PACMAN

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

• • •

Team 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)

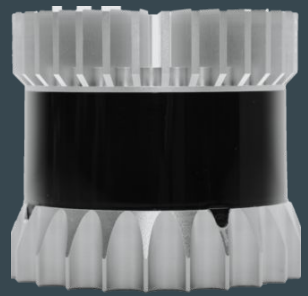
Sensors

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

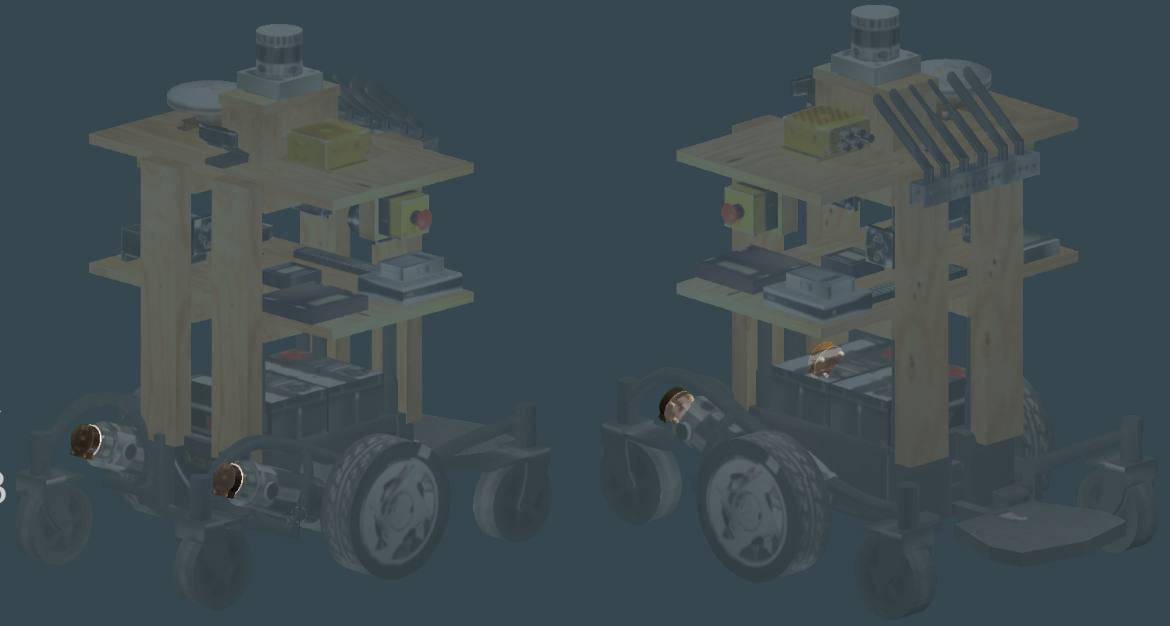


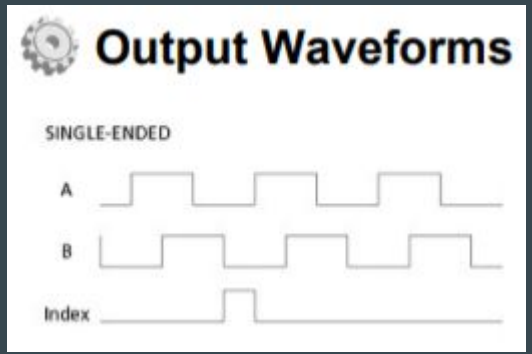
Logitech C920

- 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

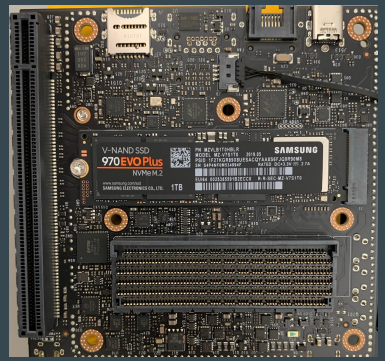


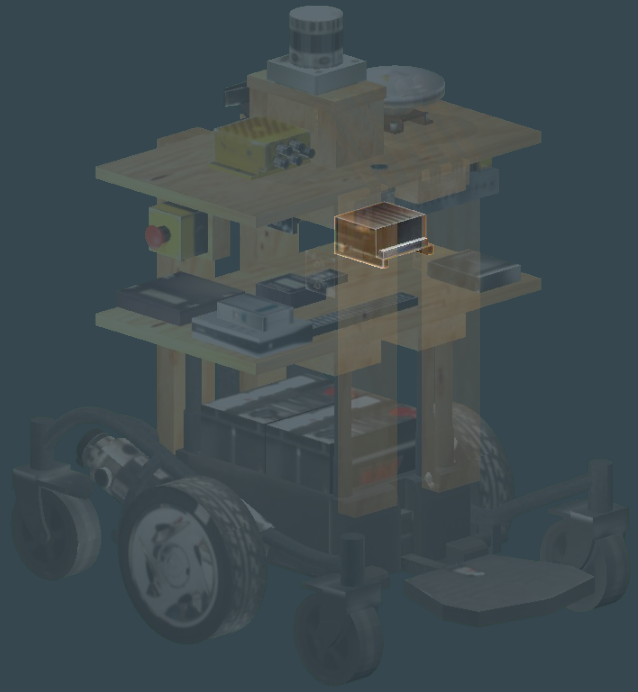


Hardware

Jetson Xavier

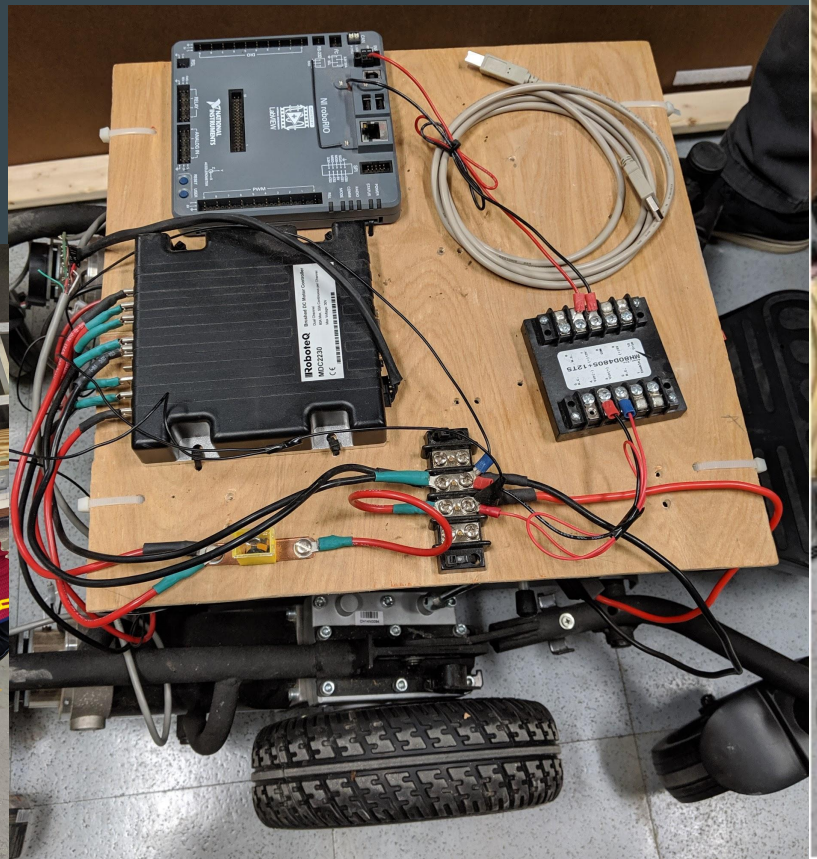
- Head Computing Node
- Excellent embedded system for AI and Computer Vision with onboard GPU
- Runs a full Linux environment

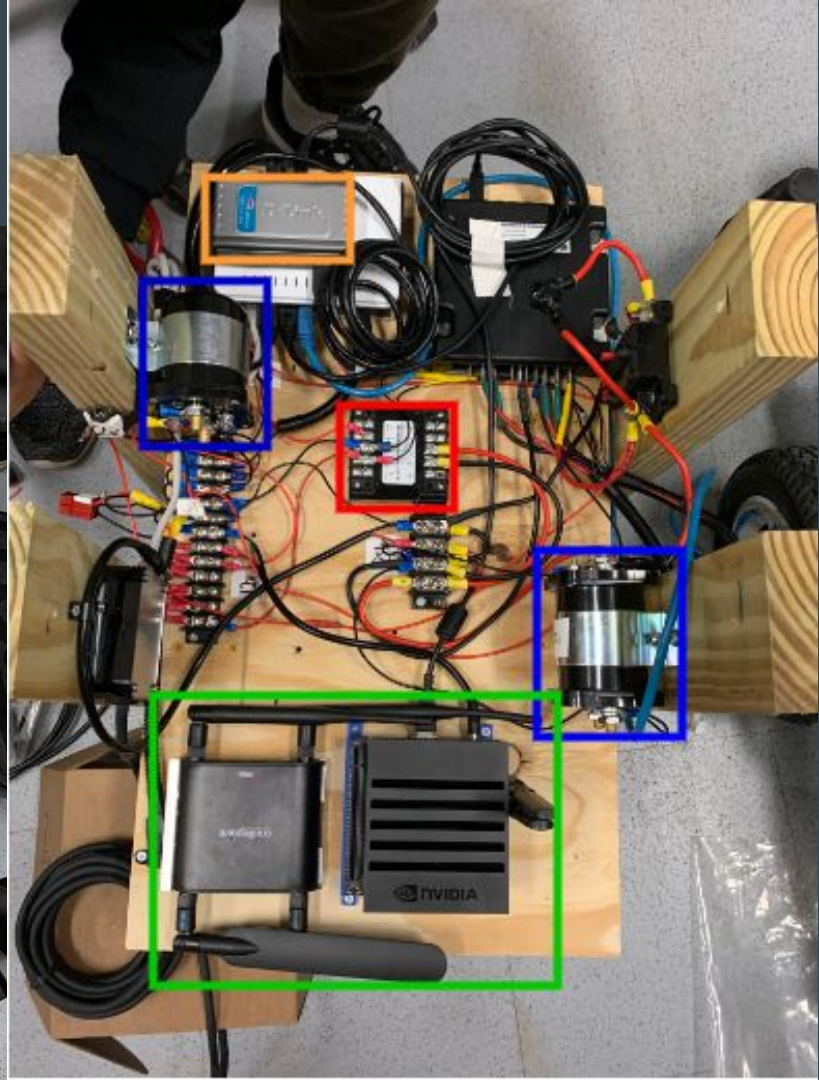




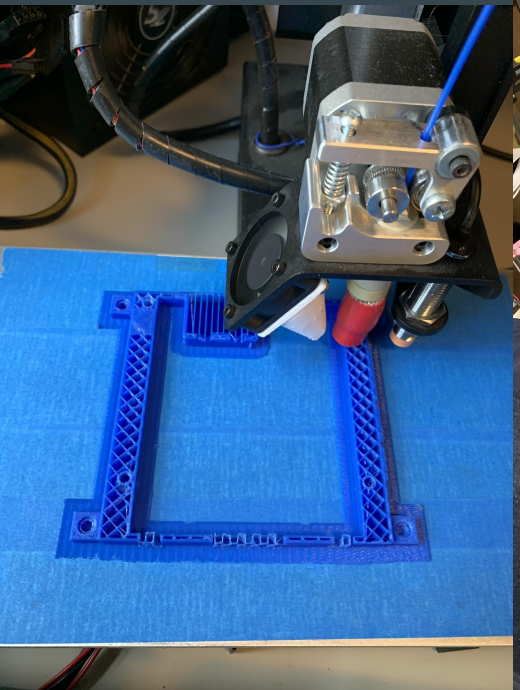
Frame

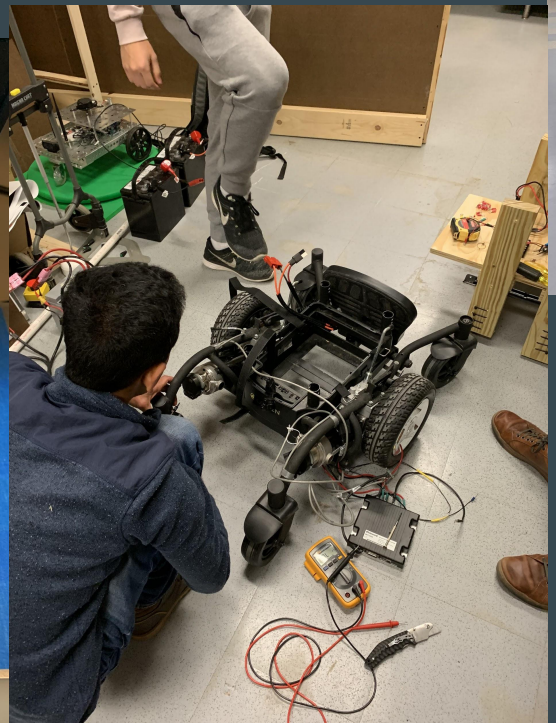

Lowe's
ProServ
Visit us at Lowes


National Instruments
NI roboRIO
IRoboteQ
MDC2230

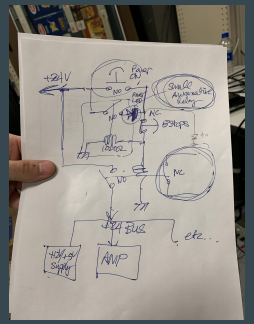

NVIDIA

Development

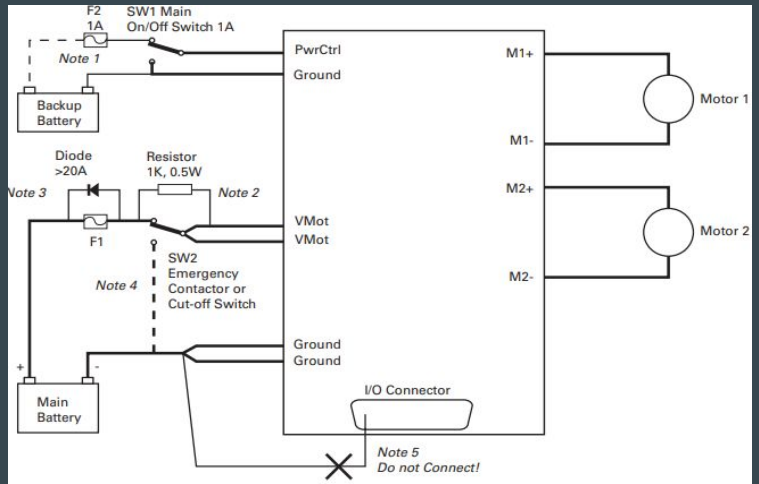


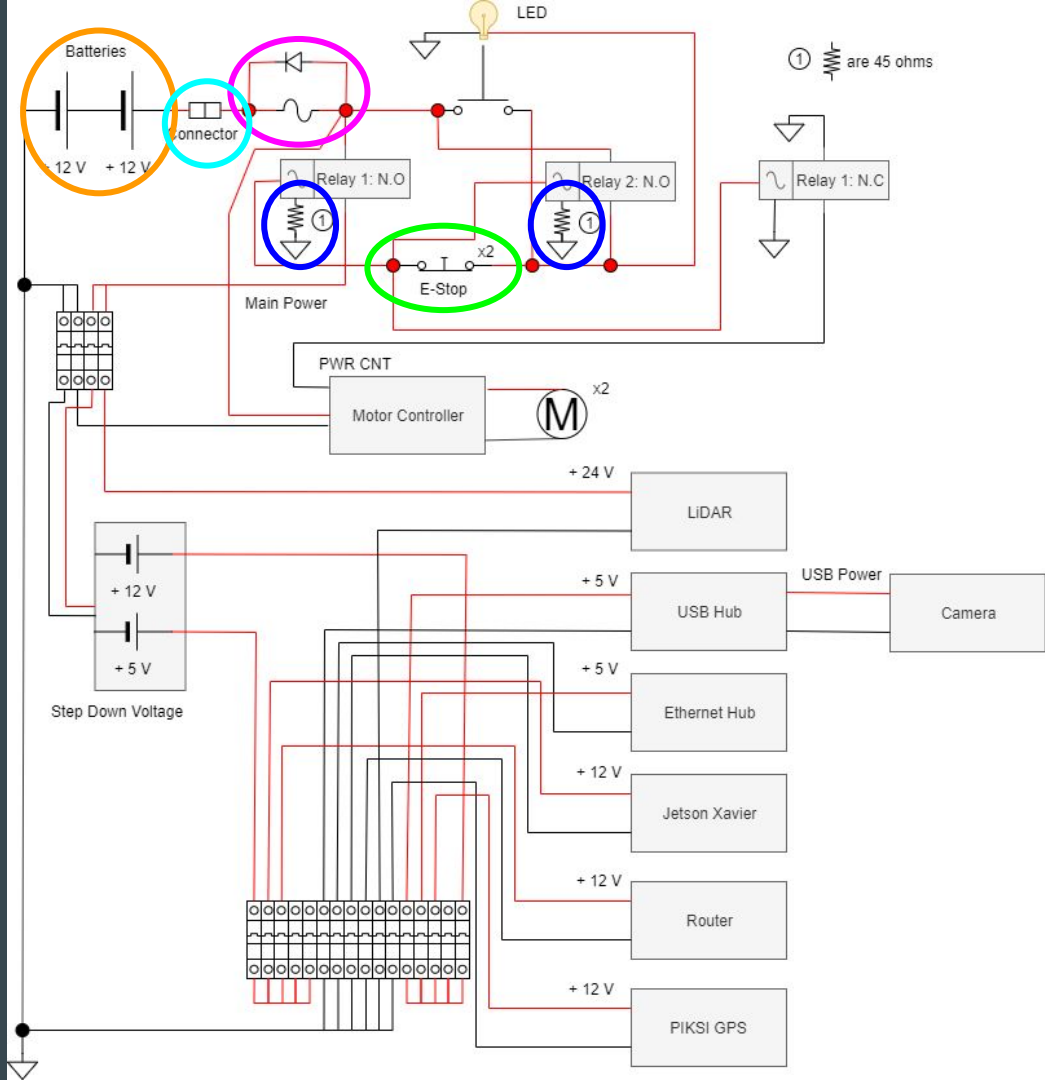


Power Circuit



- 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	
	Part Number	**Vendor**	**Quantity**	**Price per Unit**	**Total Price**	**Link**
Frame:					**32.52**	
Preasure 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	
Electronics:					**961.85**	
Breaker	1851205	Amazon	1	25.89	25.89	https://www
Diode	1N1 183A	SSDI	1	4.38	4.38	https://data
Red & Black bundle of wire (10 guage)	8541758680	Amazon	1	23.95	23.95	https://www
Red bundle of wire (18 guage)	55667423	Amazon	1	7.36	7.36	https://www
Black bundle of wire (18 guage)	55667323	Amazon	1	8.28	8.28	https://www
Relay (N.O. and N.C. ends)	WR586RS-24	Ebay	1	85.00	85.00	https://www
Relay (N.O. end)	WR586	Ebay	1	65.00	65.00	https://www
Barrel jack connectors (2.5x5.5) (2)	4330127896	Amazon	1	19.40	19.40	https://www
Wire tube 1/2" 25 ft	B07TCDTFL2	Amazon	1	9.50	9.50	https://www
Srink tube kit	B071H5XC7C	Amazon	1	9.98	9.98	https://www
Emergency stop buttons	43217-5544	Amazon	2	10.89	21.78	https://www
24/19v DC/DC converter	MWX-24-19-190W	Amazon	1	67.00	67.00	https://www
24/12v DC/DC converter	B0756W6V4F	Amazon	1	12.99	12.99	https://www
24/5v DC/DC converter	8-40T5-10A	Amazon	1	15.99	15.99	https://www
Battery power connectors	B07Q5J8BWH	Amazon	1	9.60	9.60	https://www
Batteries	M22NF-SLD-A	BatteryStuff.com	2	257.00	514.00	https://www
Terminal wire strip (2x4)	204586	Defender	1	9.99	9.99	https://www
Terminal wire strip (2x#10)	202095	Defender	2	14.99	29.98	https://www
Push button	43217-5544	Amazon	2	10.89	21.78	https://www
Hardware:					**13765.12**	
Logitec camera	960000764	Amazon	1	56.00	56.00	https://www
Playstation controller	B01LWVX2RG	Amazon	1	43.95	43.95	https://www
USB hub	ST4200USBM	Amazon	1	51.70	51.70	https://www
Ethernet hub	DAP-1522	Amazon	1	39.99	39.99	https://www
Cradlepoint technologies antenna (replacement)	EWUA0052	Amazon	1	10.29	10.29	https://www
LiDAR	OS1-16	Ouster	1	3500.00	3500.00	https://ous
Jetson Xavier	945-82972-0000-000	NVIDIA	1	699.99	699.99	https://www
Cradlepoint Router	IBR600LE-VZ	Amazon	1	269.00	269.00	https://www
PIKSI Duro GPS	Duro Inertial	Swiftnav	1	3995.00	3995.00	https://www
RoboteQ Motor Controller	MDC2230	SuperDriodRobots	1	340.00	340.00	https://www
Power chair wheel base	Compass HD	Golden Tech.	1	4599.00	4599.00	https://www
Wheel encoders	E3-512-394-IE-D-D-3	USdigital	2	80.10	160.20	https://www

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

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

Full Node Diagram

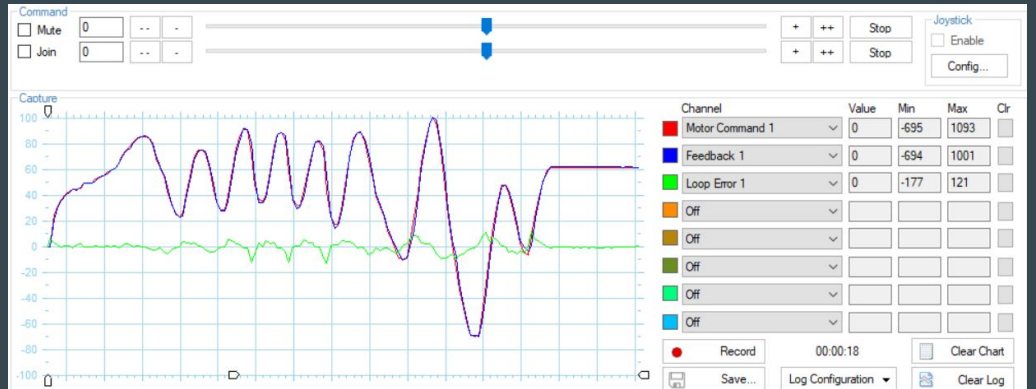
Motor Controller Custom Driver

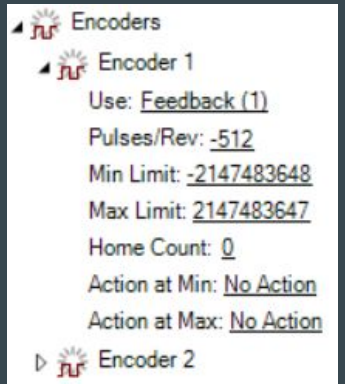
- Several ROS Drivers were tested first, none satisfactory
- So, we had to write our own
- Jake Schechter used API from Roboteq to implement core “GO” functionality
- We expanded with feedback and status updates

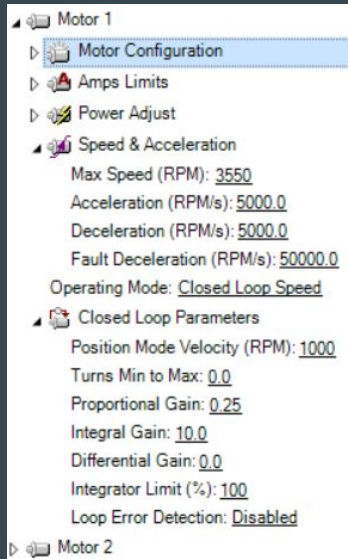
Motor Controller Utility (Roborun+)

- 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)







Differential Drive

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

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

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

v = linear velocity,

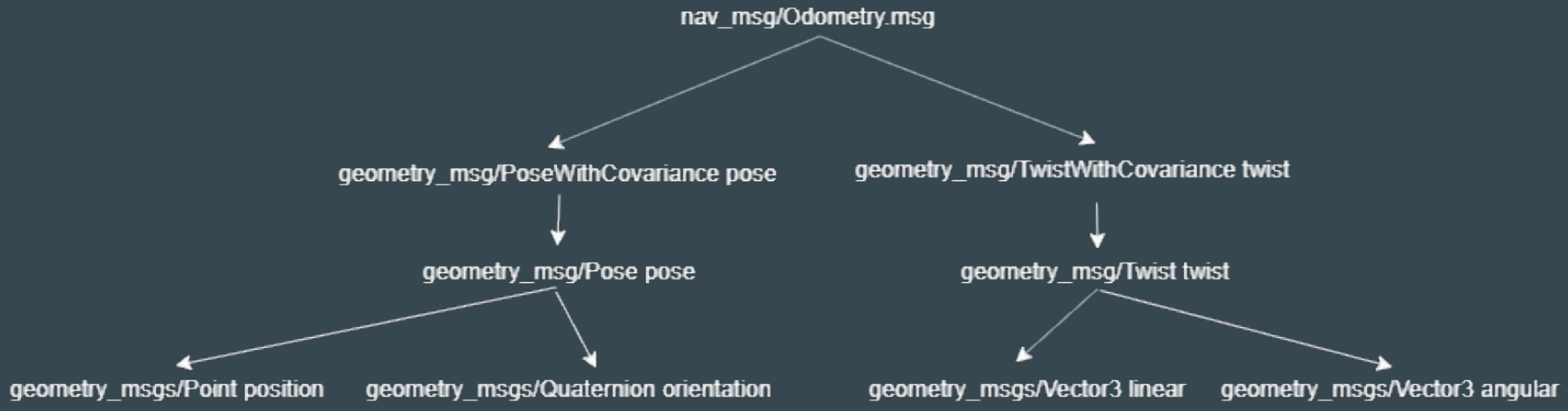
ω_z = angular velocity of rigid body,

l = length of wheel seperation,

r = radius of wheels

Path Navigation

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



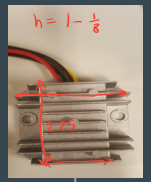
Simulation

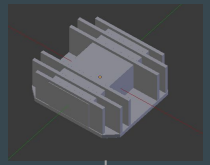
- Safe way to test our point-to-point algorithms
- Full, realistic 3D model of PACMAN
- Imported this model in ROS to have a realistic reference frame

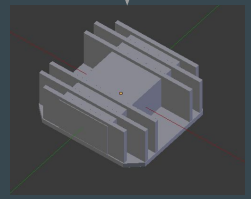
Process for Creating CAD model

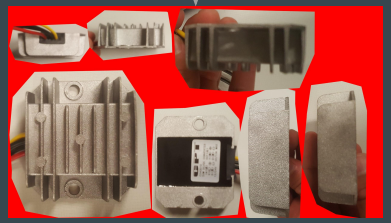
- Blender is used for CAD development
- GIMP is used for image manipulation
- The process is as follows:
  - Take Pictures
  - Measure
  - Make Mesh
  - Make Texture
  - UV wrap Mesh with Texture

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









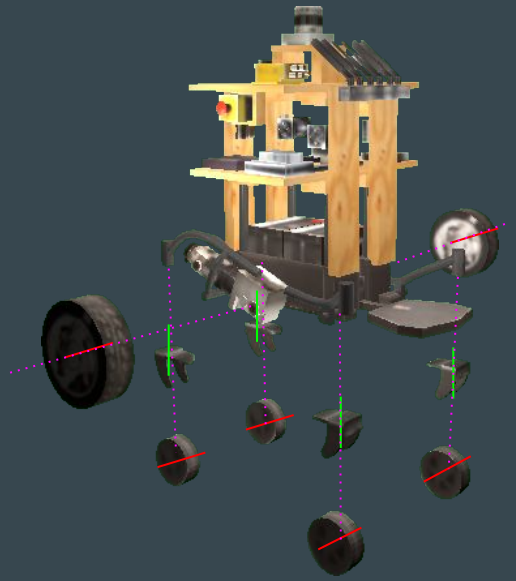
URDF (Unified Robot Description Format)

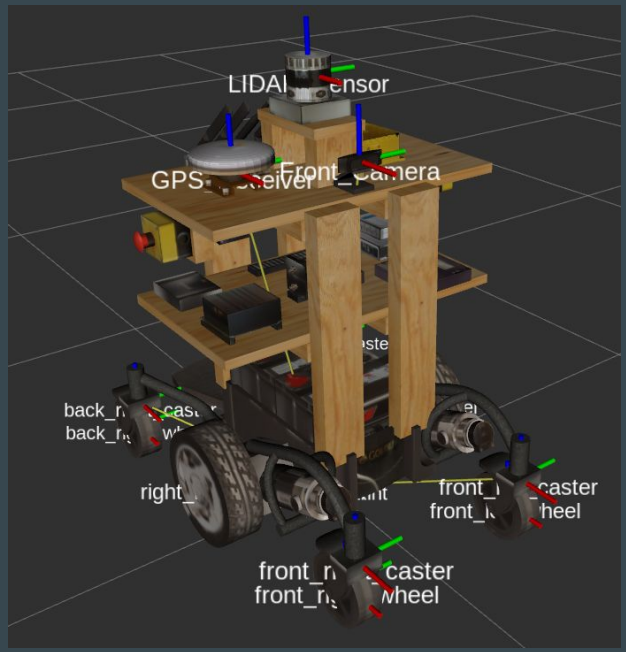
```xml
<?xml version="1.0"?>
<robot name="pacman">
    <link name="base_footprint"/>
    <link name="GPS_Receiver"/>
    <link name="LIDAR_Sensor"/>
    <link name="Front_Camera"/>


    <link name="base_link">
        <inertial>
            <origin xyz="0 0 0.157701488" rpy="0 0 0" />
            <mass value="62"/>
            <inertia ixx="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 ixx="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 ixx="0.0" ixy="0.0" ixz="0.0" iyy="0.0" iyz="0.0" izz="0.0"/>
        </inertial>
```

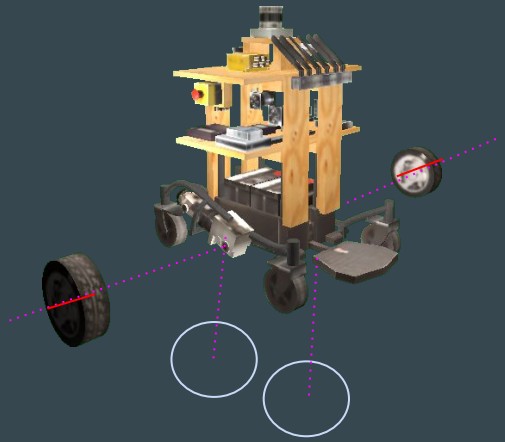




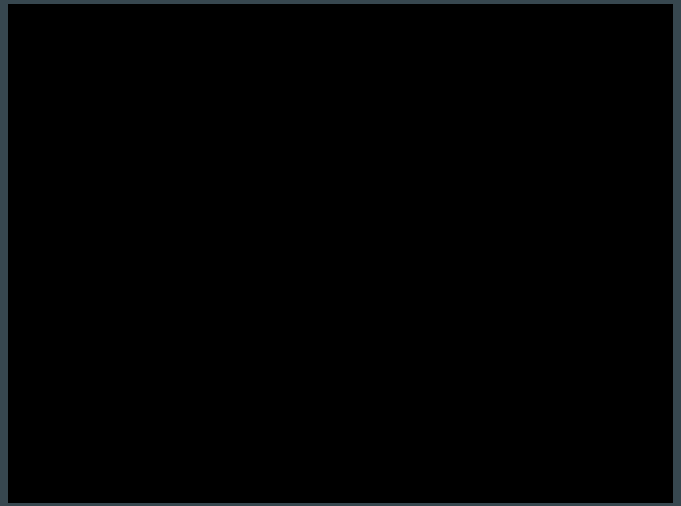
SDF (Simulation Description Format)

```xml
<?xml version="1.0" encoding="utf-8"?>

<sdf version="1.5">
     <model name="first_model">
        <static>false</static>
       <link name="base">
            <collision name="collision">
                <geometry>
                  <mesh>
                    <uri>model://pacman/meshes/base.dae</uri>
                  </mesh>
                </geometry>
                <surface>
                    <friction>
                        <ode>
                            <mu>100</mu>
                            <mu2>50</mu2>
                        </ode>
                    </friction>
                </surface>
            </collision>
          <visual name="visual">
            <geometry>
                <mesh>
                  <uri>model://pacman/meshes/base.dae</uri>
                </mesh>
            </geometry>
            <material>
                <script>
                    <uri>model://pacman/meshes/base.dae</uri>
                </script>
            </material>
          </visual>
          <inertial>
            <mass>62</mass>
          </inertial>
        </link>
        <link name="left_big_wheel">
          <pose> 0 .526654776 0 0 0 0 </pose>
                  <collision name="collision">
            <geometry>
              <mesh>
                <uri>model://pacman/meshes/big_wheel.dae</uri>
              </mesh>
            </geometry>
            <surface>
                <friction>
                    <ode>
                        <mu>15000</mu>
                        <mu2>5000</mu2>
                    </ode>
                </friction>
```

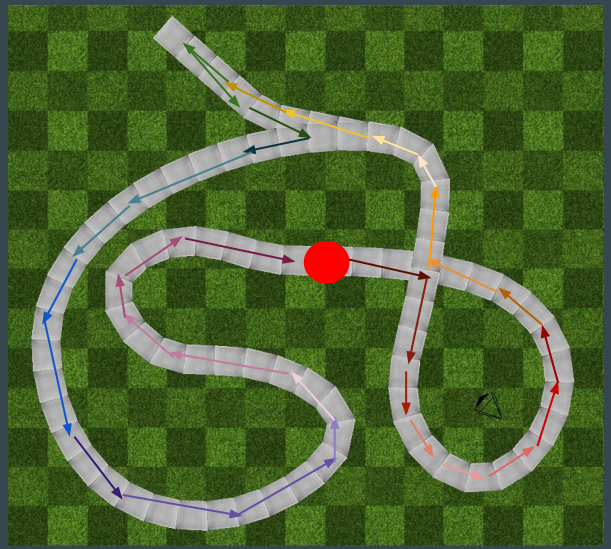


GAZEBO



Simulation Test Path

Test Path 1



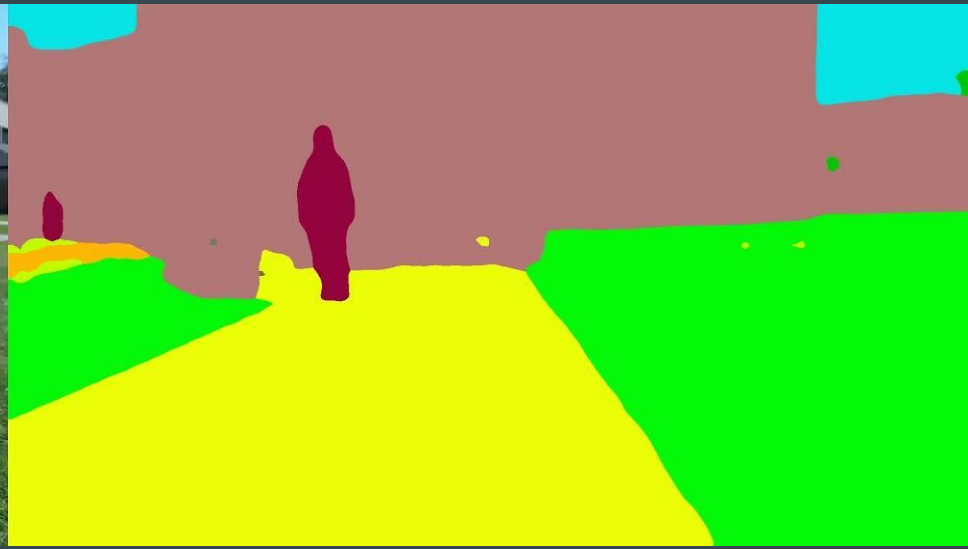
Testing Path Navigation

Path Planning

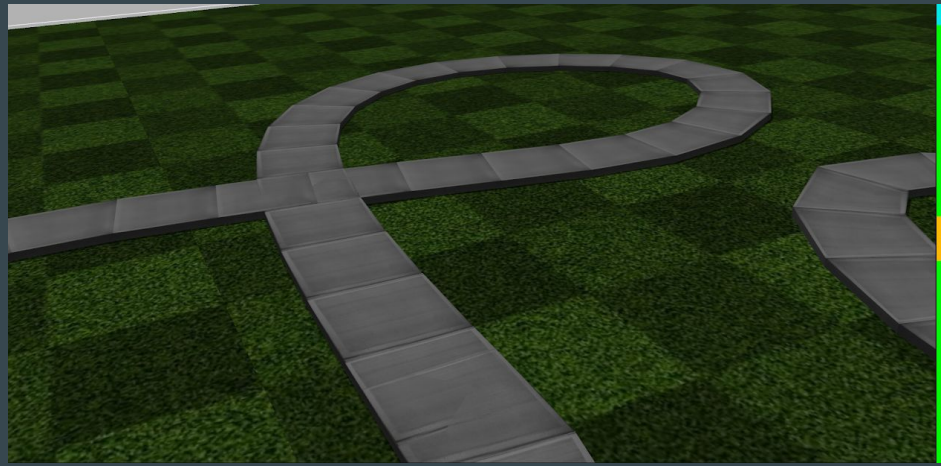
Semantic Segmentation - What is It?

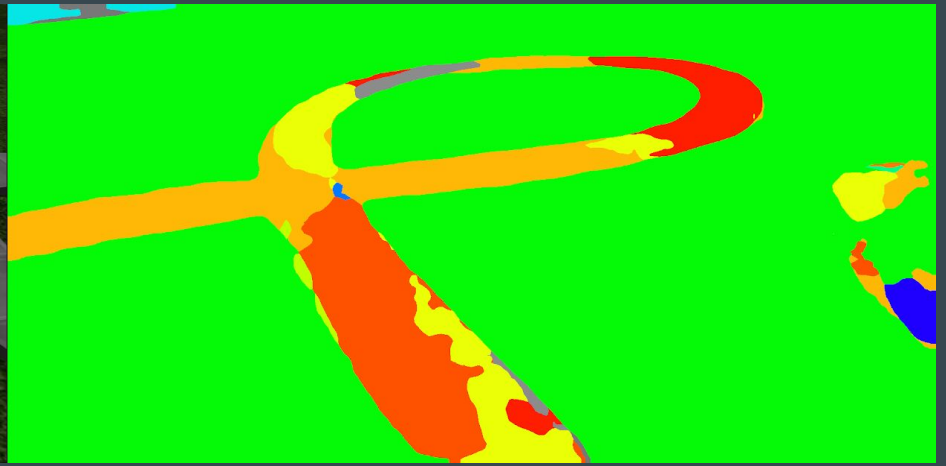
- Deep learning technique
- Uses low-level features to develop high-level features
- Per-Pixel labeling
- Open-source model out of MIT
- 150 Classes, we're only concerned with 6

Original Segmentation Results



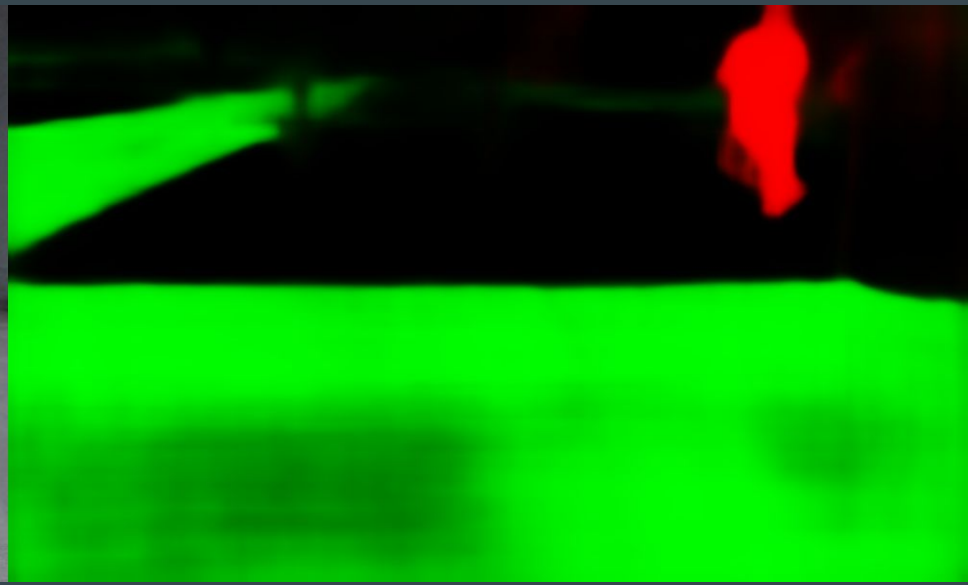
Tested Segmentation in Simulation

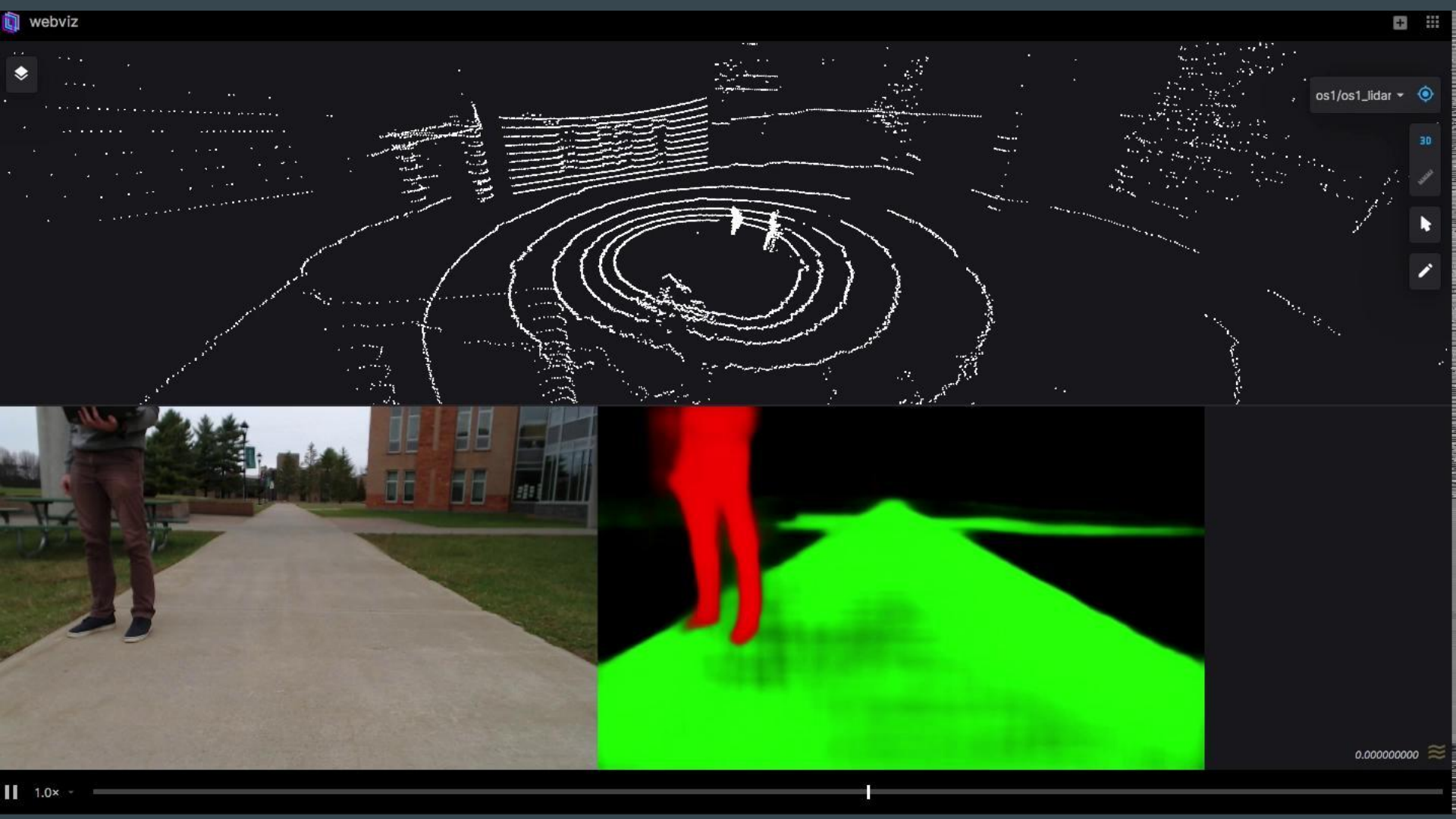




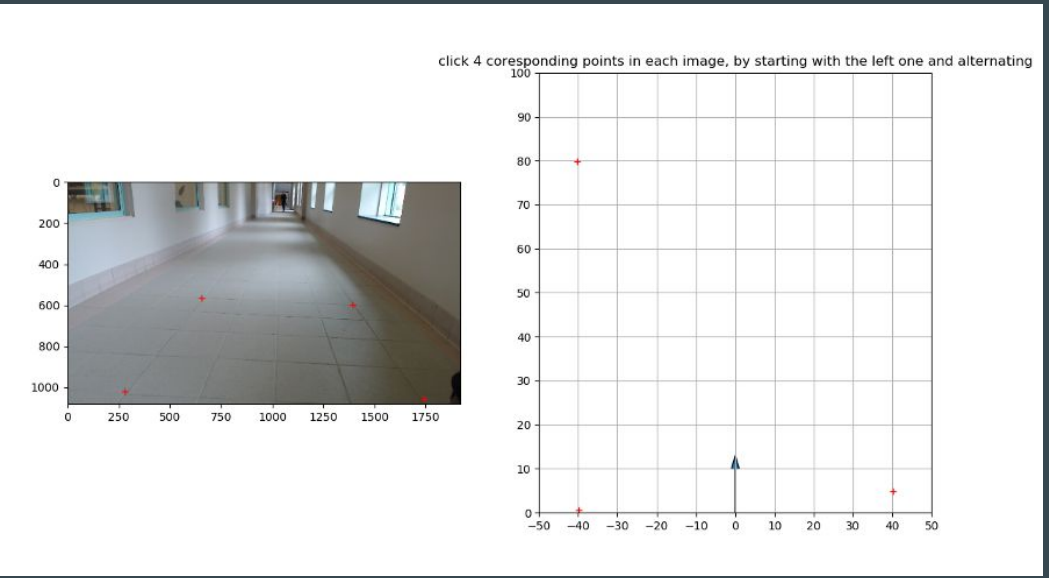
Modified Semantic Segmentation

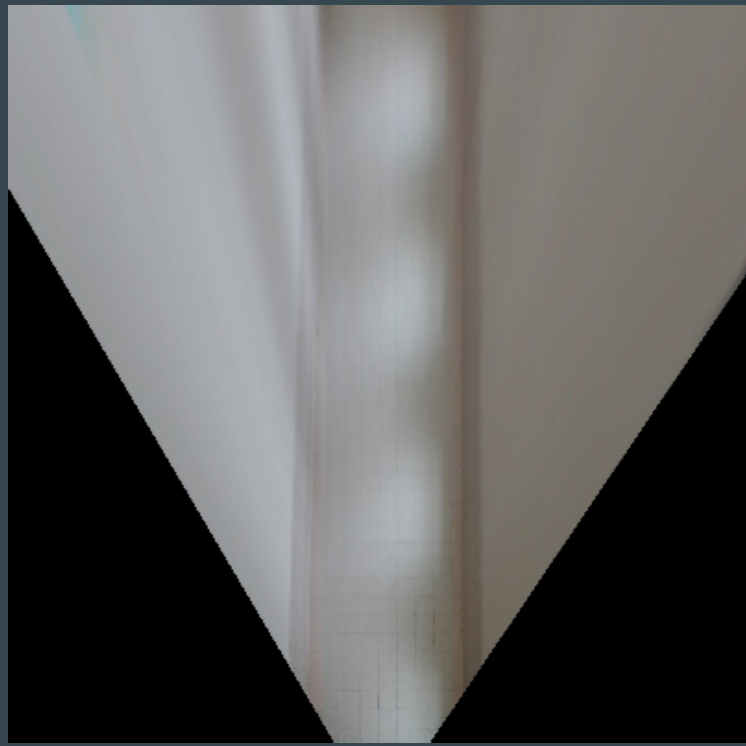
Using MITs open-source algorithm, running at 1.4 FPS at 640x480.



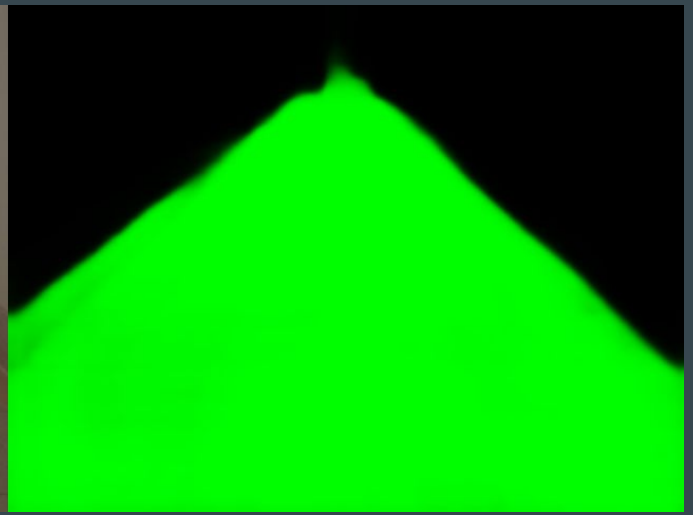

webviz
os1/os1_lidar
3D
0.000000000
1.0×

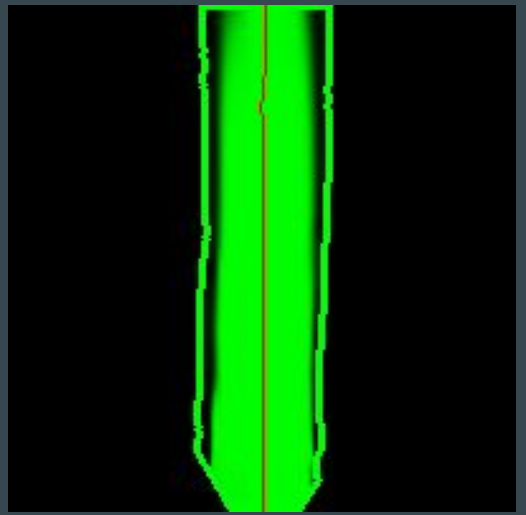
Camera Calibration - David Russell



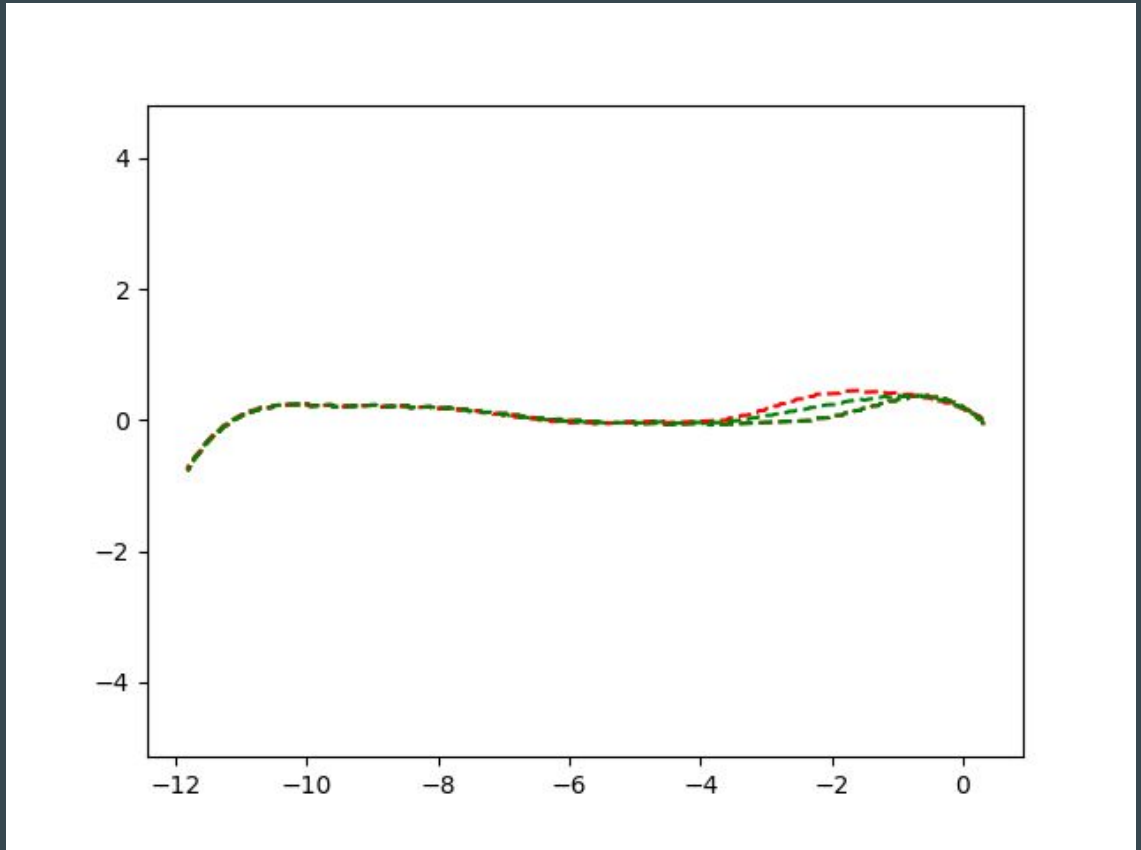


Path Planning - David Russell



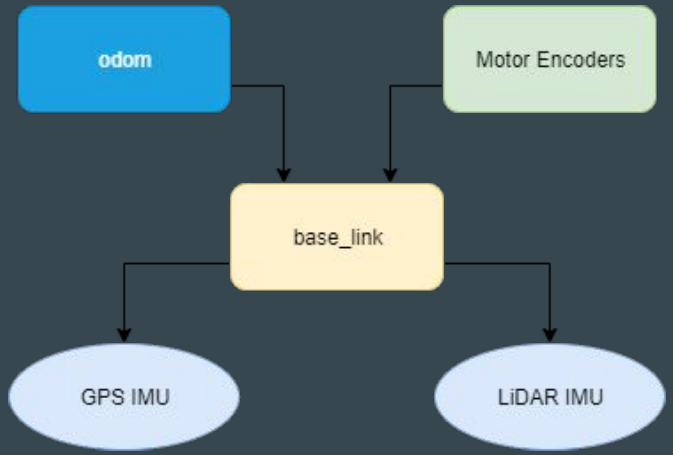


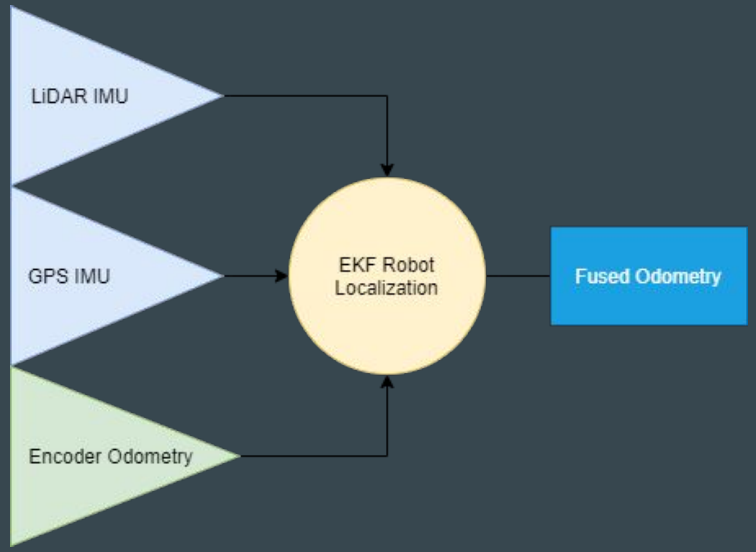
Path Planning - Adding in Memory



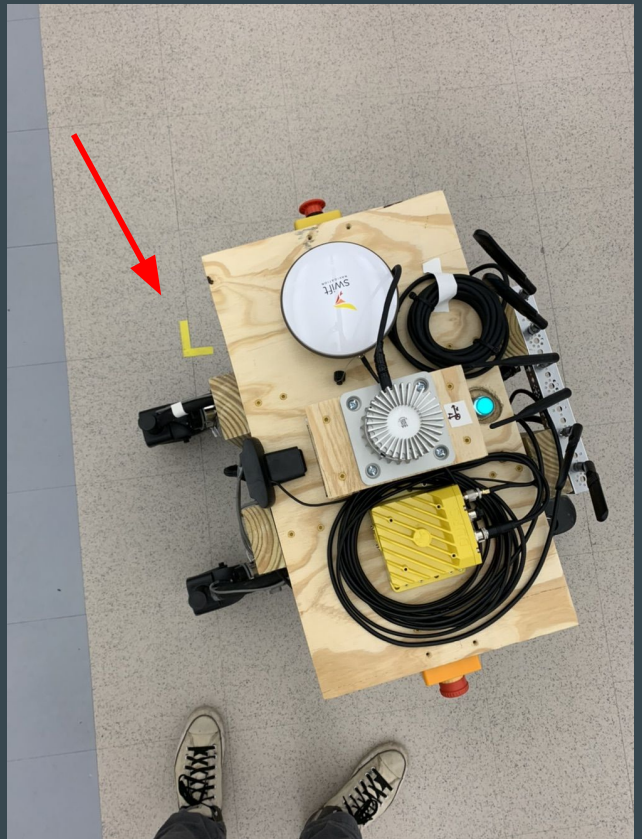
Localization via Odometry

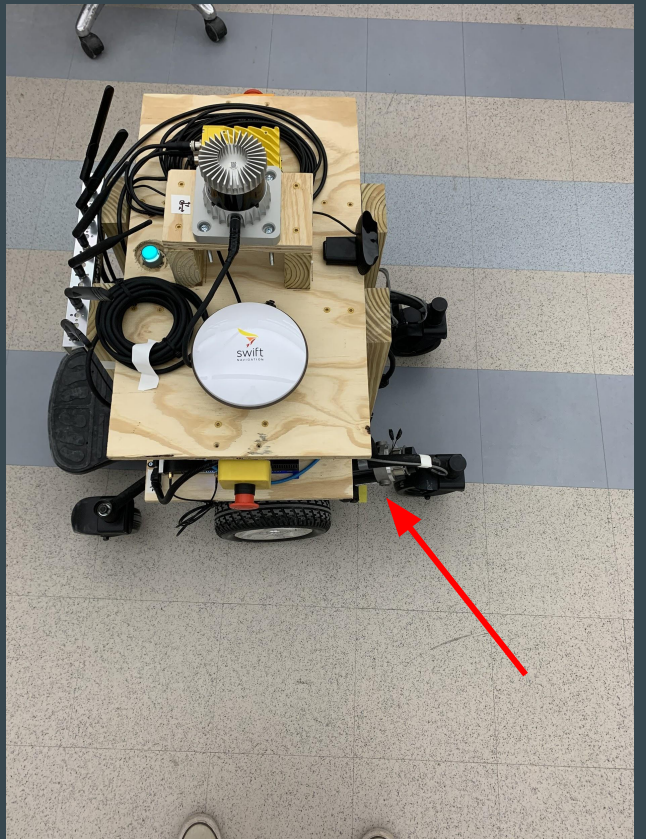
- ROS Transform (Tf) Tree - help from Sarmad Mehrdad
- Extended Kalman Filter (EKF) ROS - provided node of Robot Localization



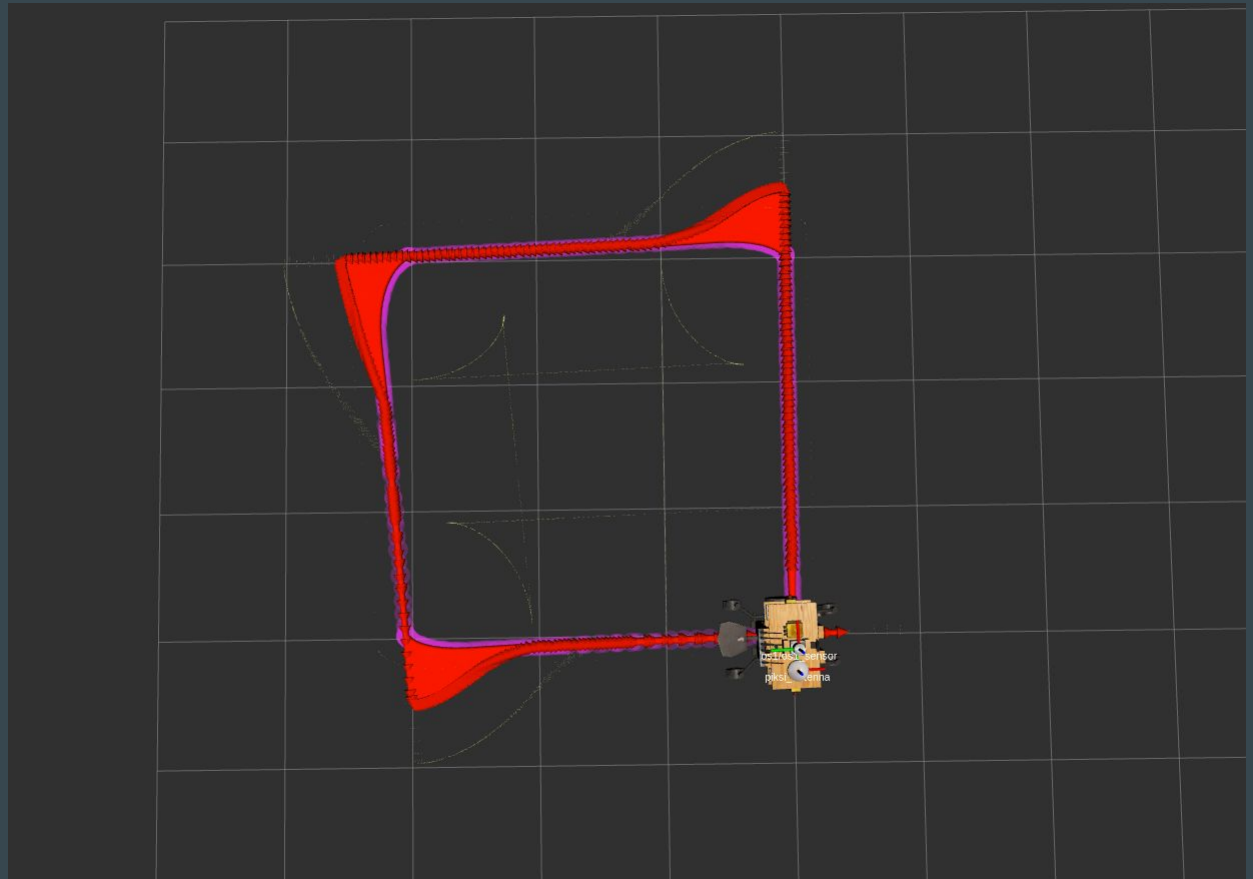


Before Fusion vs After Fusion after Three 10x10 Laps





Point-to-Point Navigation in Real World



Path Navigation (continued)

- Path Filtering

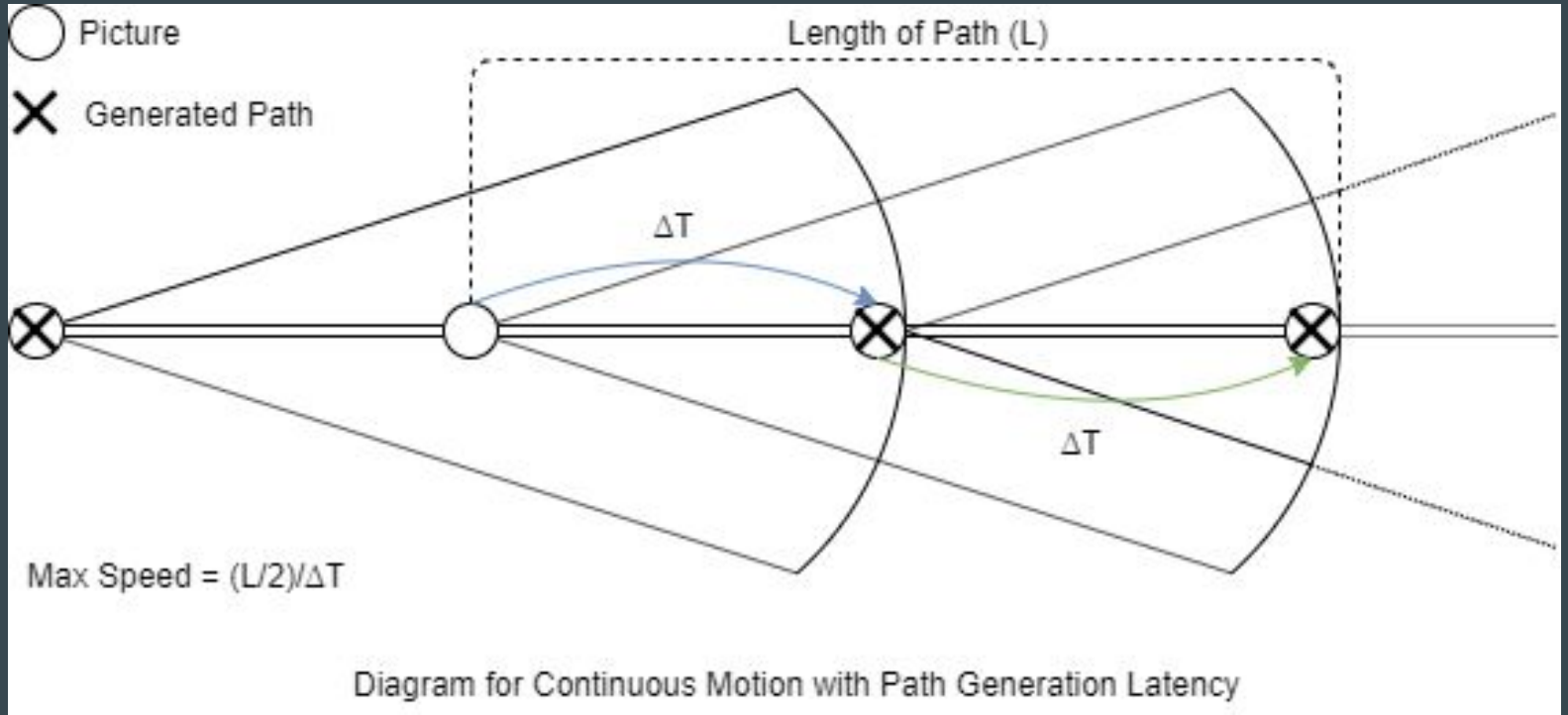
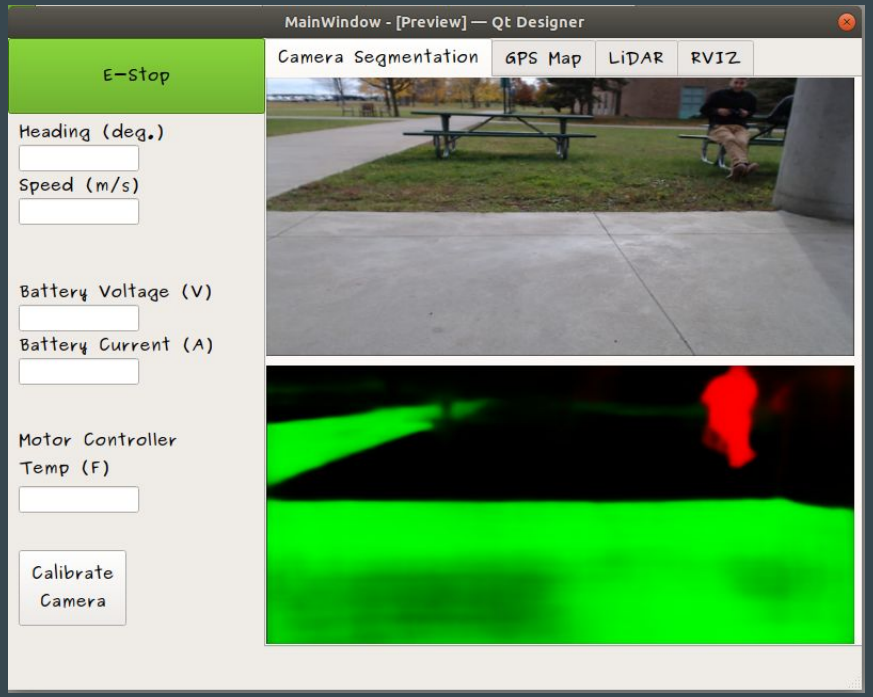


Diagram for Continuous Motion with Path Generation Latency

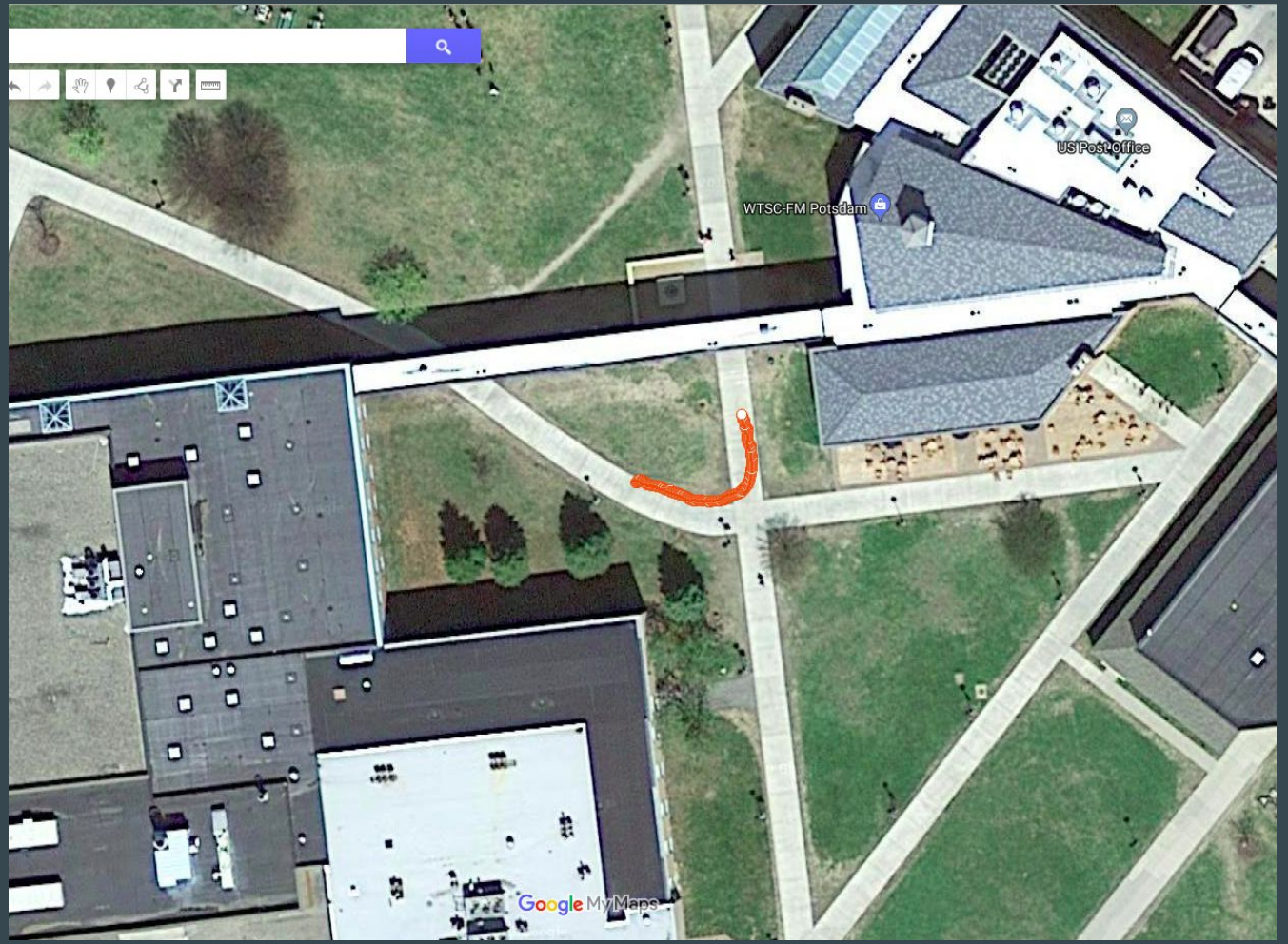
Results

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

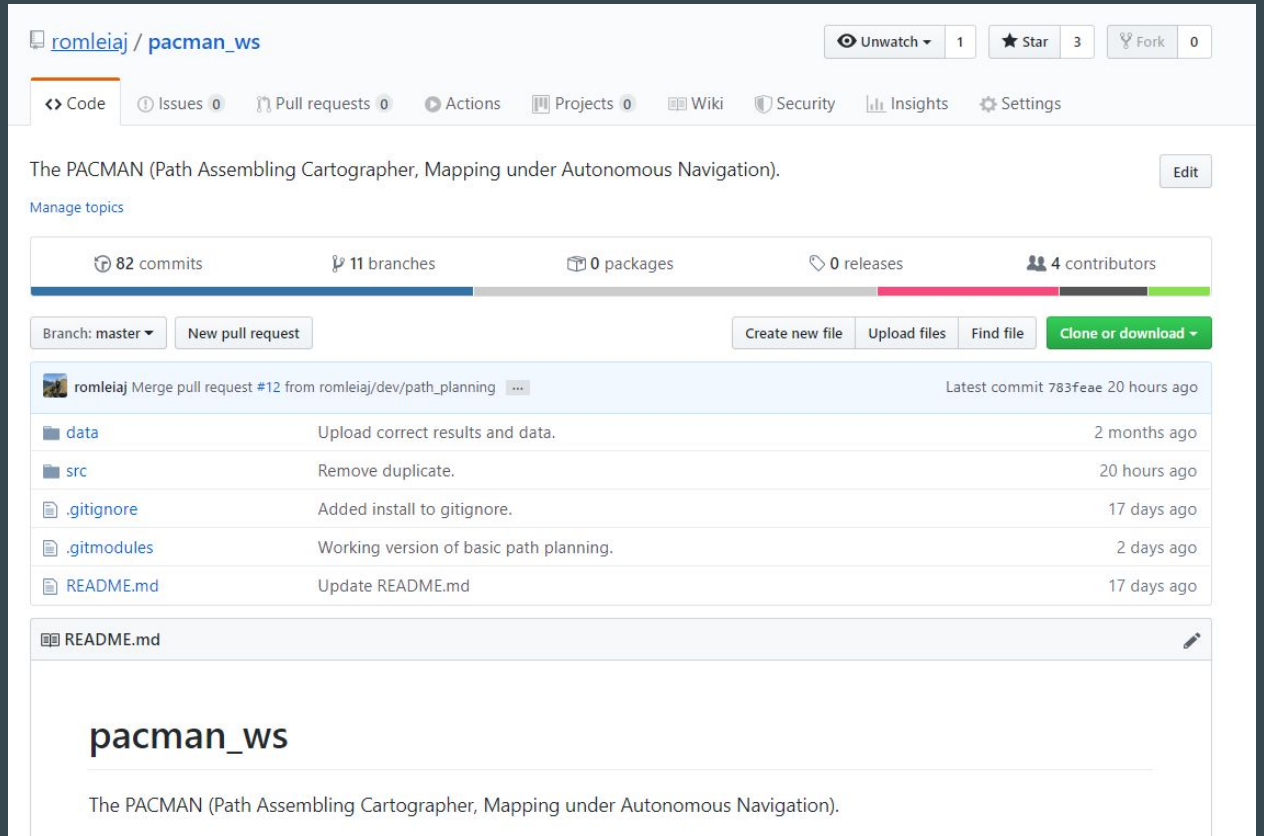
Future Work

- 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



Questions?