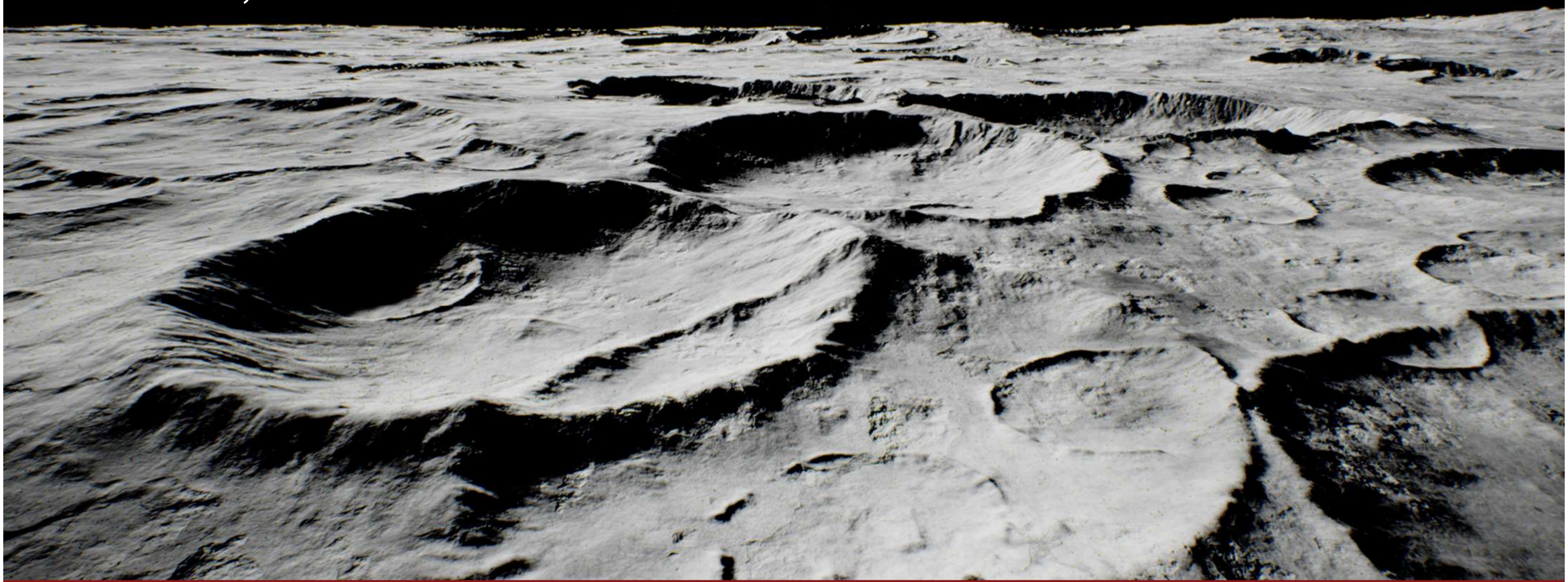
ARC Based Path Planner and Segmentation for Lunar Autonomy Challenge Contd.

THOMAS DENG, ADAM DAI



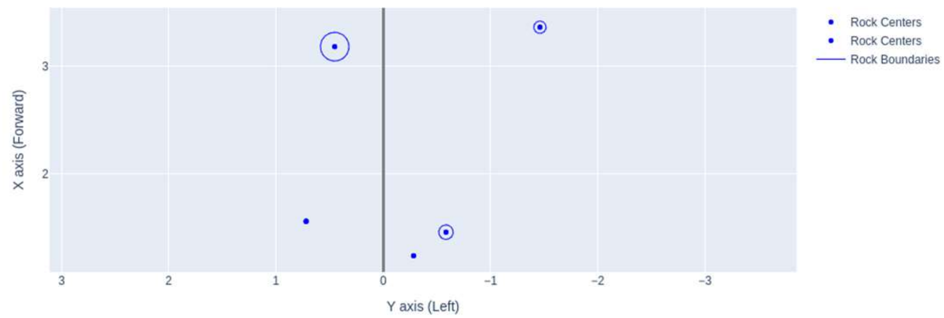
Outline/Problem Statement

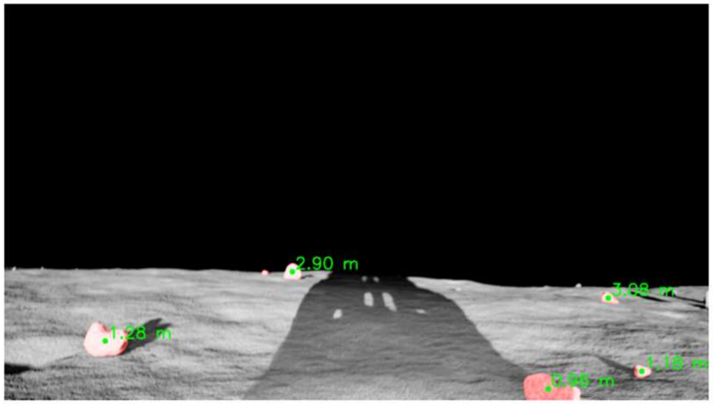
- The goal: arc-based path planning, leveraging image segmentation and depth estimation to enable autonomous navigation around segmented rocks.
- Develop a planner that effectively avoids obstacles and follows a set of global waypoints with a high degree of consistency.

Image Segmentation + Depth Estimation

- Objective: Identify and localize rocks in the camera frame to inform collision avoidance in path planning.
- Utilizes Unet, lightweight CNN segmentation, to segment rocks
- Stereo Depth Estimation: Computes rock depths using stereo disparity calculations from paired rover cameras.
- Rock Size Estimation: Determines rock radii using depth, focal length, and pixel width calculations.
- Output: Generates a local 2D rock map that integrates with trajectory planning for real-time obstacle avoidance.

Segmentation and Depth Examples



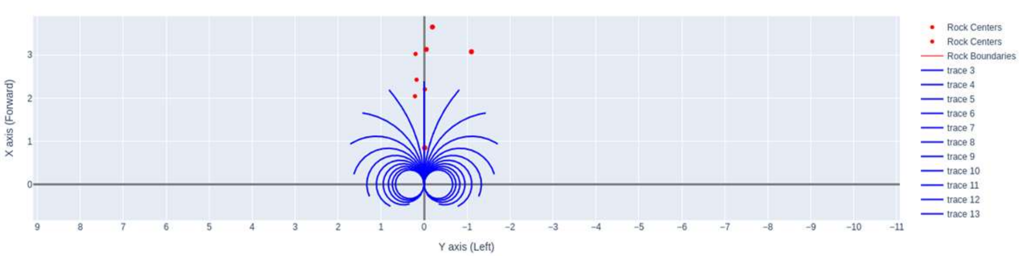


Arc-based Path Planner

- Develop a dynamically feasible, collision-free path planner that enables the rover to navigate around segmented rocks while following global waypoints.
- Uses Dubin’s car model to generate smooth, kinematically feasible paths.
- Difference Between Last Quarter:
- Instead of concatenating two sequential arc segments, just use a single arc segment, following the dynamic window approach.
- Each arc follows a constant velocity of 0.2 m/s with an angular velocity selected from five discrete values between -1 rad/s and 1 rad/s.
- Incorporates rock segmentation results and depth estimation to determine rock positions and radii.

Configuration Testing

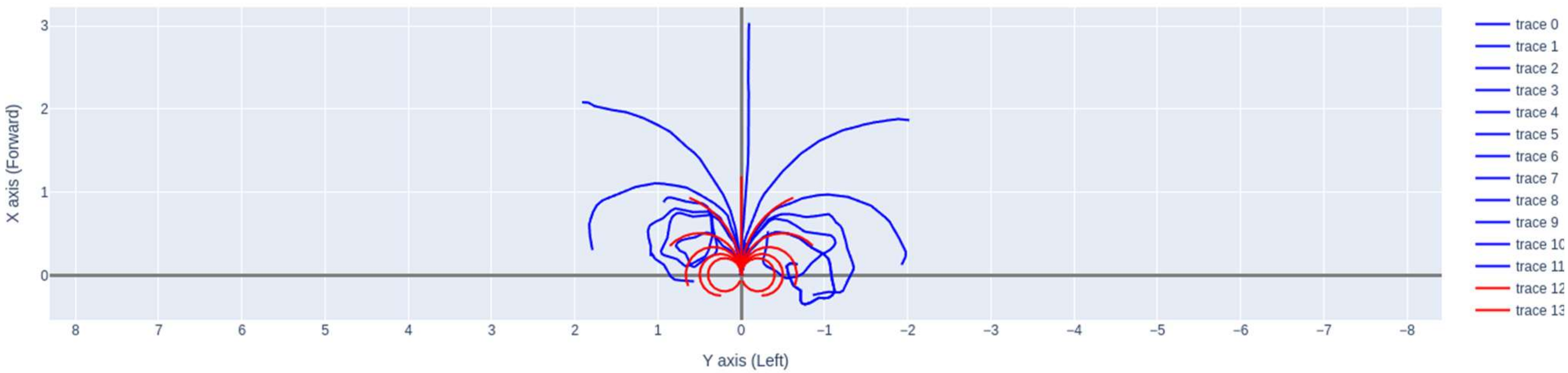
- Initial testing was conducted on a spiral path using a configuration of 20 paths and a 4-second horizon.
- However, this configuration failed to detect large obstacles in time. Extending the arc horizon to 6 seconds improved detection, but still resulted in 10 collisions and a crash after 48,804 steps.
- Eventually, we found out that the ideal time horizon would be 8 seconds long.



Improving the Dynamics Model

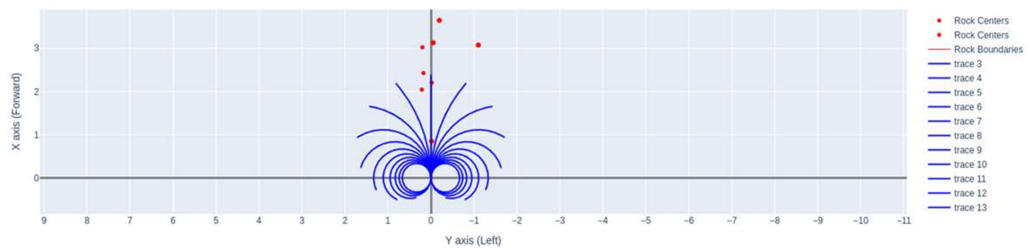
- Problem: The rover’s turning dynamics did not exactly align with the Dubin’s model, causing the rover to turn less than it should.
- To test the rover’s turning dynamics, I swept angular velocities in intervals of 0.2 from −1 to 1. I found that the planner's arc model turned more sharply than the simulator.
- Applied a scaling heuristic—multiplying planner arc curvature by a constant factor to better match the flatter turning behavior of the simulated rover.

Arc Planner Example



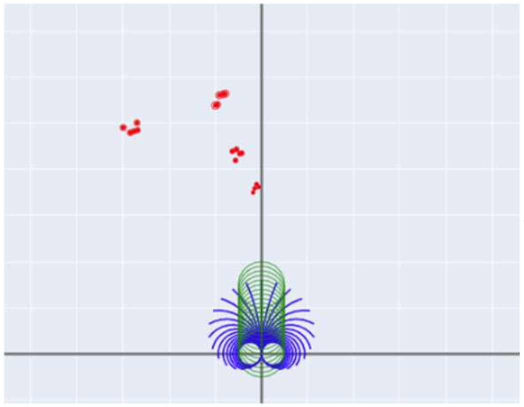
Rock Thresholds

- Another issue was the overly conservative rock detection threshold, which caused the rover to avoid even very small rocks (as small as 0.05 m).
- To address this, I increased the threshold to 0.15 m, which reduced unnecessary avoidance maneuvers.
- Eventually, through fine tuning, we found 0.08 m to be the most rock radius.



Temporal Path Planner

- Unlike the baseline planner that used only the current camera frame, this new approach integrates rock detections from the current frame and several past frames (typically the previous 5).
- This allows the rover to "remember" rocks that have left the field of view, leading to more informed decisions and improved obstacle avoidance.



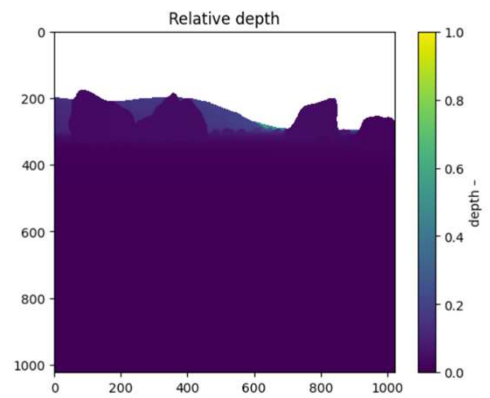
delta_pose = np.linalg.inv(pose_prev) @ pose_current

Results

- Found using 41 total paths in a range from -0.8 rad/s to 0.8 angular velocity, with a time horizon of 8 seconds resulted in the least collisions.
- Could not get Temporal Path Planner to significantly improve collision avoidance, since it did not solve collision avoidance for rocks that were not in the frame in the first place, and could also disrupt planning by creating a "wall of rocks" due to inaccurate odometry transformations of rocks.

Current Challenges and Future Works

- We identified monocular depth estimation and image segmentation as the next areas of development, rover’s side cameras are monocular.
- Could solve the problems that Temporal Path Planner attempted to solve
- Design of a monocular depth pipeline and potentially a novel architecture that jointly performs depth estimation and semantic segmentation, given the complementary nature of the two tasks.



Thank You!