Jeremy Lu

jerlomy@umich.edu || (734) 882-5330 || Ann Arbor, MI || U.S. citizen

University of Michigan, Electrical and Computer Engineering M.S., Class of 2021, GPA 4.0/4.0

National Tsing Hua University, Power Mechanical Engineering B.S., Class of 2018, GPA 4.06/4.3

SKILLS

Programming Languages: C++, Python, MATLAB, Simulink, Java, Julia, LabView
Application: Gazebo, ROS2 (Robot Operating System 2), ROS, OpenRAVE, Arduino
Sensors: LiDAR, RGB camera, IMU
Version Control: GitHub

PROFESSIONAL EXPERIENCE

ADLINK Technology Limited Taipei, Taiwan

Robotics Engineer June 2020 – August 2020

- Built mobile robot running under either ROS (Robot Operating System) or ROS2.
- Simulated the robot with Gazebo to save time and evaluate the function I built.
- Implemented LOAM (LiDAR Odometry and Mapping in Real-time) and UKF (Unscented Kalman Filter) with a Velodyne's LiDAR, VLP-16, to lower the localization error of AMCL (Adaptive Monte Carlo Localization).
- Accelerated AMCL with GPU by 30% while the pose error is about 50% less.
- Replaced TEB (Timed Elastic Band) with DWA (Dynamic Window Approach) as the controller to rise the success rate of navigation by 20%.

Advanced Robotics Limited Taipei, Taiwan

Robotics Engineer November 2018 – April 2019

- Built an AGV (Automated guided vehicle) and implemented the system on ROS to deliver items in a hotel.
- Generated the map with gmapping, laser-based SLAM (Simultaneous Localization And Mapping).
- Localized the robot with AMCL and 2D LiDAR, RPLiDAR, while the initial pose is defined by AprialTag scanned by an RGB camera.

PROJECTS

Motion Planning – RRT* October 2020

- Computed a collision-free path for a robot's arm with RRT* (rapidly-exploring random tree star) such that the arm reaches the goal without hitting any obstacle.
- Created a plugin for OpenRAVE in **C++** to accelerate.

SLAM (Simultaneous Localization And Mapping) – SuMa++ March 2020 – May 2020

- Improved the performance of SuMa++ (Efficient LiDAR-based Semantic SLAM) on KITTI dataset by 12%.
- Integrated correntropy with the semantic ICP (Iterative Closest Point) while the labels were generated from a pre-trained model with raw LiDAR data.

Computer Vision – Depth Completion March 2020 – May 2020

- Computed dense depth data from color images and sparse LiDAR data.
- Modified the architecture of DeepLiDAR with our proposed block inspired by FuseNet.
- Reduce the error by 16% compared to the original architecture.

Self-Driving Car – Navigation October 2019 – December 2019

- Simulated vehicle driving in **Matlab** to race on predefined track and avoid random obstacles

· Applied MPC (Model Predictive Control) and integrated LQR (linear-quadratic regulator) for motion planning.