Divyanshu Sahu

divyanshurs.github.io

divyanshusahu26 • divyanshurs • +1(267)-665-9340 dsahu@seas.upenn.edu

EDUCATION

2021-2023 **University of Pennsylvania** Philadelphia, PA
M.S.E Robotics. GPA: 3.93/4
Relevant Coursework: Machine Perception, Learning in Robotics, Advanced Machine Perception (deep learning for computer vision), Control and Optimization in Robotics, Feedback Controls.

2015-2019 **Institute of Technology, Nirma University** Ahmedabad, GJ
B.Tech, Mechanical Engineering, October 2019. GPA: 8.25/10

SKILLS

Languages: C/C++, Python.
Frameworks: Pytorch, Robot Operating System (ROS/ ROS2), MATLAB, SolidWorks, Gazebo, RViz, Git, Linux.

RESEARCH AND PROFESSIONAL EXPERIENCE

mLAB, Autonomous Go-Kart Group (GRASP, UPenn) Philadelphia, PA
Research Assistant, (September 2022- Current)
- **Localization**: Research focused on optimal sensor fusion techniques with GPS, IMU, camera, and Velodyne LIDAR.
- **Calibration**: Performed **lidar-camera calibration** using 3D-3D point correspondence methods. Separated cone detection pipelines using lidar and camera and fused the detections for robustness against individual sensor failure.

SkyMul Atlanta, GA
Robotics Intern, (May-August 2022) **Eohan George**
- **Led the integration of hardware and end-to-end software pipelines** to build a waypoint task schedular using the ROS navigation stack for the desired behavior of a quadruped robot.
- Developed a low-level PID controller to stabilize the asymmetrical payload on the quadruped and integrated it with the high-level modifications of the navigation stack including the **global planner and TEB controller**.

Mowito Robotic Systems Bangalore, KA
Robotics Engineer, (July 2020- June 2021) **Puru Rastogi**
- **Improved runtime efficiency** by adding truncation and oscillation reduction algorithms to the MaxL local planner.
- Developed and implemented the '**robot_follower_node'** feature, intended to make the robot follow a human using only the MaxL local planner. [GitHub]
- Added modularity to the stack modules by developing a **behavior tree framework for the navigation stack**.

PROJECTS

Image Segmentation and Object Detection (Aug-Dec 2022) [GitHub] Philadelphia, PA
- **SOLO** (**Segmenting Objects by Locations**): Implemented end-to-end instance segmentation pipeline to categorize three objects in the scene using the COCO dataset.
- **YOLO-v1** (**You Only Look Once**): Performed object detection as a single-stage object detection pipeline and analyzed its performance using the calculated mAP.
- **Faster-RCNN** (**Region-Based Convolutional Neural Network**): Implemented the two-stage object detection framework for classification by training both the region proposal network and the object detector heads.

Path planning approaches for a planar quadrotor (Aug-Dec 2022) [GitHub] Philadelphia, PA
- **Min-snap trajectory path planning**: Implemented for a planar quadrotor using a quadratic program to minimize snap of the planned trajectory in flat coordinates.
- **MPC:** Used with discretized linear dynamics to solve the trajectory optimization problem over a receding horizon.
- **iLQR**: Implemented in an iterative manner for non-linear trajectory optimization.
- **LQR**: Used a time varying formulation for trajectory tracking for a pre-defined nominal path of the quadrotor.

SLAM using Particle Filter for a humanoid robot (May 2022) [GitHub] Philadelphia, PA
- **Fused IMU and Lidar data** to perform simultaneous localization and mapping of the THOR humanoid robot.
- Updated the map based on the log probabilities of the Lidar scan given the particle position and kept a track of obstacles by updating the log odds and binary occupancy map.

Autonomous Pick and Place Challenge (Dec 2021) [GitHub] Philadelphia, PA
- Developed functionalities and grasping strategies for static and dynamic blocks using the Franka Panda Robot arm in **ROS, Gazebo, and Rviz**. Translated the simulation results to the real robot.
- Implemented a **Rapidly exploring random tree (RRT)** planner in joint space for traversing between two joint configurations and used collision checking criteria in the workspace for evaluating its feasibility.