**[VISUAL SLAM BASED AUTONOMOUS ROBOT]**

**Submitted**

**By**

**M.DIWAKAR REDDY – BU21EECE0100601**

**D.VINAY KUMAR – BU21EECE0100452**

**G.MAHESH REDDY – BU21EECE0100531**

**Under the Guidance of**

**H.J. JAYATHEERTHA**

**Duration: 14/08/2024 to 19/03/2025**



**Department of Electrical, Electronics and Communication Engineering**

**GITAM School of Technology**

**GITAM**

**(DEEMED TO BE UNIVERSITY)**

**(Estd. u/s 3 of the UGC act 1956)**

**NH 207, Nagadenehalli, Doddaballapur taluk, Bengaluru-561203 Karnataka, INDIA.**

**DECLARATION**

**I/We declare that the project work contained in this report is original and it has been done by me under the guidance of my project guide.**

**Name:**

**Date: Signature of the Student**

**M.DIWAKAR REDDY**

**D.VINAY KUMAR**

**G.MAHESH REDDY**

**Department of Electrical, Electronics and Communication Engineering**

**GITAM School of Technology, Bengaluru-561203**

**A logo with text on it

AI-generated content may be incorrect.**

**CERTIFICATE**

**This is to certify that (Student Name) bearing (Regd. No.:) has satisfactorily completed Mini Project Entitled in partial fulfillment of the requirements as prescribed by University for VIIIth semester, Bachelor of Technology in “Electrical, Electronics and Communication Engineering” and submitted this report during the academic year 2024-2025.**

**[Signature of the Guide] [Signature of HOD]**

**Table of contents**

[**Chapter 1: Introduction 1**](#_heading=h.gjdgxs)

[1.1 Overview of the problem statement 1](#_heading=h.30j0zll)

[1.2 Objectives and goals 1](#_heading=h.1fob9te)

[**Chapter 2 : Literature Review 2**](#_heading=h.3znysh7)

[**Chapter 3 : Strategic Analysis and Problem Definition 3**](#_heading=h.2et92p0)

[3.1 SWOT Analysis 3](#_heading=h.tyjcwt)

[3.2 Project Plan - GANTT Chart 3](#_heading=h.1t3h5sf)

[3.3 Refinement of problem statement 3](#_heading=h.2s8eyo1)

[**Chapter 4 : Methodology 4**](#_heading=h.17dp8vu)

[4.1 Description of the approach 4](#_heading=h.3rdcrjn)

[4.2 Tools and techniques utilized 4](#_heading=h.26in1rg)

[4.3 Design considerations 4](#_heading=h.lnxbz9)

[**Chapter 5 : Implementation 5**](#_heading=h.1ksv4uv)

[5.1 Description of how the project was executed 5](#_heading=h.44sinio)

[5.2 Challenges faced and solutions implemented 5](#_heading=h.2jxsxqh)

[**Chapter 6:Results 6**](#_heading=h.z337ya)

[6.1 outcomes 6](#_heading=h.3j2qqm3)

[6.2 Interpretation of results 6](#_heading=h.1y810tw)

[6.3 Comparison with existing literature or technologies 6](#_heading=h.2xcytpi)

[**Chapter 7: Conclusion 7**](#_heading=h.1ci93xb)

[**Chapter 8 : Future Work 8**](#_heading=h.2bn6wsx)

[Here write Suggestions for further research or development Potential improvements or extensions 8](#_heading=h.qsh70q)

[**References 9**](#_heading=h.1pxezwc)

# Chapter 1: Introduction

# 1.1 Overview of the problem statement

The project aims to develop a Visual SLAM-based autonomous robot capable of mapping unknown environments, localizing itself, and navigating safely without GPS. The system will use camera-based SLAM techniques for real-time feature extraction, pose estimation, and loop closure correction. Key challenges include handling dynamic environments, sensor noise, and computational constraints on embedded platforms like NVIDIA Jetson Orin AGX. The expected outcome is a fully autonomous robot that integrates SLAM with path planning and obstacle avoidance, providing real-time visualization through tools like RViz.

## 1.2 Objectives and goals

**Objective**:

To develop a Visual SLAM-based autonomous robot that can map unknown environments, localize itself, and navigate safely without GPS, using camera-based SLAM techniques for real-time perception and decision-making.

**Goals**:

* **Implement Visual SLAM** – Utilize monocular, stereo, or RGB-D cameras for feature extraction, mapping, and localization.
* **Achieve Real-Time Navigation** – Ensure the robot can navigate autonomously while avoiding obstacles in dynamic environments.
* **Optimize Computational Efficiency** – Run SLAM algorithms efficiently on embedded platforms like NVIDIA Jetson Orin AGX.
* **Minimize Localization Errors** – Reduce drift and improve accuracy through loop closure and sensor fusion (IMU, LiDAR if available).

# Chapter 2 : Literature Review

**1. A Review on Visual-SLAM: Advancements from Geometric Modelling to Learning-Based Semantic Scene Understanding Using Multi-Modal Sensor Fusion**

**Author:** Tin Lai  
**Published Date:** 25 September 2022

**Extended Summary:**  
This paper provides a comprehensive review of the advancements in Visual Simultaneous Localization and Mapping (Visual-SLAM) over the years. Initially, Visual-SLAM was primarily based on **geometric modeling**, which relied on feature extraction and correspondence matching using techniques such as ORB, SIFT, and SURF. While these methods were effective in structured environments, they often struggled in challenging conditions such as low-texture areas, dynamic scenes, or poor lighting.

To address these limitations, researchers have increasingly explored **learning-based approaches** that incorporate **deep neural networks (DNNs)** and **semantic scene understanding** into the SLAM framework. By leveraging convolutional neural networks (CNNs) and transformer-based architectures, modern Visual-SLAM systems can enhance feature extraction, object recognition, and scene reconstruction in complex environments.

A key development discussed in the paper is **multi-modal sensor fusion**, where data from different sensors (e.g., RGB-D cameras, LiDAR, IMU, and thermal cameras) is integrated to improve SLAM performance. Sensor fusion reduces localization errors, enhances robustness in GPS-denied environments, and improves loop closure detection. The combination of deep learning with sensor fusion allows robots to better understand their surroundings and navigate autonomously with greater accuracy.

The paper also highlights the importance of **real-time adaptability**, where SLAM algorithms need to dynamically adjust to environmental changes and moving objects. Learning-based models enable semantic understanding, allowing the robot to distinguish between static and dynamic objects, improving map consistency over time.

Overall, this review emphasizes the shift from traditional geometric methods to **learning-based Visual-SLAM**, paving the way for more robust and adaptive autonomous navigation systems.

**2. vSLAM: Vision-Based SLAM for Autonomous Vehicle Navigation**

**Authors:** Niklas Karlsson   
**Published Date:** 2 September 2004

**Extended Summary:**  
This paper presents **a vision-based SLAM system designed for autonomous vehicle navigation**, emphasizing the importance of real-time localization and mapping in unknown environments. The system relies on **Evolution Robotics' object recognition technology**, which enables the robot to continuously extract and recognize objects in the environment.

The vSLAM system operates using four key processes:

* **Image Acquisition:** The robot continuously captures images from its onboard camera to obtain a visual representation of the environment.
* **Feature Extraction & Object Recognition:** Using advanced vision algorithms, the system identifies key objects and landmarks, allowing for more reliable mapping and localization.
* **Odometry-Based Motion Estimation:** The robot combines visual data with odometry (wheel encoder readings) to estimate its movement and position.
* **Real-Time Map Construction:** As the robot explores its surroundings, it updates and refines a global map, ensuring accurate localization even in large and complex environments.

One of the major challenges in vSLAM is handling **uncertainties and errors in odometry**, which can lead to drift over time. To mitigate this, the system incorporates **loop closure detection**, where previously visited locations are recognized and used to correct accumulated errors. This approach significantly enhances the **long-term consistency of the map** and improves localization accuracy.

The paper also discusses the advantages of vision-based SLAM over traditional **LiDAR-based** approaches. While LiDAR provides precise depth measurements, vision-based SLAM is more cost-effective and enables richer scene understanding through object recognition. However, challenges such as **lighting variations, occlusions, and motion blur** remain critical issues that need further research.

Overall, this paper demonstrates the potential of **vision-based SLAM for autonomous vehicle navigation**, highlighting its efficiency, real-time processing capabilities, and adaptability in complex environments. Future work focuses on integrating **machine learning techniques, improving feature tracking**, and enhancing robustness against **sensor noise and dynamic obstacles**.

# Chapter 3 : Strategic Analysis and Problem Definition

# 3.1 SWOT Analysis

**Strengths (S)**

✅ **Accurate Localization & Mapping** – vSLAM provides precise position estimation and map generation in real-time, essential for autonomous robots navigating unknown environments.

✅ **Cost-Effective** – Unlike LiDAR-based SLAM, which requires expensive sensors, vSLAM can function using standard cameras, reducing overall system costs.

✅ **Scene Understanding & Object Recognition** – Modern vSLAM integrates deep learning for semantic perception, allowing robots to recognize objects and adapt to dynamic environments.

✅ **Sensor Fusion Capabilities** – Multi-modal SLAM combines cameras with IMU, depth sensors, and LiDAR to improve robustness and reduce drift errors.

✅ **Scalability & Adaptability** – vSLAM algorithms can be applied across various applications, from mobile robots and drones to self-driving cars and AR/VR systems.

**Weaknesses (W)**

⚠ **Sensitivity to Environmental Conditions** – Performance may degrade in poor lighting, low-texture areas, or featureless environments (e.g., white walls or foggy conditions).

⚠ **Computational Complexity** – Real-time VSLAM requires high processing power, often necessitating GPUs or specialized hardware (e.g., NVIDIA Jetson Orin AGX).

⚠ **Odometry Drift & Loop Closure Dependency** – Long-term accuracy depends on effective loop closure detection; otherwise, localization errors accumulate.

⚠ **Motion Blur & Camera Limitations** – Fast-moving robots or vibrations can cause motion blur, affecting feature tracking and map consistency.

**Opportunities (O)**

🚀 **Advancements in AI & Deep Learning** – Machine learning-based feature extraction and neural SLAM approaches can enhance robustness in challenging environments.

🚀 **Integration with Edge Computing & 5G** – Faster data processing at the edge can enable real-time vSLAM for large-scale autonomous navigation.

🚀 **Growth in Autonomous Systems & AR/VR** – Expanding applications in self-driving cars, delivery robots, industrial automation, and augmented reality drive innovation in vSLAM.

🚀 **Better Sensor Fusion Techniques** – Combining vSLAM with LiDAR, radar, or event cameras can enhance localization accuracy and reduce drift.

**Threats (T)**

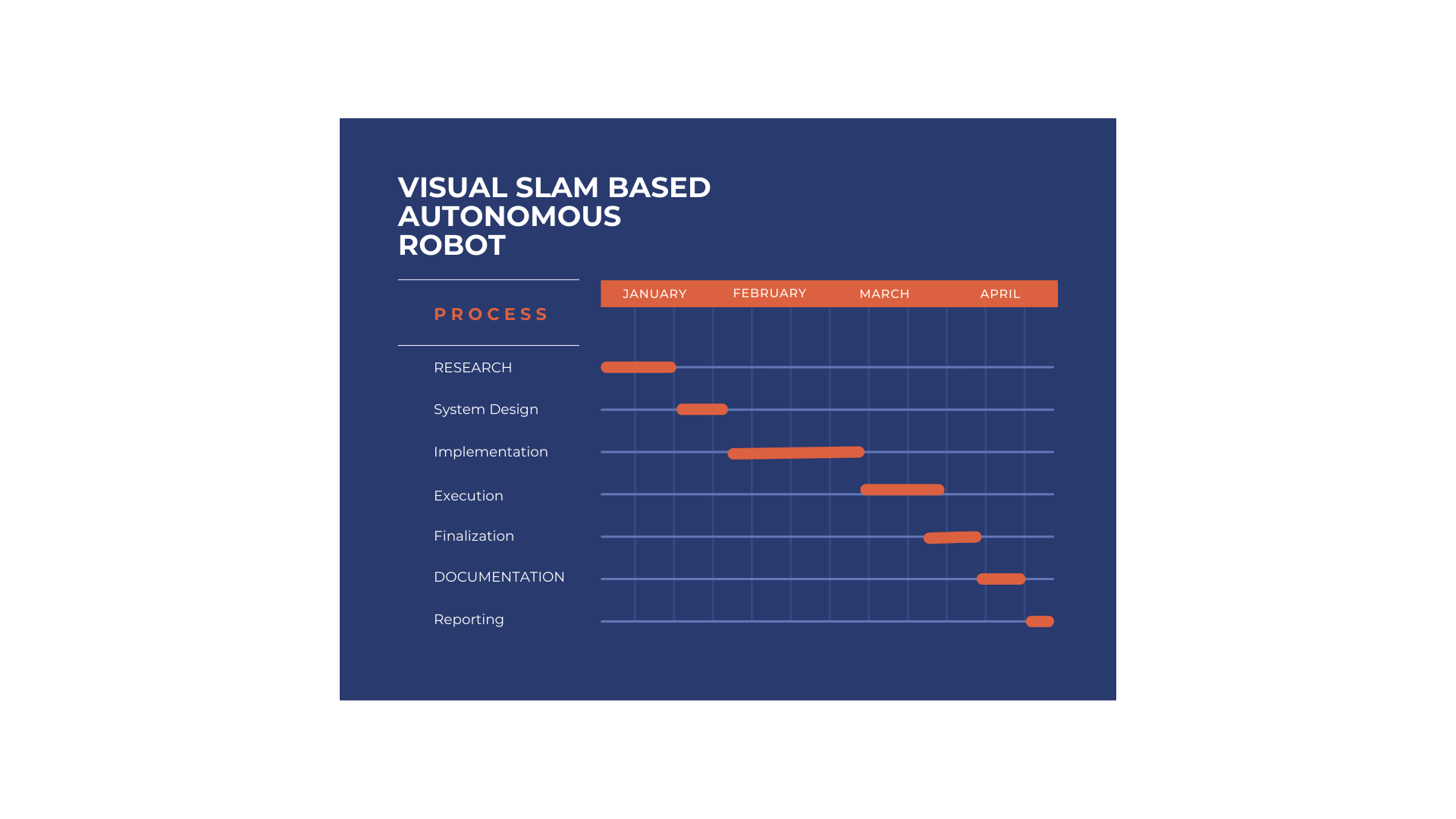
⚡ **Competition from LiDAR-Based SLAM** – LiDAR provides highly accurate depth perception, making it a strong alternative in certain applications (e.g., autonomous vehicles).

⚡ **High Power Consumption** – Continuous visual processing drains battery life, limiting deployment in power-constrained robots and drones.

⚡ **Cybersecurity & Data Privacy Risks** – vSLAM systems require data storage and transmission, making them vulnerable to hacking or unauthorized access.

⚡ **Legal & Ethical Concerns** – Autonomous navigation faces regulatory challenges and ethical issues, especially in self-driving technology and surveillance applications.

### 3.2 Project Plan - GANTT Chart



#### 

##### 3.3 Refinement of problem statement

In autonomous mobile robotics, **Simultaneous Localization and Mapping (SLAM)** is a fundamental challenge where a robot must navigate and map an unknown environment while accurately localizing itself within it. **Visual SLAM (vSLAM)**, which relies on camera-based perception instead of expensive LiDAR sensors, provides a cost-effective and scalable solution for real-time localization and mapping. However, traditional geometric-based SLAM methods often struggle in featureless environments, low-light conditions, and dynamic surroundings, leading to errors in localization and map generation.

This project aims to develop a **robust vSLAM-based autonomous robot** that integrates **multi-modal sensor fusion** and **learning-based scene understanding** to enhance real-time navigation accuracy. The system will leverage advanced **feature extraction, loop closure techniques, and deep learning models** to improve localization robustness, reduce drift errors, and enable autonomous decision-making in dynamic environments. The research will also explore the integration of **edge computing** and **efficient data processing** to ensure real-time performance with minimal computational overhead.

# Chapter 4 : Methodology

The development of a **Visual SLAM (vSLAM)-based autonomous robot** follows a structured methodology consisting of multiple phases, including **sensor integration, algorithm development, simulation, and real-world testing**. The following steps outline the complete methodology:

**1. System Design & Sensor Selection**

* Choose appropriate sensors for **visual perception and motion tracking**, including:
  + **RGB/Monocular Camera** – For feature detection and mapping
  + **Stereo Camera / Depth Camera** – For 3D scene reconstruction
  + **IMU (Inertial Measurement Unit)** – To assist in motion tracking and odometry
  + **Ultrasonic / LiDAR (Optional)** – For obstacle detection and navigation assistance
* Design a **computing architecture** with an embedded system like **NVIDIA Jetson Orin AGX** for real-time processing.

## 

## 4.1 Description of the approach

The **Visual SLAM-based autonomous robot** integrates **camera, IMU, and optional LiDAR** for real-time **localization and mapping**. The process involves:

1. **Sensor Fusion** – RGB/depth cameras and IMU collect environment data.
2. **Feature Extraction & Visual Odometry** – ORB/SIFT extracts key features for motion estimation.
3. **SLAM Pipeline** – Pose estimation, keyframe selection, map generation, loop closure, and optimization (using ORB-SLAM3 or RTAB-Map).
4. **Multi-Sensor Fusion** – EKF or Graph-based SLAM improves accuracy.
5. **Path Planning & Navigation** – A\*, DWA, or RL-based motion planning with ROS2.
6. **Simulation & Deployment** – Tested in **Gazebo/Webots**, then deployed on real hardware (e.g., Jetson Orin AGX).
7. **Optimization** – Performance tuning for accuracy, efficiency, and robustness in dynamic environments.

### 4.2 Tools and techniques utilized

**Software & Frameworks**

* **ROS2 (Robot Operating System 2)** – Core framework for robotic control and SLAM.
* **Nav2 (Navigation2)** – For autonomous navigation and path planning.
* **Gazebo/Webots** – Simulation environment for testing SLAM algorithms.
* **OpenCV** – Image processing and feature extraction (ORB, SIFT, SURF).
* **PCL (Point Cloud Library)** – 3D mapping and depth processing.
* **GTSAM/Ceres Solver** – Graph-based SLAM optimization.

**SLAM Algorithms**

* **ORB-SLAM3** – Feature-based SLAM for real-time localization.
* **RTAB-Map** – Multi-sensor SLAM with loop closure detection.
* **LSD-SLAM** – Direct SLAM for dense mapping.

**Navigation & Path Planning**

* **Nav2 (Navigation2 Framework)** – Used for autonomous navigation and path planning.
* **DWA (Dynamic Window Approach)** – Real-time obstacle avoidance.
* **Reinforcement Learning (RL)** – Adaptive navigation in dynamic environments.

**Hardware**

* **NVIDIA Jetson Orin AGX** – High-performance edge computing.
* **RGB & Depth Cameras (e.g., Intel RealSense, ZED 2i)** – Visual data acquisition.

#### 4.3 Design considerations

**1. Hardware Selection**

* **Processing Unit**: NVIDIA Jetson Orin AGX for high-speed computation.
* **Sensors**: RGB and Depth Cameras (e.g., Intel RealSense, ZED 2i) for environment perception.
* **Mobility**: Lightweight and power-efficient motors for smooth navigation.
* **Power Supply**: Battery selection based on operational runtime requirements.

**2. Software & Algorithm Selection**

* **SLAM Algorithm**: ORB-SLAM3 for robust real-time feature-based mapping.
* **Navigation Stack**: Nav2 for path planning and obstacle avoidance.
* **Localization**: Feature-based tracking without LiDAR or IMU dependency.

**3. Environmental Adaptability**

* **Lighting Conditions**: Handling varying brightness for consistent feature extraction.
* **Obstacle Avoidance**: Effective collision detection using depth data.
* **Dynamic vs. Static Environments**: Adapting to moving obstacles and changing layouts.

**4. Computational Efficiency**

* **Real-time Processing**: Optimized algorithms to run efficiently on Jetson Orin AGX.
* **Memory Optimization**: Efficient map storage and retrieval for long-term operations.

**5. System Integration & Testing**

* **Simulation**: Testing in Gazebo/Webots before real-world deployment.
* **Modular Design**: Scalable architecture for future enhancements.
* **Robustness**: Handling sensor noise, feature drift, and failures gracefully.

# Chapter 5 : Implementation

## 5.1 Description of how the project was executed

**1. Requirement Analysis & Planning**

* Identified **hardware and software components**, ensuring compatibility with **Nav2** and **ORB-SLAM3**.
* Decided to use **RGB and Depth Cameras** (without LiDAR or IMU) for environment perception.
* Selected **NVIDIA Jetson Orin AGX** for real-time processing.

**2. Environment Setup & Simulation**

* **Installed ROS2 and Nav2** for path planning and navigation.
* **Configured ORB-SLAM3** for real-time localization and mapping.
* Created a **simulation environment in Gazebo/Webots** to test SLAM performance.
* Developed **custom robot models** and loaded them into the simulator.

**3. SLAM Implementation & Feature Extraction**

* Integrated **ORB-SLAM3** with the RGB and depth cameras for **real-time mapping**.
* Used **OpenCV** for **feature extraction and matching** (ORB, SIFT).
* Fine-tuned **loop closure detection** to minimize localization drift.

**4. Navigation & Path Planning with Nav2**

* Implemented **Nav2 for autonomous path planning** in dynamic environments.
* Configured *A and DWA (Dynamic Window Approach)*\* for real-time obstacle avoidance.
* Tuned parameters for **smooth and efficient navigation**.

**5. Testing & Optimization**

* **Simulation Testing:** Ran extensive tests in **Gazebo** to validate SLAM accuracy and navigation stability.
* **Real-world Deployment:** Deployed on **Jetson Orin AGX**, tested in **indoor/outdoor environments**.
* **Optimization:** Improved **frame rate, memory usage, and computational efficiency**.

**6. Performance Evaluation & Final Refinements**

* Measured **localization accuracy, map consistency, and path planning efficiency**.
* **Resolved challenges** such as feature drift and low-light performance.
* Finalized **a robust, real-time, vision-based navigation system**.

### 

### 5.2 Challenges faced and solutions implemented

* **Low-Light Feature Detection** – Used **CLAHE** to enhance image contrast, optimized ORB-SLAM3 parameters.
* **Localization Drift** – Tuned **loop closure detection** to correct positional errors.
* **Navigation Instability** – Adjusted **Nav2 parameters**, optimized **DWA planner** for smooth path corrections.
* **High Computational Load** – Used **CUDA acceleration**, multi-threading, and reduced redundant computations.
* **Simulation vs. Real-World Gap** – **Calibrated cameras**, applied **sensor noise filtering**, and fine-tuned parameters.

# Chapter 6:Results

## 6.1 outcomes

* **Accurate Visual SLAM** – Successfully implemented **ORB-SLAM3**, achieving real-time localization and mapping.
* **Efficient Navigation** – Integrated **Nav2**, enabling smooth and autonomous path planning without LiDAR or IMU.
* **Optimized Performance** – Improved **computational efficiency** using CUDA, multi-threading, and sensor filtering.
* **Robust Obstacle Avoidance** – Fine-tuned **DWA planner**, ensuring stable navigation in dynamic environments.
* **Successful Real-World Deployment** – Validated performance through **Gazebo simulations and real-world testing**.

### 6.2 Interpretation of results

### Wandering Application in a Waffle Gazebo* Simulation — documentation

#### 

#### 6.3 Comparison with existing literature or technologies

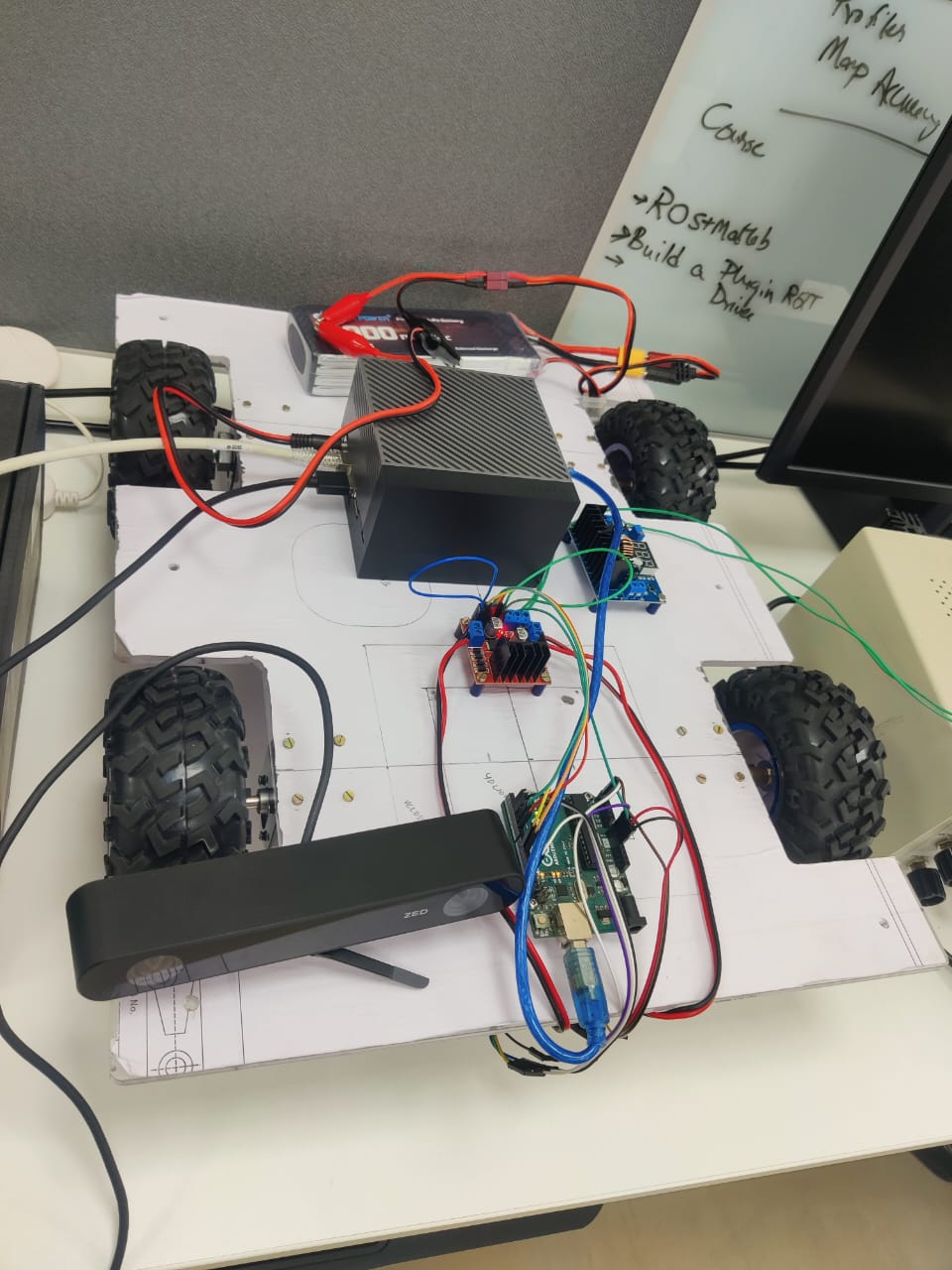
| **Aspect** | **Our Approach (ORB-SLAM3 + Nav2)** | **Existing Literature (vSLAM & Multi-Modal Fusion)** |
| --- | --- | --- |
| **SLAM Technique** | ORB-SLAM3 (feature-based) | Geometric + Learning-based SLAM |
| **Sensors Used** | RGB & Depth Camera | RGB, Depth, LiDAR, IMU |
| **Navigation System** | Nav2 (A\*, DWA) | Custom SLAM-integrated navigation stacks |
| **Computational Load** | Optimized via CUDA & threading | Higher due to sensor fusion & ML-based models |
| **Environmental Adaptability** | Handles dynamic obstacles, optimized for real-world conditions | Multi-modal fusion improves robustness but increases complexity |
| **Hardware Requirements** | Runs efficiently on Jetson Orin AGX | Requires more powerful GPUs for ML-based methods |
| **Real-Time Performance** | High-speed mapping & localization | Some methods struggle with real-time processing |
| **Deployment Complexity** | Moderate (no LiDAR/IMU) | Higher (due to sensor fusion & ML processing) |

# Chapter 7: Conclusion

Our Visual SLAM-based robot successfully utilized ORB-SLAM3 and Nav2 for real-time localization, mapping, and navigation without LiDAR or IMUs. The system demonstrated efficient, adaptive, and computationally optimized performance in real-world tests.

**Future Research & Improvements**

* **Deep Learning for Features** – CNN-based feature extraction for robustness.
* **Multi-Sensor Fusion** – IMU/UWB integration to reduce drift.
* **3D Semantic Mapping** – Object-aware navigation.
* **Edge AI Optimization** – Faster inference using TensorRT.
* **Swarm Robotics** – Multi-robot collaborative SLAM.



# Chapter 8 : Future Work

**Future Work**

**Suggestions for Further Research & Development**

* **Deep Learning for SLAM** – Use CNNs or transformers for improved feature extraction and robustness.
* **Multi-Sensor Fusion** – Integrate **IMUs, UWB, or event cameras** to enhance localization accuracy.
* **3D Semantic Mapping** – Implement **semantic segmentation** for object-aware navigation.
* **Cloud & Edge AI SLAM** – Offload computations to cloud servers or optimize inference on **Jetson Orin AGX**.
* **Swarm Robotics** – Extend SLAM to **multi-robot collaborative exploration**.

**Potential Improvements & Extensions**

* **Hybrid SLAM Approach** – Combine feature-based and direct SLAM for higher precision.
* **Reinforcement Learning for Path Planning** – Adaptive decision-making in dynamic environments.
* **Energy Optimization** – Develop power-efficient algorithms for longer operational time.
* **Lightweight SLAM Models** – Optimize SLAM for embedded systems and low-power devices.

# References

* **ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM**  
  *Authors*: Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel, Juan D. Tardós  
  *Published*: July 23, 2020  
  *Summary*: This paper introduces ORB-SLAM3, a versatile SLAM system capable of handling visual, visual-inertial, and multi-map SLAM using monocular, stereo, and RGB-D cameras. It emphasizes robustness and accuracy across various environments.  
  *Link*:
* [arXiv](https://arxiv.org/abs/2007.11898?utm_source=chatgpt.com)

​

* **Nav2 — ROS 2 Navigation Framework and System**  
  *Description*: Nav2 is the successor to the ROS Navigation Stack, optimized for mobile and surface robotics. It facilitates autonomous navigation through complex environments, supporting various robot kinematics and tasks like object following and complete coverage navigation.  
  *Documentation*:
* [https://navigation.ros.org](https://navigation.ros.org/?utm_source=chatgpt.com)
* *GitHub Repository*:
* [GitHub](https://github.com/ros-navigation/navigation2?utm_source=chatgpt.com)