Self-Driving Vehicle: Final Presentation

Group Members: Sandeep Alankar, Zhuohuan Li, Anthony Siu, Adas Bankauskas, Malav Majmudar, Abia Mallick, Aayush Agnihotri, Lohith Bodipati

Zhuohuan Li (GR)

Anthony Siu (UG)

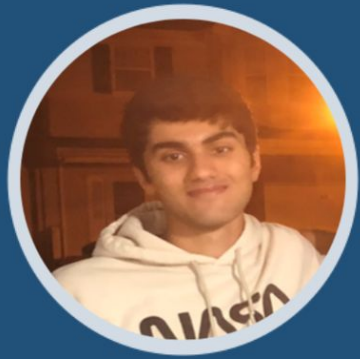
Sandeep Alankar (UG)

Lohith Bodipati (HS)

Adas Bankauskas (UG)

Abia Mallick (UG)

Malav Majmudar (UG)



Aayush Agnihotri (HS)

Who we are

Useful Terms

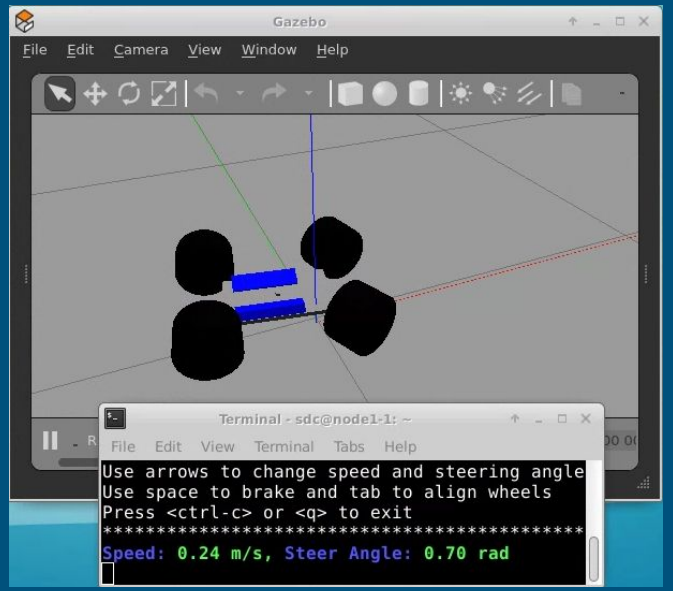
- ROS
  - Robot Operating System
  - Set of software libraries and tools used to build robot applications
- Chrome Remote Desktop
  - Allows a user to remotely access another computer
- Gazebo
  - 3D simulator that offers the ability to simulate robots operating in complex, digital environments
- Neural Network
  - Computational learning system that uses a network of different functions to understand and translate a data input of one form into a desired output

Project Objectives

- Build a fully functional self-driving vehicle
- Incorporate ROS control into simulated car software
- Write AI/machine learning algorithms for self-driving behavior
- Use Gazebo to map out simulations
- Build a physical model at WINLAB and test its autonomy in a real environment

Gazebo Simulator

- Created basic self-driving model in Gazebo
- Tested Ackermann steering
- Controlled digital model with keyop.py script
  - Adjustable speed and steering angle



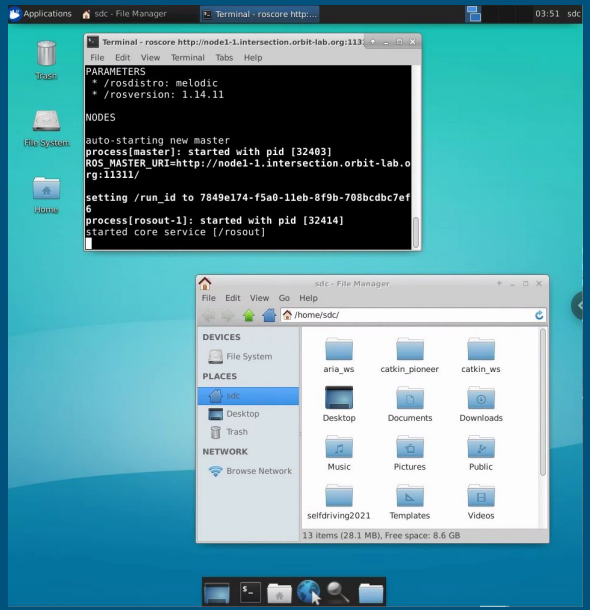
Combining RealSense Point Clouds

- Accessed four RealSense camera positioned around model city intersection
- Created 3D image from each perspective and started to experiment with combining images by creating and transforming individual point clouds
  - Affine transform

Pioneer 3-DX

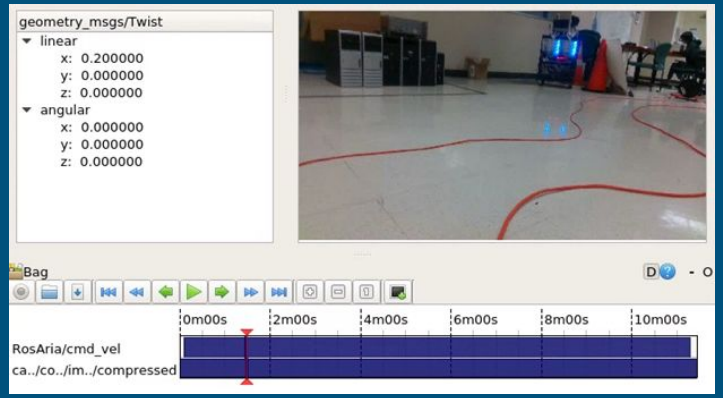
- The primary robot used to collect data
  - Contains onboard sensors, ROS compatibility, remote control
- Directly controlled using RosAria interface
- RealSense Depth Camera attached to top

```
**********************************************************************
*                    ROSARIA CLIENT INTERFACE                        *
*                                                                    *
*             Welcome to the RosAria client interface!               *
*                                                                    *
*        [1] go_three_second                                         *
*        [2] spin_clockwise                                          *
*        [3] spin_counterclockwise                                   *
*        [4] teleop                                                  *
*        [5] enable/disable print_state                              *
*        [6] enable_motors                                           *
*        Press [Q] to close the interface                            *
**********************************************************************
```

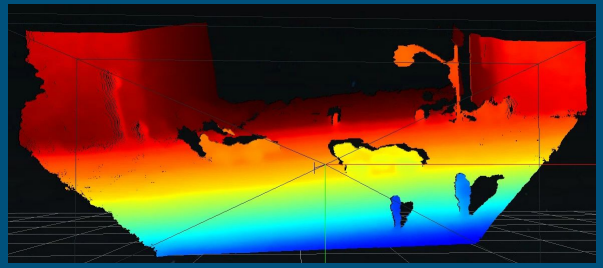


Recording Training Data

- Using RosAria to wirelessly drive Pioneer 3-DX
- Recording bag file using Rosbag package
- Subscribed to control and image topics
  - Rosaria/cmd_vel
  - camera/depth/image_raw/compressed
- Converted bag files to image (.ndz) files that were fed into 4 convolutional layers of neural network



Future Plans



- The depth sensor of the RealSense camera could provide additional training data
- Reduce the number of clients necessary to communicate between the user and the robot
- Allow for remote subscription to camera without sacrificing robot control security
- Configure RosAria onto smaller mobile robots and test self-driving behavior alongside Pioneer 3-DX in city intersection

Special Thanks

Project Advisor:

Jennifer Shane

Internship Supervisor:

Ivan Seskar

Any Questions?