7. Unmanned Aerial Vehicles (UAVs) and their application in geomorphic mapping”. In: Apr. 2015.

[4] Jorge Torres-Sánchez et al. “Mapping the 3D structure of almond trees using UAV acquired photogrammetric point clouds and object-based image analysis”. In: *Biosystems Engineering* 176 (2018), pp. 172–184. ISSN: 1537-5110. DOI: https://doi.org/10.1016/j.biosystemseng.2018.10.018. URL: https://www.sciencedirect.com/science/article/pii/S1537511018306391.

[5] 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’*. 1997, pp. 146–151. DOI: 10.1109/CIRA.1997.613851.

[6] Wenchao Gao et al. “An improved Frontier-Based Approach for Autonomous Exploration”. In: *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. 2018, pp. 292–297. DOI: 10.1109/ICARCV.2018.8581245.

[7] Fabio Bruno et al. “Experimentation of structured light and stereo vision for underwater 3D reconstruction”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 66 (July 2011), pp. 508–518. DOI: 10.1016/j.isprsjprs.2011.02.009.

[8] Shimon Ullman. “The Interpretation of Structure from Motion”. In: *The Interpretation of Visual Motion*. 1979, pp. 133–175.

[9] Armin Hornung et al. “OctoMap: An efficient probabilistic 3D mapping framework based on octrees”. In: *Autonomous Robots* 34 (Apr. 2013). DOI: 10.1007/s10514-012-9321-0.

[10] Y.K. Hwang and N. Ahuja. “A potential field approach to path planning”. In: *IEEE Transactions on Robotics and Automation* 8.1 (1992), pp. 23–32. DOI: 10.1109/70.127236.

[11] Sertac Karaman and Emilio Frazzoli. *Sampling-based Algorithms for Optimal Motion Planning*. 2011. arXiv: 1105.1186 [cs.RO].

[12] Nicolas Pugeault Oscar Mendez Maldonado Simon Hadfield and Richard Bowden. “Next-Best Stereo: Extending Next-Best View Optimisation For Collaborative Sensors”. In: *Proceedings of the British Machine Vision Conference (BMVC)*. Ed. by Edwin R. Hancock Richard C. Wilson and William A. P. Smith. BMVA Press, Sept. 2016, pp. 65.1–65.12. ISBN: 1-901725-59-6. DOI: 10.5244/C.30.65. URL: https://dx.doi.org/10.5244/C.30.65.

[13] stevemartin. *Modify the Laser angle scan*. URL: https://answers.ros.org/question/310758/how-can-i-modify-the-laser-angle-scan/ (visited on 02/27/2021).

[14] John Canny. “A Computational Approach to Edge Detection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.6 (1986), pp. 679–698. DOI: 10.1109/TPAMI.1986.4767851.

[15] Ryan Kortvelesy. *A Star Software Library*. URL: https://github.com/Acciorocketships/MAPF (visited on 02/25/2021).

APPENDIX

A. RQT Graph

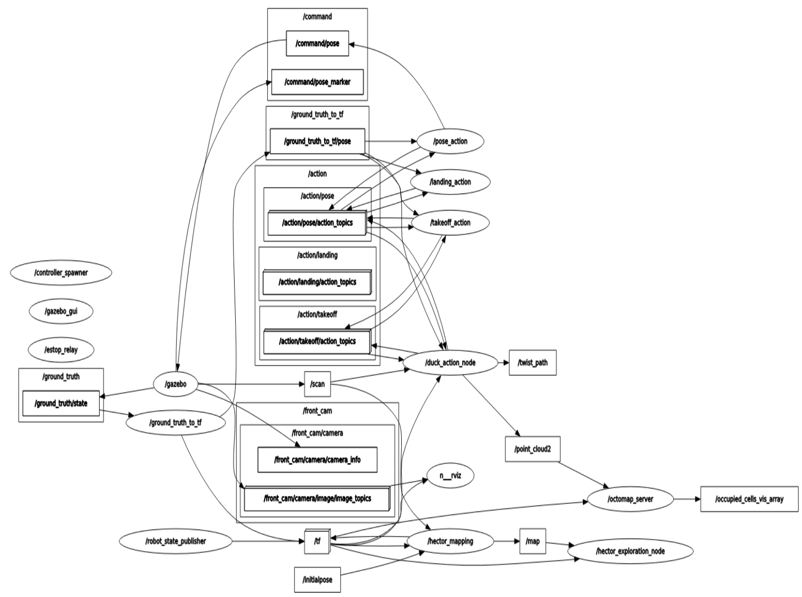


Fig. 8: ROS RQT Graph