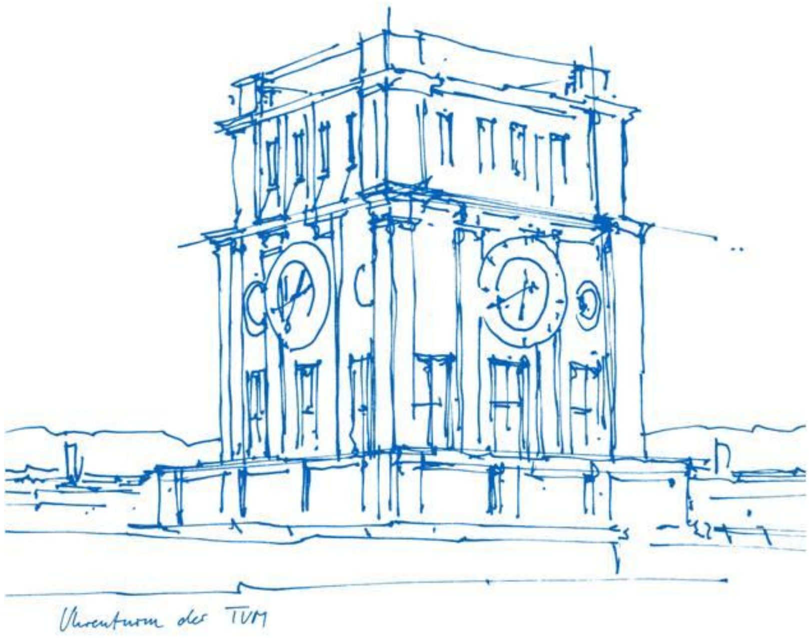
Technische Universität München

TUM AAS LRG 6300 Autonomous Systems

2023/24 WS Group Project Sub-Terrain Challenge Group 4



Group 4: Cem Kucukgenc, Baran Ozer, Erencan Aslakci, Serdar Soyer, Hunkar Suci

Group 4

Cem Kucukgenc

Baran Ozer

Serdar Soyer

Erencan Aslakci

Hunkar Suci

Outline

- Introduction
- Packages
  - Navigation
  - State Machine
  - Vision
- Discussion
- Conclusion

Introduction

Goal: Autonomous GPS-denied frontier exploration with drone for Unity cave simulation while locating the lanterns.

System Architecture

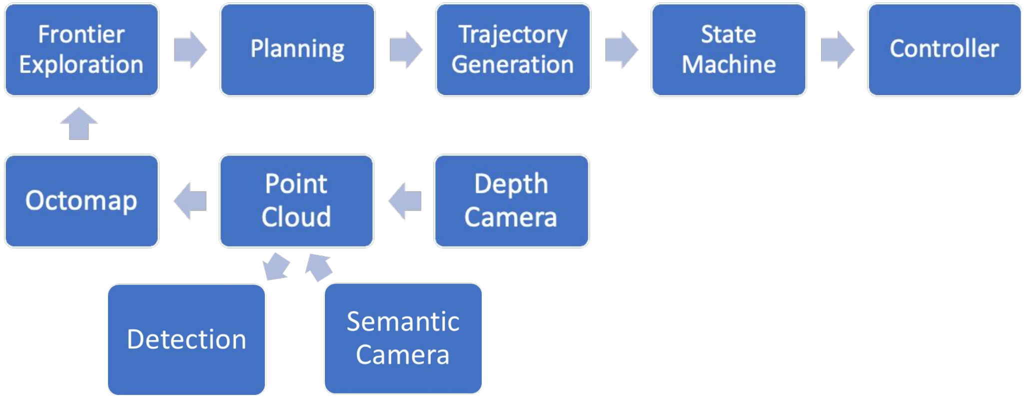


Figure 1: System Architecture

Introduction - RQT Graph

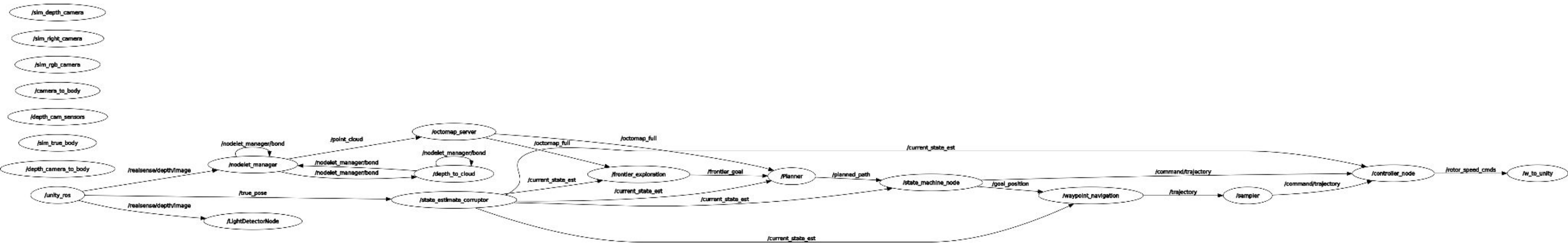


Figure 2: RQT Graph

Navigation Package

1) Frontier Exploration
2) Path Planner
3) Trajectory Generation

Figure 3: The drone

Figure 4: The drone in the cave

Navigation Package - Frontier Exploration

Frontier-based Exploration

- Dynamically identify and navigate towards frontiers, the boundaries between explored and unexplored areas.

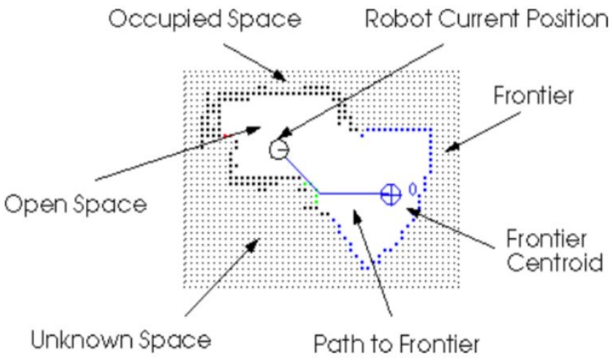


Figure 5: Frontier-based Exploration of unknown environment [1]

Navigation Package - Frontier Exploration

Frontier-based Exploration System Architecture

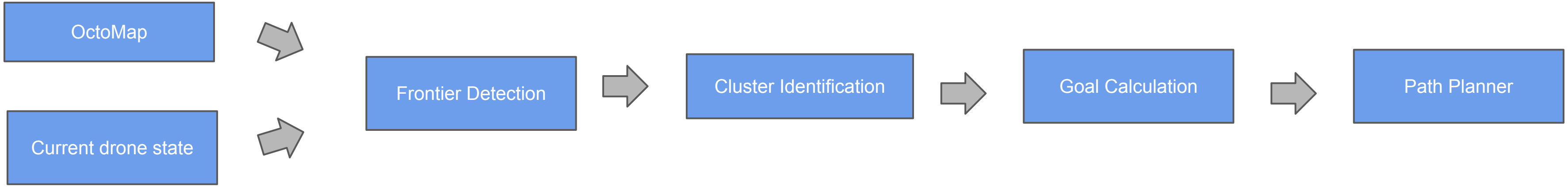


Figure 6: System Architecture of Frontier Exploration

Navigation Package - Frontier Exploration

Frontier Detection and Goals

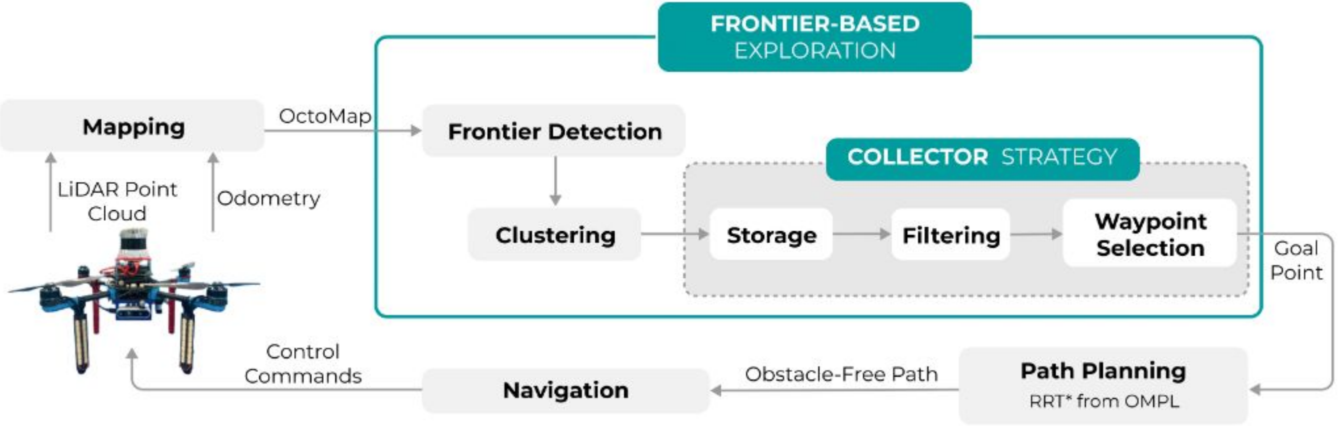


Figure 7: Overall schematic diagram of the 3D exploration [2]

Navigation Package - Frontier Exploration

Frontier-based Exploration Strategy and OctoMap

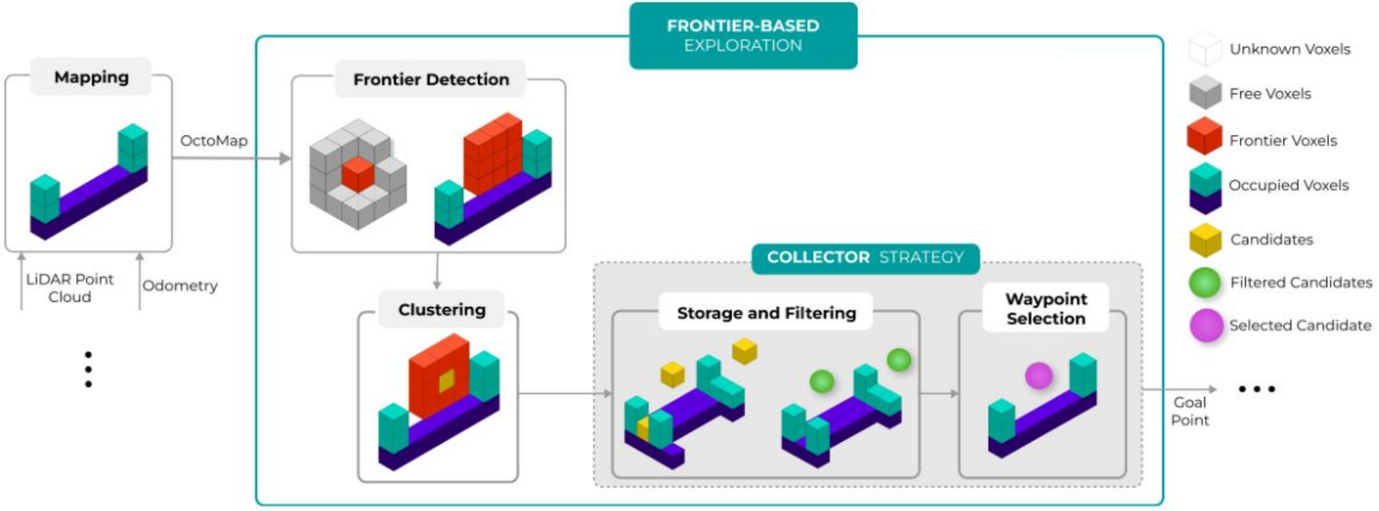


Figure 8: Frontier-based exploration strategy on the small part of the OctoMap [2]

Navigation Package - Path Planner

Path Planner System Architecture

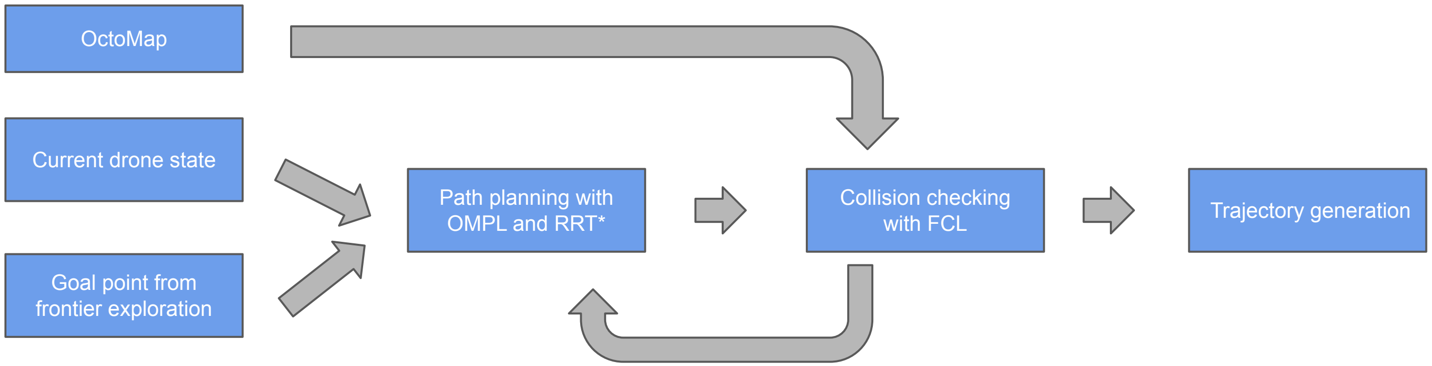


Figure 9: System Architecture of Path Planner

Navigation Package - Path Planner

- OMPL (Open Motion Planning Library) to define the configuration space
  - Drone state is represented as a point
  - Problem definition to specify start and goal states
  - Path length as optimization criterion
  - RRT* (Rapidly-exploring Random Tree) to find a collision-free path from start to goal states
    - sampling-based motion planning algorithm
    - incrementally builds a tree to explore the configuration space

- FCL (Flexible Collision Library) for collision checking
  - Drone: represented as a box
  - Surroundings: from OctoMap
  - Checks whether desired state is colliding with the surroundings

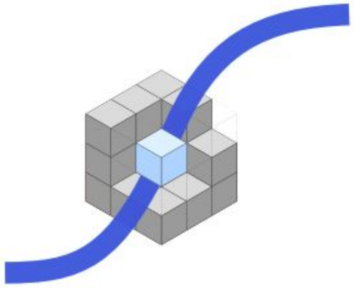


Figure 10: Collision check methodology [2]

Navigation Package - Trajectory Generation

- mav_trajectory_generation package
  - for minimum snap trajectory generation

- Boundary conditions:
  - Start Point = current position from current_state_est topic
  - Start Velocity = current velocity from current_state_est topic
  - End Point = desired position from path planner
  - End Velocity = (0,0,0)

Navigation Package - Trajectory Generation

- Initially, trajectory generation focused only on 3D positions (x, y, z).
    - Created problems in path planning

- Then trajectory generation improved to 4D by including the yaw information
    - Yaw calculated as the angle of the vector starting from the current position to the desired point
    - Drone facing in the proper direction ensured

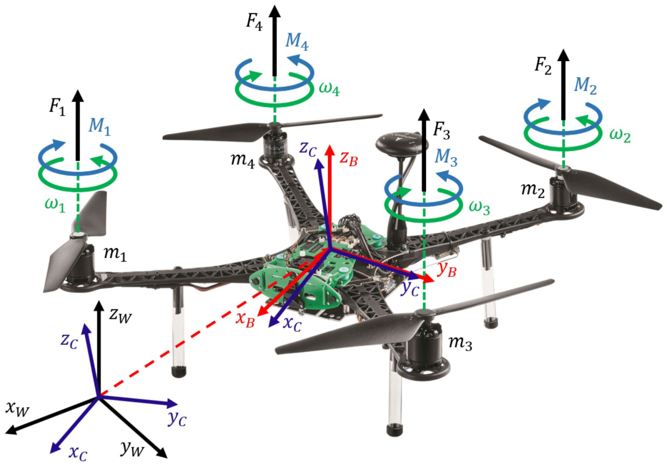


Figure 11: Drone Body Axes [3]

Navigation Package - Trajectory Generation

- After the trajectory is calculated with the settings mentioned before, it is sampled and sent to the geometric controller.

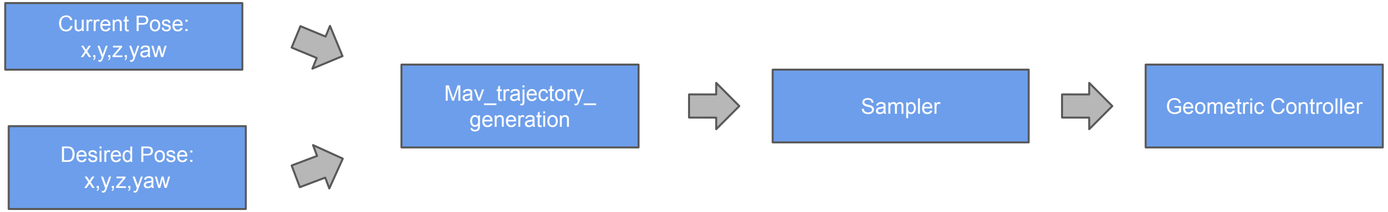


Figure 12: System architecture of trajectory generation

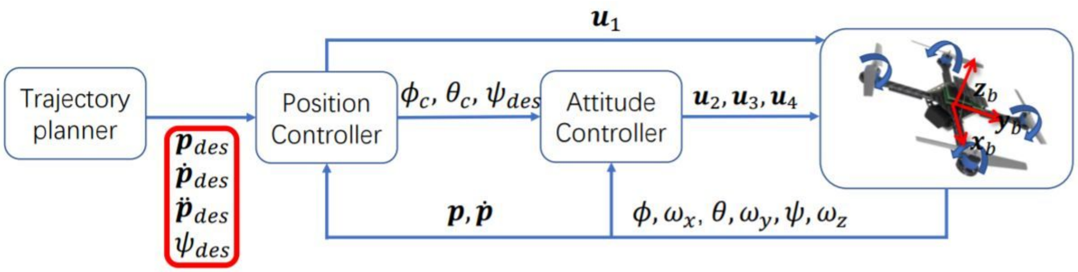


Figure 13: Geometric Controller [4]

State Machine Package

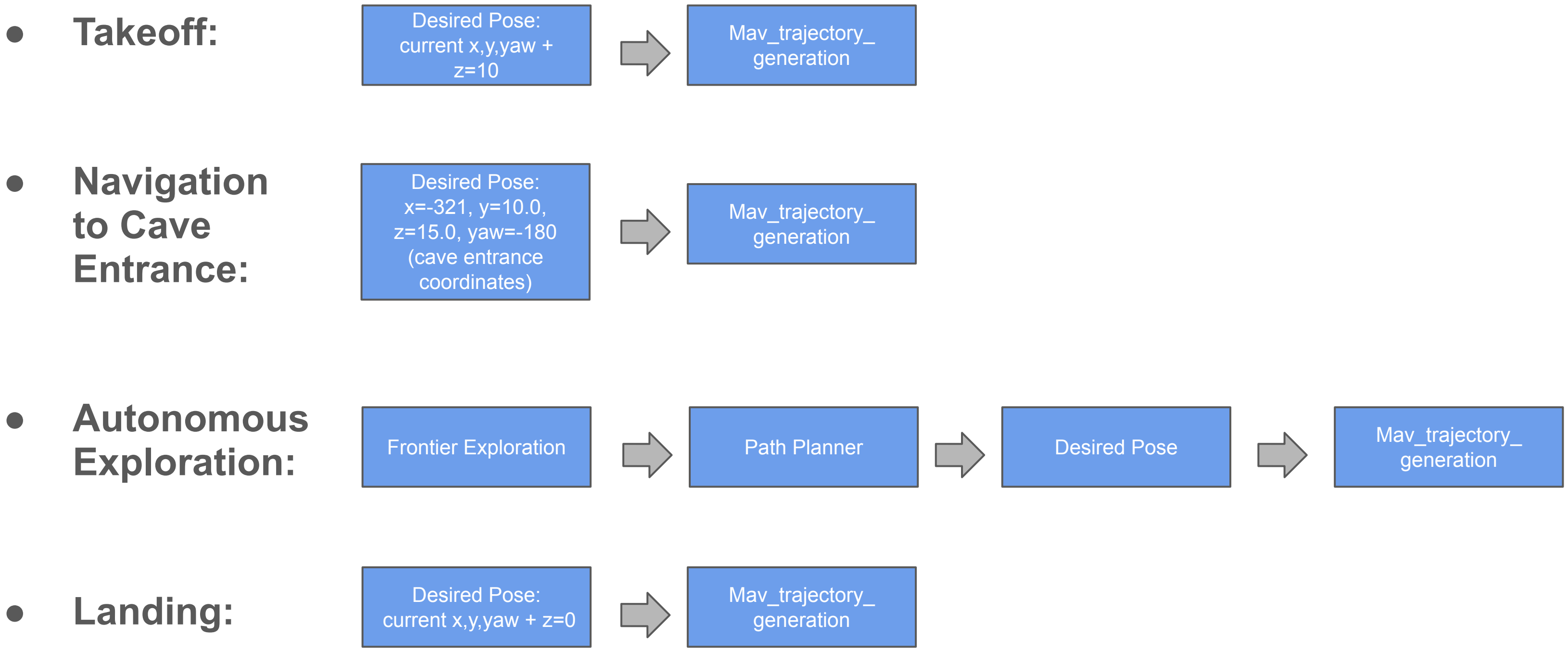


Figure 14: System architecture of state machine

Vision Package

1) Mapping
2) Light Detector

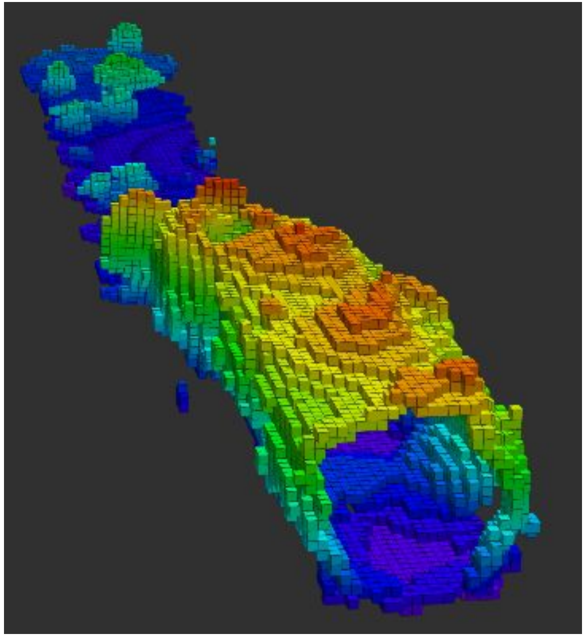


Figure 15: OctoMap

Figure 16: Lantern

Vision Package - Mapping

- **Point cloud generation** converts depth images to 3D point clouds
- **Octomap** then takes the generated point cloud from depth image and represents 3D spaces as **free, occupied, and unknown.**

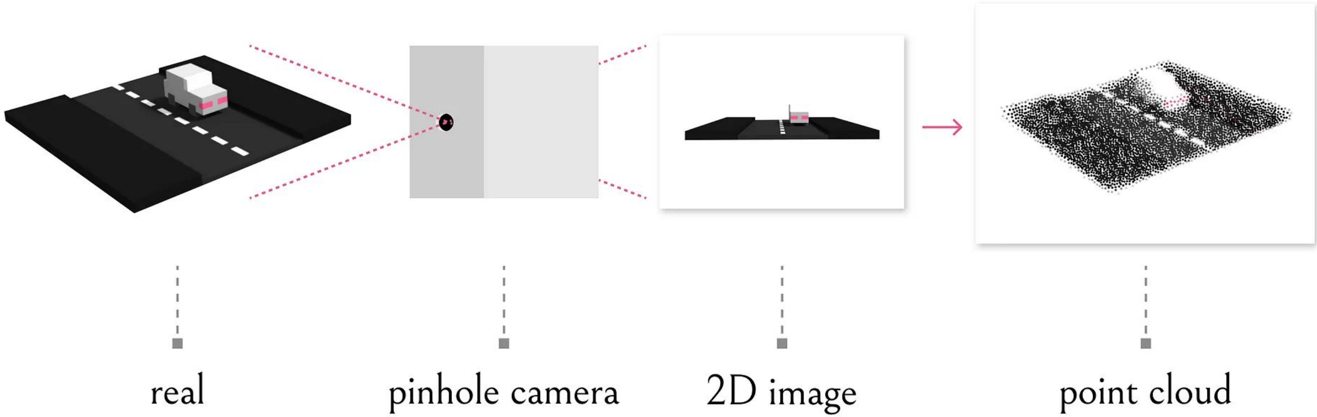


Figure 17: Depth image to point cloud conversion [5]

Vision Package - Mapping

Octomap Parameters:

- Latch = false
  - Suitable for dynamic data
- Resolution = 2
  - Tuned for performance and detail
- Max range = 32
  - Optimal max range as 32

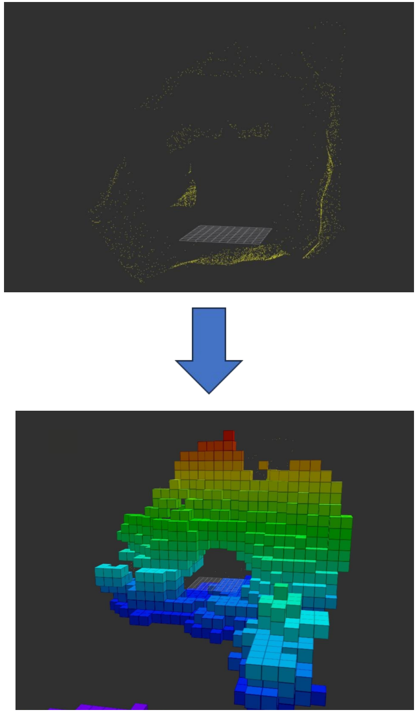


Figure 18: Point cloud to voxel grid representation

Vision Package - Light Detector

- To detect and localize light sources in the environment using depth and semantic images.

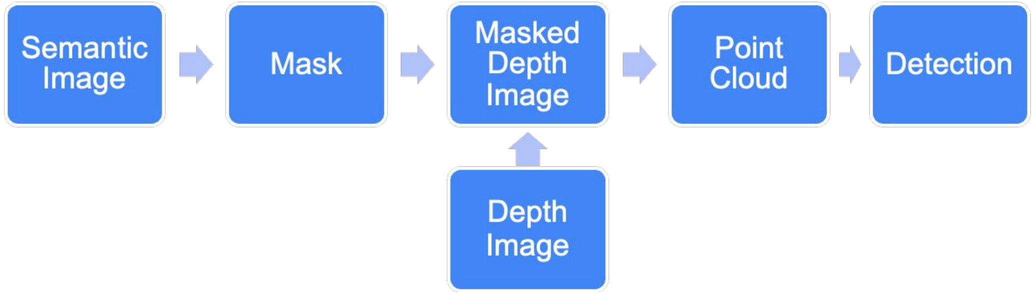


Figure 19: System architecture of light detector

Light 1

x = -59
y = 1
z = 9

Light 2

x = -599
y = -11
z = 6

Light 3

x = -743
y = -230
z = 2

Light 4

x = -1044
y = -158
z = -37

Figure 20: Light seen by semantic camera

Discussion - Achievements

- All the core parts expected in the assignment have been completed.

✅ A **state machine** for your robot, managing the take-off, travelling and landing at the goal location.

✅ A **perception pipeline** that converts the depth image, first to a point cloud and second to a voxel-grid representation of the environment.

✅ A **path planner** that generates a path through the environment to the goal location.

✅ A **trajectory planner** that plans a trajectory based on the found path

Discussion - Issues

- Fully autonomous exploration of the cave while simultaneously generating OctoMap of its surroundings was done, but:
    - Segmentation fault occurs during frontier exploration and path planning
    - When frontier exploration restarted drone continues its mission
- Cave voids
    - Causing errors in frontier exploration
    - Patching reduced those errors
    - Remaining voids still create problems
- Landing function
    - Working if executed
    - Since segmentation fault occurs, it was not executed automatically

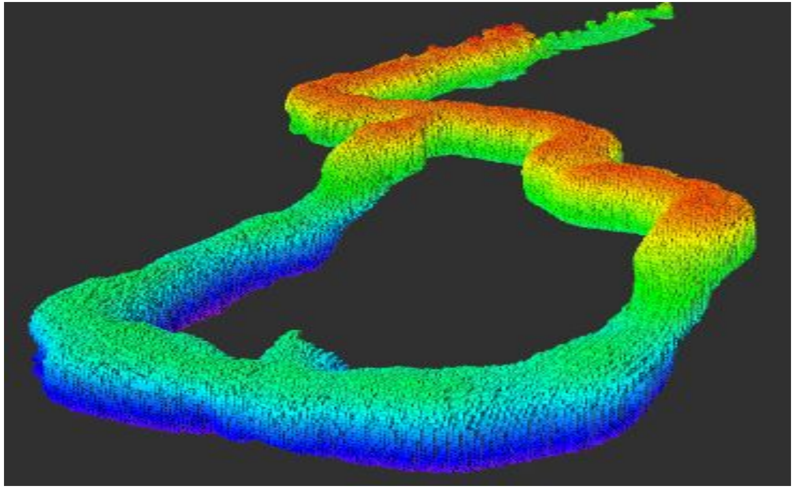


Figure 21: Explored cave

Conclusion

This project integrates ROS, OpenCV, PCL, OMPL, FCL, and MAV trajectory generation to achieve autonomous drone navigation. It overcomes dynamic frontier exploration mission with path planning, a robust state machine and vision-based detection.

Future Work:

- Artificial Intelligence models for increased environmental perception and autonomous decision-making
- Development of algorithms for more sophisticated navigation and obstacle avoidance, enabling the drone to adapt to complex environments.
- Frontier exploration with multiple drones.

References

[1] Vu Phi Tran, Matthew A. Garratt, Kathryn Kasmarik, Sreenatha G. Anavatti, Shadi Abpeikar, Frontier-led swarming: Robust multi-robot coverage of unknown environments, Swarm and Evolutionary Computation, Volume 75, 2022, 101171, ISSN 2210-6502, https://doi.org/10.1016/j.swevo.2022.101171.

[2] Caiza, I. D. C., Milas, A., Grova, M. a. M., Pérez-Grau, F. J., & Petrović, T. (2023). Autonomous exploration of unknown 3D environments using a Frontier-Based collector strategy. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2311.12408

[3] Arshad, M., Ahmed, J., & Bang, H. (2023). Quadrotor path planning and polynomial trajectory generation using quadratic programming for indoor environments. *Drones*, *7*(2), 122. https://doi.org/10.3390/drones7020122

[4] ELEC5660. (n.d.). https://gaowenliang.github.io/HKUST-ELEC5660-Introduction-to-Aerial-Robots/project/p1p4.htm

[5] Yodayoda. (2020b, December 11). From depth map to point cloud. Medium. https://medium.com/yodayoda/from-depth-map-to-point-cloud-7473721d3f

Thank you!

Any questions?