# Indoor Robotics

## Performance of Automotive Sensors in Indoor Settings

**Babak Farahmand, Daniel Luft-Martinez, Mayowa Borisade, Paige Riola, Shealtiel Mulder, Walter Condori**

**Special thanks to:**

**Shuhei Takahashi**, **Josh Frankfurth**, **Jolton Dsouza** & **Rob Tucker** of Continental,
Our professor **Richard Jullig**,
Our TAs **Chandraniel** & **Arindam**,
And **Veronica Hovanessian**!

## Abstract

We experimented with a fusion of automotive lidar, radar, and camera sensors in order to evaluate their potential for indoor use. These sensors are integrated with a Husky UGV robot; this setup allows us to collect data for tasks such as object detection & classification. We then evaluate the accuracy of our results by comparing our finalized object classifications to the actual environment.

## Overview

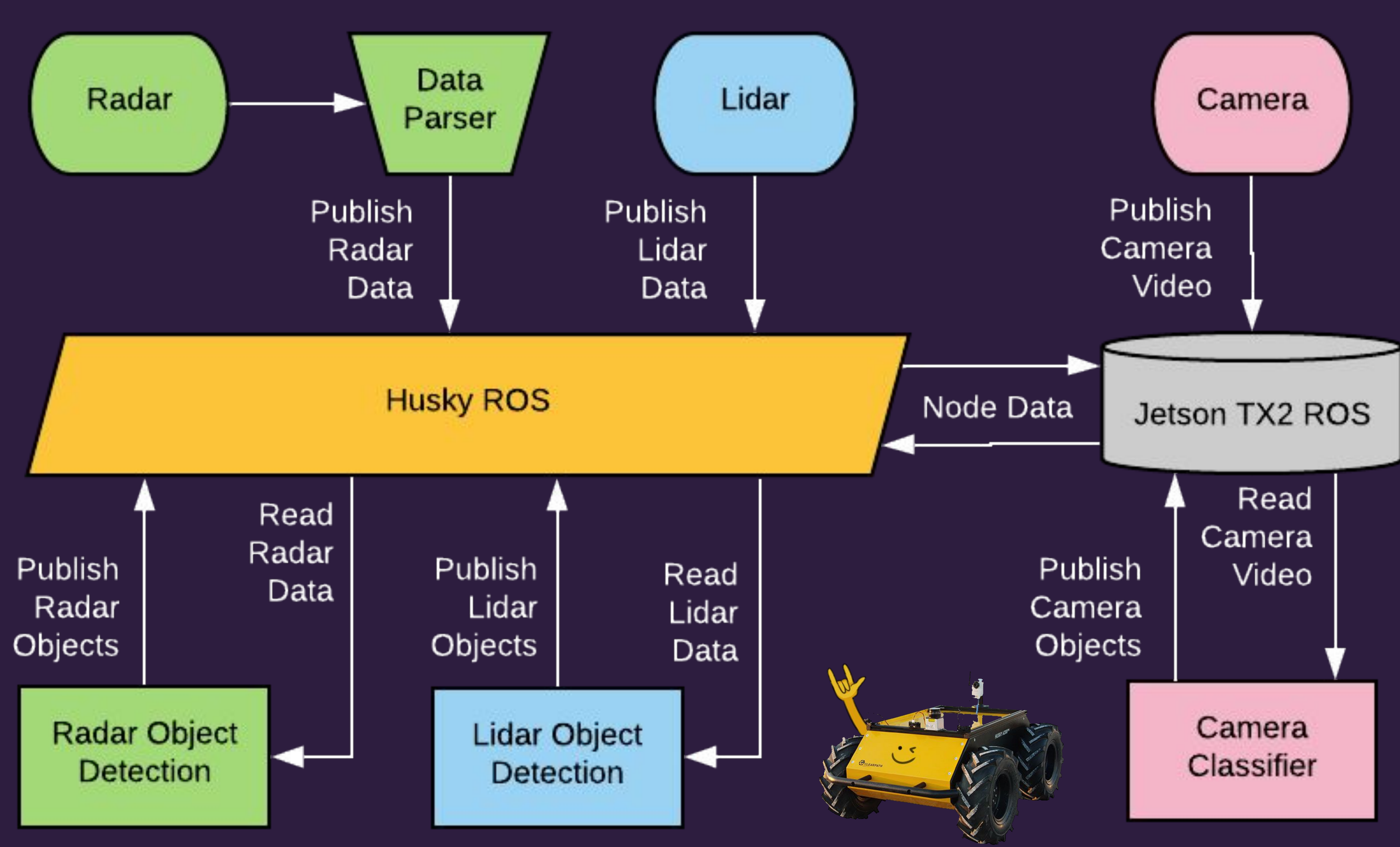
**Continental** is an automotive parts manufacturing company with a variety of **sensors** suited for tasks such as collision detection and adaptive cruise control. With an expanding robotics market, Continental is interested in marketing these same devices for use in **indoor environments**. Our goal is to analyze the performance of Continental sensors, in indoor settings, and determine the **best combination of their sensors** for indoor robotics use.

## Challenges

We began with the pretense of having physical access to the hardware. However, we faced many setbacks in acquiring and accessing the hardware, including legal delays for on-campus hardware access, and and campus closure due to COVID-19. Each new setback changed our workflow, shifted our goals, and reset our progress. We only managed partial hardware access towards the project's end. In response, we shifted focus to our data processing methods.

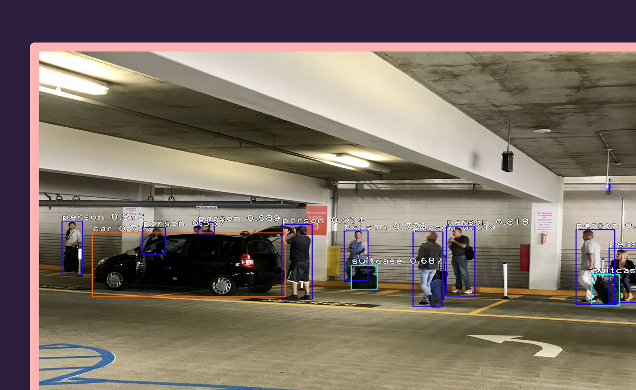
## Approach

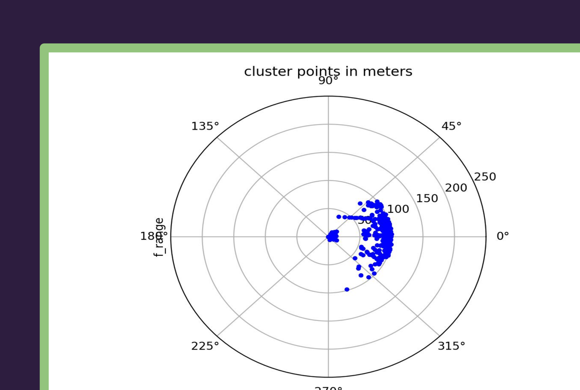
Data from our sensors is transferred to our middleware, ROS (Robot Operating System), located both on the Husky and the Jetson TX2, an external AI-computing device. ROS consolidates our sensor data for processing multiple data streams simultaneously. Each of the sensors publishes it's data to ROS through executables called nodes, which follow a publisher-subscriber pattern. We then perform radar-lidar object detection and live video classification, with the results published back to ROS. Ultimately, we're combining the data to map object labels to a 3D model of the environment within ROS. We can then determine the accuracy of the model by comparing it to the actual environment.





Edge Detection


Object Classification


Radar Visualization

## Conclusion

We were able to accomplish a variety of processing methods. For the camera, we implemented live video edge detection and object classification. In addition, we achieved video data streaming within ROS, which we can apply these processes to. For radar, we were able to parse the raw data directly from our radar sensor, produce a 2D visualization of the data, and detect objects. For lidar, we implemented 2D mapping within a simulated environment.

Given more time, we would integrate all of our processes into a single ROS-centered system, and produce a full 3D visualization of our processed data within ROS. Upgrading to the most recent ROS release would simplify the integration of our Python 3 code with our overall system.