Autonomous Systems Project

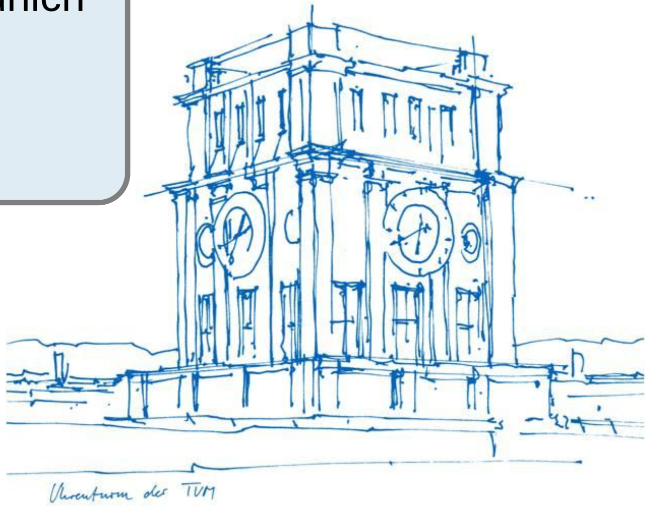
Ramkrishna Chaudhari, Mohamad Alattar

Berhan Sofuoglu, Fabian Sommer

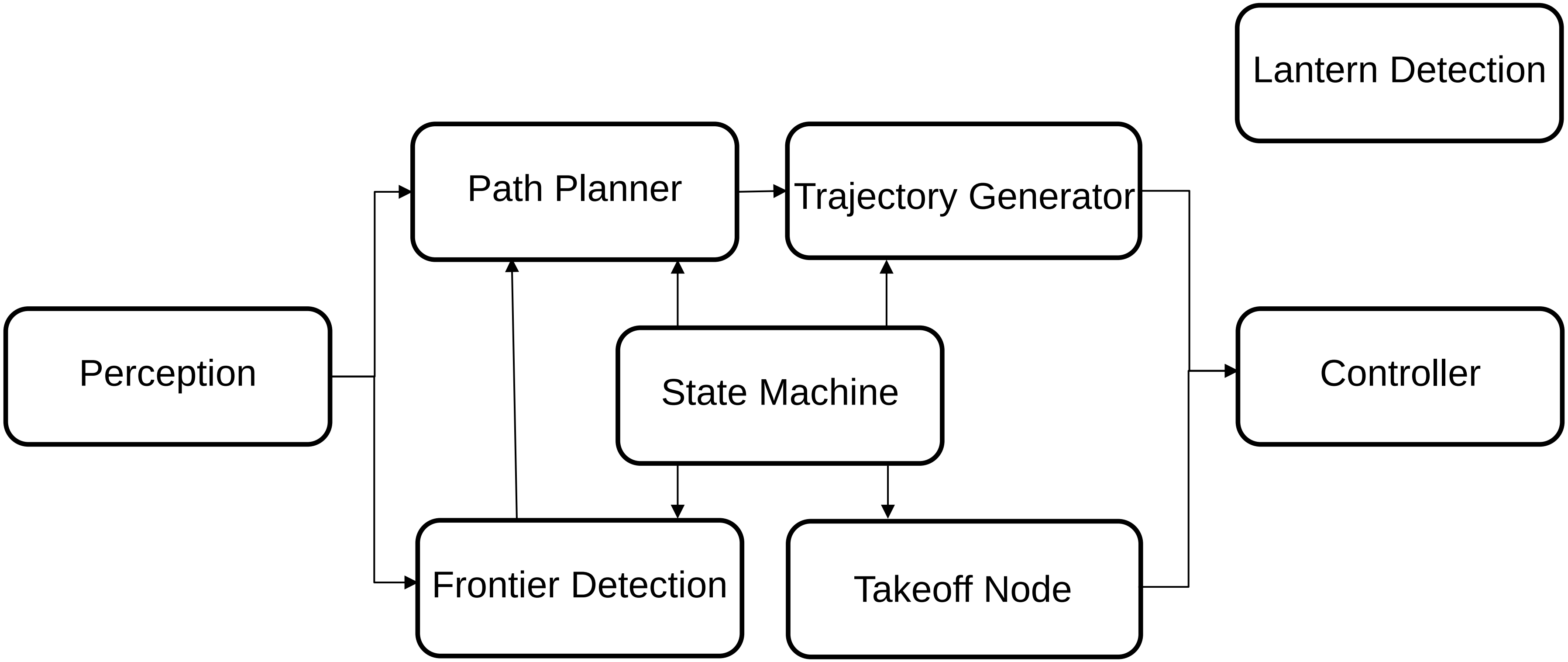
Team 4

Technical University of Munich

March 4th, 2025



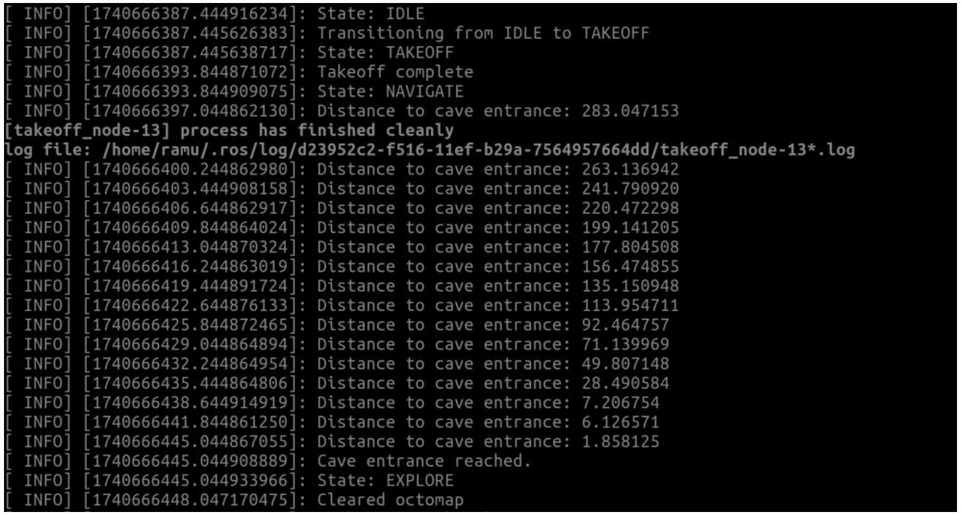
General Architecture



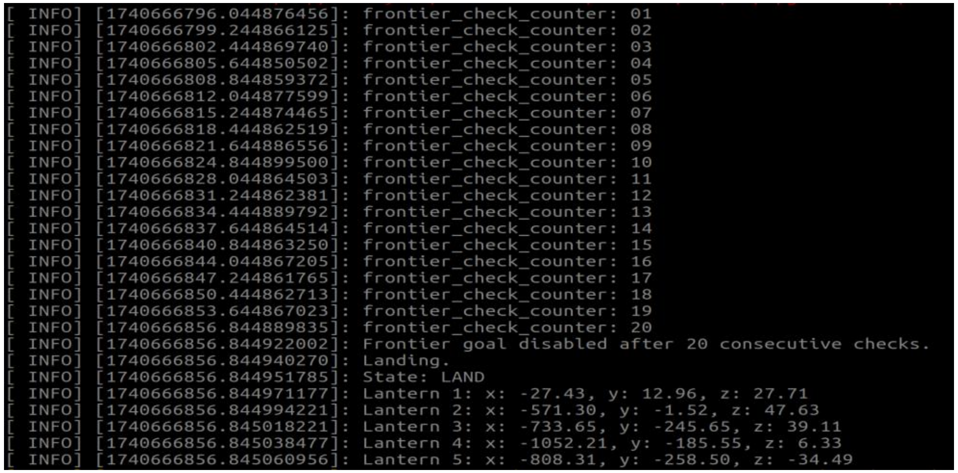
State Machine

- Track goals and their thresholds
- Transition the states depending on whether goals are met or not
- STATES
  - IDLE*
  - TAKEOFF
  - NAVIGATE
  - EXPLORE
  - LAND
- Supply the necessary communication for the mission (e.g. Frontier Detection, Lantern Detection, Takeoff Node)
- Debugging and testing for the mission if necessary

*placeholder state



```
[ INFO] [1740666387.444916234]: State: IDLE
[ INFO] [1740666387.445626383]: Transitioning from IDLE to TAKEOFF
[ INFO] [1740666387.445638717]: State: TAKEOFF
[ INFO] [1740666393.844871072]: Takeoff complete
[ INFO] [1740666393.844909075]: State: NAVIGATE
[ INFO] [1740666397.044862130]: Distance to cave entrance: 283.047153
[takeoff_node-13] process has finished cleanly
log file: /home/ramu/.ros/log/d23952c2-f516-11ef-b29a-7564957664dd/takeoff_node-13*.log
[ INFO] [1740666400.244862980]: Distance to cave entrance: 263.136942
[ INFO] [1740666403.444908158]: Distance to cave entrance: 241.790920
[ INFO] [1740666406.644862917]: Distance to cave entrance: 220.472298
[ INFO] [1740666409.844864024]: Distance to cave entrance: 199.141205
[ INFO] [1740666413.044870324]: Distance to cave entrance: 177.804508
[ INFO] [1740666416.244863019]: Distance to cave entrance: 156.474855
[ INFO] [1740666419.444891724]: Distance to cave entrance: 135.150948
[ INFO] [1740666422.644876133]: Distance to cave entrance: 113.954711
[ INFO] [1740666425.844872465]: Distance to cave entrance: 92.464757
[ INFO] [1740666429.044864894]: Distance to cave entrance: 71.139969
[ INFO] [1740666432.244864954]: Distance to cave entrance: 49.807148
[ INFO] [1740666435.444864806]: Distance to cave entrance: 28.490584
[ INFO] [1740666438.644914919]: Distance to cave entrance: 7.206754
[ INFO] [1740666441.844861250]: Distance to cave entrance: 6.126571
[ INFO] [1740666445.044867055]: Distance to cave entrance: 1.858125
[ INFO] [1740666445.044908889]: Cave entrance reached.
[ INFO] [1740666445.044933966]: State: EXPLORE
[ INFO] [1740666448.047170475]: Cleared octomap
```



```
[ INFO] [1740666796.044876456]: frontier_check_counter: 01
[ INFO] [1740666799.244866125]: frontier_check_counter: 02
[ INFO] [1740666802.444869740]: frontier_check_counter: 03
[ INFO] [1740666805.644850502]: frontier_check_counter: 04
[ INFO] [1740666808.844859372]: frontier_check_counter: 05
[ INFO] [1740666812.044877599]: frontier_check_counter: 06
[ INFO] [1740666815.244874465]: frontier_check_counter: 07
[ INFO] [1740666818.444862519]: frontier_check_counter: 08
[ INFO] [1740666821.644886556]: frontier_check_counter: 09
[ INFO] [1740666824.844899500]: frontier_check_counter: 10
[ INFO] [1740666828.044864503]: frontier_check_counter: 11
[ INFO] [1740666831.244862381]: frontier_check_counter: 12
[ INFO] [1740666834.444889792]: frontier_check_counter: 13
[ INFO] [1740666837.644864514]: frontier_check_counter: 14
[ INFO] [1740666840.844863250]: frontier_check_counter: 15
[ INFO] [1740666844.044867205]: frontier_check_counter: 16
[ INFO] [1740666847.244861765]: frontier_check_counter: 17
[ INFO] [1740666850.444862713]: frontier_check_counter: 18
[ INFO] [1740666853.644867023]: frontier_check_counter: 19
[ INFO] [1740666856.844889835]: frontier_check_counter: 20
[ INFO] [1740666856.844922002]: Frontier goal disabled after 20 consecutive checks.
[ INFO] [1740666856.844940270]: Landing.
[ INFO] [1740666856.844951785]: State: LAND
[ INFO] [1740666856.844971177]: Lantern 1: x: -27.43, y: 12.96, z: 27.71
[ INFO] [1740666856.844994221]: Lantern 2: x: -571.30, y: -1.52, z: 47.63
[ INFO] [1740666856.845018221]: Lantern 3: x: -733.65, y: -245.65, z: 39.11
[ INFO] [1740666856.845038477]: Lantern 4: x: -1052.21, y: -185.55, z: 6.33
[ INFO] [1740666856.845060956]: Lantern 5: x: -808.31, y: -258.50, z: -34.49
```

Lantern Detection

- Lantern detection via sematic camera –> MONO8 data converted to OpenCV matrix
- Compute centroid of the detected lantern pixels -> average pixel coordinates of lantern
- Convert 2D position into 3D position with depth camera

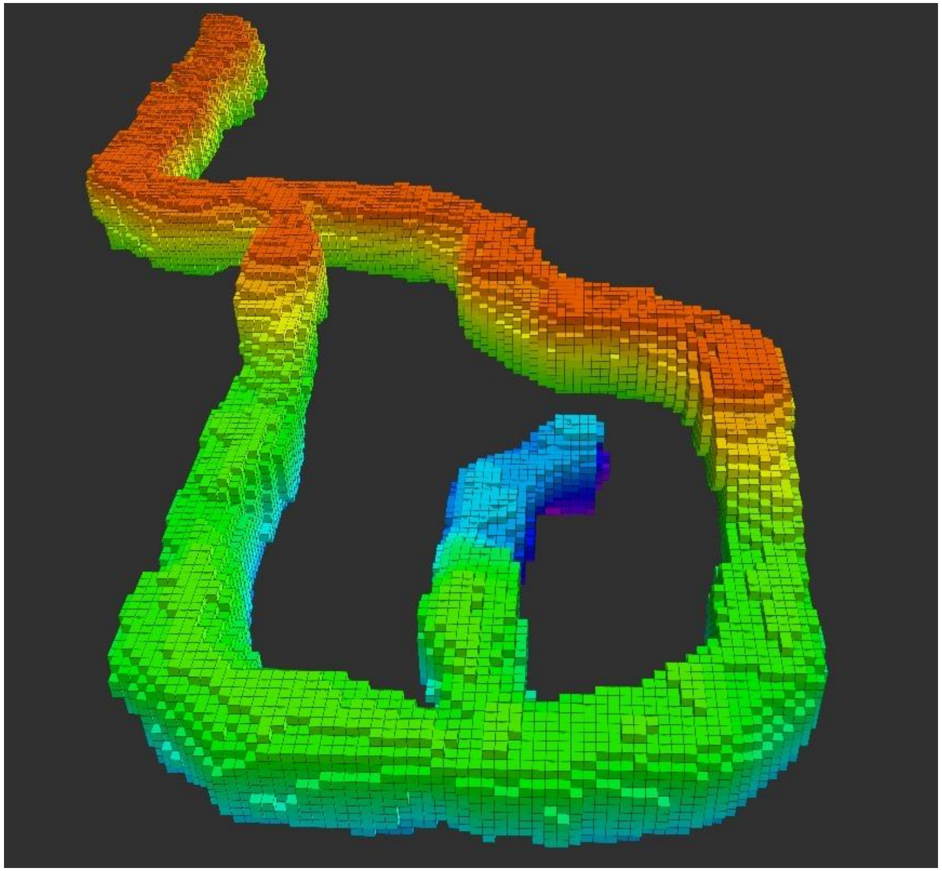
$$X = \frac{(\bar{x} - c_x)\, Z_{\text{depth}}}{f_x}, \quad Y = \frac{(\bar{y} - c_y)\, Z_{\text{depth}}}{f_y}, \quad Z = Z_{\text{depth}}$$

- Transform the 3D camera frame coordinates into global frame using tf2_ros

$$\begin{pmatrix} X_{\text{global}} \\ Y_{\text{global}} \\ Z_{\text{global}} \end{pmatrix} = \mathbf{T}_{\text{global}\leftarrow\text{camera}} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

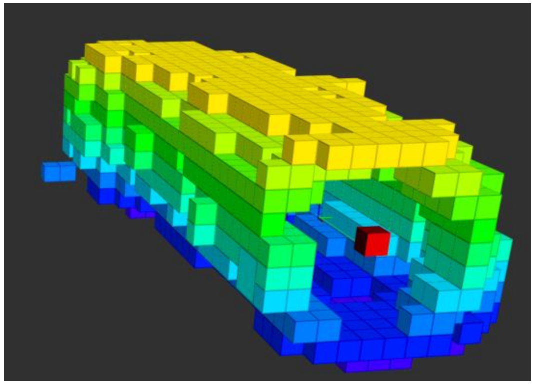
- Check if the detected lantern is a duplicate using the euclidean distance from previously detected lanterns
- Only register lantern as detected if its near the drone

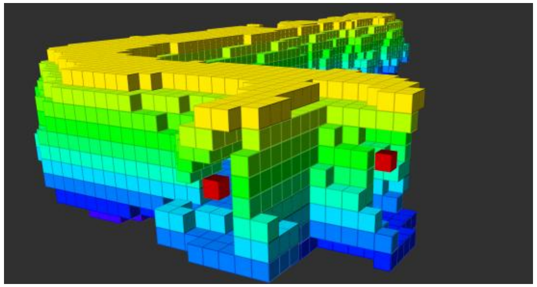
Perception

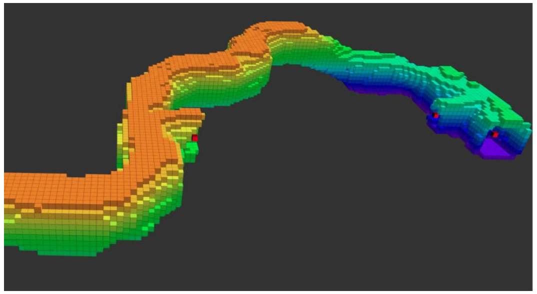


Frontier Detection

- Boundary between explored and unexplored regions in a mapped environment.
- Uses OctoMap to get the Free spaces knowledge between known and unknown spaces.
- Clusters them using the Mean-Shift-Clustering.





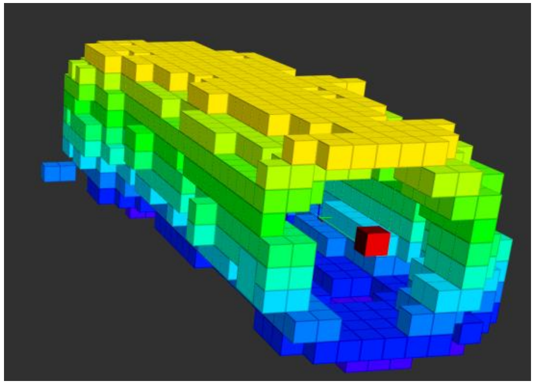


Frontier Scoring and Selection

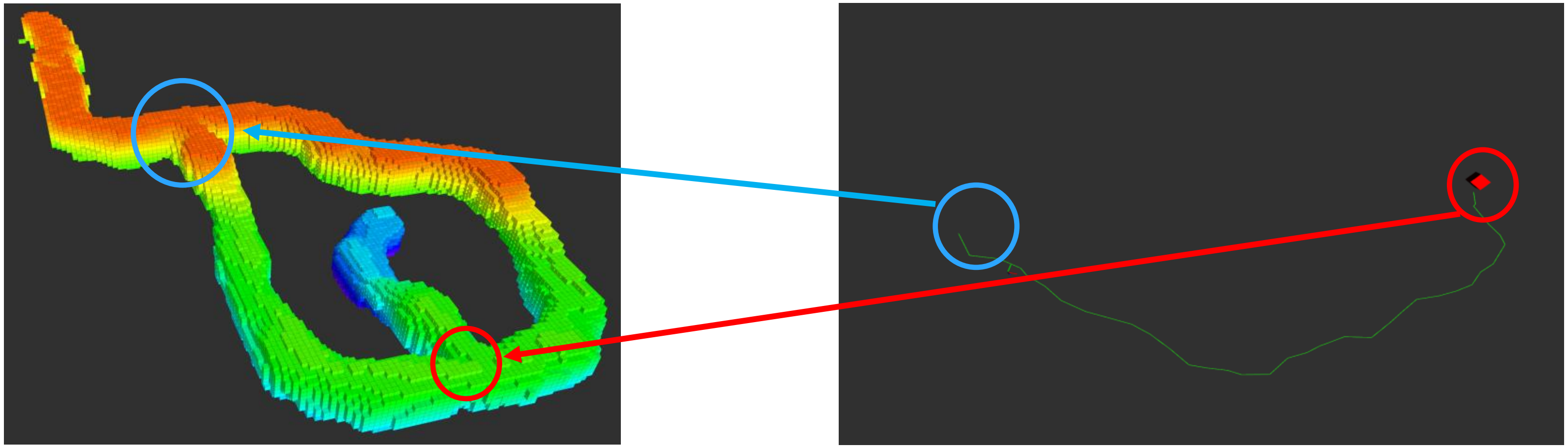
$$\boxed{S(f) = -k_d \cdot d(f) + k_n \cdot N(f) - k_y \cdot |\theta(f) - \theta_{drone}|}$$

- ➢ $d(f)$ is the Euclidean distance from the drone to the frontier,
- ➢ $N(f)$ is the number of adjacent frontiers
- ➢ $\theta(f)$ is the yaw alignment difference
- ➢ $k_d(1)$, $k_n(0.1)$, $k_y(55.0)$ are weight coefficients

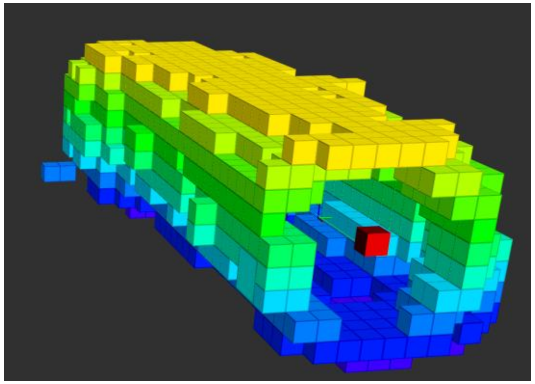
Path Planning [OMPL → RRT*]



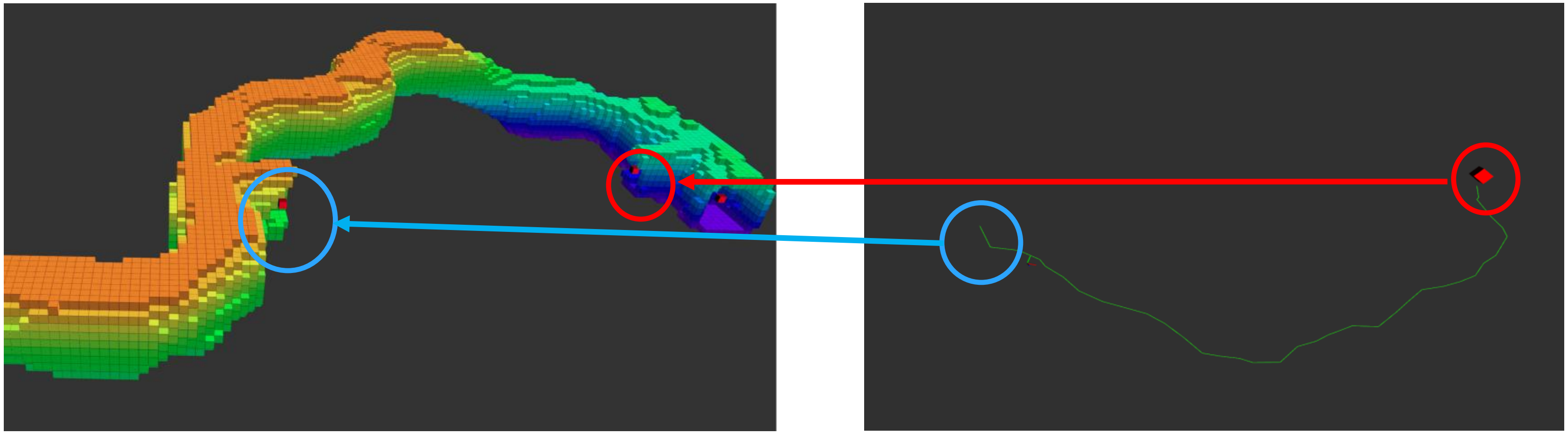
- Generates a path from current position to the frontier_goal
- Takes the Octomap into account
- A random sample is drawn from the free space
- The nearest node in the existing tree is identified
- The new edge is validated against the occupancy map to ensure obstacle-free movement
- If the new path provides a shorter cost, nearby nodes are reconnected to optimize the path
- Ray-casting techniques are used to check for occlusions along potential path segments



Path Planning [OMPL ➡ RRT*]



- Generates a path from current position to the frontier_goal
- Takes the Octomap into account
- A random sample is drawn from the free space
- The nearest node in the existing tree is identified
- The new edge is validated against the occupancy map to ensure obstacle-free movement
- If the new path provides a shorter cost, nearby nodes are reconnected to optimize the path
- Ray-casting techniques are used to check for occlusions along potential path segments



Trajectory Generator

- Uses Jerk-minimization polynomial

$$p(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

- Constraints for continuity in position and velocity

$$\begin{aligned} p(0) &= p_0, \quad & p(T) &= p_T, \\ p'(0) &= v_0, \quad & p'(T) &= v_T \end{aligned}$$

- Speed determined based on segment length
  - ○ minimum time of 1s to avoid high speeds
  - ○ Maximum speed of 6.67 m/s

$$T = \max(1.0, d \times 0.15)$$

- If a more efficient path is introduced while the drone is flying in a trajectory, a new trajectory is generated along the more efficient path

Controller

- Uses the "/desired_state" and "current_state_est" topics to implement the Geometric Controller proposed by Lee et al. [1]
- No finetuning was necessary, the controller tuning paramters proposed in the base source code worked well for us

[1] T. Lee, M. Leok and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," *49th IEEE Conference on Decision and Control (CDC)*, Atlanta, GA, USA, 2010, pp. 5420-5425

Conclusion and Limitations

- Successful in exploring the Cave and finding all the lanterns autonomously:

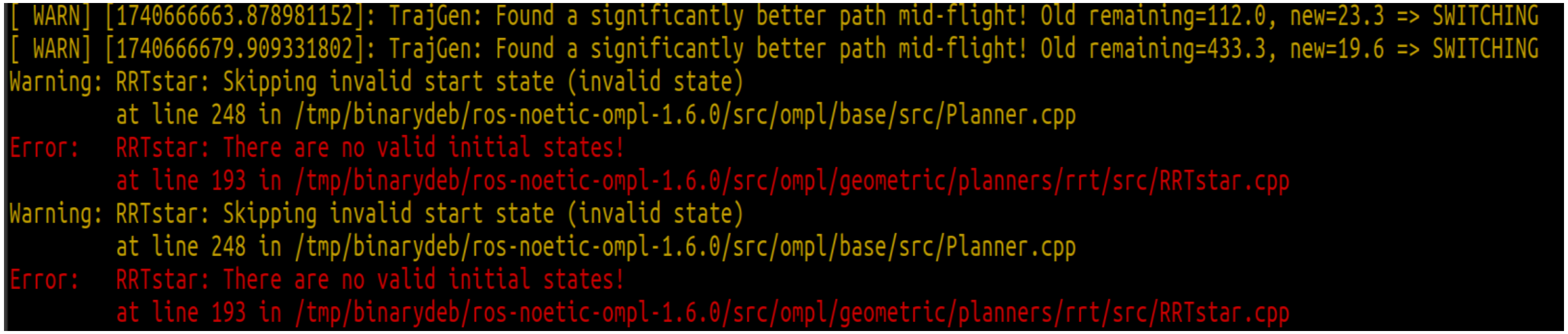
Lanterns	x	y	z
Lantern 1	-27.43	12.96	27.71
Lantern 2	-571.30	-1.52	47.63
Lantern 3	-733.65	-245.65	39.11
Lantern 4	-1052.21	-185.55	6.33
Lantern 5	-808.31	-258.50	-34.49

Limitations

- Nondeterministic path from pathplanner

  -Frontier scoring and selection

  -Current yaw of the drone
- Error/Warning from Path planner



```
[ WARN] [1740666663.878981152]: TrajGen: Found a significantly better path mid-flight! Old remaining=112.0, new=23.3 => SWITCHING
[ WARN] [1740666679.909331802]: TrajGen: Found a significantly better path mid-flight! Old remaining=433.3, new=19.6 => SWITCHING
Warning: RRTstar: Skipping invalid start state (invalid state)
         at line 248 in /tmp/binarydeb/ros-noetic-ompl-1.6.0/src/ompl/base/src/Planner.cpp
Error:   RRTstar: There are no valid initial states!
         at line 193 in /tmp/binarydeb/ros-noetic-ompl-1.6.0/src/ompl/geometric/planners/rrt/src/RRTstar.cpp
Warning: RRTstar: Skipping invalid start state (invalid state)
         at line 248 in /tmp/binarydeb/ros-noetic-ompl-1.6.0/src/ompl/base/src/Planner.cpp
Error:   RRTstar: There are no valid initial states!
         at line 193 in /tmp/binarydeb/ros-noetic-ompl-1.6.0/src/ompl/geometric/planners/rrt/src/RRTstar.cpp
```

THANKS YOU FOR YOUR ATTENTION

ANY QUESTIONS?