MADHU NARRA CHITTIBABU

madhunc117@gmail.com | linkedin.com/in/madhunc | github.com/Madhunc5229 | Website: MadhuNC | +1 (240)-259-1980

Domain Skills: Computer Vision, Localization, Path Planning, Controls, Software Engineering

WORK EXPERIENCE

Toyota Material Handling North America (TMHNA) – *Robotics Software Engineer* Sept 2023 – Present | Syracuse, NY

- Accelerated a 3D LiDAR scan matcher using **nonlinear optimization** from ~50 seconds per scan to <1 second convergence and enabled 10 Hz matcher output. Achieved **97.31%** convergence under defined success criteria. Submitted an invention disclosure for a novel mapping technique.
- Built an adaptive **Kalman filter tuning** pipeline to fuse wheel encoder and IMU data for vehicle odometry. Improved average positional RMSE by **47.5%** across 9 runs per vehicle covering diverse driving scenarios.
- Integrated odometry and LiDAR localization into a REP-105 compliant localization engine and deployed it on an NVIDIA Jetson **Orin Nano**. Validated against ground truth in a 38,750 sq ft warehouse and achieved **0.5 m** average translational accuracy at **10 Hz**.
- Designed a modular autonomy **software architecture**, enabling plug-in localization engines without downstream changes. Presented a production-truck demo of location-based behavior control using two localization engines.
- Led a small cross-functional team to deliver an in-house **Autonomous Mobile Robot** (AMR) stack on a production vehicle using ROS2 Nav2. Integrated custom localization and perception to achieve autonomous point-to-point navigation.
- Led vehicle integration and **platform bring-up** on a production pallet truck. Installed and validated 3D LiDAR, cameras, E-stop, and CAN breakouts using electrical schematics.
- Implemented **ROS2 CANopen** drivers using the vehicle's CANopen object dictionary. Enabled ROS2-to-CAN command and telemetry for on-vehicle control and integration.
- Tuned **Smac Hybrid A*** and the **Regulated Pure Pursuit** controller for pallet-truck navigation performance in Nav2.
- Improved robot to server communication by migrating ROS2 middleware to **Zenoh** and tuning data broadcasting. Reduced network traffic by **98.3%**.
- Appointed as the **North America (NA) representative** for Toyota's GT (Global Technology) initiative. Defined NA requirements and benchmarks for GT-VSLAM (Visual Simultaneous Localization and Mapping), evaluated 6 solutions and drove adoption of one solution.

Humatics – *Robotics Software Engineer* Jun 2022 – Aug 2022 | Waltham, MA

- Implemented dynamic transform functionality for a mobile 6-DOF robotic arm to enable object manipulation during motion.
- Performed dynamic **3D pose estimation** on **sparse point cloud data** obtained by **radio frequency triangulation** on asynchronous UDP datagrams from a base station at frequency of **125Hz**. Obtained a **RMSE** of **0.001m**. Click *here* for demo video.

Caterpillar Inc. – *Design Engineer* Jul 2018 – May 2021 | Bangalore, India

- Developed engineering designs for **high-pressure** machinery in Oil & Gas industry using **Solidworks**, **AutoCAD**, **ANSYS** and **CFD** analysis.
- Effectively led a team, implemented **automation** for quote drawings preparation, resulting in a **10-hour** reduction in work time.

EDUCATION

University of Maryland, College Park, MD | *M. Eng, Robotics* | *GPA: 3.73 / 4.0* Aug 2021 – May 2023

PES University, Bangalore, India | B. Tech, Mechanical Engineering Aug 2014 – May 2018

SKILLS & ABILITIES

Programming Languages: C, C++, Python, MATLAB

Software Tools/Platforms/Practices: ROS2, Docker, PCL, Zenoh, BehaviorTree.CPP, Ceres-Solver, NVIDIA Jetson, CloudCompare, Linux, PyTorch, OpenCV, Open3D, Gazebo, RViz, Raspberry-Pi, CMake, CI/CD, Gtest, Git, UML, AIP (Agile Iterative Process)

TECHNICAL PROJECTS

Perception:

Real-Time Semantic Segmentation | *Deep Learning, Segmentation* | *Python, PyTorch* | *GitHub*

- Developed a custom **encoder-decoder** architecture using **DenseNet169** for real-time **scene understanding** on the **Cityscapes** dataset, Improved the **mIoU** from **45%** to **77.5%** by implementing a **custom Multi-Class Focal Loss** in PyTorch to address class imbalance.

Affine Structure from Motion | *3D Geometry, Structure from Motion, Optical Flow* | *Python, OpenCV* | *GitHub*

- Implemented **Harris corner** detection and **Lucas-Kanade** optical flow algorithms to accurately to **detect** and **track** 500 feature points.
- Recovered **3D point cloud** and camera path by factorizing the measurement matrix and using **Cholesky** decomposition for disambiguation.

Autonomous Mobile Robot | *SLAM, Sensor Fusion* | *Raspberry-Pi, Pi-Camera* | *YouTube*

- Designed and built a wheeled robot for autonomous collection of materials from a cluttered environment, using a **localization** method that **fuses** data from **IMU**, **ultrasonic sensor**, and wheel odometry, and an object **detection** and **navigation** schema using **machine vision**.

ICP for Shape Alignment from 2D Point Cloud | *Point Clouds* | *Python, NumPy, OpenCV* | *GitHub*

- Implemented **I**terative **C**losest **P**oint algorithm to align two sets of 2D points using global image transformation with Euclidean distance as the **point-to-point** distance metric. Obtained an average **L2 error** (difference of **Euclidean distance transform** of two images) of **50**.

Monocular Visual SLAM | *3D geometry, VSLAM* | *Python, OpenCV* | *GitHub [On-going]*

- Performed **Zhang's** camera calibration, reconstructed a 3D scene and simultaneously obtained the camera poses from a given set of images using their feature points correspondence by performing **triangulation**, **PnP** using RANSAC and **nonlinear optimization** methods.

Uncalibrated Stereo Vision | *3D Depth Estimation* | *Python, OpenCV* | *GitHub*

- Utilized ORB feature detection and estimated **Fundamental Matrix** using **RANSAC**, extracted camera poses from **Essential Matrix** and disambiguated the poses by checking the **Chirality** condition and performing **linear triangulation**. Implemented pixel-wise block matching with Sum of Squared Differences (**SSD**) on the rectified images to generate the **disparity** and **depth** map.

Malaria Detection using Convolution Neural Network (CNN) | *Deep Learning, Classification* | *Python, PyTorch* | *GitHub*

- Developed a Computer Aided Detection System for the classification of Malaria infected cells in thin blood smear images using a custom CNN network trained on the *National Library of Medicine Malaria Dataset* achieved a F1 score of **96.02%**.

AR-Tag Detection and Pose estimation | *Augmented Reality* | *Python, NumPy, OpenCV* | *GitHub*

- Designed a pipeline for the detection and pose estimation of a custom AR-Tag along with superimposing 2D image & 3D cube on it using Homography, Camera Projection Matrix for mixed reality applications.

Motion/Path Planning:

RRT-star Implementation for Mobile Delivery Robot | *Path Planning* | *Python, ROS* | *GitHub*

- Developed a RRT-star (**2500** nodes) path planning module in ROS to efficiently navigate a delivery robot (Turtlebot) between **six** delivery points and **one** charging point. Built a custom 10mX10m Gazebo world and utilized OpenCV for real-time path visualization.

A-star Algorithm implementation for Turtlebot | *Path Planning* | *Python, ROS* | *GitHub*

- Developed an A-star based planning module in ROS for Turtlebot3 with **5 non-holonomic** actions and used **weighted Euclidean distance** as the Heuristic to improve the computation speed of the algorithm. Used Gazebo and OpenCV for real-time path visualization.

Design and Simulation of UR10e Robot Manipulator | *Motion Planning* | *Python, ROS, Gazebo, RViz, MoveIt* | *GitHub*

- Designed a **6-DOF** manipulator using **SolidWorks**, simulated it using the **URDF** model in RViz & Gazebo, implemented **MoveIt** package for planning and executed robot actions in RViz, and validated inverse and forward **kinematics** calculations through simulations.

C++/ROS:

Human Detection and Tracking | *Object Detection* | *C++, OpenCV* | *GitHub*

- Developed a C++ module using pretrained **Yolov5s** model for human detection, employing **Agile** Iterative Process with **Doxygen** documentation, **Gtest** unit testing for **100%** code coverage, continuous integration, sprint planning and backlog notes within **two**-weeks.

Obstacle avoidance using Turtlebot3 | *Software Development for Robotics* | *ROS2 Humble, C++* | *GitHub*

- Implemented obstacle avoidance navigation in ROS2 for Turtlebot3 by reading LiDAR output, utilizing standard agile methodology.

MicroMouse Maze Solver | *Path Planning* | *C++* | *GitHub*

- Programmed Depth-First Search (DFS) and backtracking for path planning of a robot to reach the goal node in a varying 16X16 maze.

Urban Search & Rescue (US&R) | *Perception, Path Planning* | *C++, ROS* | *GitHub*

- Programmed **two** Turtlebots to perform US&R operation, one explores the map, broadcasts the location of victims to the other bot, which rescues the victims using **SimpleActionClient** server. Used ROS **move_base** for navigation of the bots in the Gazebo world.

Autonomous Warehouse Inspection | *Software Development for Robotics* | *ROS2, C++* | *GitHub*

- Developed an autonomous robot, using ROS2 **Nav2** stack in Gazebo capable of defect/hazardous-event detection on a factory floor.
- Implemented the Agile process, including **UML** diagrams, unit tests, and a project package with **GitHubCI** and **CodeCov** code coverage.