International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

# SLAM Mapping Robot for Autonomous Navigation

**Akshay V**
Dept. of Electronics and Communication Engineering
K. S. Institute of Technology
Bengaluru, India

**B Rohith**
Dept. of Electronics and Communication Engineering
K. S. Institute of Technology
Bengaluru, India

**Bhumika V**
Dept. of Electronics and Communication Engineering
K. S. Institute of Technology
Bengaluru, India

**Govindan G**
Dept. of Electronics and Communication Engineering
K. S. Institute of Technology
Bengaluru, India

**Dr. Saleem S Tevaramani**
Dept. of Electronics and Communication Engineering
K. S. Institute of Technology
Bengaluru, India

***Abstract* : Autonomous robots are increasingly used in delivery, surveillance, and industrial automation, yet navigating unknown indoor environments without predefined maps or GPS remains a challenge. This paper presents an autonomous mobile robot that maps its environment, navigates safely, and physically interacts with objects through an integrated fetch-and-deliver task. The system performs Simultaneous Localization and Mapping (SLAM) on a Raspberry Pi, fusing data from an RPLiDAR A1, an Inertial Measurement Unit (IMU), an ultrasonic sensor, and a camera to build an occupancy map while estimating the robot's pose in real time. A* provides global path planning and the Dynamic Window Approach (DWA) provides local obstacle avoidance, while an ESP32 drives the motors through an L298N driver. The camera detects the target object and determines its location; the robot approaches it, and a robotic arm with a gripper grasps it, carries up to 150 g, and releases it at the designated location. The process runs as a closed loop of sensing, localization, planning, and action until the task is complete. Results from the prototype indicate that the robot builds a usable map, follows planned paths while avoiding obstacles, and completes the fetch-and-place sequence. The discussion shows that fusing complementary sensors improves perception reliability and that combining navigation with manipulation extends the robot beyond passive movement, although indoor operation, sensor range, and limited onboard computing constrain performance. The work demonstrates integrated perception, localization, planning, and manipulation on a low-cost embedded platform, with potential in warehouse automation and assistive robotics.**

***Keywords : SLAM, autonomous mobile robot, LiDAR, sensor fusion, path planning, A*, DWA, robotic manipulation, Raspberry Pi, ROS***

## I. INTRODUCTION

Why do autonomous robots need SLAM? Robots are increasingly deployed in delivery, surveillance, warehouse automation, healthcare assistance, and industrial material handling, where they must move through spaces without human guidance. Indoors, GPS is often unreliable or unavailable, and a prior map may not exist, so the robot must understand its surroundings by itself. Simultaneous Localization and Mapping (SLAM) addresses this by building a map of an unknown environment while estimating the robot's own position within it [2]. Open-source packages such as G-Mapping, Hector SLAM, Cartographer, and SLAM Toolbox have made 2D LiDAR SLAM widely available [3]–[6], and ROS provides the common software layer that connects them to the rest of the robot [1].

What limits traditional navigation? Systems that depend on pre-programmed paths or static maps adapt poorly to dynamic, unstructured environments. SLAM-based navigation instead lets the robot perceive its surroundings in real time and combine the resulting map with path planning, using A* for global routing [13] and the Dynamic Window Approach (DWA) for local obstacle avoidance [14], as in modern navigation stacks [15]. These capabilities become essential when the robot must do more than move, such as identify, retrieve, and deliver objects, because that requires perception, decision-making, and physical interaction



(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

to work together.

What does this work propose? This paper presents an autonomous mobile robot that maps its environment, localizes itself, navigates safely, and performs an object fetch-and-deliver task. How is it built? A Raspberry Pi acts as the central processing unit and fuses data from LiDAR, an Inertial Measurement Unit (IMU), ultrasonic sensors, and a camera. LiDAR provides the spatial measurements for map construction, the IMU improves localization by estimating orientation and motion, the ultrasonic sensors add close-range obstacle detection, and the camera supports object detection. How does the robot interact with objects? Once the camera locates the target, the robot approaches it, grasps it with a robotic arm and gripper, carries it to the delivery location, and releases it. This turns the robot from a passive navigator into an active agent, and it aims to show a low-cost, scalable, and modular approach to autonomous

## II. RELATED WORK

### A. 2D LiDAR-Based SLAM

Grisetti et al. [3] proposed a Rao-Black wellized particle filter for grid mapping, using adaptive resampling and an odometry-informed proposal to reduce the number of particles; it needs odometry, and its cost grows with particle count and map size. Kohlbrecher et al. [4] (Hector SLAM) relied on fast scan matching of high-rate LiDAR data without odometry, but it has no explicit loop closure and can degrade in long, featureless corridors. Hess et al. [5] (Cartographer) built submaps with branch-and-bound scan matching for real-time loop closure, at the cost of heavier computation and many parameters to tune. Macenski and Jambrecic [6] introduced SLAM Toolbox, a graph-based mapping package that supports long-term mapping and map serialization, though it targets mapping rather than complete task execution.

### B. Visual and RGB-D SLAM

Mur-Artal and Tardós [7] presented ORB-SLAM2 for monocular, stereo, and RGB-D cameras with loop closing and relocalization; its sparse feature map is not directly usable as an occupancy grid for planning, and tracking can fail in low-texture or dynamic scenes. Labbé and Michaud [8] developed RTAB-Map, which uses appearance-based loop closure with memory management for long-term online operation, but it is demanding for low-power boards. Bloesch et al. [9] proposed CodeSLAM, a compact learned code for dense depth that is optimized jointly with camera pose, although depth quality can drop when the environment differs from the training data.

### C. LiDAR-Inertial Fusion

Zhang and Singh [10] introduced LOAM, which splits the problem into high-frequency odometry and low-frequency mapping for real-time 3D LiDAR mapping, but drift accumulates over large-scale trajectories. Shan and Englot [11] (LeGO-LOAM) made this lightweight for embedded use by exploiting the ground plane, which limits it to ground vehicles with 3D LiDAR. Shan *et al.* [12] (LIO-SAM) tightly coupled LiDAR, IMU, and optional GPS in a factor graph with a local sliding window, reducing drift, but it requires a 3D LiDAR, a 9-axis IMU, and precise LiDAR-IMU extrinsic calibration.

### D. Navigation and Path Planning

Hart *et al.* [13] defined the A* heuristic search, which returns a minimum-cost route on a known map but does not consider robot dynamics. Fox *et al.* [14] proposed the Dynamic Window Approach (DWA), which samples velocities within the robot's dynamic limits for reactive obstacle avoidance; it can become trapped in local minima and depends on a global plan. Macenski *et al.* [15] described Nav2, a modular ROS 2 navigation stack with planners, controllers, and recovery behaviors; it addresses navigation only, so manipulation lies outside its scope.

### E. Benchmarking of Open-Source SLAM

Santos *et al.* [16] compared 2D SLAM techniques available in ROS, including G-Mapping, Hector SLAM, Karto SLAM, Core SLAM, and Lago SLAM, and showed that results depend on the environment and sensor setup. Yagfarov *et al.* [17] compared 2D LiDAR SLAM algorithms against precise ground truth, and Filipenko and Afanasyev [18] compared several SLAM systems for an indoor mobile robot. These studies focus on mapping accuracy and do not evaluate integrated navigation with object handling.

### F. Research Gap

Table I compares representative works with the proposed system. To the best of our review, none of the surveyed works combines (i) a low-cost Raspberry Pi and ESP32 control architecture, (ii) front and rear LiDAR fused with IMU and ultrasonic sensing, (iii) real-time occupancy-grid mapping with A* global and DWA local planning, and (iv) a robotic arm and gripper for fetch-and-place in a single indoor unit. This is the gap the proposed robot is designed to fill.

TABLE I.: COMPARISON OF SELECTED RELATED WORKS

| Ref. | Key feature | Limitation |
|---|---|---|
| [3] | Rao-Black well sized particle filter grid mapping | Needs odometry; cost grows with particles |
| [4] | Scan-matching SLAM without odometry | No loop closure; weak in featureless corridors |
| [5] | Submaps with real-time loop closure | High computation; heavy tuning |
| [7] | Real-time monocular, stereo, and RGB-D SLAM | Sparse map; fails in low-texture scenes |



(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

| | | |
|---|---|---|
| [8] | Long-term appearance-based mapping | Heavy for low-power boards |
| [12] | Tightly coupled LiDAR-IMU-GPS factor graph | Needs 3D LiDAR and precise calibration. |
| [15] | Modular navigation stack | Navigation only; no manipulation |
| [16] | ROS 2D SLAM comparison | Mapping accuracy only |
| **Proposed** | ***Dual LiDAR, IMU, and ultrasonic fusion; A + DWA; arm-gripper fetch-and-place (150 g)**** | **Experimental validation to be reported** |

## II. PROBLEM IDENTIFICATION AND OBJECTIVES

### A. Problem Identification

Modern autonomous systems face multiple challenges, including computational limitations affecting real-time decision-making, sensor drift and calibration errors reducing accuracy, and wireless latency delaying communication. Additionally, limited perception can lead to navigation issues, while data association and parameter tuning impact reliability. Dynamic environments and scalability constraints further complicate performance, highlighting the need for robust and efficient autonomous navigation solutions.

Objectives

The objectives of this work are to:

- To combine LiDAR, IMU, ultrasonic, and camera data for accurate mapping and to reduce drift and blind spots.
- To achieve fast, safe navigation using a Raspberry Pi and an ESP32 for real-time onboard control, with A* and DWA for route planning and obstacle avoidance.
- To enable autonomous fetch-and-deliver using a robotic arm and gripper in a modular design that is easy to upgrade.

## III. PROPOSED SYSTEM ARCHITECTURE

Fig. 1 shows the block diagram of the proposed SLAM-based mobile robot. The system has four functional parts: a sensing unit that perceives the environment, a central processing unit that maps and plans, a motion-control unit that drives the wheels, and a manipulation and communication layer that handles objects and remote access. The Raspberry Pi is the hub, receiving data from all the sensors and sending commands to the actuators.

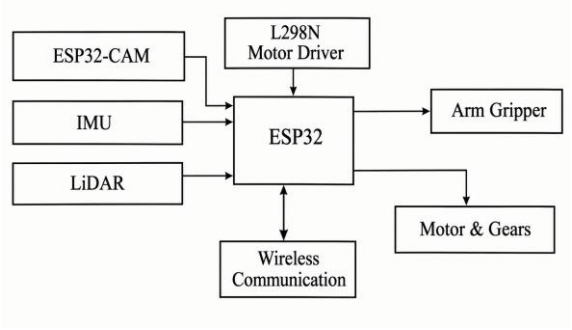



Fig. 1. Block diagram of the proposed SLAM Mapping Robot for Autonomous Navigation

### A. ESP32

The ESP32 receives movement commands from the Raspberry Pi and turns them into motor control signals. It acts as a dedicated low-level controller, so the Raspberry Pi can concentrate on SLAM and planning. The L298N motor driver is a dual H-bridge that takes these low-power signals and supplies the current the motors need, setting direction and speed. The motors and gears move the robot along the planned path and provide the torque needed to carry the payload.

### B.ESP32 Camera

The ESP32-CAM is used in the SLAM robot to capture real-time images of the surrounding environment. It helps in visual monitoring, obstacle identification, and understanding nearby objects. The camera provides visual information that supports autonomous navigation and improves the robot's environmental awareness. It is a compact, low-cost solution for robotic vision applications.

### C.Arm Gripper

The robotic arm and gripper receive their commands from the Raspberry Pi. Once the robot is positioned near the target, the arm grasps the object, the robot carries it to the delivery location, and the gripper releases it. The payload capacity is up to 150 g.

### D .LiDAR

The LiDAR is the main mapping sensor of the robot. It emits laser pulses and measures the time each pulse takes to return after reflecting off a surface, which gives the distance to that surface. As the sensor rotates, it collects a 360° set of distance readings called a scan. The Raspberry Pi uses these scans in the SLAM algorithm to build the occupancy-grid map and to estimate the robot's position within it. LiDAR gives accurate range data and is not affected by lighting conditions, which makes it reliable for indoor mapping. It can miss transparent or highly reflective surfaces such as glass, which is one reason the ultrasonic sensor is added.



(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

E.IMU

The IMU measures the robot's own motion rather than the environment. It typically combines an accelerometer, which measures linear acceleration, with a gyroscope, which measures rotation rate, and some units also include a magnetometer for heading. From these readings the system estimates the robot's orientation and how it is moving. This helps localization in two ways. It supplies motion estimates between LiDAR scans, and it corrects errors during turns or on uneven floors, where scan matching alone can drift. The IMU accumulates error over time, so its data is fused with the LiDAR data and not used alone.

F. Wireless Communication

The wireless communication block links the Raspberry Pi to a remote device in both directions. It lets the user send commands and monitor the robot's status, map, and task progress, which supports remote operation and debugging.

H.LN298N Motor Driver

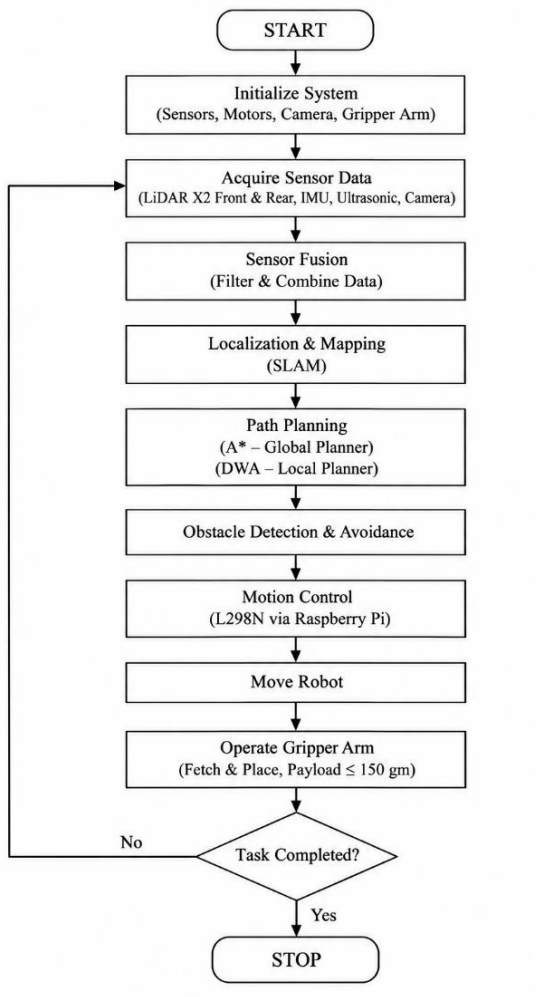
The L298N Motor Driver is used in the SLAM robot to control the speed and direction of DC motors. It acts as an interface between the microcontroller and motors, allowing forward, backward, left, and right movements. It supports autonomous navigation by controlling the robot's wheel movements based on commands from the controller.

I.Motors & Gears

DC motors and gears are essential components of the SLAM robot, responsible for providing movement and controlled navigation. The DC motors convert electrical energy into mechanical energy to rotate the wheels. Gear mechanisms help adjust the speed and torque according to the robot's movement requirements. They provide sufficient torque for smooth movement and better stability on different surfaces. The motors are controlled through the L298N motor driver, which receives commands from the microcontroller. By controlling motor speed and direction, the robot can move forward, reverse, turn left, and turn right, supporting autonomous navigation and mapping.

## IV.METHODOLOGY

The working of the system is summarized by the flowchart in Fig. 2.




On start-up, the Raspberry Pi initializes the sensors, motors, camera, ESP32, and gripper arm so that every component is ready before autonomous operation begins. The robot then acquires data from the front and rear LiDAR units, the IMU, the ultrasonic sensor, and the camera. These readings are filtered and combined in a sensor-fusion stage, and the fused data is passed to the SLAM module, which builds a real-time occupancy-grid map while estimating the robot's position within it. Using this map, the A* algorithm computes an optimal global route to the target location, and the Dynamic Window Approach (DWA) adjusts the robot's velocity in real time so that it follows the route within its motion limits. Obstacles detected by the sensors are checked against the planned path, and the robot modifies its motion to avoid a collision when one is found. The Raspberry Pi then sends movement commands to the ESP32, which drives the motors through the L298N motor driver, and the robot moves toward the target. Once it is positioned correctly, the gripper arm performs the fetch-and-place operation, grasping the object and releasing it at the delivery location with a payload of up to 150 g. Finally, the system checks whether the task is completed. If it is not, the cycle returns to sensor data acquisition and repeats, so the robot keeps tracking changes in its environment; if it is, the robot stops.



(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

## IV. EXPECTED BENEFITS AND LIMITATIONS

## V. EXPECTED BENEFITS AND LIMITATIONS

### A.Expected Benefits

Table II summarizes how the proposed design differs from a conventional navigation system that relies on pre-programmed paths or static maps. Building the map with SLAM lets the robot operate in an unknown indoor space without prior information or GPS. Fusing the front and rear LiDAR, IMU, and ultrasonic data is intended to improve localization accuracy and obstacle detection compared with a single sensor. Combining A* global planning with DWA local planning allows the robot to follow an efficient route while reacting to obstacles in real time. The robotic arm and gripper extend the robot from navigation alone to fetch-and-place tasks, and the Raspberry Pi and ESP32 architecture keeps the design low-cost and modular. These benefits follow from the design and remain to be quantified experimentally.

TABLE II: DESIGN COMPARISON WITH A CONVENTIONAL ROBOT SYSTEM

| Aspect | Conventional system | Proposed robot |
|---|---|---|
| Map | Static or pre-programmed | Built in real time by SLAM |
| Sensing | Single or limited sensors | Dual LiDAR, IMU, ultrasonic, camera fusion |
| Path Planning | Fixed path | A* global and DWA local planning |
| Task capability | Navigation only | Navigation with fetch-and-place (150 g) |
| Control architecture | Single controller | Raspberry Pi for processing, ESP32 for motor control. |

### A. Limitations

The prototype is intended for indoor environments, and its payload is limited to 150 g. Its sensing has known weaknesses: LiDAR can miss transparent or highly reflective surfaces such as glass, the ultrasonic sensor has a narrow field of view and lower accuracy, and IMU readings drift over time. The Raspberry Pi has limited computing power, which restricts how dense the mapping and perception can be, and the arm offers limited precision for delicate objects. The performance of the system has also not yet been validated experimentally.

The SLAM robot has limitations such as sensor accuracy, limited battery backup, and computational requirements. Environmental factors like poor lighting, reflective surfaces, and obstacles may affect mapping accuracy. The ESP32-CAM has limited image-processing capabilities. Additionally, uneven surfaces, motor errors, and limited processing power can reduce navigation efficiency and real-time performance.

## V.RESULTS

The figure shows the robot model and its environment simulated in ROS using Gazebo. The robot's navigation path, starting position, and goal location are visualized to demonstrate autonomous navigation.

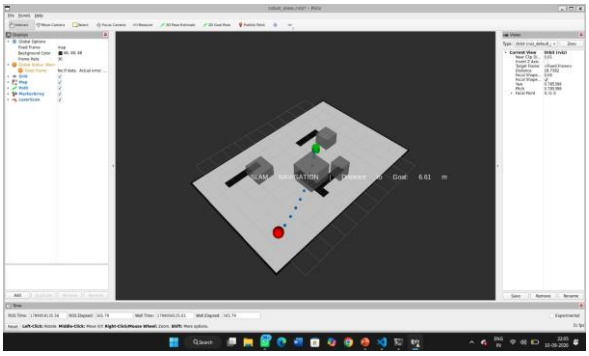


Fig.3 SLAM ROS Prototype

The developed robotic system successfully integrates sensor data from LiDAR, IMU, and ultrasonic sensors for autonomous navigation. The Raspberry Pi performs SLAM-based mapping and localization, while the ESP32 and L298N control the motor movement. The robot can detect obstacles, navigate through the mapped environment, and perform basic object-handling operations using the robotic arm and gripper. The system demonstrates the feasibility of combining mapping, navigation, and object manipulation in a single autonomous mobile robot.

## VI. CONCLUSION

The proposed SLAM-based autonomous mobile robot integrates mapping, localization, obstacle avoidance, navigation, and object-fetching capabilities into a single low-cost robotic platform. The system uses a Raspberry Pi as the main processing unit, integrated with an RPLiDAR A1, an Inertial Measurement Unit (IMU), an ultrasonic sensor, and a Raspberry Pi camera to perceive and understand the surrounding environment. The collected sensor data supports environment mapping, robot localization, and obstacle detection. A* path planning is used to determine an efficient route toward the target, while the Dynamic Window Approach (DWA) supports local navigation and obstacle avoidance.

## VII. FUTURE SCOPE

The prototype can be extended in several directions. Experimental validation in different indoor layouts should report mapping accuracy, navigation success rate, and fetch-and-place reliability. Perception can be improved with AI-based object detection and deeper tightly coupled fusion of LiDAR, IMU, and camera data, as in [9], [12], including semantic mapping for dynamic environments. The arm can be upgraded to handle objects of varying shapes, sizes, and weights



(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

beyond the current 150 g payload, and voice or app-based control can improve usability. Building on frameworks such as [15], the system can be extended to multi-robot collaboration and outdoor navigation, and applied to warehouse logistics, healthcare assistance, and smart homes as a scalable service robot.

## REFERENCES

[1] J. Wang, L. Chen, H. Tanaka, and K. Morris, "Thermal-Visual/Infrared Fusion for Robust SLAM in Degraded Environments," *IEEE Robotics and Automation Letters*, vol. 10, no. 3, pp. 1456–1463, 2025.

[2] V. K. Vigneshwaran, R. R. Das, and D. R. Sona, "Mapping and Navigation of Autonomous Robot using SLAM Algorithm," in *Proc. IEEE Int. Conf. on Advances in Computing and Communications (ICACC)*, pp. 1–6, 2025.

[3] N. Kumar, F. FJ, A. S. A. Doss, and N. G. M. Thao, "Hybrid Perception in Autonomous Mobile Robots: Integrating 2D and 3D SLAM with ROS2," in *Proc. IEEE Int. Conf. on Consumer Technology (ICCT)*, pp. 65–72, 2025.

[4] S. Chen, Y. Liu, W. Zhang, and Q. Zhao, "Transformer-based Loop Closure Detection for 3D LiDAR SLAM," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 1123–1130, 2025.

[5] R. Patel, A. Gupta, T. Rosa, and M. Weber, "Bio-Inspired Neuromorphic Visual SLAM with Event Cameras," *IEEE Transactions on Robotics*, vol. 41, no. 2, pp. 567–578, 2025.

[6] V. Arumugam and V. Alagumalai, "Development and Enhancement of ROS-based SLAM Methods for Navigation in Dynamic Environment," in *Proc. IEEE Green Industrial Electronics and Sustainable Technologies (GIEST)*, pp. 1–8, 2024.

[7] X. Wang, "Mobile Robot Simulation of Path Planning and Map Building," in *Proc. IEEE Int. Conf. on Intelligent Robotics and Control Engineering (IRCE)*, pp. 1–6, 2024.

[8] J. Long and B. Zhuang, "Multi-robots Path Planning and Mapping for Exploring Unknown Environment," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 1–7, 2024.

[9] H. Borthakur and S. Panja, "Integration of Various ROS Packages for Autonomous Navigation and Mapping of Mobile Robot," in *Proc. IEEE Int. Conf. on Robotics, Intelligent Automation and Control Technologies (RIACT)*, pp. 1–6, 2024.

[10] V. Priya, V. Balambika, and M. Achudhan, "Computational Implementation and Optimization of ROS-based SLAM Techniques for Wheeled Mobile Robot Navigation in Dynamic Environments," in *Proc. IEEE Parallel, Distributed and Grid Computing (PDGC)*, pp. 827–831, 2024.

[11] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM3: An Accurate Open Source Library for Visual, Visual-Inertial, and Multimap SLAM," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 1421–1437, 2022.

[12] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: an Opensource Library for Real-Time Metric-Semantic SLAM," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 123–130, 2022.

[13] M. Marchel, K. Nowicki, and J. Wrona, "A Comparative Study of 2D LiDAR SLAM Algorithms on ROS 2," in *Proc. IEEE Int. Conf. on Intelligent Systems (IS)*, pp. 422–428, 2022.

[14] J. Huang, S. Yang, Z. Zhao, Y. Liu, and X. Zhang, "A Survey of Visual SLAM Methods in Dynamic Environments," *IEEE Access*, vol. 9, pp. 12345–12360, 2021.

[15] X. Li, Y. Zhang, Y. Wang, and J. Liu, "DSL-SLAM: Dual Stage Loop Closure Detection for Visual SLAM," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 789–796, 2021.

[16] W. Wang, Z. Zhang, H. Wang, and Z. Chen, "RF-LIO: An Automated LiDAR Inertial Odometry With Robust Feature Matching," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1234–1241, 2021.

[17] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly Coupled Lidar Inertial Odometry via Smoothing and Mapping," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 5135–5142, 2020.

[18] J. Zhang, M. Henein, R. Mahony, and V. Ila, "VDO-SLAM: A Visual Dynamic Object-aware SLAM System," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 1–7, 2020.

[19] Y. Zhao, S. Gao, P. Wang, and S. Li, "GSLAM: A General SLAM Framework and Benchmark," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 1–8, 2020.

[20] J. Czarnowski, T. Laidlow, R. Clark, and A. J. Davison, "Deep Factors: Real-Time Probabilistic Dense Monocular SLAM," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2020



(This work is licensed under a Creative Commons Attribution 4.0 International License.)