



# PACMAN

**P**ath **A**ssembling **C**artographer, **M**apping under **A**utonomous **N**avigation

• • •

**T**eam 5 - Very Good Robotics

Adam Romlein, Avery Oefinger, Max Cutugno, David Russell

# Purpose / Problem Statement

- Navigate sidewalks around campus autonomously
- Create HD Maps
- Geolocate sidewalks
- Collect data for other autonomous devices on campus (RT 1000 autonomous snow plow)



# Sensors

# Swiftnav Duro

- Real-Time Kinematic GPS
- Meter resolution, can use Base Station on campus to get cm resolution
- Ruggedized to IP-67, military-grade
- Integrated IMU



# Ouster OS1-16

- 360 degree, 16 beam LiDAR with 33 degree FoV
- 120m range
- Useful for Mapping and collision detection
- Integrated IMU



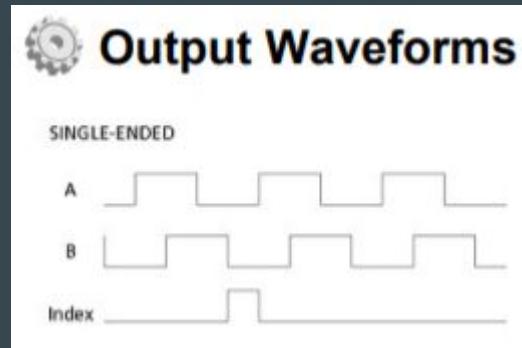
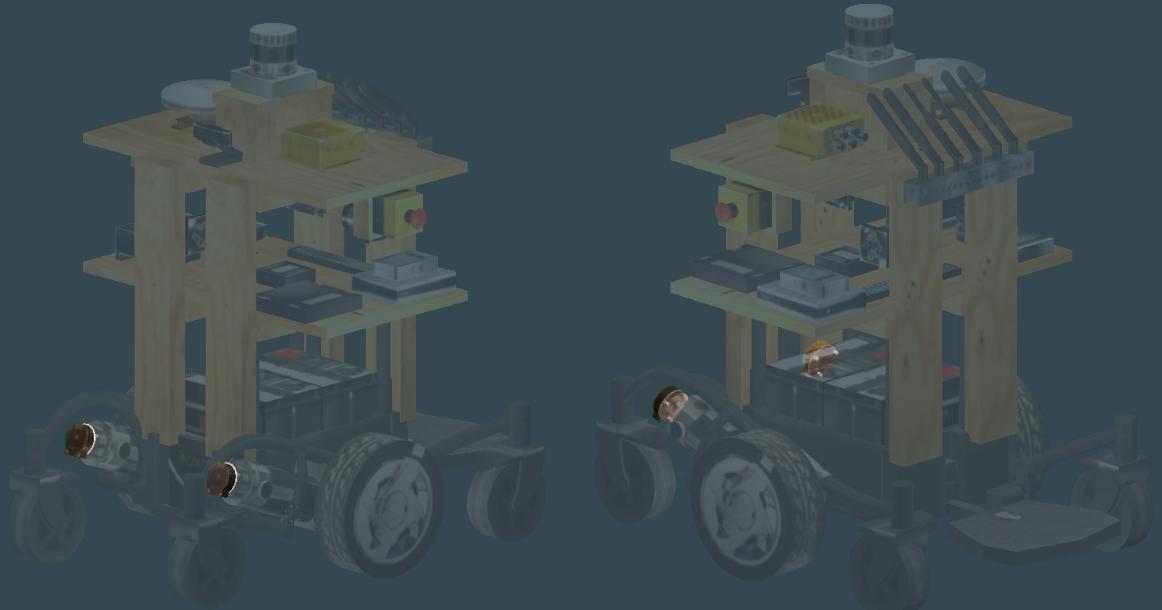
# Logitech C920

- Standard UHD Webcam
- Used for navigation of sidewalks
- All Computer Vision is through this



# US Digital Optical Encoders

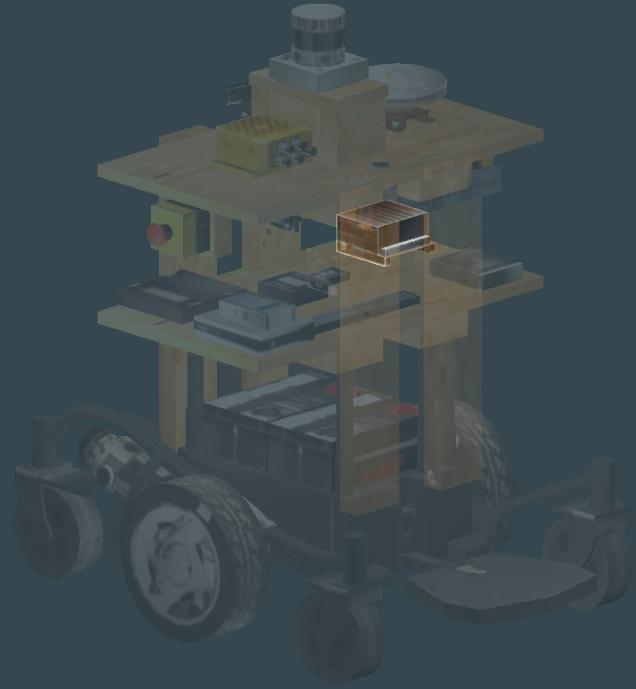
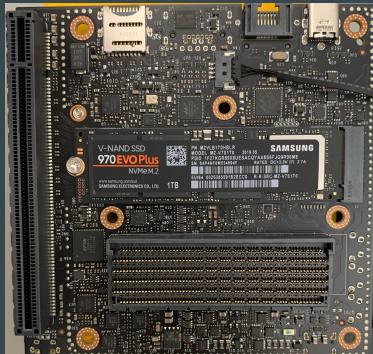
- Quadrature Encoder
- 2048 pulses/rev resolution
- Dual Phase channel A & B



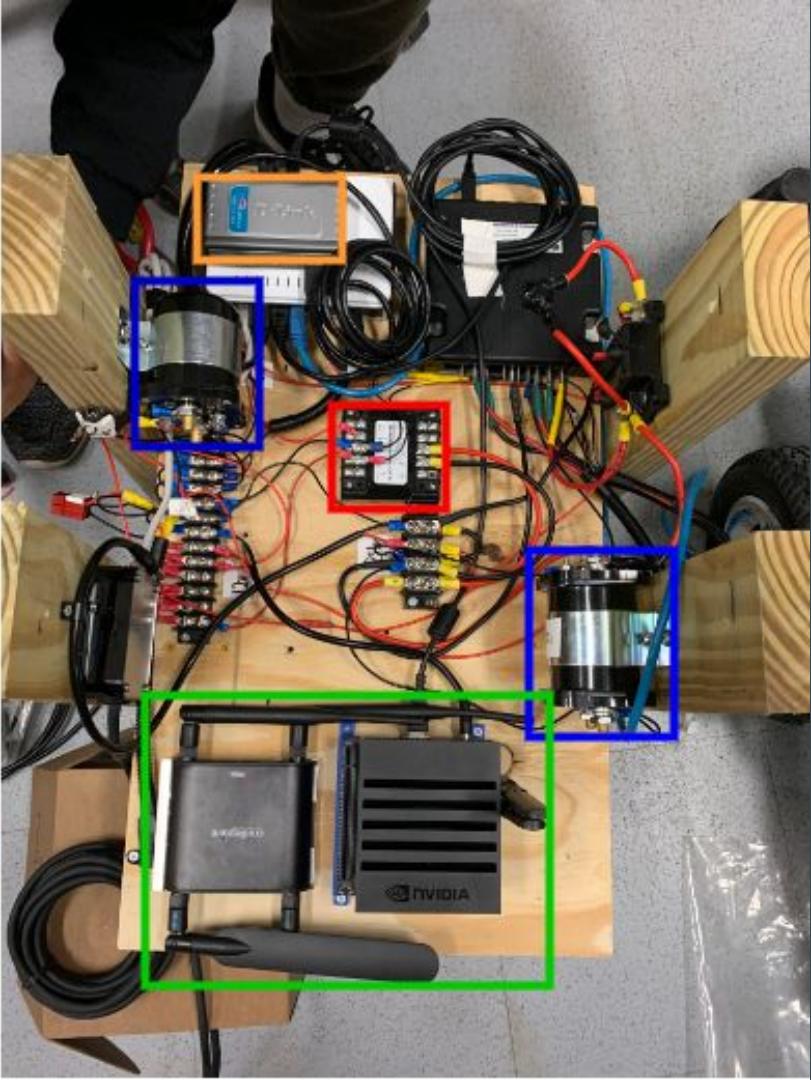
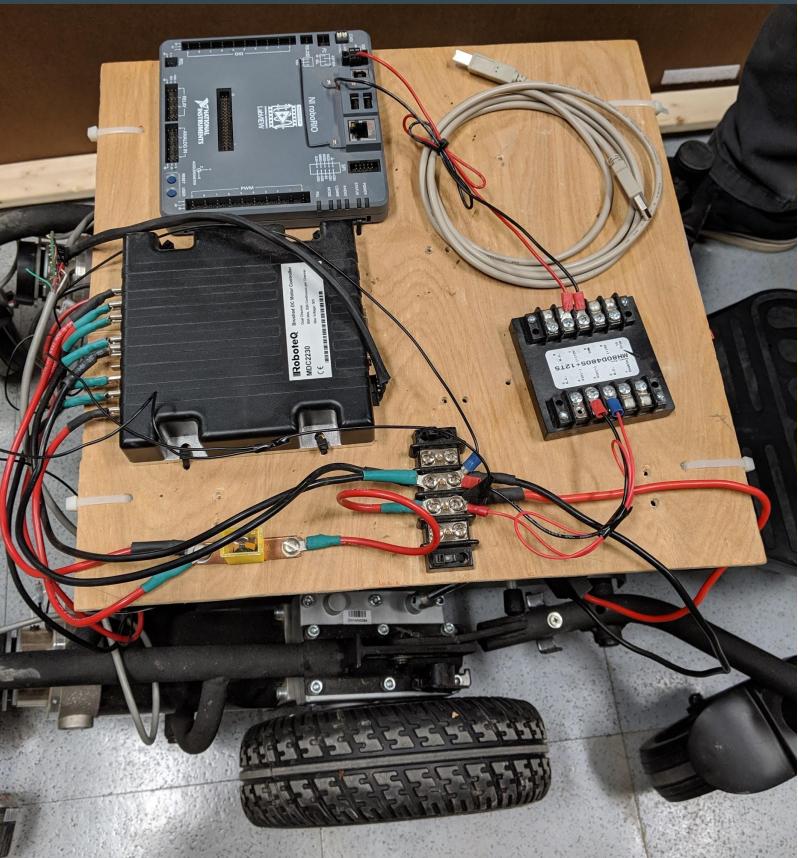
# Hardware

# Jetson Xavier

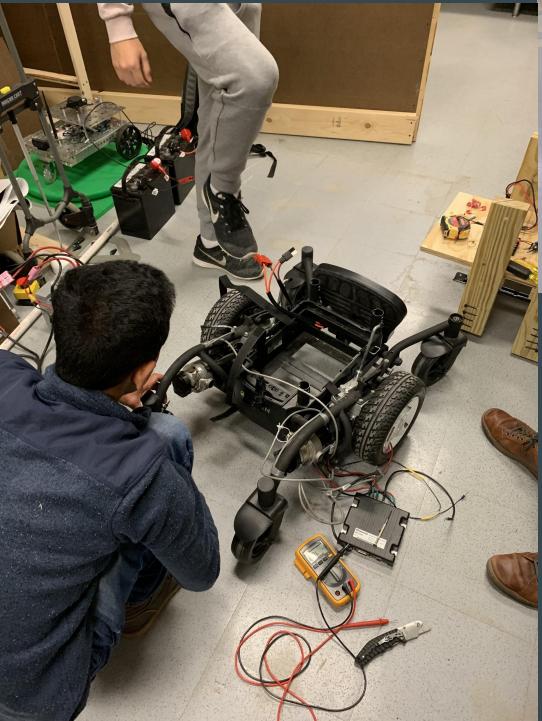
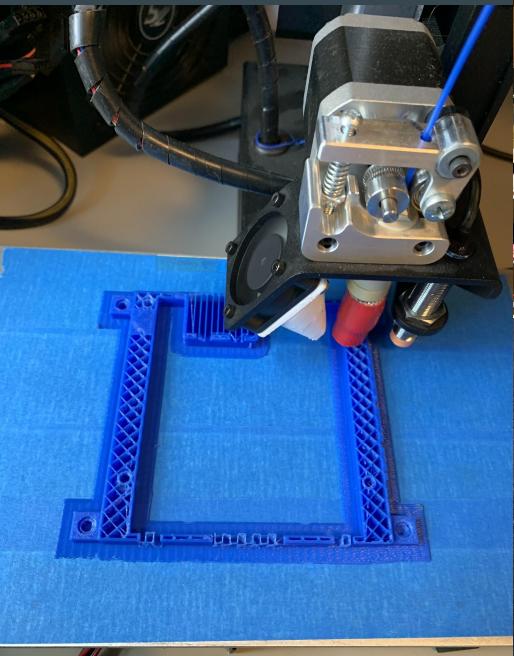
- Head Computing Node
- Excellent embedded system for AI and Computer Vision with onboard GPU
- Runs a full Linux environment



# Frame

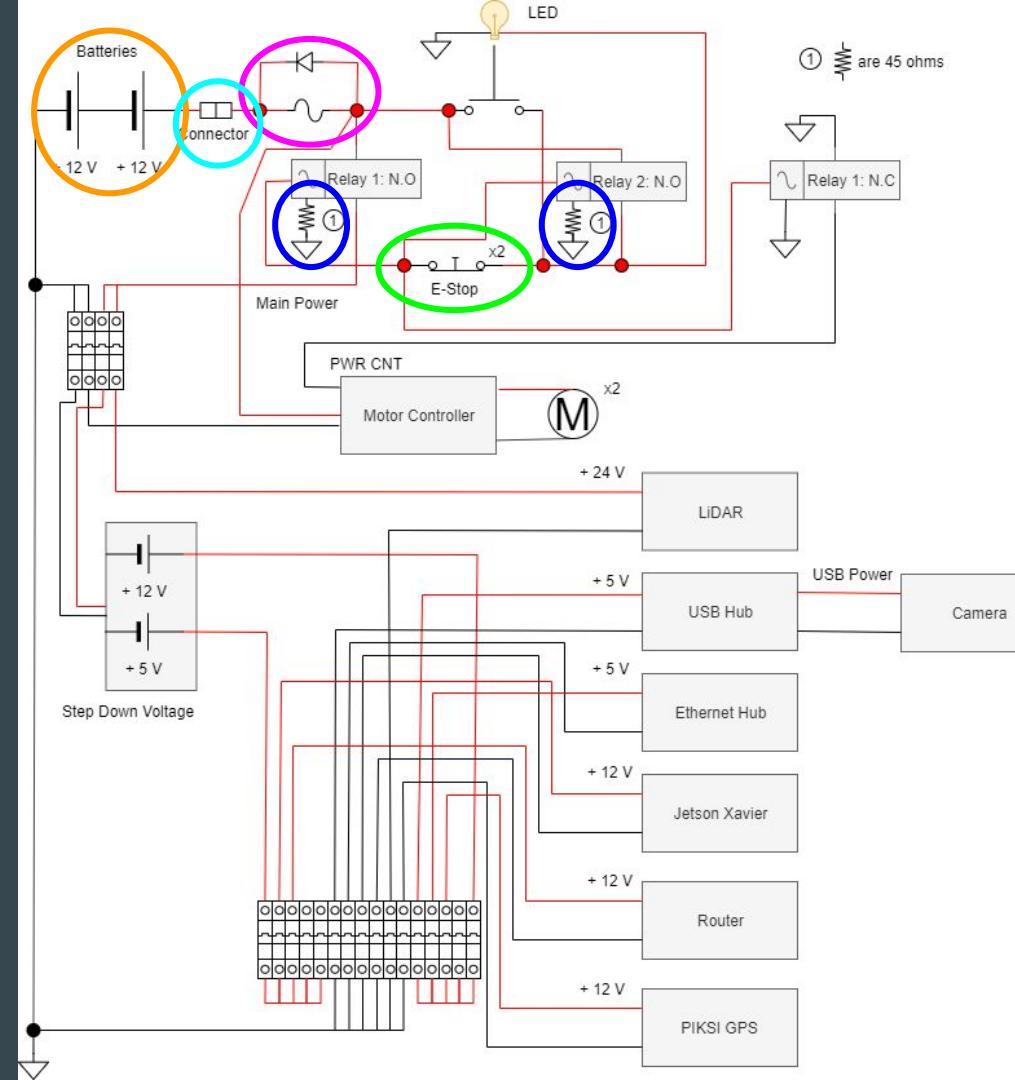
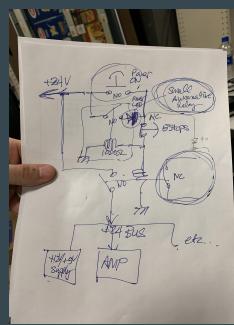
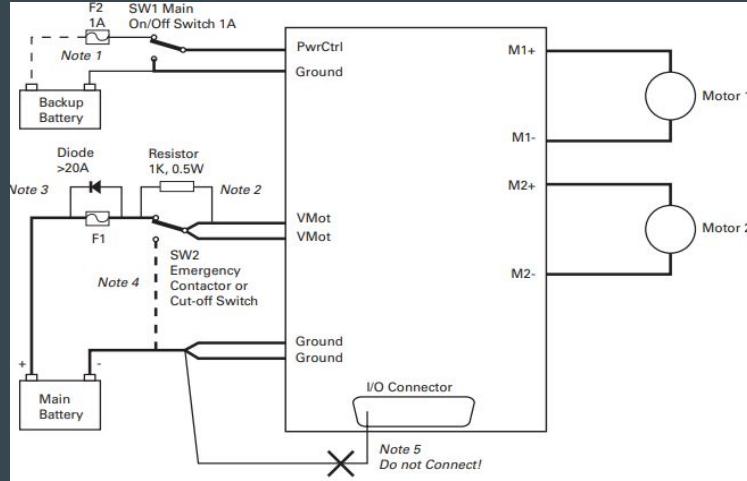


# Development



# Power Circuit

- Safety
- Heat Dispersion
- Reliable Power



# Expenses

Prices

Frame: \$32.52

Electronics: \$961.85

Components: \$13,765.12

- These prices are how much it would take to completely rebuild the project. A lot of these parts were already owned.

					Total Price:	14759.49
					Total Price	Link
<b>Frame:</b>						<b>32.52</b>
Pressure treated wood 2-4-8	468930	Lowe's	2	4.47	8.94	
Sanded pine plywood 23/32-2-4	7710	Lowe's	1	13.62	13.62	
Black cable ties (20)	230994	Lowe's	1	3.98	3.98	
Deck screws 10x3 (#)	755730	Lowe's	1	5.98	5.98	
<b>Electronics:</b>						<b>961.85</b>
Breaker	1851205	Amazon	1	25.89	25.89	<a href="https://www.amazon.com">https://www.amazon.com</a>
Diode	1N1 183A	SSDI	1	4.38	4.38	<a href="https://www.sdi.com">https://www.sdi.com</a>
Red & Black bundle of wire (10 guage)	8541758680	Amazon	1	23.95	23.95	<a href="https://www.amazon.com">https://www.amazon.com</a>
Red bundle of wire (18 guage)	55667423	Amazon	1	7.36	7.36	<a href="https://www.amazon.com">https://www.amazon.com</a>
Black bundle of wire (18 guage)	55667323	Amazon	1	8.28	8.28	<a href="https://www.amazon.com">https://www.amazon.com</a>
Relay (N.O. and N.C. ends)	WR586RS-24	Ebay	1	85.00	85.00	<a href="https://www.ebay.com">https://www.ebay.com</a>
Relay (N.O. end)	WR586	Ebay	1	65.00	65.00	<a href="https://www.ebay.com">https://www.ebay.com</a>
Barrel jack connectors (2.5x5.5) (2)	4330127896	Amazon	1	19.40	19.40	<a href="https://www.amazon.com">https://www.amazon.com</a>
Wire tube 1/2" 25 ft	B07TCDTFL2	Amazon	1	9.50	9.50	<a href="https://www.amazon.com">https://www.amazon.com</a>
Shrink tube kit	B071H5XC7C	Amazon	1	9.98	9.98	<a href="https://www.amazon.com">https://www.amazon.com</a>
Emergency stop buttons	43217-5544	Amazon	2	10.89	21.78	<a href="https://www.amazon.com">https://www.amazon.com</a>
24/19v DC/DC converter	MWX-24-19-190W	Amazon	1	67.00	67.00	<a href="https://www.amazon.com">https://www.amazon.com</a>
24/12v DC/DC converter	B0756W6V4F	Amazon	1	12.99	12.99	<a href="https://www.amazon.com">https://www.amazon.com</a>
24/5v DC/DC converter	8-40T5-10A	Amazon	1	15.99	15.99	<a href="https://www.amazon.com">https://www.amazon.com</a>
Battery power connectors	B07Q5J8BWH	Amazon	1	9.60	9.60	<a href="https://www.amazon.com">https://www.amazon.com</a>
Batteries	M22NF-SLD-A	<a href="https://www.batterystuff.com">BatteryStuff.com</a>	2	257.00	514.00	<a href="https://www.batterystuff.com">https://www.batterystuff.com</a>
Terminal wire strip (2x4)	204586	Defender	1	9.99	9.99	<a href="https://www.defender.com">https://www.defender.com</a>
Terminal wire strip (2x#10)	202095	Defender	2	14.99	29.98	<a href="https://www.defender.com">https://www.defender.com</a>
Push button	43217-5544	Amazon	2	10.89	21.78	<a href="https://www.amazon.com">https://www.amazon.com</a>
<b>Hardware:</b>						<b>13765.12</b>
Logitec camera	960000764	Amazon	1	56.00	56.00	<a href="https://www.amazon.com">https://www.amazon.com</a>
Playstation controller	B01LWVX2RG	Amazon	1	43.95	43.95	<a href="https://www.amazon.com">https://www.amazon.com</a>
USB hub	ST4200USBM	Amazon	1	51.70	51.70	<a href="https://www.amazon.com">https://www.amazon.com</a>
Ethernet hub	DAP-1522	Amazon	1	39.99	39.99	<a href="https://www.amazon.com">https://www.amazon.com</a>
Cradlepoint technologies antenna (replacement)	EWUA0052	Amazon	1	10.29	10.29	<a href="https://www.amazon.com">https://www.amazon.com</a>
LIDAR	OS1-16	Ouster	1	3500.00	3500.00	<a href="https://www.ouster.com">https://www.ouster.com</a>
Jetson Xavier	945-82972-0000-000	NVIDIA	1	699.99	699.99	<a href="https://www.nvidia.com">https://www.nvidia.com</a>
Cradlepoint Router	IBR600LE-VZ	Amazon	1	269.00	269.00	<a href="https://www.amazon.com">https://www.amazon.com</a>
PIKSI Duro GPS	Duro Inertial	Swiftnav	1	3995.00	3995.00	<a href="https://www.swiftnav.com">https://www.swiftnav.com</a>
RoboteQ Motor Controller	MDC2230	SuperDroidRobots	1	340.00	340.00	<a href="https://www.superdroidrobots.com">https://www.superdroidrobots.com</a>
Power chair wheel base	Compass HD	Golden Tech.	1	4599.00	4599.00	<a href="https://www.goldentech.com">https://www.goldentech.com</a>
Wheel encoders	E3-512-394-IE-D-D-3	USdigital	2	80.10	160.20	<a href="https://www.usdigital.com">https://www.usdigital.com</a>

# Comparison

- The Clearpath Husky is a ROS platform
  - Does not have sensors (just encoders)

	Husky	PACMAN
Cost	\$24,000	\$14,760
Max load capacity	75 lbs	450 lbs
Speed	1.0 m/s	1.66 m/s



**Husky A200™ UGV mobile base**

**22 000,00€**

**ADD TO CART**

# Software

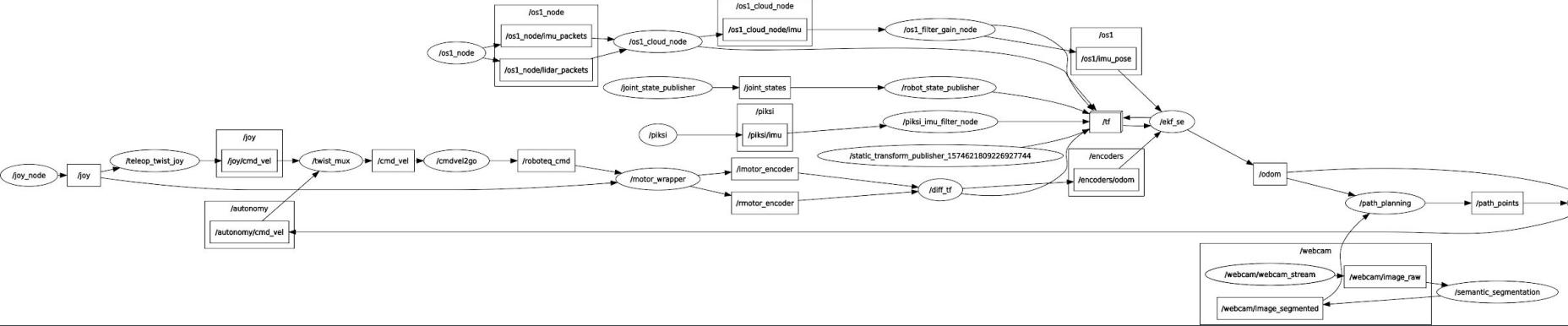
# ROS (Robot Operating System)

- Entire system is developed through ROS
- ROS is a message-passing middleware
- Useful for abstracting driver integration from higher-level development
- Can develop in both C++ and Python - each process is a node



Open Source Robotics Foundation

# Full Node Diagram



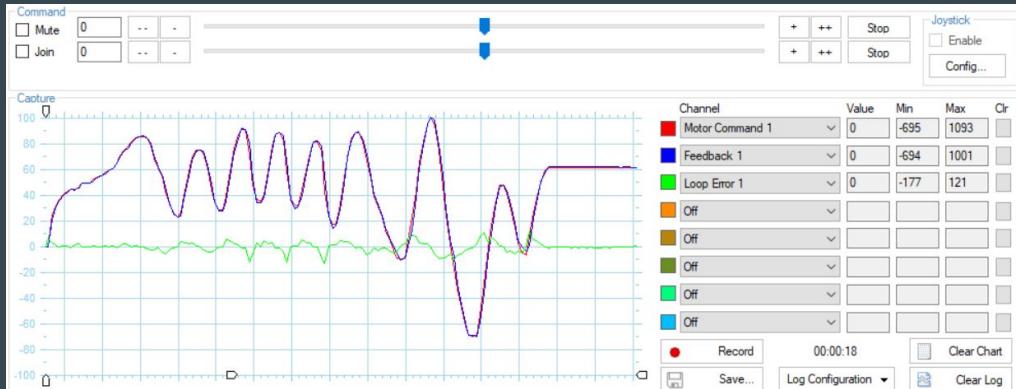


# Motor Controller Custom Driver

- Several ROS Drivers were tested first, none satisfactory
- So, we had to write our own
- Jake Schechter used API from Roboteq to implement core “GO” functionality
- We expanded with feedback and status updates

# Motor Controller Utility (Roborun+)

- Used utility for encoder PID control.
  - Image of RoboRun Utility and graph of wheel response to encoders
- Used utility to convert “go units” to quantitative speed (m/s).  
Ended up using qualitative data to achieve result: timed experiment of 50 turns
- Gear ratio (GR) of motors: 26.25 (ideal)  
After experiment was 28.608 (8.98% error)



The figure shows the RoboRun Utility software interface. On the left, the 'Motor Configuration' section is expanded, showing parameters like Max Speed (RPM), Acceleration (RPM/s), Deceleration (RPM/s), and Fault Deceleration (RPM/s). Below it, the 'Closed Loop Parameters' section is expanded, showing Position Mode Velocity (RPM), Turns Min to Max, Proportional Gain, Integral Gain, Differential Gain, Integrator Limit (%), and Loop Error Detection. On the right, the 'Encoders' section is expanded, showing details for Encoder 1 and Encoder 2, including Use, Pulses/Rev, Min Limit, Max Limit, Home Count, Action at Min, and Action at Max. The 'Motor 1' section is also visible at the bottom.

**Motor Configuration**

- Max Speed (RPM): 3550
- Acceleration (RPM/s): 5000.0
- Deceleration (RPM/s): 5000.0
- Fault Deceleration (RPM/s): 50000.0

**Closed Loop Parameters**

- Position Mode Velocity (RPM): 1000
- Turns Min to Max: 0.0
- Proportional Gain: 0.25
- Integral Gain: 10.0
- Differential Gain: 0.0
- Integrator Limit (%): 100
- Loop Error Detection: Disabled

**Encoders**

**Encoder 1**

- Use: Feedback (1)
- Pulses/Rev: -512
- Min Limit: -2147483648
- Max Limit: 2147483647
- Home Count: 0
- Action at Min: No Action
- Action at Max: No Action

**Encoder 2**

**Motor 1**

# Differential Drive

- Need to turn
- Input is taken as an angular and linear velocity
- Turned into rad/s
- Tested equations PS4 Controller through ROS

$$\omega_{L \text{ wheel}} = \frac{2v - l\omega_z}{2r}$$

$$\omega_{R \text{ wheel}} = \frac{2v + l\omega_z}{2r}$$

$v$  = linear velocity,

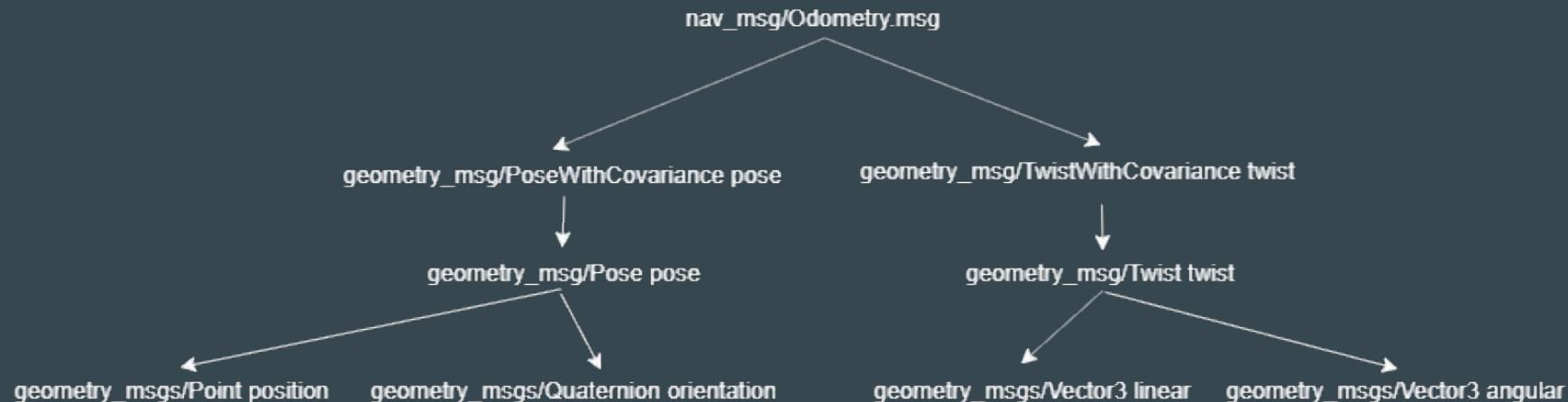
$\omega_z$  = angular velocity of rigid body,

$l$  = length of wheel separation,

$r$  = radius of wheels|

# Path Navigation

- Linear and angular speed control
- Move through a queue of waypoints till end of queue.



# Simulation

- Safe way to test our point-to-point algorithms
- Full, realistic 3D model of PACMAN
- Imported this model in ROS to have a realistic reference frame

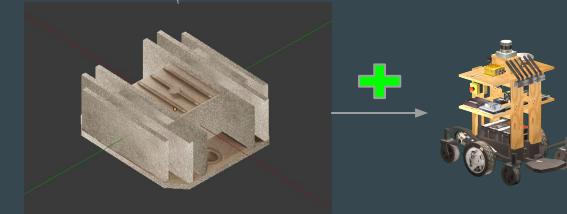
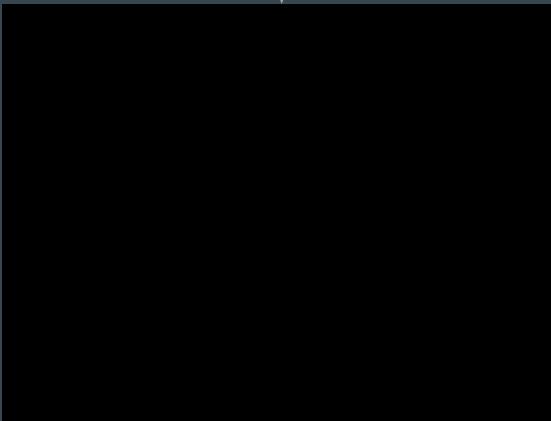
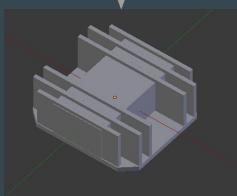
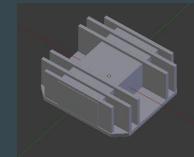
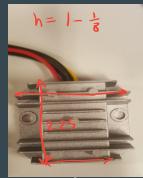
# Process for Creating CAD model



- Blender is used for CAD development
- GIMP is used for image manipulation
- The process is as follows:
  - Take Pictures
  - Measure
  - Make Mesh
  - Make Texture
  - UV wrap Mesh with Texture



# Example: Addition of DC/DC converter to the CAD Model



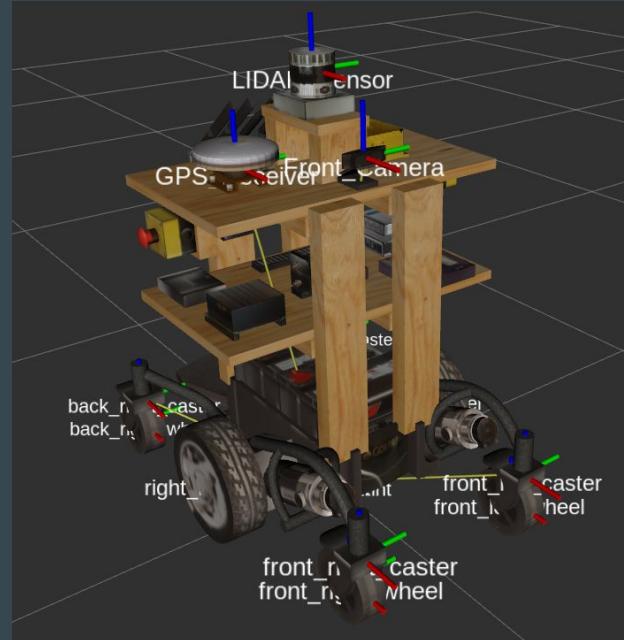
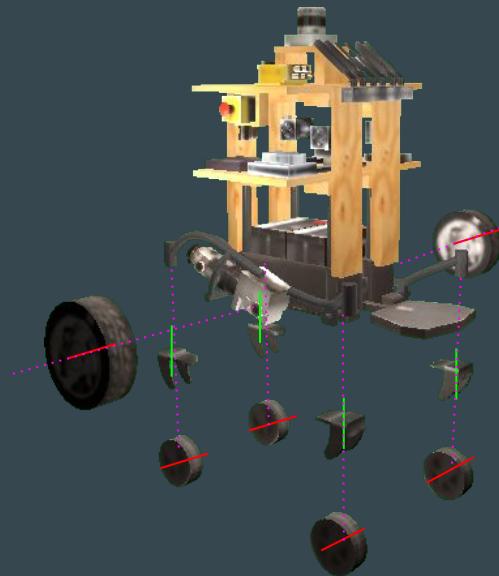
# URDF (Unified Robot Description Format)

```
<?xml version="1.0"?>
<robot name="pacman">
  <link name="base_footprint"/>
  <link name="GPS_Receiver"/>
  <link name="LIDAR_Sensor"/>
  <link name="Front_Camera"/>

  <link name="base_link">
    <inertial>
      <origin xyz="0 0 0.157701488" rpy="0 0 0" />
      <mass value="62"/>
      <inertia ixz="0.0" ixy="0.0" ixz="0.0" iyy="0.0" iyz="0.0" izz="0.0"/>
    </inertial>
    <visual>
      <geometry>
        <mesh filename="file://$(arg modelDirectory)/materials/textures/base.dae"/>
      </geometry>
    </visual>
    <collision>
      <geometry>
        <mesh filename="file://$(arg modelDirectory)/materials/textures/base.dae"/>
      </geometry>
    </collision>
  </link>

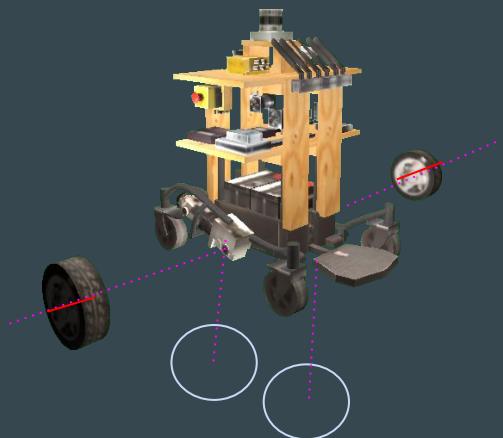
  <link name="left_big_wheel">
    <inertial>
      <origin xyz="0 0 0" rpy="0 0 0" />
      <mass value="0"/>
      <inertia ixz="0.0" ixy="0.0" ixz="0.0" iyy="0.0" iyz="0.0" izz="0.0"/>
    </inertial>
    <visual>
      <origin xyz="0 -0.2646172 -0.123838716" rpy="0 0 0" />
      <geometry>
        <mesh filename="file://$(arg modelDirectory)/materials/textures/left_wheel.dae"/>
      </geometry>
    </visual>
    <collision>
      <origin xyz="0 -0.2646172 -0.123838716" rpy="0 0 0" />
      <geometry>
        <mesh filename="file://$(arg modelDirectory)/materials/textures/left_wheel.dae"/>
      </geometry>
    </collision>
  </link>

  <link name="right_big_wheel">
    <inertial>
      <origin xyz="0 0 0" rpy="0 0 0" />
      <mass value="0"/>
      <inertia ixz="0.0" ixy="0.0" ixz="0.0" iyy="0.0" iyz="0.0" izz="0.0"/>
    </inertial>
```



# SDF (Simulation Description Format)

```
1  <?xml version="1.0" encoding="utf-8"?>
2
3  <sdf version="1.5">
4      <model name="first model">
5          <static>false</static>
6          <link name="base">
7              <collision name="collision">
8                  <geometry>
9                      <mesh>
10                         <uri>model://pacman/meshes/base.dae</uri>
11                     </mesh>
12                 </geometry>
13                 <surface>
14                     <friction>
15                         <ode>
16                             <mu>100</mu>
17                             <mu2>50</mu2>
18                         </ode>
19                     </friction>
20                 </surface>
21             </collision>
22             <visual name="visual">
23                 <geometry>
24                     <mesh>
25                         <uri>model://pacman/meshes/base.dae</uri>
26                     </mesh>
27                 </geometry>
28                 <material>
29                     <script>
30                         <uri>model://pacman/materials/base.sdf</uri>
31                     </script>
32                 </material>
33             </visual>
34             <inertial>
35                 <mass>62</mass>
36             </inertial>
37         </link>
38         <link name="left big wheel">
39             <pose> 0 .526654776 0 0 0 0 </pose>
40             <collision name="collision">
41                 <geometry>
42                     <mesh>
43                         <uri>model://pacman/meshes/big_wheel.dae</uri>
44                     </mesh>
45                 </geometry>
46                 <surface>
47                     <friction>
48                         <ode>
49                             <mu>15000</mu>
50                             <mu2>5000</mu2>
51                         </ode>
52                     </friction>
```



# Simulation Test Path

Test Path 1



Testing Path Navigation



# Path Planning

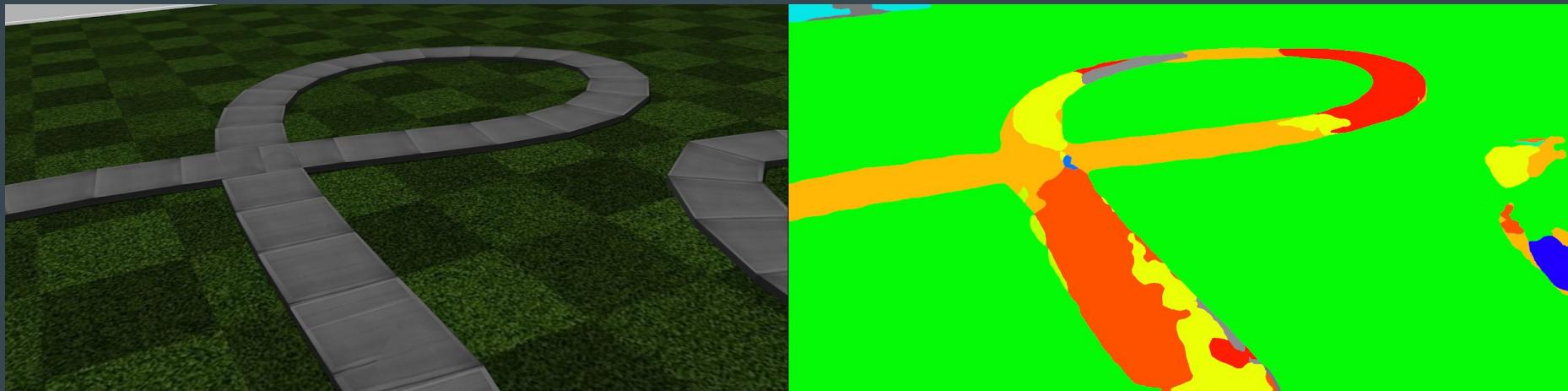
# Semantic Segmentation - What is It?

- Deep learning technique
- Uses low-level features to develop high-level features
- Per-Pixel labeling
- Open-source model out of MIT
- 150 Classes, we're only concerned with 6

# Original Segmentation Results

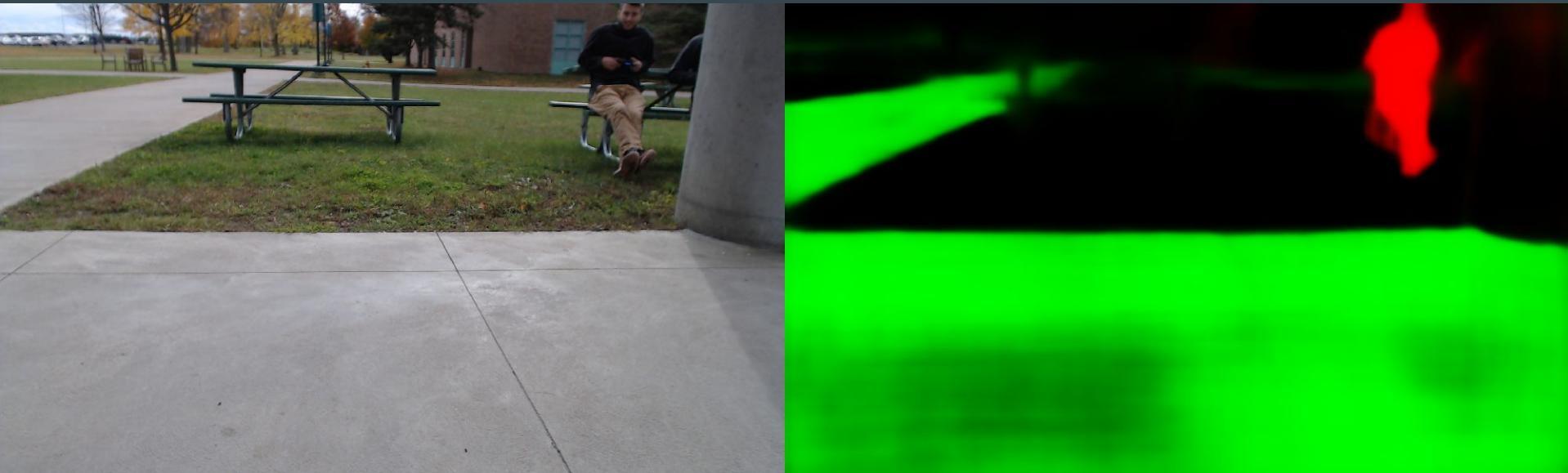


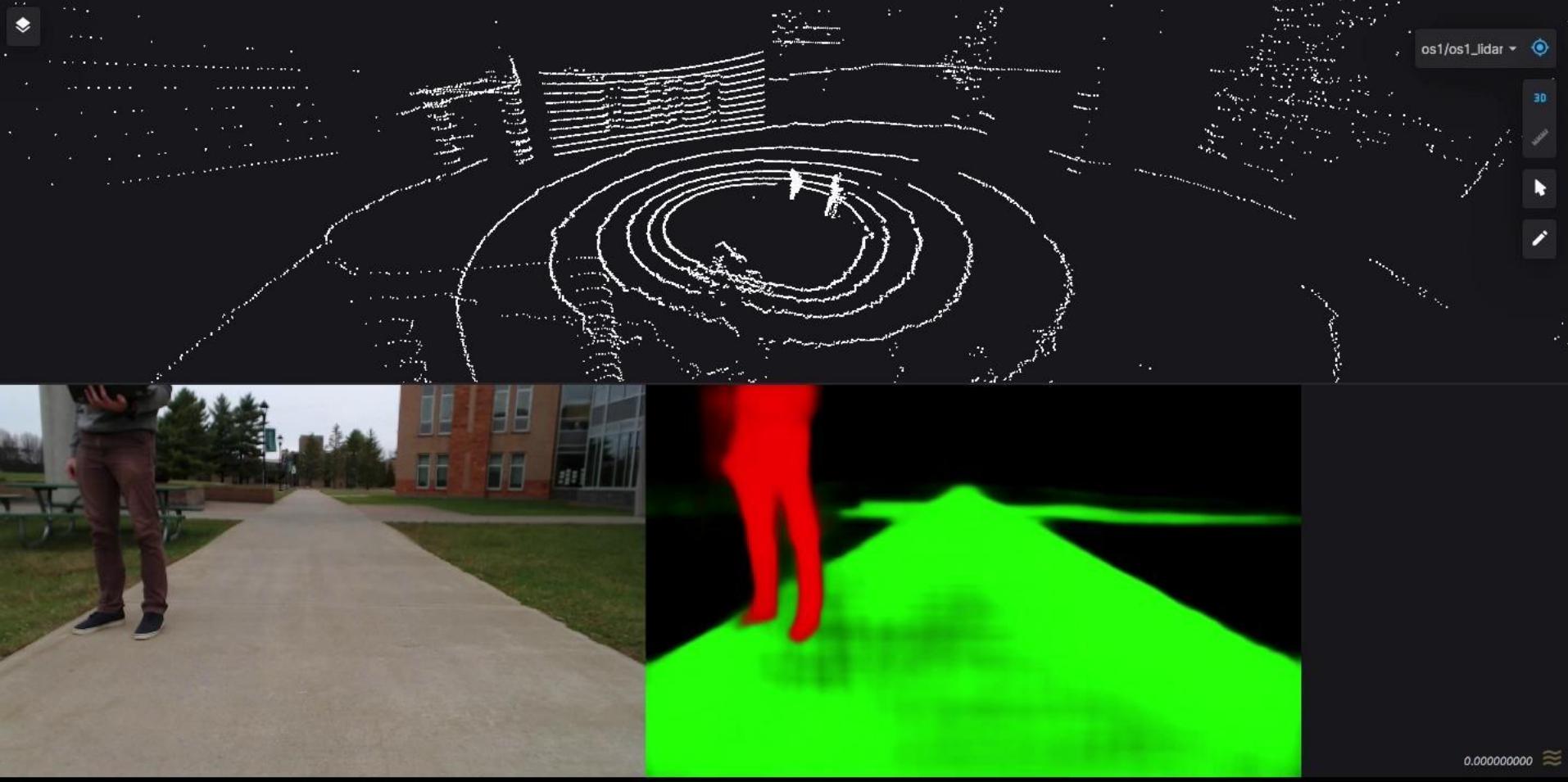
# Tested Segmentation in Simulation



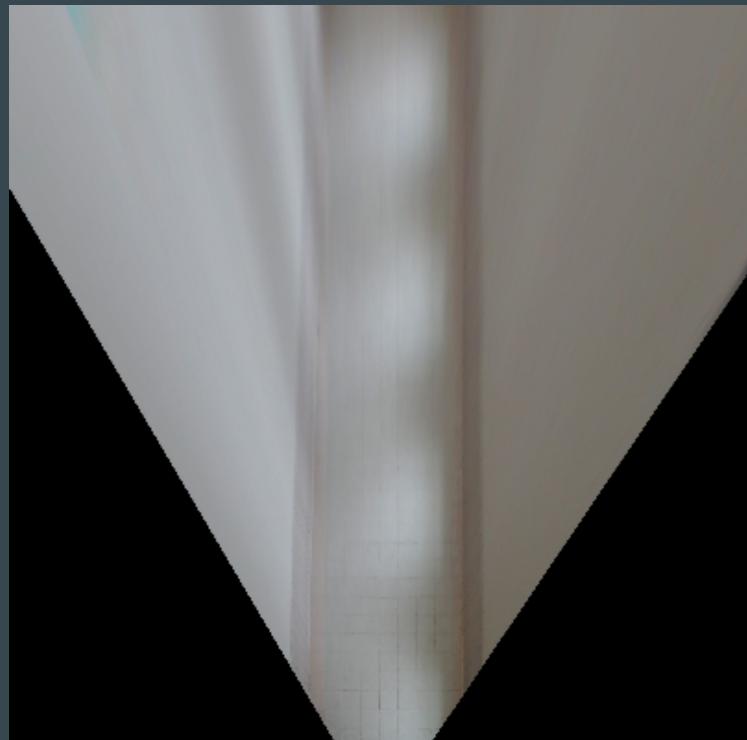
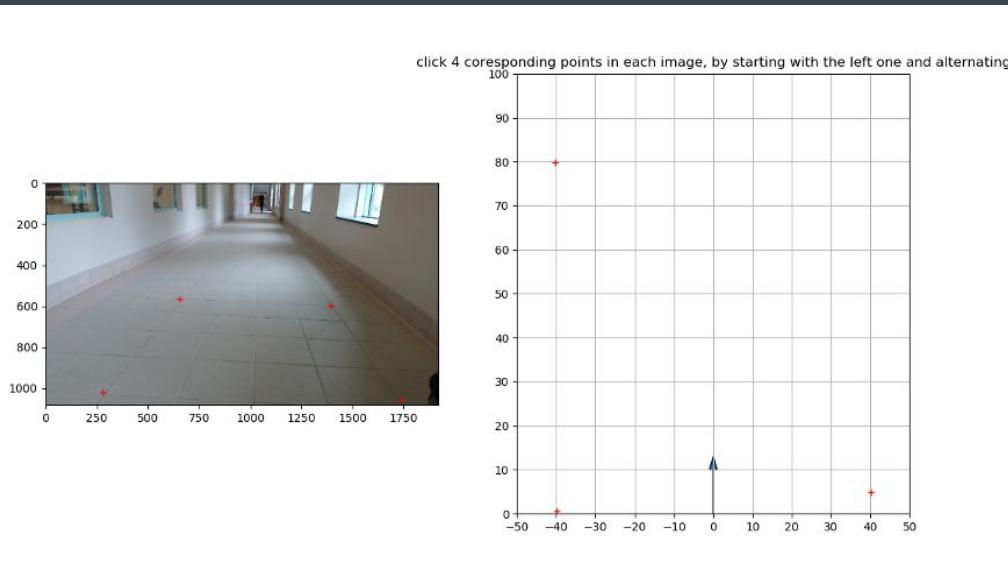
# Modified Semantic Segmentation

Using MITs open-source algorithm, running at 1.4 FPS at 640x480.

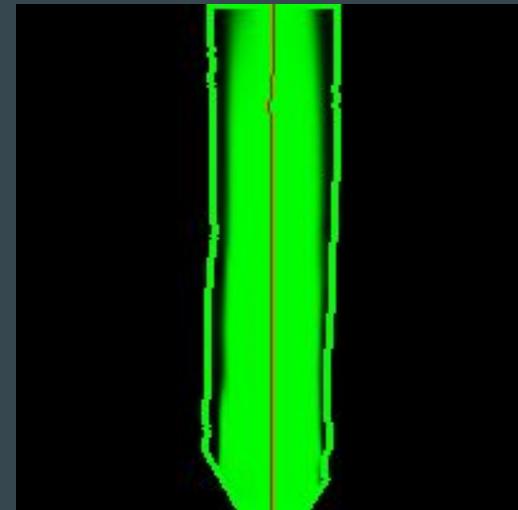
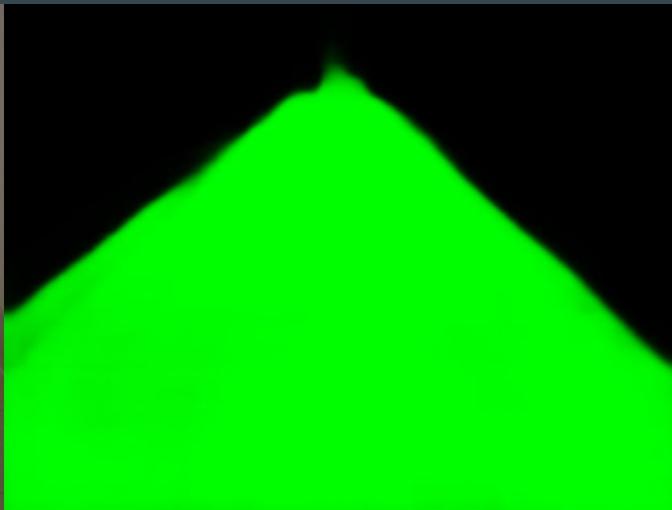




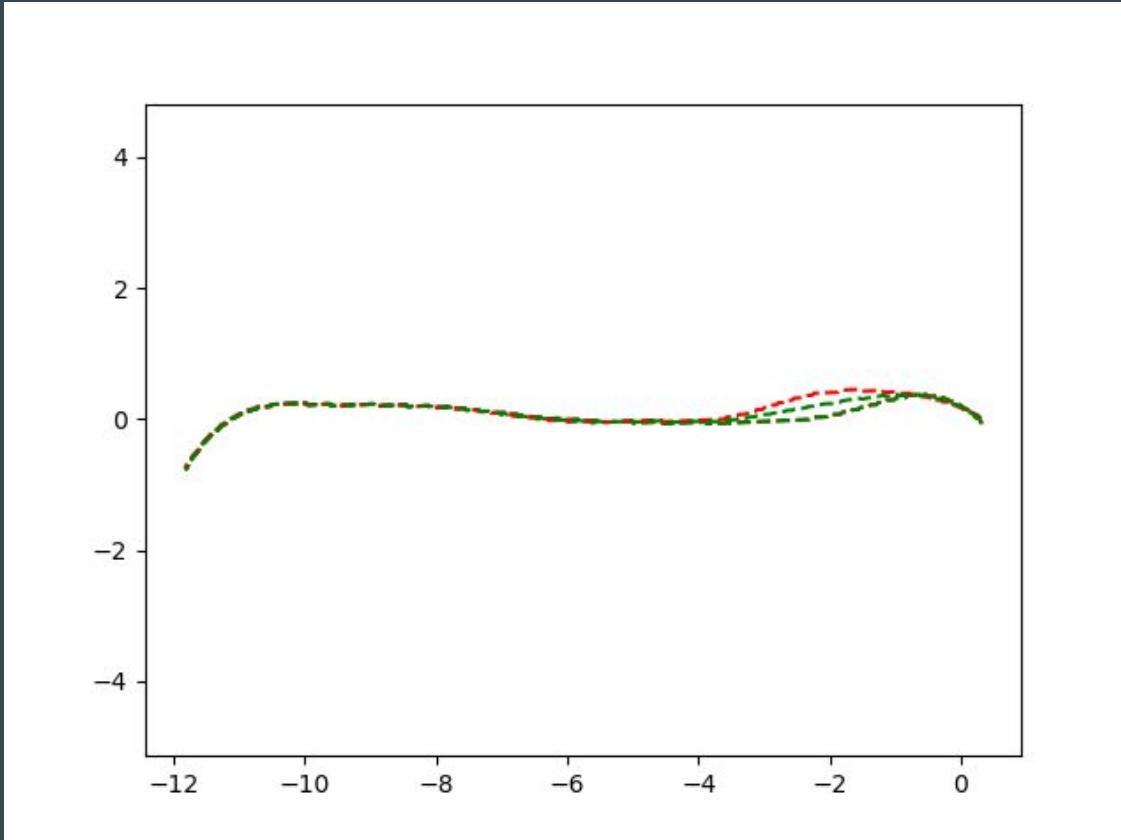
# Camera Calibration - David Russell



# Path Planning - David Russell

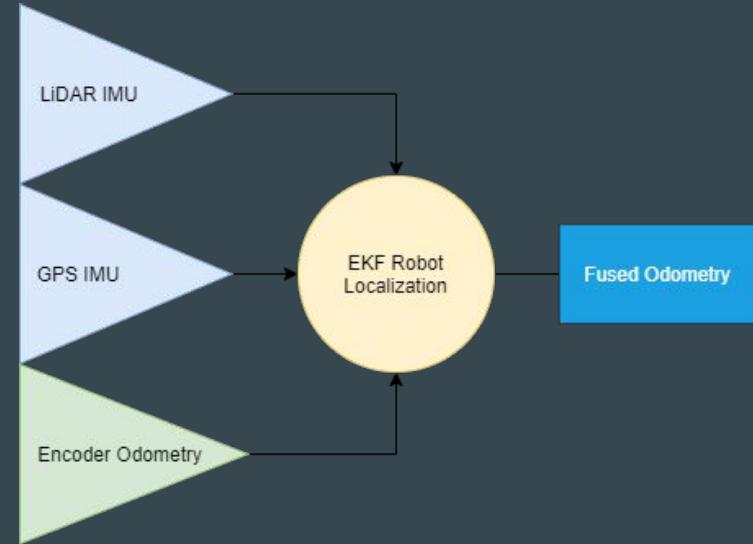
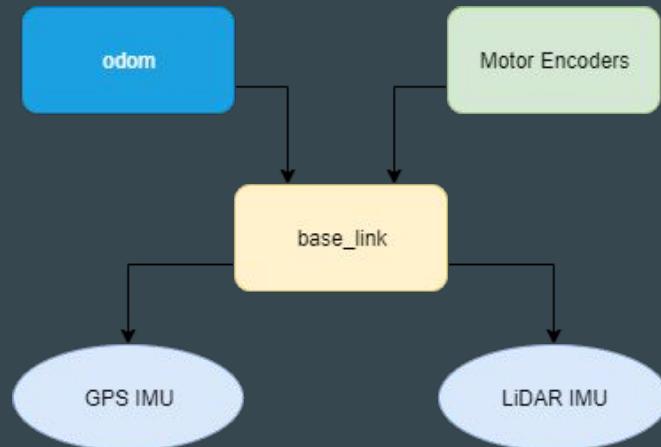


# Path Planning - Adding in Memory



# Localization via Odometry

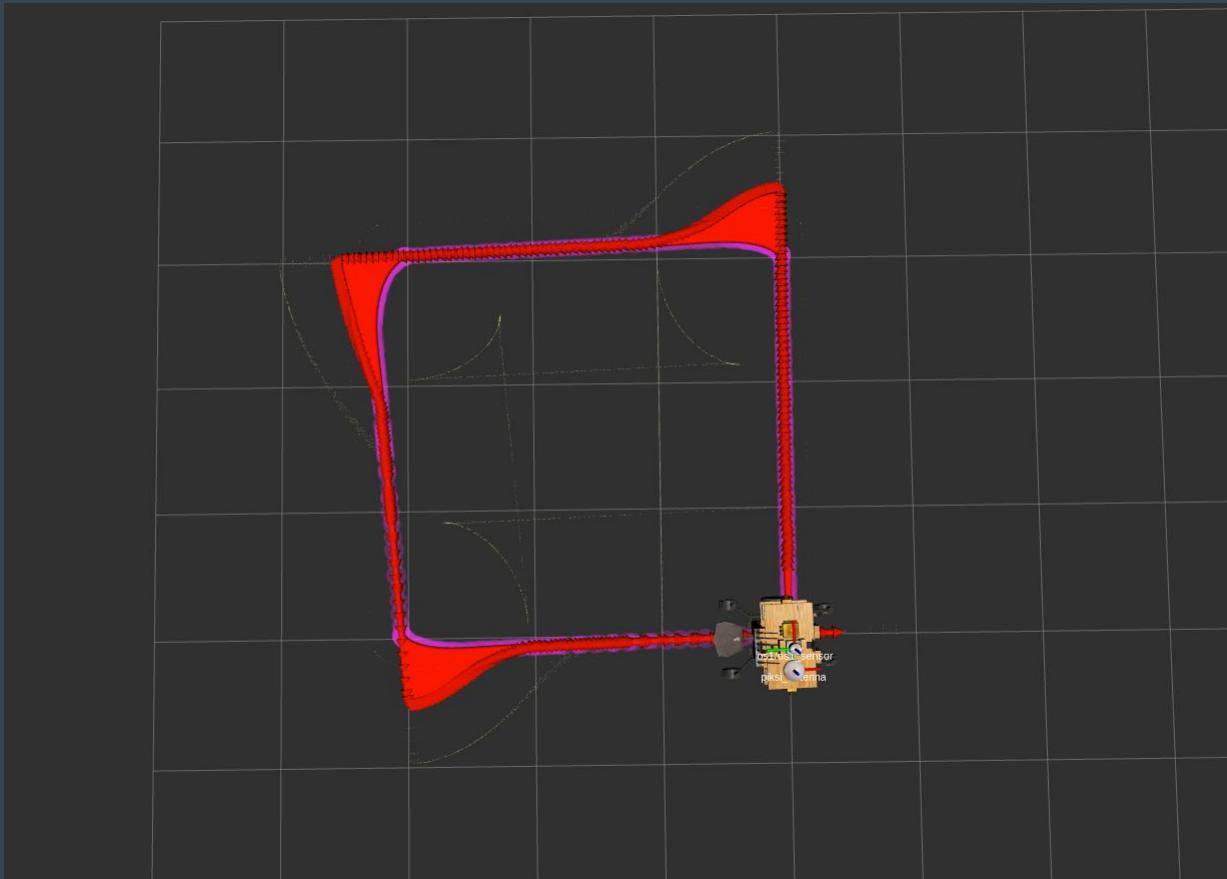
- ROS Transform (Tf) Tree - help from Sarmad Mehrdad
- Extended Kalman Filter (EKF) ROS - provided node of Robot Localization



# Before Fusion vs After Fusion after Three 10x10 Laps

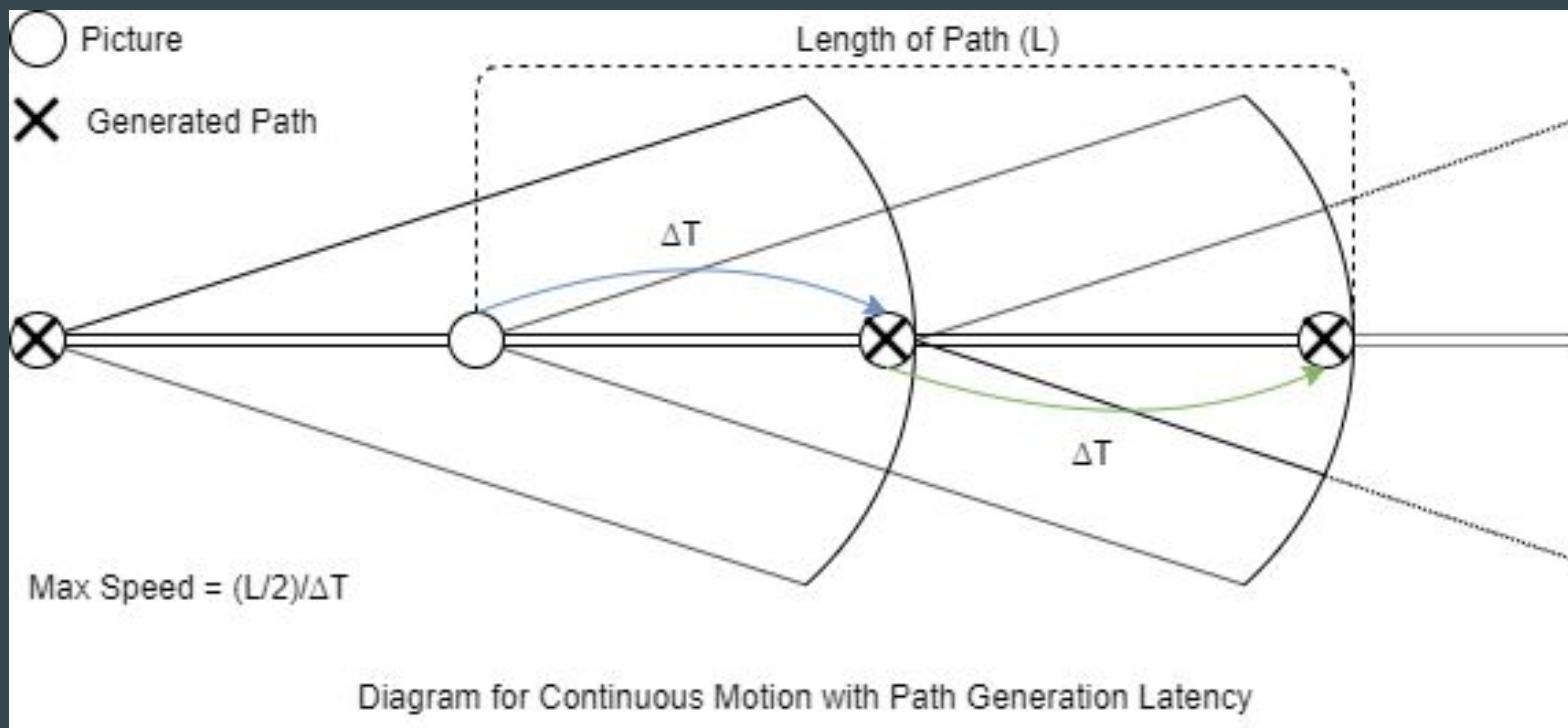


# Point-to-Point Navigation in Real World



# Path Navigation (continued)

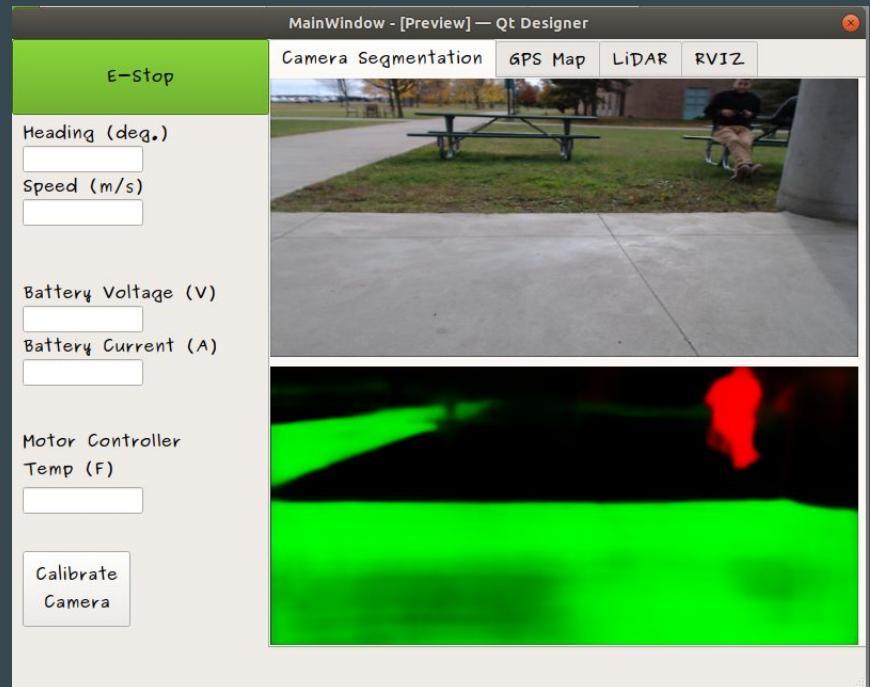
- Path Filtering



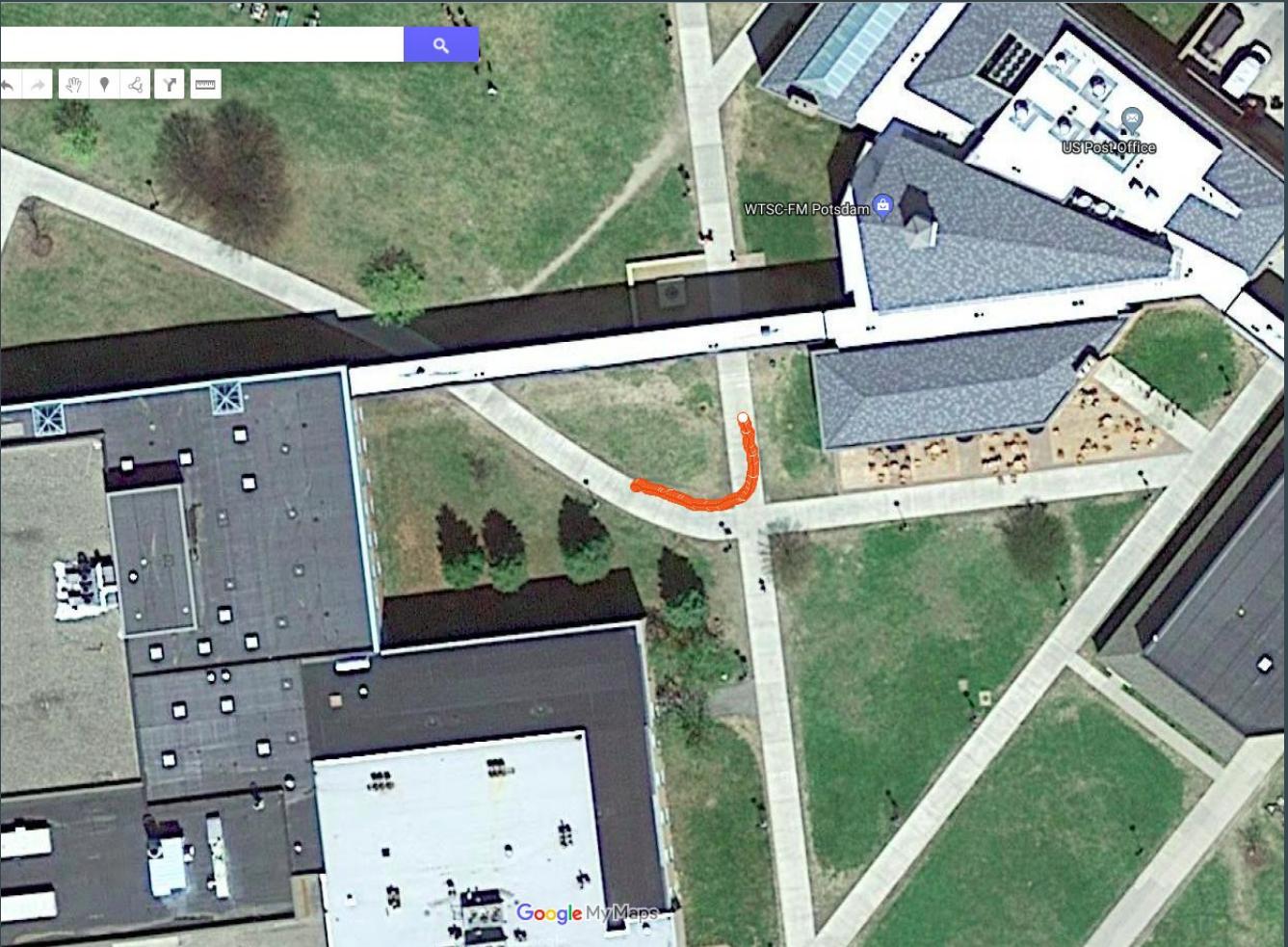
# Results

# GUI

- Provides views of sensor data in user friendly manner
- Users will be able to software Estop the vehicle through the GUI
- Users will be able to calibrate the camera through the GUI



# GPS Track



# Real-World Path Following



# Future Work

- Implement object detection from LiDAR data
- Build 3D maps from LiDAR data
- Create GUI to visualize geolocated sidewalk (In progress)

# Thank You

- Professor Carroll
- Professor Khondker
- Professor Sonar
- Hannah DeFazio
- Sarmad Mehrdad
- Jake Schechter

# Check out our GitHub!

[https://github.com/romleiaj/pacman\\_ws](https://github.com/romleiaj/pacman_ws)

The screenshot shows the GitHub repository page for `pacman_ws`. The repository is owned by `romleiaj`. The main header includes the repository name, a star count of 3, and a fork count of 0. Below the header, there are tabs for Code, Issues (0), Pull requests (0), Actions, Projects (0), Wiki, Security, Insights, and Settings. A summary bar at the top provides metrics: 82 commits, 11 branches, 0 packages, 0 releases, and 4 contributors. The Branch dropdown is set to master. A New pull request button is visible. Below the summary, a list of recent commits is shown:

File / Action	Description	Time Ago
romleiaj Merge pull request #12 from romleiaj/dev/path_planning ...	Latest commit 783feae	20 hours ago
data	Upload correct results and data.	2 months ago
src	Remove duplicate.	20 hours ago
.gitignore	Added install to gitignore.	17 days ago
.gitmodules	Working version of basic path planning.	2 days ago
README.md	Update README.md	17 days ago

At the bottom of the page, there is a section for the `pacman_ws` package, which contains the same description and a link to the repository.

# Questions?