Smit Dumore

College Park, MD, USA smitd@umd.edu +1 (240) 463 5001 Portfolio LinkedIn

EDUCATION

University of Maryland, College Park, *Masters in Robotics (GPA 4.0/4.0)* 08/2022 – Expected (05/2024)
College Park, MD

Control of Mobile Robots, Perception for Robotics, Machine learning, Planning for Robotics.

Vishwakarma Institute of Technology, Pune, India, 2017 – 2021 | India
BTech. Mechanical Engineering (GPA 8.48/10.0)
Machine Design, Mechatronics, Kinematics of Mechanisms, Differential Equations.

PROFESSIONAL EXPERIENCE

Open Source Developer, *Open Robotics* 11/2023 – present | Remote, USA
- Collaborating with **Steve Macenski** to enhance **ROS Nav2** functionalities.
- Designing a loopback simulator for ROS2, simplifying Gazebo for streamlined high-level behavioral testing.
- Improving documentation for various ROS2 packages to enhance user and developer accessibility.

Robotics Software Engineer, *Botysnc* 07/2021 – 06/2022 | India
- Developed a **Hyper Accurate Docking Algorithm** for an **AMR** (Autonomous Mobile Robot) for autonomous recharging in **C++14.** Experimented with **Iterative Closest Point** scan matching and reflective tape based docking. The docking accuracy obtained was ±2cm. **(video)** .
- Worked on Landmark based **Extended Kalman Filter** for localisation of AMR in long corridors. Obtained RMSE of 0.2m and 5 degrees.
- Successfully tuned and tested custom **Navigation Stack** for lifting and tugging applications upto 2 tonnes in various industrial sites.
- Developed a Teleoperator package in **C++** for controlling an AMR with a joystick.

PROJECTS

Autonomous Racing Planning and Control stack, 03/2022
Vishwakarma Institute of Technology, Pune, India
- Implemented real-time **RRT and RRT*** path planning algorithms using C++11 for local planning in a head-to-head autonomous racing car. Implemented optimization technique using **Kd-Tree** to improve the algorithm's performance *(github)*
- Implemented a **Pure Pursuit** local planner for the vehicle to follow a global path on the racetrack.
- Implemented a **Model Predictive Controller (MPC)** to find optimal control inputs for trajectory tracking and obstacle avoidance. *(github)*
- Used **OSQP** library to obtain a solve time of **30ms** on a Nvidia Jetson NX for a linear MPC subject to linear constraints and a quadratic cost function.

Stereo Visual SLAM, *University of Maryland, College Park. (github)* 06/2023 – present
- Successfully implemented Stereo Visual SLAM to estimate globally consistent camera trajectory and build a sparse 3D map of the environment.
- Utilized **GFTT** algorithm for feature detection accurate feature identification across frames.
- Employed **triangulation** to accurately determine the 3D positions of keypoints.
- Implemented **Lucas Kanade** optical flow techniques for feature tracking.
- Utilized the **g2o** library for **graph optimization**, implementing **Bundle Adjustment** as a backend technique to obtianed refined camera poses and 3D feature locations

Reinforcement Learning Pacman Agent, 01/2023
University of Maryland, College Park. (github)
- Implemented a **BFS , Best first Search, Astar, Dijkstra** path finding algorithm to search Ghosts in a Pacman environment.
- Modelled the Pacman environment as a **MDP** (Markov decesion processes) and used Value Iteration to maximise score of Pacman against stochastic and adversarial ghosts.
- Used Q-learning to learn optimal actions in a state to maximise Pacman score.

Dynamic Window Approach Local Planner, 10/2022
University of Maryland, College Park. (github)
- Developed a kino-dynamic local planner for a turtlebot using the Dynamic Window Approach.
- Planner is capable of dodging **dynamic obstacles**.
- Planner generates paths that are **kinematically feasible** and locally optimal.

YOLO From Scratch, *University of Maryland, College Park. (github)* 09/2023 – present
- Developed a custom implementation of the YOLO v1 (**You Only Look Once**) object detection.
- Implemented single-pass object detection and regression-based bounding box prediction using **Convolution Neural Networks** (CNN).
- Utilized YOLO architecture with grid cells and anchor boxes for object detection.
- Calculated **Mean Average Precision** (mAP) to assess model accuracy across object classes.

SKILLS

Programming Languages and Tools
C++, Python, MATLAB, Julia, ROS, OpenCV, PCL, Rviz, Gazebo, pytorch