ROHITH VIKRAM SARAVANAN

+1 (202) 604 0766 - rohithvs@umd.edu - Linkedin - Github

EDUCATION

University of Maryland **College Park, MD**
Master of Engineering in Robotics *January 2023 - December 2024*
Coursework – Computer Vision, Planning for Autonomous Robots, Robot Modelling, Deep learning frameworks, Autonomous Aerial Robotics, Agile Robotics for Industrial Automation Competition, Reinforcement Learning

Anna University **Chennai, India**
Bachelor of Engineering in Mechanical Engineering *August 2017 - June 2021*

TECHNICAL SKILLS

Programming Languages and Tools: Python, C++, MATLAB, Bash, Git, Google Colab, Linux Ubuntu
Software and Simulation: PyTorch, ROS2, Gazebo, IsaacSim, PX4-Autopilot, Q-Ground Control, YOLOv5, Nav2, SolidWorks, Rviz, MoveIt, SLAM, Docker, QEMU
Libraries: Numpy, OpenCv, Pandas

WORK EXPERIENCE

Robotics Engineering Intern

Void Robotics, Florida, United States *June 2024 - August 2024*

- Simulated a custom autonomous robot for precise waypoint navigation in RViz using the Nav2 stack. Developed ROS nodes to bridge IMU and GPS data between ROS and Ignition Gazebo.
- Enhanced odometry accuracy by optimizing fusion parameters in the robot localization package, using an extended Kalman filter to integrate wheel encoder and IMU data, significantly reducing odometry errors and resolving drift issues.
- Set up a Docker-in-Docker (DinD) environment that enabled remote users to develop robotics applications on x86 architecture within Docker, while also supporting ARM64-based applications for the Jetson Orin. This setup allowed multiple developers to work efficiently across architectures without requiring direct access to the Jetson Orin hardware. ***[Nav2, C++, State Estimation, Sensor Fusion, Extended Kalman Filter, IMU, Docker, QEMU]***

PROJECTS

Perception for Autonomous Robots, *Perception*

- **Implemented Red Ball Trajectory Detection and Analysis:** Utilized OpenCV to filter the red channel from video frames, accurately detected the ball's coordinates, and applied Standard Least Squares to fit a parabolic trajectory.
- **Panoramic Image Stitching:** Created a panoramic image by stitching 4 images using SIFT for feature extraction and matching.
- **Stereo Vision Implementation:** Developed a stereo vision system by performing camera calibration, image rectification, and depth image computation from datasets containing two images taken at different positions. ***[OpenCV, RANSAC, Homography, SIFT Detector, Image Stitching, Stereo Vision, Camera Calibration]***

Autonomous Navigation on TurtleBot3 using DQN, *Robot Learning*

- Developed a TurtleBot3 navigation system using Deep Q-Learning to find optimal paths and avoid dynamic obstacles using LIDAR, achieving over 67 percent success in a simulated environment.
- Designed a complex reward function to enhance decision-making accuracy.
- Trained and tested the DQN model for effective navigation and collision avoidance towards randomly assigned goals. ***[Reinforcement Learning, DQN, Dynamic Obstacle Avoidance, ROS2]***

Drone Localization using Multiple Apriltags, *Aerial Robotics*

- Executed precise 3D positioning of the drone using trilateration with known AprilTag coordinates.
- Validated the accuracy by comparing trilaterated positions with data from the drone's IMU, ensuring reliable localization. ***[PX4-Autopilot, Q-Ground Control, Trilateration, Object Tracking]***

Self Docking Robot, *Computer Vision*

- Utilized a custom-trained YOLOv5 algorithm for obstacle detection, implemented obstacle avoidance, and integrated QR code scanning for intelligent docking capabilities. ***[YOLOv5, Self Docking, Obstacle Avoidance]***

Implementation of the Path planning algorithms on a Differential Drive TurtleBot3 robot, *Planning*

- Implemented the BFS, Dijkstra, A-star algorithm for path planning of a differential drive (non-holonomic) mobile robot (Turtlebot) in a custom obstacle map using OpenCV.
- Successfully integrated the path planning algorithms with ROS2 to simulate the turtlebot robot following the optimal path to move from the start position to the goal position. ***[Motion Planning, BFS, Dijkstra, A* algorithm, Differential drive TurtleBot3, ROS2, Gazebo Simulation]***