Obstacle Detection for a Nuclear Pipe Inspection Robot
Final Report: Robot Autonomy

Josh Bennett	Henry Chen	Joe Phaneuf	David Evans	Adam Driscoll	Advisor:
jjbennet	zihanc	jphaneuf	dje1	jdriscoll	Red Whittaker

Background

There are several nuclear facilities across the United States that are preparing to be demolished. In order to reuse the facilities' land and for the health of surrounding residents, pipes containing nuclear contaminants must be removed and disposed of properly. The trick is not all pipes contain nuclear contaminants. In the interest of saving costs, demolition personnel need to map out the radiation intensity of the pipes. Mapping out the pipes is not only dangerous but difficult for a human to do accurately. Using a robot, the demolition personel can safely determine the radioactivity of the pipes and use this information to precisely remove the radioactive pipe. This is the goal of Radpiper, a nuclear pipe inspection robot.

Figure 1: Radpiper Platform

Radpiper is a project by Red Whittaker that is designed to drive down a straight pipe and produce a radiation map of the pipe. The robot uses a standard sodium iodide sensor to measure the radiation in the pipe, a laser and encoders to measure distance traveled, and a lidar to help navigate through the pipe. Figure 1 shows an image of the actual Radpiper platform.

A challenging task for Radpiper is perception of the inside of the pipe. The robot must be able to navigate safely through a pipe. It must identify untraversable obstacles in the pipe like mounds of sediment, holes in the bottom of the pipe, or broken pipe. It also must identify when it has reached the end of the pipe to prevent the robot from crashing into a wall or falling out of the pipe. Additionally, it must accurately estimate the distance to the obstacle to maximize coverage. To perceive the pipe, Radpiper currently uses the lidar sensor to generate a three-dimensional point cloud. Our team will use this point cloud to identify untraversable obstacles and end-of-pipe conditions for Radpiper.

Problem Statement

We will work to identify five untraversable obstacle conditions and two end-of-pipe conditions. The five untraversable obstacle conditions are large deposits, the soft Y, missing floor, junction, and pipe expansion. Each of the conditions are illustrated in Figure 2.

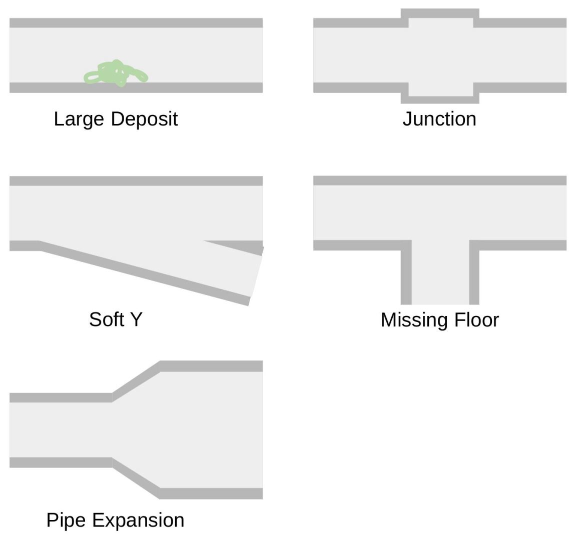


Figure 2: Pipe Obstacle Conditions

The two end-of-pipe conditions are a closed pipe and open pipe. Both conditions are illustrated in Figure 3.

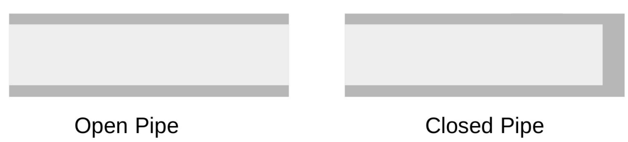


Figure 3: end-of-pipe Conditions

A key challenge to developing this system is making our algorithm efficient in terms of processing power and memory capacity. Radpiper's computer does not have a GPU or other powerful computing hardware, so we must be able to process the point cloud and make a decision with as little computational overhead as possible. An additional challenge to the project in general is the lack of testing infrastructure. We do not have access to the pipes at the nuclear facility, so it will be difficult to validate our algorithm on actual obstacle and end-of-pipe conditions. Instead, we will set up mock obstacle conditions inside of a sample pipe in the Field Robotics Center.

Methodology: System Outline

Our system takes in a point cloud from the Radpiper's lidar and returns a status to Radpiper indicating that it is safe to proceed, or the type of obstacle detected and distance to the obstacle if it is not safe to proceed.

We start by using the point cloud to find the equation of a cylinder that represents the pipe. If a cylinder cannot be generated with enough confidence, then we know we are nearing the end of the pipe. If a cylinder is able to fit the pipe, we can project the cylinder down the pipe and slice the point cloud into smaller segments. We can then inspect each of these segments closer. If the segments do not fit the cylinder well enough we can perform further analysis to determine the type of obstacle we are encountering. With this approach we can return both the type of obstacle found and the distance to the obstacle. Figure 4 shows a functional architecture describing the relationships between the components.

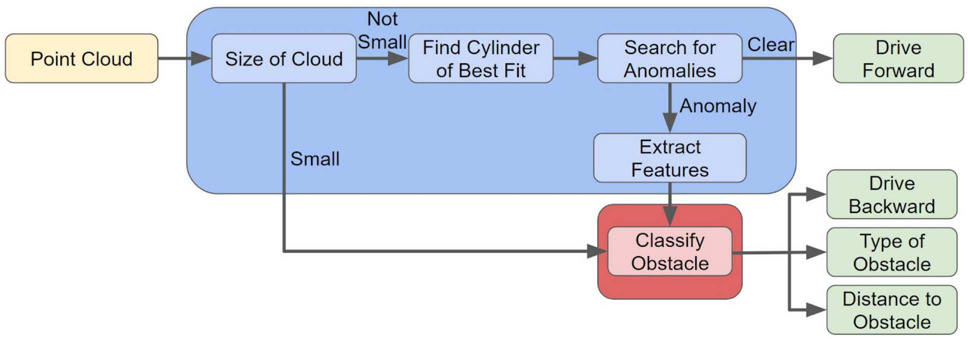


Figure 4: System Architecture Outline

Specifically, we have implemented all functionality within the blue area and red areas, with heavy utilization of the Point Cloud Library (PCL). The point cloud is already being generated by the Radpiper robot, and the Radpiper team will handle the planning and control given the obstacle status flags we are providing.

Methodology: Distance to Anomaly

Our system must be able to determine the distance from RadPiper to an anomaly in the pipe in order to determine the correct behavior. The anomalies we are interested in identifying are anything that causes the pipe to be non-cylindrical whether that's an the end of the pipe, a pile of untraversable sediment, or a hole in the bottom of the pipe. To do this, we sliced the pipe into n sections of our desired slice width resolution as shown in figure [insert figure number here]. We then want to find the nearest section with an anomaly and return the distance to that section.

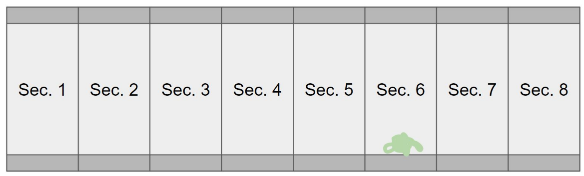


Figure 5: Pipe sliced into sections

Our team explored several strategies for searching these sections, each with its own advantages and disadvantages. When evaluating these strategies, our team valued low computational complexity and the strategy's consistency in providing the correct distance to the anomaly.

Strategy 1: Incremental RANSAC

The first strategy we evaluated was a naive RANSAC approach. Our team found that it was easy to identify an anomaly using RANSAC techniques by fitting a cylinder model to a point cloud section and checking the proportion of inliers fit by the model.

With this in mind, the first strategy incrementally performed RANSAC on each section of of the pipe checked the proportion of inliers to find sections that contained anomalies. We started at the nearest sliced section to RadPiper, fitted a cylinder model, and evaluated the fit to determine if the section contained an anomaly. We evaluate the fit by checking the proportion of points in the section fit by the model. If the proportion is below a certain threshold, then we classify the pipe section as containing and anomaly. If the section contains an anomaly, we return the distance to the anomaly. If no anomaly was found, we moved onto the next pipe section and continued the process.

The process can be shown in visually in figure 6. A slice near RadPiper is initially selected as shown highlighted in red. This section is fit by a cylinder model and evaluated. Because no anomaly is found in step 1, the system moves onto the pipe section in step 2. This continues until an anomaly is found as shown in step 4.

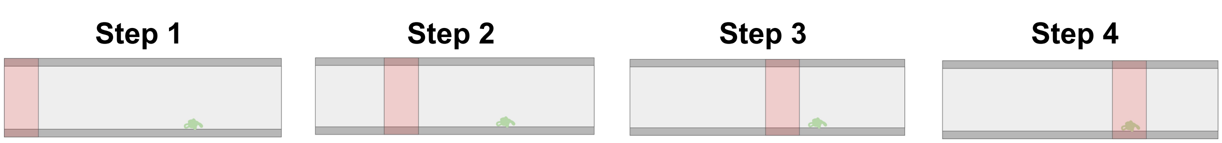


Figure 6: Incremental RANSAC Visualization

This strategy is attractive because it is straightforward to implement and provides consistent anomaly detection. Unfortunately, this strategy is computationally expensive because a RANSAC model must be fit to every section of your pipe that is your resolution size. With this in mind, we moved onto strategy 2.

Strategy 2: RANSAC and Project

Our team thought performing RANSAC on every single section of pipe was unnecessary. Instead we could use RANSAC to fit a single model across several sections of the pipe to fit a cylinder model to a large section of the pipe. We could then check each section of pipe against this cylinder model.

Visually, this is shown in figure 7. An initial cylinder model is fit on pipe with RANSAC. This model this then stepped forward and checked against other pipe sections shown in yellow. This continues until an anomaly is found.

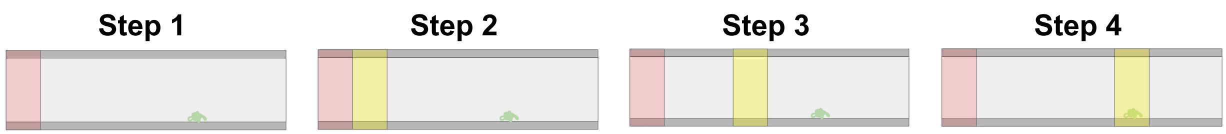


Figure 7: RANSAC and Project Visualization

This strategy is significantly more computationally efficient than the incremental RANSAC approach, however this model is not robust variations in the cylinder. If the initial RANSAC cylinder fit is slightly off, like shown below, then the cylinder model will not project forward along the correct pipe axis. This causes a the cylinder model to not be representative of the correct pipe section, and thus will inaccurately trigger an anomaly reading on a free section of pipe. Figure 8 shows the inaccurate forward projection of the cylinder model.

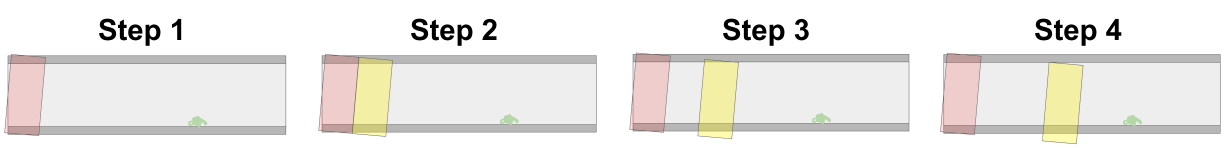


Figure 8: RANSAC and Project Failure

To meet this shortcoming, our team next looked at a strategy to take advantage of the ability of RANSAC to consistently fit a cylinder model while reducing computational complexity. We arrived at strategy 3.

Strategy 3: Binary-search RANSAC

Given a pipe section, RANSAC can accurately determine if the section represents a cylinder. However RANSAC is not good for determining if another section it has not seen represents a cylinder. Therefore, we want to run RANSAC as few times as possible.

Our solution is to perform a binary search along the pipe, performing RANSAC across various lengths of pipe to eventually identify a section of pipe with an anomaly. We do this by first fitting a cylinder model to the entire pipe. If the model meets the inlier proportion threshold, we trust there are no obstacles in any of the sections of pipe. If the model fails the inlier proportion threshold, we then split the sections into a forward and and back half of the pipe and perform RANSAC on each section. This RANSAC, evaluate, divide process continues until a section of our minimum slice width fails our cylinder fit.

Visually this is shown in figure 9. The entire pipe is initially fit with a cylinder model by RANSAC as shown in red. The division of this section is shown by a smaller section being fit next, and this continues until the anomaly is found in our minimum slice width.

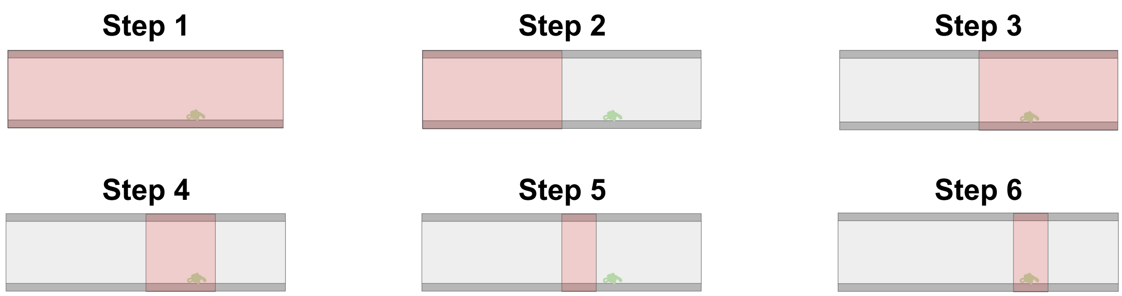


Figure 9: Binary Search RANSAC Visualization

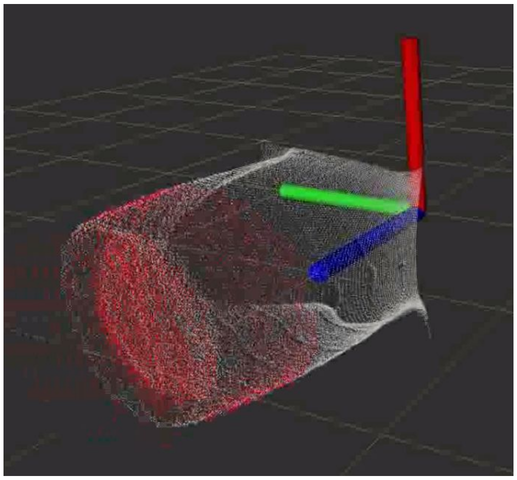
Because we used RANSAC to fit individual sections of the pipe, we were able to achive much more consistent anomaly detection results while still reducing our computational complexity by performing a binary search. For these reasons, we implemented the binary-search RANSAC approach in our final system.

Methodology: Classification

The distance to anomaly subsystem provides an estimate of the coefficients describing the pipe as a cylinder by means of RANSAC. When an anomaly is detected, a slice of the input point cloud near the anomaly is extracted. The cylinder coefficients provide a cylinder axis and cylinder radius.

From these parameters, points in the extracted point cloud can be classified as being part of the cylinder, inside the cylinder, or outside the cylinder. Points that are part of the cylinder fall within a threshold distance of the modelled radius. Points corresponding to positive obstacles and closed pipe conditions fall under a threshold distance of the modelled radius. Points corresponding to negative obstacles and and open pipe conditions fall above a threshold distance of the modelled radius.

Figure 10 shows extracted point clouds for classification for closed and open pipe conditions. Figure 11 shows histograms of radial residuals for closed and open pipe conditions.



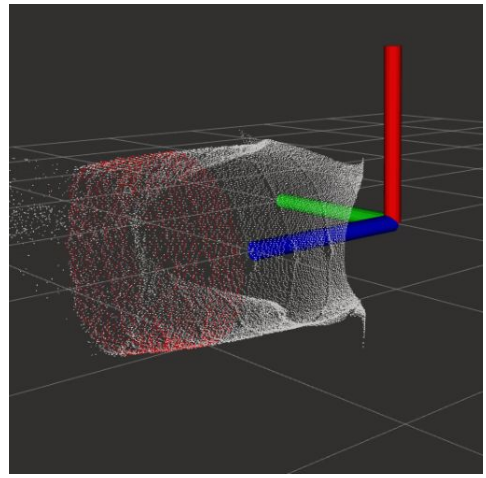
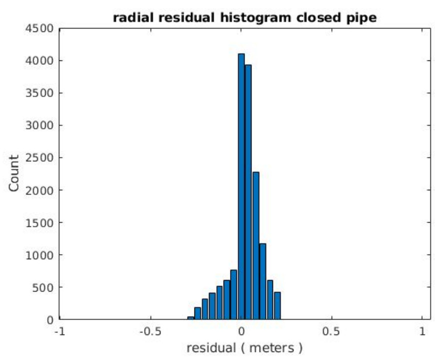


Figure 10: Closed (left) and open (right) pipe nomaly point cloud (red) extracted from input point clouds (white)



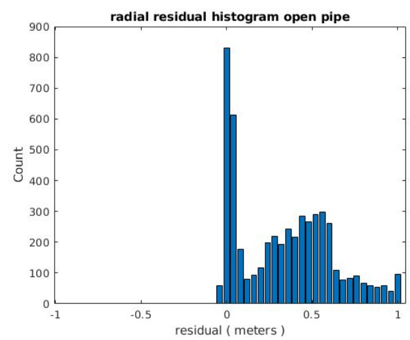


Figure 11: Point-Radius residual histograms for closed pipe (left) and open pipe (right)

After a point cloud containing an anomalous condition is extracted, and radial inliers extracted from that, the percentage of outliers is used to classify open pipe, closed pipe, and positive obstacle events.

Results:

Using the methodology described above, we were able to correctly locate and identify the three stop conditions for which data was provided. Using the pointcloud, the location of the hazard was calculated with a resolution of 0.125 meters. To benchmark the classification subsystem, 20 seconds of each available bagged data file (open pipe , closed pipe, obstacle in closed pipe, 2 instances of obstacle in open pipe) was processed by our algorithm and classifications compared against ground truth labels. Table 1 shows the resulting confusion matrix for classifications.

Figures 12 and 13 show snapshots of the algorithm in action on open and closed pipes. The input and classified point clouds (white and red respectively) are shown adjacent to a prediction output and the classification definitions. The prediction outputs show RadPiper safeguard messages, of which only the decision and distance are relevant for this project. Distances are in meters, relative to the origin of the sensor. Ground truth data for distance measurements is unavailable for further analysis.

Predicted Actual	**Closed**	**Open**	**Positive Obstacle**
Closed	52	0	0
Open	0	80	8
Positive Obstacle	0	7	188
Total Accuracy	**95.5%**		

Table 1: Anomaly classification confusion matrix

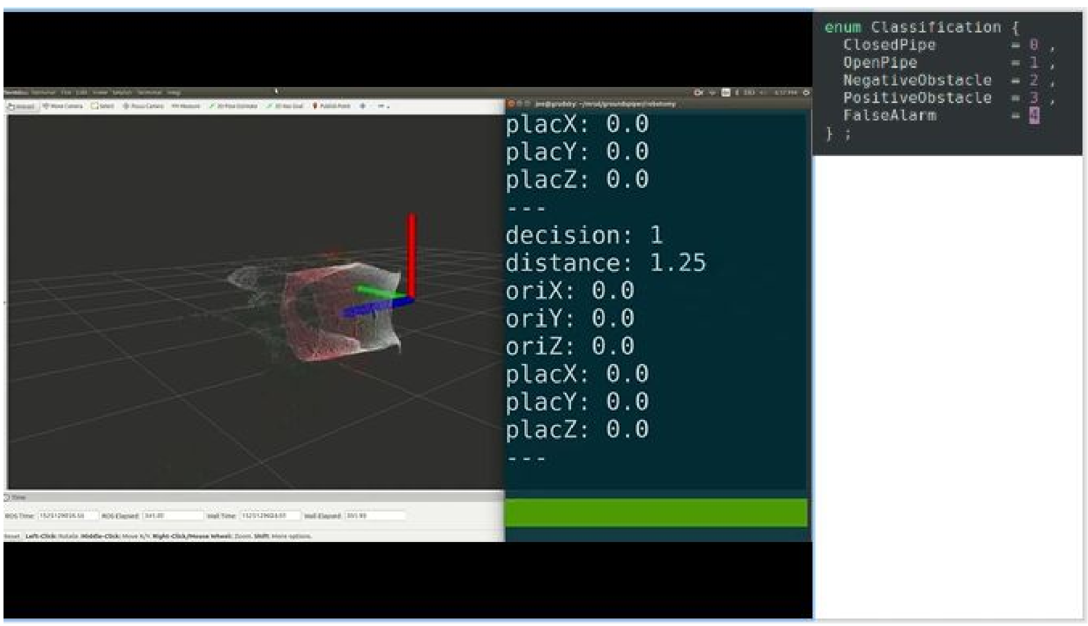


Figure 12: An open pipe condition being localized and classified.

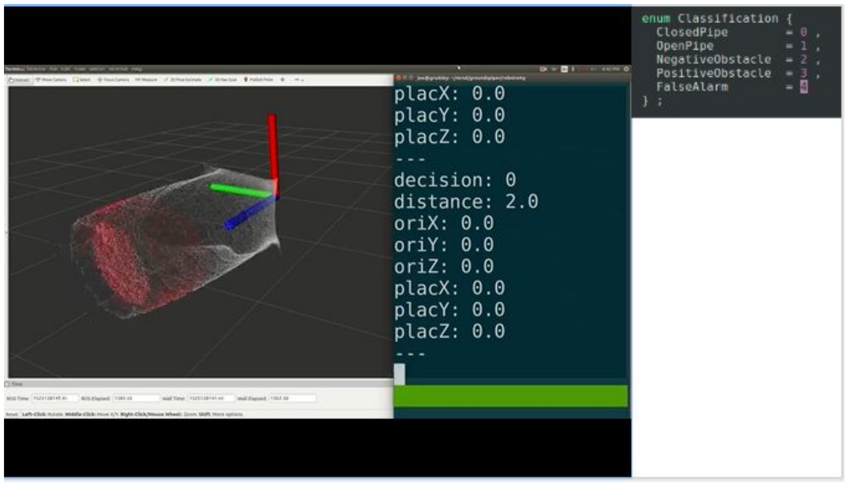


Figure 13: A closed pipe condition being localized and classified

Future Work:

In the future, downsampling of the pointcloud via a voxelgrid can be used to improve the speed of the processing and make the system more robust to noise.

A custom RANSAC model that also takes the length of the cylinder into account may also be useful, as it would provide the longest section of pipe that is appears to be unobstructed. This prevents the need to perform RANSAC multiple times on different lengths of pipe, since the end of the pipe outputted by the custom model would indicate the beginning of the obstacle.

Finally, the point cloud used for detection and classification of anomalies is derived from a depth image. Using only a depth image, geometric image based methods can be used to anomalies within the pipe, especially in the case of a positive obstacle. This will also avoid the need to project the depth image pixels into a 3D point cloud, saving compute time.

Conclusion

We have generated a methodology for efficiently detecting, classifying and measuring distance to anomalies in radioactive pipes. The first improvement over existing work is generic anomaly detection, allowing a single classification step for each new input point cloud. The classification step is structured for ease of implementing new classifications, as the system requirements change in the future. The second critical contribution is speed of anomaly detection, where the binary RANSAC search method will run a single RANSAC fit in non-anomalous pipe conditions.

Final Video:
https://youtu.be/31fhP1hTfz4