ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DEPARTMENT OF ELECTRICAL, ELECTRONIC AND INFORMATION ENGINEERING

"GUGLIELMO MARCONI" (DEI)

SECOND CYCLE DEGREE IN AUTOMATION ENGINEERING

DYNAMIC GENERATION OF A GLOBAL ENVIRONMENTAL MODEL FOR CONNECTED VEHICLES

Master Thesis in Autonomous and Mobile Robotics

Supervisor

Prof. Dr. Gianluca Palli

Defended by

Elshan Moradkhani

Co-Supervisor

Falk Dettinger, M.Sc.
(University of Stuttgart)

Graduation Session March 2025

Academic Year 2023/2024

University of Stuttgart
Institute of Industrial Automation and
Software Engineering

Dynamic Generation of a Global Environmental Model for Connected Vehicles

Submitted at the University of Stuttgart by
Elshan Moradkhani

M.Sc. in Automation Engineering at University of Bologna
Erasmus Exchange Student at University of Stuttgart

Supervisor: Prof. Gianluca Palli (University of Bologna)

Co-Supervisor: Falk Dettinger M.Sc. (University of Stuttgart)

0 Table of Contents

Table of Figures

Table of Tables

Table of Abbreviations

AV / AD	Autonomous Vehicles / Autonomous Driving
BEV	Bird's-Eye View
BRIEF	Binary Robust Independent Elementary Features
CAM	Co-operative Awareness Message
CARLA	CAR Learning to Act
CAV	Connected and Autonomous Vehicles
CNN	Convolutional Neural Network
CPM	Co-operative Perception Message
DMM	Direct Map Merging
DRL	Deep Reinforcement Learning
DSRC	Dedicated Short-Range Communication
DBF	Digital Beamforming
EKF	Extended Kalman Filter
ETSI	European Telecommunications Standard Institute
FAST	Features from Accumulated Segment Test
FoV	Field of View
FPFH	Fast Point Feature Histograms
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPU	Graphical Processing Unit
HD	High-Definition
ICP	Iterative Closest Point
IMU	Inertial Measurement Unit
IoU	Intersection over Union
IPM	Inverse Perspective Mapping
LDM	Local Dynamic Map
LiDAR	Light Detection and Ranging
LOAM	LIDAR Odometry and Mapping
LeGO-LOAM	Lightweight and Ground-Optimized LIDAR Odometry and Mapping
LRR	Long-Range Radar
MIMO	Multi-Input-Multi-Output
MRR	

MODT	Mid-Range Radar
MTM	Moving Object Detection and Tracking
NDT	Map Transformation Matrix
NLOS	Normal Distributions Transform
NRConv	Non-Line-of-Sight
OBU	Noise Resistant sub-manifold Convolution
ORB	On-Board Unit
PCD	Oriented FAST and Rotated BRIEF
PFM	Point Cloud Data
PSM	Probabilistic Feature Matching
PRF	Polar Scan Matching
RANSAC	Pulse Repetition Frequency
RL	Random Sample Consensus
RSU	Reinforcement Learning
RMSE	Roadside Unit
RTK-GPS	Root Mean Square Error
SAE	Real-time Kinematic GPS
SIFT	Society of Automotive Engineers
SLAM	Scale-Invariant Feature Transform
SMCN	Simultaneous Localization and Mapping
SRR	Situation Metrics Computing Network
StVD	Short-Range Radar
SUMO	Stochastic Voxel Discard
TMC	Simulation of Urban Mobility
UDP	Traffic Management Centers
V2I	User Datagram Protocol
V2P	Vehicle-to-Infrastructure
V2V	Vehicle-to-Pedestrian
V2X	Vehicle-to-Vehicle
VoI	Vehicle-to-Everything
V-SLAM	Value of Information
VSLs	Visual Simultaneous Localization and Mapping
	Virtual Supporting Lines

Abstract

This study explores the enhancement of autonomy in connected vehicles through improved cooperative mapping techniques. By dividing large areas into smaller sections for independent exploration, vehicles can generate precise local maps using integrated sensors such as LIDAR, cameras, and GNSS. The focus is on urban and outdoor environments, utilizing Vehicle-to-Vehicle (V2V) communication rather than Vehicle-to-Everything (V2X) or Vehicle-to-Infrastructure (V2I). Two main map merging methods are evaluated: Direct Map Merging (DMM) and Indirect Map Merging (IMM). DMM offers real-time capabilities and simplicity, while IMM provides improved accuracy and flexibility but is computationally intensive. The study employs the CARLA simulator to validate the proposed cooperative 3D mapping system, utilizing vehicles equipped with advanced sensors. Performance metrics such as fitness score, RMSE, and C2C distance are used to evaluate the effectiveness of the system in different scenarios.

1 Basis:

In today's internet-connected world, vehicles are revolutionizing transportation by communicating with each other and infrastructures in real time. This enhances efficiency, prevents accidents, and saves lives. Connected vehicle technologies enable higher autonomy levels, crucial for both autonomous driving and Advanced Driver Assistance Systems (ADAS). They also reduce human errors, major causes of accidents, and traffic congestion. The Connected Autonomous Vehicle (CAV) promotes safety by raising awareness among vehicles. Emerging vehicle connectivity supports various communications like V2V and V2I, improving transportation efficiency. Autonomous driving systems utilize sensors for environment perception, ensuring safer travel through accurate mapping and localization.

Motivation and Objectives:

To make connected vehicles more autonomous, it is needed to carefully record and map their surroundings. Since sensor signals can be affected by the environment, we use sensor fusion to improve detection reliability. However, sensors have limits in detecting the environment fully. So, communication protocols help to share information among vehicles, transmitting only clear objects due to network limits. This creates uncertainty in the environment model, which can be solved by sending relevant sensor data. This data is turned into a model showing the surroundings. Typically, SLAM algorithms are used for this. They use local sensor data to figure out the vehicle's position and the position of objects around it. There are various mapping methods, like occupancy grid maps or semantic maps. We explore ways to create a detailed global map for networked vehicles, considering different map formats with different details and information types.

Structure and Main Tasks:

In the near future, autonomous vehicles are expected to take over various transportation roles, replacing traditional vehicles. Expected trends indicate the availability of both Autonomous Vehicles (AVs) and connected vehicles (CVs), possibly together, because it is expected to be inexpensive. While AVs and CVs can be equipped with a variety of sensors, a critical functionality involves their ability to accurately position and measure distances between vehicles and objects and map their environment in detailed. The reliability and accuracy of connected vehicle systems, as well as road safety, rely on these sensors and their measurements.

The primary objectives of autonomous driving involve addressing key questions: Where am I? Where are the others? How do I reach my destination? These inquiries find solutions through key elements such as Localization and Mapping, Sensing, Planning, and Driver State Monitoring respectively. The Society of Automotive Engineers (SAE) classifies vehicular autonomy into six levels (0 to 5), depending on the level of human driver involvement during operation. Researchers are aiming for level 5 where cars can operate without any human input.[1]

Table 1: Description of Autonomy Level [2]

Level 0 Driver Only	Level 1 Assisted	Level 2 Partially Automated	Level 3 Highly Automated	Level 4 Fully Automated	Level 5 Self - Driving
	Feet-off	Hands-off	Eyes-off	Eyes-off	Mind-off
Perform longitudinal and lateral tasks continuously	Perform longitudinal or lateral tasks continuously	Driver monitors systems continuously	Driver does not monitor systems continuously	Driver is required in defined use cases	Driver is not required during the entire journey
Driver Monitors the Environments			Machine Monitors the environment		

Connected and Autonomous Vehicles (CAVs) involve three main tasks. Perception, Planning and Control. The autonomy system of driverless vehicles relies on integration of perception and planning, each consisting of distinct subsystems. In the domain of autonomous vehicular operations, the comprehensive control of vehicle motion encompasses both longitudinal and lateral directions. This necessitates the execution of a control task that involves actively managing both the powertrain and steering system to ensure optimal and effective motion. The ultimate goal in the realm of Advanced Driver Assistance Systems (ADAS) and Autonomous Driving is to emulate human driver's capability to sense, reason and act to achieve beyond human driving performance.[1]

1.1.1 Perception

The perception layer gathers data from various sensors, undertaking tasks such as establishing the vehicle's global and local position, creating an environmental map, detecting, classifying, and tracking obstacles or participants. Connected and Autonomous Vehicles need both onboard and communication sensors to work well. These sensors also need to interact smoothly with the environment, like roads and other vehicles, for CAVs to operate effectively. The connectivity facilitated by vehicle communication enables the perception layer to exchange critical information with other road users, promoting a collaborative driving. This Perception Layer is typically segmented into three essential components: Sensor Fusion, Environmental Perception, and Localization and Mapping. The aim of these components is to ensure robust and reliable perception, as well as precise localization and mapping. These elements are vital in making accurate and dependable decisions for vehicle control. Essentially, a reliable Perception Layer is crucial for autonomous vehicles to operate well, influencing their ability to navigate and interact safely and intelligently with the various dynamic driving scenarios. [1]

1.1.2 Planning:

When the perception layer provides data about the surroundings and the vehicle's position, the planning layer translates that information into practical driving actions. This highlights the crucial role of planning in ensuring smooth and efficient driving in autonomous systems. The planning layer determines the optimal global route by utilizing remote map data that includes road and traffic information, covering maneuver, path, and trajectory planning. Through real-time decision-making, it calculates a locally optimal trajectory, influencing servo control for precise vehicle actuation. This comprehensive planning process, initiated after receiving the environmental model and ego-vehicle position from the perception layer, allows the planning layer to transform information into the desired driving action.[1]

1.1.3 Control:

Within the control layer, precise commands are calculated to guide vehicle actuators, ensuring the reliable execution of the intended trajectory. This crucial layer effectively guides vehicle movements with precise servo control and adept trajectory tracking. The motion control block manages both longitudinal and lateral motions, smoothly integrating with powertrain and steering controls. This integrated approach ensures the vehicle responds optimally, achieving precise trajectory tracking in diverse driving scenarios.[1]

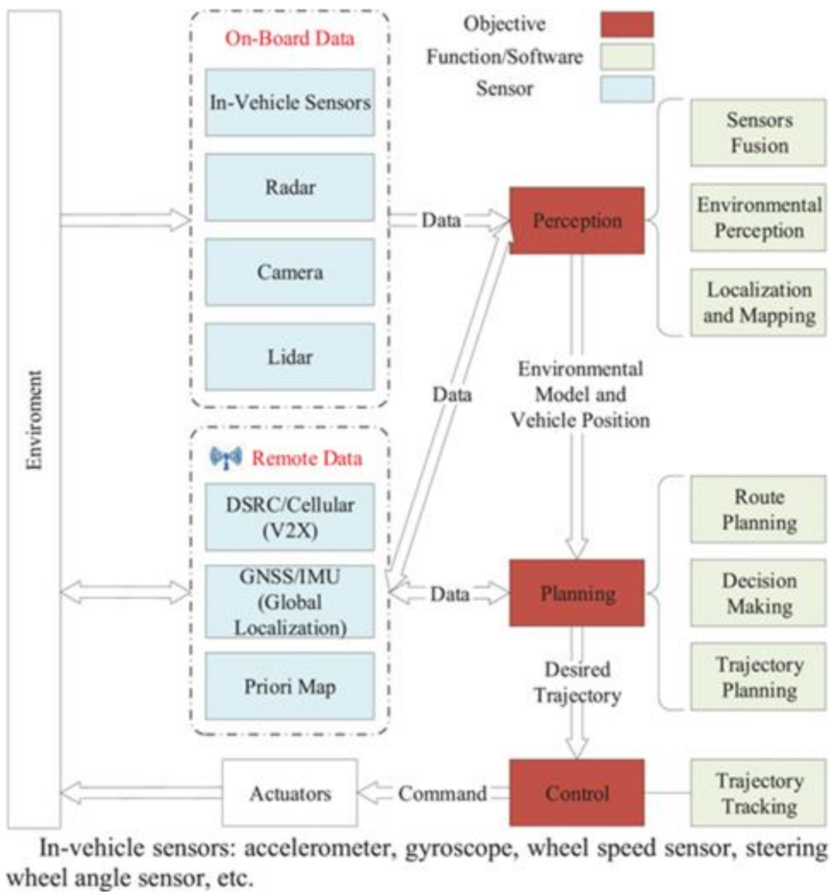


Figure 1: Connected Autonomous Vehicles Architecture **[1]**

1.1.4 What is not covered in this Thesis

Within the scope of this project, the primary emphasis is on Perception and map generation, deliberately excluding discussions related to planning and control blocks.

1.2 Sensing Technologies:

Connected Autonomous Vehicles (CAVs) heavily rely on sensors for data collection and communication with electronic devices. Perception, in alongside sensors, plays a pivotal role by creating a model of the environment, detecting obstacles, recognizing traffic signals, identifying road markings, and achieving accurate localization.

Sensors are categorized as onboard sensors for direct data collection and communication sensors, dependent on communication quality. After extracting environmental data, processes for localization and mapping are conducted to fulfil the perception task. Furthermore, the effectiveness of sensors significantly impacts CAVs' safety and efficiency by providing a comprehensive understanding of the surrounding environment. This allows the vehicle to make informed decisions and navigate diverse scenarios with precision.

In the domain of Autonomous Driving, sensors play a crucial role in determining a vehicle's position and orientation. GNSS, LiDAR, Radar, IMU, and cameras collect data to comprehend the vehicle's surroundings. Sensors are categorized as either exteroceptive (sensing the environment) or proprioceptive (measuring internal conditions). They are also classified based on energy usage, with passive sensors (e.g., cameras, GNSS, and inertial sensors) not emitting energy, and active sensors (e.g., LiDAR and Radar) emitting energy to perceive the environment. This diverse sensor array is vital for accurate and reliable localization, facilitating higher-level decision-making for vehicles.

In autonomous vehicles, sensors such as Radar, LiDAR, and Cameras contribute to surrounding sensing by perceiving road conditions, traffic signs, weather, obstacles, and the driver's state. These sensors can either be active, emitting signals, or passive, perceiving existing signals. Selecting the appropriate sensor involves considering factors like the required information type, the suitability of an active or passive sensor, budget constraints, and the decision to use a single or multiple sensors. [3]

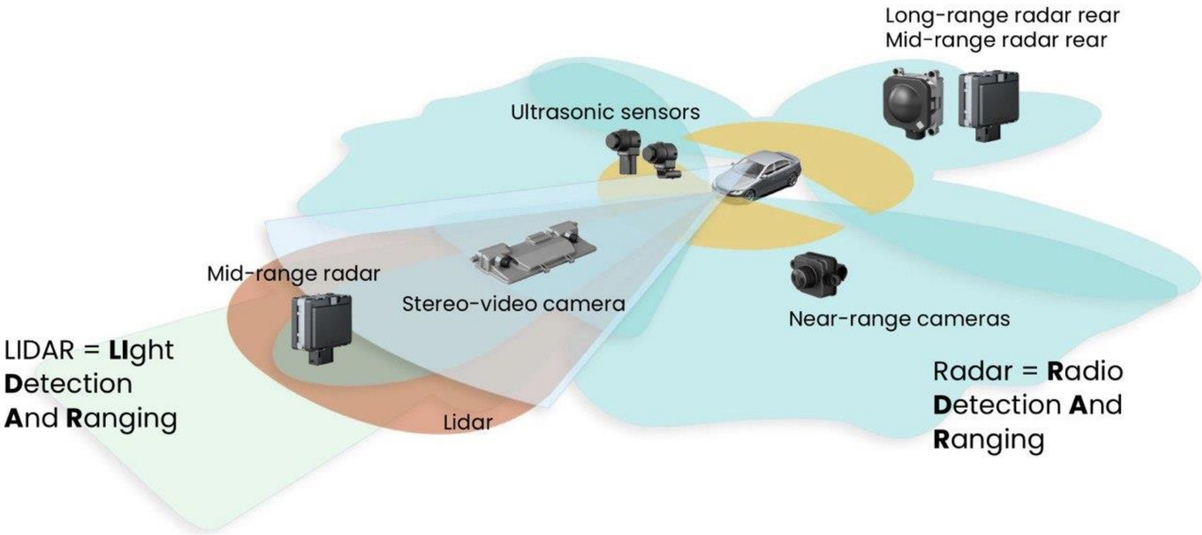


Figure 2: Sensor suite resembling deployed on highly automated vehicles [image source: Bosch]

1.2.1 On-Board Sensors:

Onboard sensors are like the eyes and ears of a vehicle. They collect information directly from the car's surroundings, helping it navigate and make smart decisions. These sensors include cameras, LiDAR, radar, GNSS, and IMU. Cameras see and recognize things, LiDAR creates 3D maps, radar detects objects, GNSS provides global positioning, and IMU helps understand motion. Together, these sensors make sure the car can drive safely and react wisely to what is happening on the road in real time.[1]

1.2.2 In-vehicle Sensors

In autonomous vehicles, critical sensors encompass the accelerometer, gyroscope, wheel speed sensor, and steering wheel angle sensor. The accelerometer measures changes in the vehicle’s speed, while the gyroscope tracks rotation or angular velocity, for steering. These often unite in the Inertial Measurement Unit (IMU). On each wheel, the wheel speed sensor keeps an eye on how fast it is spinning. This is really important for safety systems like antilock brakes and traction control. Finally, the steering wheel angle sensor shows how much the steering wheel is turning. This is super important for controlling the car and making things like electric power steering and lane departure warnings work in Advanced Driver Assistance Systems (ADAS). These sensors, acting as the car's sensory toolkit, aid in navigation and decision-making, ensuring a smooth and controlled ride. In this case, proprioceptive sensors check the ego-vehicle's current state, commonly using pre-installed units like odometers, IMUs, gyroscopes, and data from the controller area network (CAN) bus.[1]

In addition to the mentioned sensors like the accelerometer, gyroscope, wheel speed sensor, and steering wheel angle sensor, there are several other in-vehicle sensors. These include Rain Sensors, Light Sensors, Temperature Sensors, Fuel Level Sensors, Tire Pressure Sensors, and Biometric Sensors with interior cameras. Rain Sensors automatically adjust wiper speed in response to detected rain or moisture on the windshield. Light Sensors autonomously control headlights by gauging ambient light levels. Temperature Sensors monitor interior and exterior temperatures to contribute to climate control. Fuel Level Sensors indicate the amount of fuel in the tank, aiding drivers in monitoring gas levels. Tire Pressure Sensors alert drivers when tire pressure deviates from recommended levels.

In advanced systems, Biometric Sensors and interior cameras play a vital role. They monitor the driver, adapting settings based on preferences, and detecting signs of fatigue or drowsiness to enhance safety. These applications collectively contribute to an improved driving experience, with ongoing advancements continuing to refine and expand the capabilities of in-vehicle sensors.

The specific sensors in a vehicle can vary depending on the make, model, and the level of technology integrated into the vehicle. Advances in automotive technology continue to introduce new sensors and enhance existing ones to improve safety, efficiency, and overall driving experience.

1.2.3 Radar

Radar plays a crucial role in automotive applications by measuring the position and velocity of objects relative to the vehicle. Unlike LIDAR and cameras, radar is robust in poor visibility conditions (all-weather capability) enhancing the reliability of autonomous vehicles. However, it may face challenges in extreme weather. Three main types of automotive radars exist:

- Short-Range Radar (SRR) at 24 GHz is great for parking and close detection, like Blind Spot and Cross Traffic Alert, with a wide view.
- Mid-Range Radar (MRR) at 76-77 GHz is for warnings like Forward Collision and Emergency Braking. It focuses sharply with precise antennas and a narrower field of view (FoV) ensuring your safety on the road.

  Long-range radar (LRR) at 76-77 GHz for front radar, adaptive cruise control and long-range object detection. This radar system possesses narrower field of view with operation range of 250 meters. [4]

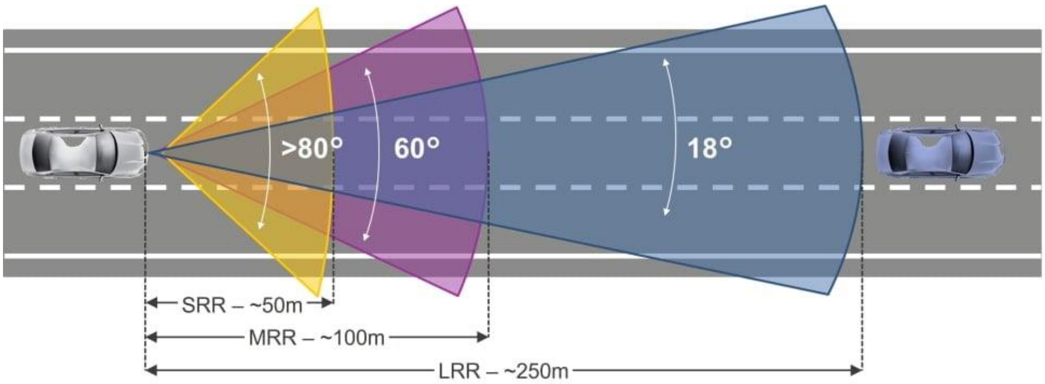


Figure 3: Typical range and Field of View for automotive Radar[4]

Radar measurements include range, relative velocity, and direction estimation. Techniques like Digital Beamforming (DBF) and multi-input-multi-output (MIMO)[5] improve radar capabilities. DBF enhances angular resolution, while MIMO radar, using multiple receivers and transmitters, reduces clutter and enhances coverage. Radars are extensively studied for autonomous vehicle development, covering multiple target detection, ego-motion estimation, radar perception, self-localization, lane prediction, and pedestrian detection. Ego-motion estimation utilizes Doppler information for 2D motion estimation of the vehicle and velocity measurement. Radar-grid, a recent technique, builds a detailed 3D representation for tasks like SLAM, landmark extraction, and sensor fusion. Radars predict driving lanes in the absence of efficient optical sensors. Pedestrian detection involves micro-Doppler signatures and feature extraction methods. Radar-grid, multiple target detection, and sensor fusion are identified as future research directions, with radar-grid offering potential improvements in perception capabilities and cost reduction for AV. [1]

Figure 4: Front Radar Sensor (Left) and Mid-range sensor (Right) [image source Bosch]

1.2.4 LIDAR

LIDAR relies on laser technology to determine the spatial position of objects in its environment, achieved through the emission of laser impulses and the application of the time-of-flight method. The sensor calculates the distance of scanned points from its center. Renowned for its high frequency, precision, extensive range, and robustness to lighting influences, LIDAR stands out as a versatile technology capable of operating both indoors and outdoors, producing either 2D or 3D point clouds. LIDAR improves scene understanding with better detail compared to radar due to higher resolution, achieved through numerous scans and dense scan points. In autonomous vehicles, LIDAR is crucial, using laser beams to measure distances and create detailed point cloud maps around the vehicle. LIDAR has two main ways of measuring distance: pulse measurement (time-of-flight) and phase measurement. Pulse measurement, like Velodyne's widely used product, is great for long distances (up to 800m) and is popular in autonomous vehicles. [6] On the other hand, phase measurement provides faster data rates and accuracy but only works well at shorter distances (less than 100m). Critical parameters affecting LIDAR in autonomous vehicles include eye safety, data receive rate, range resolution, frame rate, and maximum pulse repetition frequency (PRF). Emphasizing human-eye safety, LIDAR ensures invisible, harmless laser emissions. It excels as a primary data source, collaborating with cameras, sonars, and radars, offering varied environmental representations. Despite a lower frame rate, LIDAR's 360-degree field of view, adaptability to lighting and adverse weather conditions and precision make it a favoured sensor in autonomous vehicles. Perception using LIDAR involves segmentation, fragmentation clustering, and tracking. Segmentation groups LIDAR measurement points based on predefined thresholds, often incorporating target distance. Fragmentation clustering identifies object types through physical features like size and shape, while tracking relies on methods such as Kalman filters. LIDAR plays a crucial role in recognising objects like road markings, pedestrians, cyclists, and cars. Using reflection intensity analysis, specifically the modified Otsu method, it ensures robust road marking recognition in diverse lighting conditions [7].

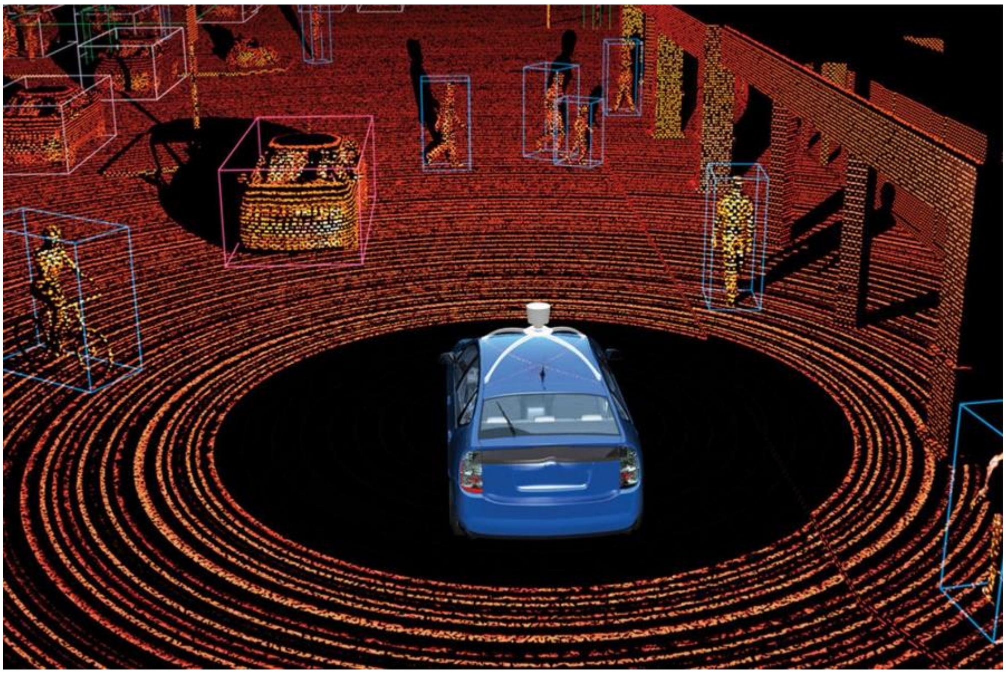


Figure 5: Visualization of a LIDAR point cloud [8]

1.2.5 Camera

Cameras play a pivotal role in autonomous vehicles by generating a 2D image of the surrounding environment through the detection of electromagnetic waves emitted by objects. Unlike active sensors such as LIDAR, radar, and ultrasonic, cameras are passive, relying on detecting energy without emitting any. Despite being a low-cost option, cameras are computationally intensive and find application in various areas, including vehicle, pedestrian, lane marking, and traffic sign detection in autonomous vehicles. There are three main types of cameras used in autonomous driving: single, stereo, and infrared.

Figure 6: Monocular vision vs. Stereo Vision [9]

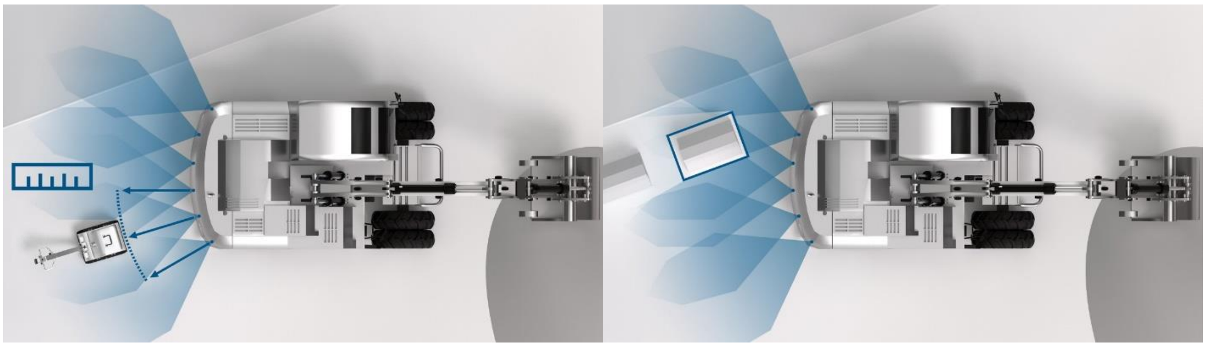
Infrared cameras are specifically designed for pedestrian detection during nighttime conditions. Single and stereo cameras employ different methods for vehicle detection. Appearance-based methods, applied to both single and stereo cameras, analyze features like symmetry, edge, and headlights for vehicle detection. Motion-based methods, used primarily with single cameras, face limitations due to the lack of direct depth information. For stereo cameras, stereo matching, appearance-based methods (such as v-disparity and u-disparity), and motion-based methods (including optical flow and occupancy grids) contribute to vehicle detection and scene segmentation. Pedestrian detection involves model-based, motion-based, appearance-based, and part-based methods. Lane marking detection with cameras follows a series of steps, including pre-processing, color processing, ROI selection, edge detection, and lane detection. Traffic sign detection with cameras involves segmentation, shape feature extraction, and detection using various methods like Hough transform and cascaded classifiers. Despite significant advancements in digital cameras and video processing, challenges persist, such as detecting partially occluded vehicles and improving traffic sign recognition using high-definition (HD) map information. Ongoing research is needed to address these challenges and enhance the robustness of camera-based perception systems in autonomous vehicles.

Figure 6: from Left, Stereo Camera, Multi-Purpose single camera, Near-Range Camera [Photos from Bosch]

1.2.6 Ultrasonic Sensor:

Ultrasonic sensors utilize sound waves for obstacle detection and are particularly effective in parking scenarios and low-speed manoeuvres. Emitting ultrasonic waves, these sensors measure the distance to surrounding objects based on the time it takes for the waves to reflect back. This data is then processed to provide real-time feedback to the driver. Ultrasonic sensors help vehicles to park better by accurately measuring how close they are to obstacles. This makes parking easier and allows for smooth and controlled movement in confined spaces. Strategically placing ultrasonic sensors all around vehicles ensures they detect obstacles from every angle, offering complete coverage similar to Blind Spot Detection. So, these sensors help drivers handle tricky situations in cities and parking lots by giving them a better understanding of their surroundings. They are useful tools for urban driving and parking scenarios. [10]

The ultrasonic sensor plays a crucial role in detecting objects near the vehicle, especially in areas where the camera may have limitations. It processes this information to create maps with altitude data. Furthermore, it provides valuable input to the system, enabling the categorization of objects and determining feasible routes. We also still use the ultrasonic sensor to identify the distance with the object in front of the car and even when parking or behind the car. [11]



(a) (b)

(c)

Figure 7: Ultrasonic Sensor Distance Measurement (a) and Object Localization (b) by Ultrasonic Sensors [10]

Table 2: Sensor Classification and Comparison

Sensors	Advantages	Disadvantages
Camera	• High angular resolution • Good for Classification • Human Vision Resemblance	• Feasible to bad weather and light • No direct range and velocity measurement • Bad depth Estimation
LIDAR	• High angular resolution • 360° visibility • Robust to lighting conditions	• Expensive • Feasible to fog and bad weather conditions • No velocity measurement
Radar	• Long Range • Robust to bad weather and lighting • Hidden Installation • Low cost • Velocity Measurement	• Low Resolution
Ultrasound	• Cheap and cost effective • Mostly used for parking functions • Only for near range (<8)	• Short Range • Gives no insight about the object

1.3 Communication Sensors:

The communication sensors can receive additional data through communication. In this case, the communication sensors suffer more data delay or loss problems, compared to the on-board sensors. The typical communication sensors in CAV include Global Navigation Satellite System (GNSS), Dedicated short-range communication (DSRC) and Cellular technology. Communication and connectivity are enabling technologies for intelligent transportation systems. Perception makes self-driving vehicles aware of their surroundings, similar to senses for human drivers. Multi-vehicle cooperation, awareness of obstacles outside the line of sight, and forecasts require communication. Today's core technologies are DSRC and cellular communication (4G and 5G) [1].

1.3.1 Global Navigation Satellite System (GNSS):

Global Navigation Satellite Systems (GNSS) are crucial for determining vehicle position and velocity on a global scale, playing a vital role in various Intelligent Transportation System (ITS) applications such as Autonomous Vehicles (AV), advanced driver assistance systems (ADAS), toll collection, and traffic management.

GNSS, categorized as communication sensors, communicates with satellites to determine the vehicle's global position using a receiver or antenna. [1]

Different types of GPS receivers offer varying levels of accuracy:

- Standard GPS receivers: achieve 3-8 meters accuracy, while
- Differential GPS (DGPS): enhances it to 1-3 meters by utilizing ground-based stations.
- Real-time kinematic GPS (RTK-GPS) achieves centimetre-level accuracy but it is costly

Despite its advantages, GNSS has limitations [12], including:

- Susceptibility to obstacles causing multipath and non-line-of-sight (NLOS) issues.
- Multipath issues involve reflected signals, addressable through specialized techniques.

NLOS requires mitigation strategies or fusion with other sensors. To address limitations like low update frequency and long-term GPS outages, sensor fusion is utilized. Fusion involves incorporating data from Inertial Measurement Units (IMUs), in-vehicle sensors, cameras, radar, LIDAR, and digital maps. GPS and IMU fusion is common for high-frequency updates, while Bayesian filters fuse in-vehicle sensors for enhanced positioning accuracy [1].

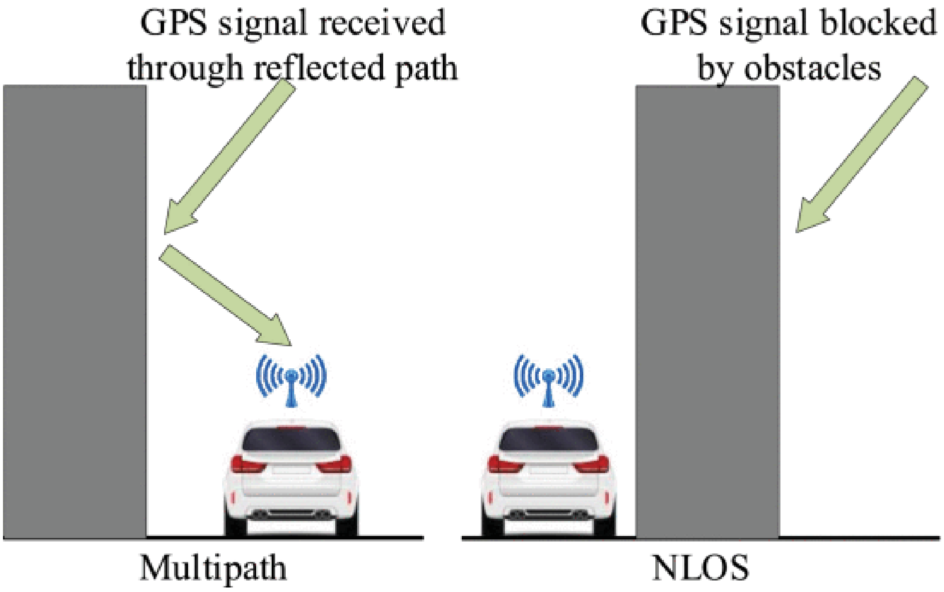


Figure 8: Obstacle influence on GPS signals[1]

1.3.2 Dedicated Short Range Communications (DSRC):

Dedicated Short-Range Communication (DSRC) stands as a pivotal wireless communication technology for Vehicle-to-Everything (V2X) communication, designed for short-range real-time communications between vehicles and roadside infrastructures. DSRC operates in the 5.9GHz frequency band and is primarily used oft vehicle-to-vehicle (V2V) and vehicle-to-infrastructures (V2I) communications. It would be useful to improve road safety, traffic efficiency and overall transportation system. [1]

DSRC stands out from 5G cellular networks due to its low-latency end-to-end communication, making it a reliable option for safety applications. However, for enhanced data speeds, Long Term Evolution (LTE) or C-V2X is employed instead of DSRC. Despite DSRC's significant role, emerging cellular technologies like 5G are considered, and the choice depends on regional standards, regulations, and industry collaboration. [1]

Connected Vehicle (CV) technology operates through two primary components: the Road-side Unit (RSU) and the On-Board Unit (OBU). The OBU is situated within the vehicle, while the RSU is placed either at intersections or alongside roads. Through the OBU, a CV constantly shares essential vehicle details multiple times per second and can also receive messages from nearby CV-equipped vehicles. These incoming messages, referred to as Basic Safety Messages (BSMs), are utilized to assess the trajectories of both the present and nearby vehicles, aiding in the detection of potential future incidents. [13]

DSRC, or Dedicated Short-Range Communication, has key features and applications in intelligent transportation systems, including safety applications, traffic flow management, cell phones for Vehicle-to-Pedestrian (V2P) communication, intersection collision avoidance, electronic toll collection, and the development of connected vehicle systems. Despite its maturity, DSRC faces challenges like scalability, latency, and reliability degradation in certain conditions. Solutions involve incorporating additional technologies, addressing security and privacy concerns, and mitigating construction and maintenance costs. [13]

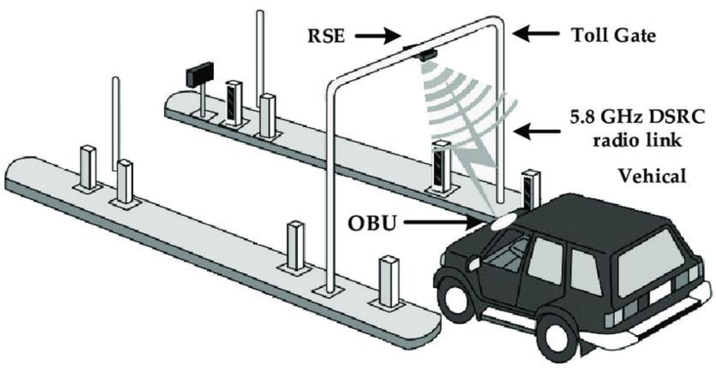


Figure 9: DSRC system communication with roadside equipment [12]

1.4 Vehicle Connectivity:

Connected Vehicles (CV) use Vehicle-to-Everything (V2X) communication technology to interact with other vehicles and networks, including Vehicle-to-Vehicle (V2V), Vehicle-to-Pedestrian (V2P), and Vehicle-to-Infrastructure (V2I) communication. Through dedicated short-range communications (DSRC), CVs can transmit Context-Aware Messages (CAM) that include information about the host vehicle's speed, heading, and brake status. These messages facilitate communication between vehicles, warning drivers about potential crashes and hazards [1].

C-V2X, or Cellular Vehicle-to-Everything, uses existing cellular networks to facilitate direct vehicle-to-vehicle communication and network-based communication. It operates in the 5.9 GHz band and cellular networks, ensuring seamless communication across different environments [14].

Recent developments within the 3rd Generation Partnership Project (3GPP) aim to enhance cellular V2X technology for faster and more efficient communication. While the latest 3GPP LTE releases have introduced cellular V2X for direct vehicle communication, there's an ongoing discussion about whether to adopt Dedicated Short-Range Communications (DSRC) or cellular V2X [14].

However, both approaches encounter challenges, particularly in managing message congestion on the safety channel. This congestion could compromise communication reliability, especially in busy traffic scenarios, potentially posing safety risks. Additionally, existing V2V networks struggle with limited capacity to handle numerous vehicles and data rates, hindering advanced applications like cooperative Advanced Driver Assistance Systems (ADAS) and platooning [14].

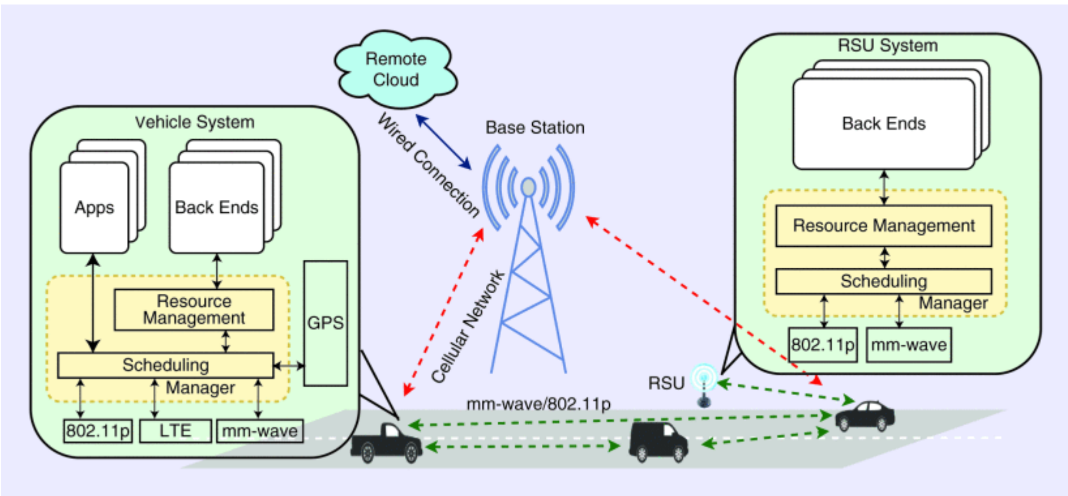


Figure 10: Overview of Vehicular Communication Environment [15]

1.4.1 Vehicle-to-Vehicle (V2V) Communication:

V2V, or Vehicle-to-Vehicle communication, involves vehicles directly sharing real-time information such as speed, position, and direction. This communication enhances road safety by providing drivers with timely details about nearby vehicles, potential hazards, and traffic conditions. It aims to prevent accidents and improve overall traffic flow.

DSRC (Dedicated Short-Range Communications) operates in the 5.9 GHz frequency band, facilitating direct communication between vehicles to share vital information like location and speed. V2V communication typically utilizes dedicated wireless technologies, with Dedicated Short-Range Communications (DSRC) being the most common standard based on the IEEE 802.11p standard.
C-V2X (Cellular Vehicle-to-Everything) uses existing cellular networks to enable both direct vehicle-to-vehicle communication and network-based communication, offering versatility in the 5.9 GHz band and cellular networks.

1.4.2 Vehicle-to-Infrastructure (V2I) Communication:

V2I, or Vehicle-to-Infrastructure communication, refers to the exchange of information between vehicles and infrastructure elements, such as traffic signals, road signs, and other components of the transportation system. This communication is a key aspect of intelligent transportation systems (ITS) and is part of the broader concept of V2X (Vehicle-to-Everything) communication [16].

In V2I communication, vehicles and infrastructure components use wireless technologies to share data, which can include information about traffic conditions, road hazards, traffic signal status, and other relevant details. This real-time exchange of information enables vehicles to make more informed decisions, optimize traffic flow, enhance safety, and improve overall transportation efficiency [16].

V2I communication plays a crucial role in the development of smart and connected transportation systems, contributing to advancements in areas such as autonomous driving, traffic management, and the overall effectiveness of transportation networks.
Key components and devices used in V2I communication include:

1.4.3 On-Board Units (OBUs):

These are devices installed in vehicles to enable communication with roadside infrastructure. OBUs typically consist of a communication module, such as Dedicated Short-Range Communication (DSRC) or Cellular Vehicle-to-Everything (C-V2X) [16].

1.4.4 Roadside Units (RSUs):

Installed along roadways or highways to facilitate communication with vehicles. RSUs communicate traffic conditions, road hazards, and signal timing to vehicles. RSUs are part of V2I (Vehicle-to-Infrastructure) communication systems, allowing the exchange of information between vehicles and the roadside infrastructure [16].

1.4.5 Traffic Management Centers (TMCs):

Centralized facilities optimize traffic flow and safety by providing real-time information to vehicles. This is a centralized facility where traffic-related data is collected, monitored, and managed. TMCs play a crucial role in traffic control and management, coordinating signals, responding to incidents, and optimizing traffic flow. They use various technologies and data sources, including information from roadside units, to make informed decisions and improve overall traffic efficiency [17].

1.4.6 Vehicle-to-Pedestrian (V2P) Communication:

V2P stands for Vehicle-to-Pedestrian, which is a specific subset of communication within the broader V2X (Vehicle-to-Everything) framework. V2P technology enables communication between vehicles and pedestrians, contributing to improved safety and awareness in urban environments.[16]

In a V2P system, vehicles equipped with communication capabilities can exchange information with pedestrians and vice versa. This communication can take various forms, such as warnings, alerts, or notifications to enhance environmental understandings and prevent potential accidents such as [18]:

- **Pedestrian Warnings**: Vehicles can transmit warnings to pedestrians about their presence, when the pedestrian might be in a blind spot or not easily visible to the driver.

- **Crosswalk Safety**: Pedestrians can receive notifications from nearby vehicles, indicating whether it is safe to cross the road or if a vehicle is approaching.

- **Intersection Safety**: The technology can enhance safety at intersections by alerting both drivers and pedestrians about each other's presence, reducing the risk of collisions.

- **Emergency Situations**: In emergency situations, such as a vehicle approaching at high speed or a pedestrian in a hazardous location, V2P communication can provide real-time alerts to all parties involved.

1.5

1.5 Localization and Mapping

1.5.1 Localization:

Localization in Autonomous Connected Vehicles (ACVs) refers to the ability of the vehicle to determine its precise position and orientation within its environment. Accurate localization is a crucial aspect of autonomous driving, as it enables the vehicle to understand its surroundings, plan optimal routes, and make informed decisions. Localization is achieved through the integration of various sensors in which explained in the previous sections [19].

Autonomous vehicle localization is categorized into the three main approaches:

- **Traditional Approaches (First Category):**
  - Utilize exteroceptive sensor data.
  - Employ Bayes-filter-based techniques.
  - Include road marks and landmarks for navigation.

- **Machine Learning Approaches (Second Category):**
  - Explore contemporary methods.
  - Leverage machine learning for enhanced localization.

- **Communication-Based Localization (Third Category):**
  - Focus on vehicle-to-vehicle or vehicle-to-infrastructure communication (V2X).

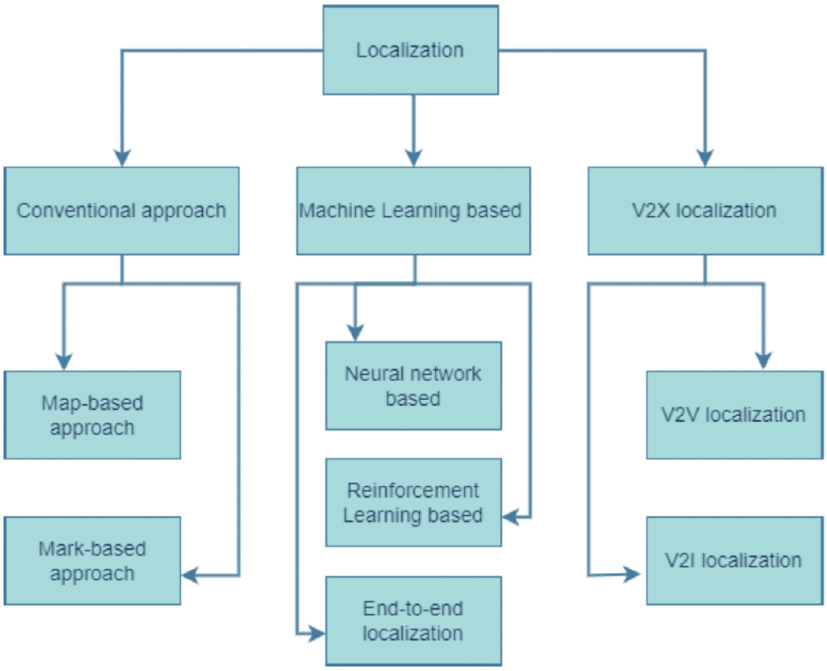


Figure 11: Localization Categories

1.5.1.1 *Conventional Localization:*

Traditional Localization relies on cost-effective sensors like cameras, LiDAR, and Radar for visual perception and scene detection in diverse weather conditions. These sensors provide information that is compared with existing maps during autonomous operation to determine the vehicle's location. Landmarks, road marks, IMU, GNSS, gyroscope, and odometer are sometimes utilized to enhance localization robustness. Conventional map-based localization often suffers from changes in maps during nighttime or harsh weather. Often, features inside the prior map look different at various times of the day and night or in different seasons. For example, lane marking on a road could be partially invisible or fully covered by snow in snowy weather. In that situation, it is important to use sensors with minimal errors [19].

Two main categories exist within traditional localization approaches:

- **Map-based**: In the context of Simultaneous Localization and Mapping (SLAM), the map-based approach encounters difficulties related to error accumulation, high computational demands, fast data transmission necessities, and high resource utilization. Utilizing High-Definition (HD) maps alongside Light Detection and Ranging (LiDAR) sensors provides a more precise and effective solution. Techniques such as point cloud data accumulation, map matching, and feature extraction contribute to accurate vehicle localization. However, assessing map-matching performance becomes challenging in complex urban environments due to the high data volume and computational requirements. This complexity arises from the presence of both semi-static and dynamic objects in urban environments [19].

- **Mark-based**: Locating objects in urban areas is facilitated by using landmarks and road marks. Landmarks like trees, traffic light poles, and tall buildings, along with methods like LiDAR and stereo cameras, enable accurate vehicle localization. Lane markings, guardrails, and various road markings, detected through innovative algorithms and LiDAR point clouds, enhance localization accuracy. Traditional Localization utilizes cost-effective sensors for scene detection, comparing information with maps for vehicle location. Two main categories, map-based and mark-based, differ in their reliance on detailed maps or positions of markings for localization. Vision-only localization using a monocular camera achieves high accuracy through map matching [19].

In summary, map-based approaches focus on generating and localizing within predefined maps, while mark-based methods use elements like road markings and landmarks for vehicle localization. SLAM is briefly mentioned but considered beyond the current review's scope.

1.5.1.2 *Machine-Learning-Based Localization:*

Machine learning is a growing field that helps solve real-world problems, like making autonomous driving safer. In tasks such as spotting pedestrians, recognizing road markings, and locating vehicles, machine learning, especially deep learning, is valuable. These methods often work better than traditional approaches and are sometimes used to make traditional methods even better. [19]

Three main categories exist within the machine-learning based localization approach:

- **Neural Network Approach:**
  Neural networks, particularly convolutional neural networks (CNNs) and recurrent neural networks (RNNs), play a pivotal role in advancing vehicle localization. They excel in generating maps and addressing challenges such as dynamic objects and temporary obstacles [20]. Additionally, they aid in filtering out removable objects from 3D point clouds, thereby facilitating map generation and vehicle localization. Deep learning algorithms, particularly CNNs, can be applied to visual localization, enabling precise localization across diverse weather conditions through techniques like Deep Visual Global Localization (Deep VGL) [21]. CNN-based approaches are instrumental in measuring uncertainty and enhancing accuracy. For instance, CoordiNet predicts camera pose from a single image while providing uncertainty estimation [22]. Furthermore, deep learning techniques extend to specialized tasks such as curb detection, as exemplified in, offering accurate and cost-effective solutions for lateral localization using monovision fisheye cameras [23].

- **Reinforcement Learning Approach:**
  Reinforcement learning (RL) and Deep Reinforcement Learning (DRL) are powerful AI models increasingly applied in automotive applications to train machines through their environment and mistakes. DRL, popularized by Google DeepMind, offers various techniques for autonomous vehicle tasks. CNN and RNN are suggested for perception and localization tasks, while RL is suitable for planning and control [24].

  Techniques such as RL-AKF, DQLL, and RL-L2O have been proposed to improve localization, lane level localization, and LiDAR-based 3D object detection and localization, respectively. RL methods are in the beginning stages for localization tasks, offering potential for future research to complete the entire localization process using RL or DRL, rather than just improving results. Furthermore, the integration of RL into localization processes offers a unique advantage over traditional supervised learning approaches by not requiring extensive labeled data. This aspect opens up avenues for innovative research in refining localization algorithms and improving their robustness in real-world settings. [19]

- **End-to-End Localization:**

  End-to-end localization, or behavior reflex, optimizes driving using a single network for various tasks. This network takes input from sensors and produces steering and wheel commands, treating all tasks as one machine learning problem. In contrast, modular pipelines use interconnected modules for perception, localization, planning, and control. [19]

  Researchers have proposed various techniques for end-to-end localization, such as using deep attention mechanisms to find stable features for visual localization. They also explore different levels of localization, starting from road-level to lane-level, using data from sources like OpenStreetMap and detectors like YOLO. [25]

  End-to-end approaches can be categorized based on whether they map sensor data to control or localization. Pose estimation-based visual localization falls into the latter category.

1.5.1.3 *V2X Localization:*

V2X enables vehicles to interact with their surroundings in Intelligent Transport Systems (ITS). This collaboration is made possible through internet connectivity and various wireless communication methods. V2X localization is a crucial technology that helps vehicles accurately locate themselves among other vehicles, infrastructure, and objects. It is essential for connected and autonomous vehicles to understand their surroundings and make quick decisions, improving safety and efficiency on the road.[19]

- ***V2V Localization***

  V2V localization, or Vehicle-to-Vehicle localization, is a technology that enables vehicles to determine their precise locations in relation to other vehicles. It allows vehicles to communicate with each other and exchange information about their positions, speed, and direction. This shared data enables a vehicle to estimate its position without the necessity of high-precision sensors. V2V localization is crucial for enhancing safety on the road by enabling vehicles to detect potential collisions, coordinate maneuvers, and avoid accidents. It is a key component of connected and autonomous vehicle (CAV) systems, to improve situational awareness and decision-making capabilities. The techniques encompass various methods aimed at enhancing localization accuracy and robustness in vehicular systems. These include a doubled-layer consistency check, which ensures robust localization, a V2V communication-based positioning system integrating GPS receivers and ranging sensors for precise location determination, and the utilization of RFID systems to further enhance localization accuracy. Additionally, considerations are made regarding the impact of road configurations on V2V-based localization, highlighting the need for adaptability in different environments.[19]

- ***V2I Localization***

  V2I, which stands for vehicle-to-infrastructure communication, is crucial in smart transportation systems. Unlike V2V (vehicle-to-vehicle) communication, V2I relies on fixed infrastructure such as roadside units (RSUs) for better positioning accuracy and dependable communication. This method offers benefits like precise location tracking because RSUs stay in fixed positions, ensuring consistent communication, and facilitating the exchange of important details like weather updates and traffic conditions. Various V2I localization techniques include: IR-UWB, DOA, TOA, TDOA, and AOA estimation. These techniques have the potential to revolutionize transportation systems, making them safer and more efficient for everyone.

1.5.2 Mapping and Map Types:

Understanding location dynamics is crucial for effectively addressing challenges in our surroundings. Maps serve as vital tools for both human drivers and autonomous vehicles, aiding decision-making during vehicle control. While physical and digital maps enable drivers to navigate their vehicles, autonomous vehicles require detailed maps for informed decision-making, unlike human drivers who may rely on simpler abstract maps. As autonomous vehicles become more common, the importance of maps increases further.

Maps play a critical role in autonomous driving by offering unique capabilities, including the ability to perceive occluded areas without being affected by environmental conditions, thanks to their infinite range. They also provide accurate real-time information about the surroundings, acting as an additional sensor in autonomous driving systems. Beyond navigation, maps support various functions such as self-localization, vehicle control, motion planning, perception, and system management. They offer static and dynamic information such as road grade, speed limits, and traffic conditions, aiding in self-localization by integrating sensor data and predicting satellite signal availability for accurate positioning. Creating detailed maps is central to ensuring autonomous vehicles are aware of their location [26].

Autonomous driving depends on three key types of map data:

- Topological
- Geometric
- Semantic information

1.5.2.1 *Topological maps:*

Topological maps display connections between things, such as roads, which can be used to plan energy-efficient routes over large distances. Maps that represent the spatial relationships and connectivity between different locations and features without explicitly detailing geometric or visual information. These maps focus on capturing the topological structure of the environment rather than its precise geometry. Advanced methods using deep learning and aerial images can reveal even more connections than traditional maps, helping follow traffic rules on smaller streets. [26]

1.5.2.2 *Geometric Maps:*

Geometric information in urban environments includes three main categories: permanent, temporary, and dynamic features. Permanent features, such as buildings and signs, are crucial for obstacle avoidance and accurate localization within a city. Temporary elements like roadworks, although not always mapped, can significantly influence vehicle sensors. Additionally, dynamic features like moving vehicles require real-time tracking to anticipate their trajectories and plan safe navigation. Effective mapping of permanent structures lays the foundation for detecting and responding to temporary and dynamic changes with precision. [26]

1.5.2.3 *Semantic Maps:*

Semantic information gives meaning to features like road speed limits, lane specifics, and road types. It is a detailed representation of the surrounding environment of the vehicle, and marked with semantic information about different elements such as lanes, traffic signs, pedestrians, vehicles, and other objects relevant to navigation and decision-making. It helps make smart decisions while driving, such as understanding when it is safe to turn left at a traffic light. It is important to connect semantic info with the actual shape of roads. To do this well, geometric data should match up with semantic details. For instance, if a road has different speed limits, it should be split into separate segments. Clear and structured geometry makes it easier to include semantic information accurately. This kind of info is crucial for autonomous driving maps, providing detailed data for safe navigation. New computer vision techniques, like automatic semantic mapping, make it easier to integrate this info into autonomous driving systems. [26]

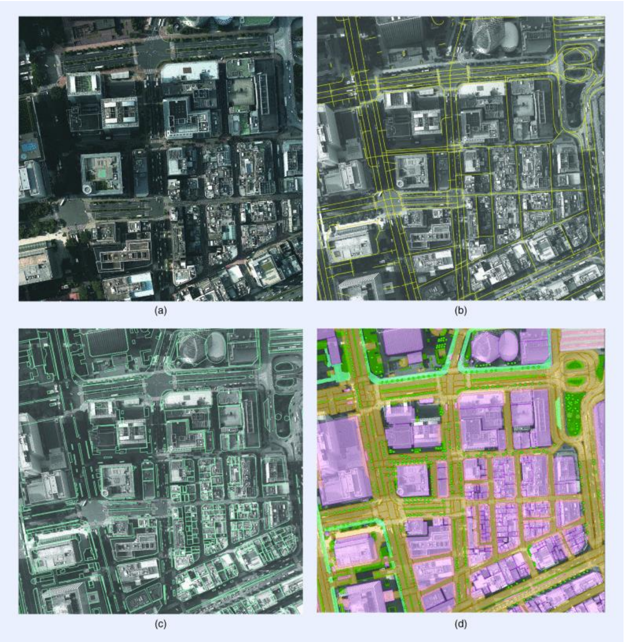


Figure 12: Map formats: satellite image (a), topology (b), geometry (c) and semantic information (d)

1.5.3 Combined Approaches:

1.5.3.1 *SLAM Based Approaches*:

Simultaneous Localization and Mapping (SLAM) is crucial for robots and Autonomous Vehicles (AVs) to create a map of their surroundings and determine their position, especially where there is no existing map. Accurate self-localization ensures safe navigation for self-driving vehicles, helping them understand their environment and navigate effectively.

Self-driving cars often use a mix of methods for efficiency, adaptability and finding the right balance. However, this presents challenges, such as managing large map data and ensuring up-to-date maps. SLAM is vital because it prevents mistakes, particularly where there is no existing map, such as inside buildings with mobile robots. Over years, SLAM has greatly improved, becoming more reliable and scalable. It serves as the backbone for many applications, assisting robots and self-driving cars in planning paths and avoiding getting lost. These enhancements make self-driving technology safer and more reliable in real-world scenarios.

1.5.3.2 *LIDAR SLAM:*

LiDAR-SLAM integrates LiDAR sensors and mapping algorithms to enable real-time mapping and localization for robots and autonomous systems. By emitting laser beams and measuring their return time, LiDAR sensors create detailed 3D maps of surroundings. Its primary goal is precise positioning in unfamiliar environments while mapping them accurately. Challenges include robustness in varying conditions like lighting and dynamic obstacles, essential for safe navigation and decision-making. Advances in LiDAR technology and computational power have made LiDAR-SLAM more practical, with sensor fusion improving performance by combining LiDAR data with inputs from cameras or IMUs. It encompasses various types, from 2D LiDAR-SLAM for indoor environments to 3D LiDAR-SLAM for outdoor surveying. Overall, LiDAR-SLAM is indispensable for navigating unknown or changing environments in robotics and autonomous systems, evolving continually for broader applications.

1.5.3.3 Visual SLAM:

Visual SLAM is an advanced robotics technique for real-time navigation in unknown environments. It utilizes cameras or LiDAR sensors to create a detailed 3D map and determine its precise location. The camera continuously captures images to help the robot or autonomous system to understand its surroundings. V-SLAM relies primarily on visual sensors for cost-effectiveness and rich environmental data. However, it may face challenges in low-light conditions, high computational costs and visual ambiguities. Despite these challenges, advancements in deep learning show promise in improving V-SLAM performance, especially in scenarios involving variable illuminations, occlusions, and dynamic elements. Additional sensors like IMUs or LiDAR can enhance V-SLAM performance by providing orientation and movement data. Depth sensors aid in motion and positioning estimation when combined with cameras. [27]

In practical terms, Visual SLAM enables a robot to navigate and map unknown environments simultaneously. It achieves this through two methods: identifying specific features in the environment and calculating its position based on changes in captured images. Various approaches to V-SLAM exist, including feature-based, direct, and RGB-D SLAM, each with its own advantages and evaluated using publicly available datasets. However, challenges persist, particularly in low-texture and noisy environments. Common visual sensors used in V-SLAM include monocular, stereo, RGB-D, and event cameras. [27]

In summary, Visual SLAM integrates advanced sensors, real-time mapping, and precise localization, demonstrating the fusion of robotics and computer vision for superior performance in navigating unfamiliar spaces.

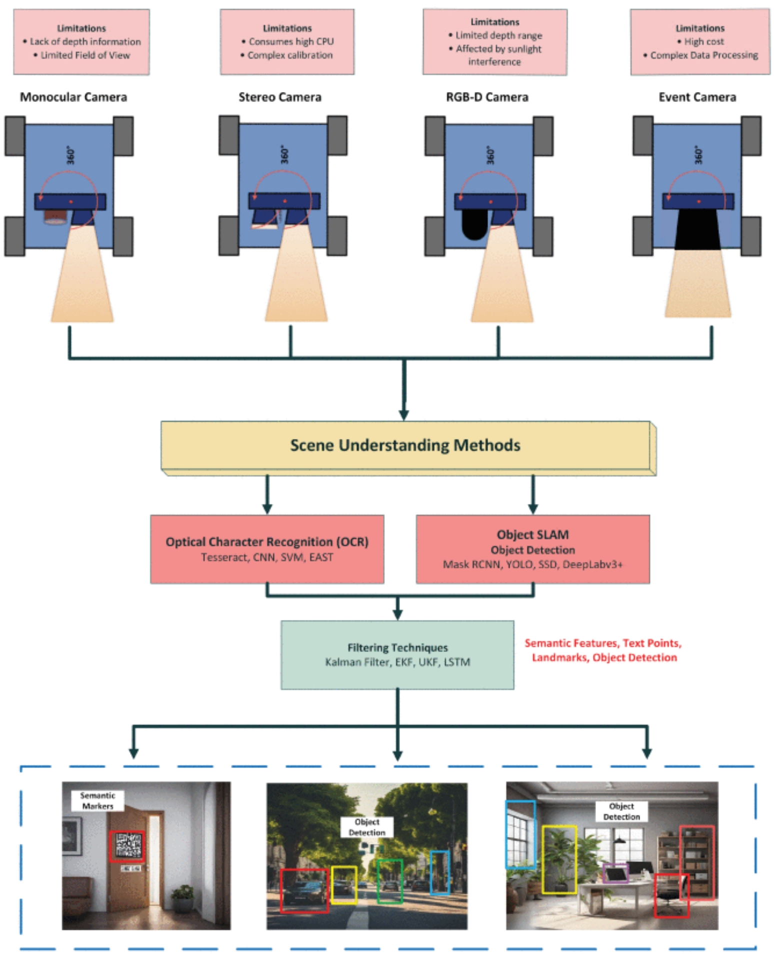


Figure 13: Overview of Scene understanding using V-SLAM in outdoor and indoor environments [27]

1.5.3.4 *Cooperative SLAM:*

To tackle transportation issues, we propose innovative resource-sharing among vehicles, particularly in Cooperative Connected and Autonomous Vehicles (CAVs). Equipped with advanced sensors and computing capabilities, these CAVs collaborate to understand driving environments, promoting road safety and efficiency. This system fosters collaboration among stakeholders across different transport systems.

In this context, Cooperative Simultaneous Localization and Mapping (C-SLAM) emerges as a crucial technique. It involves multiple robots working together to create a map of the environment, merging individual maps into a cohesive global one. Real-time cooperative SLAM enhances 3D LiDAR mapping accuracy, leveraging the wide view and precise distance measurement capabilities of LiDAR sensors, which are preferred over vision sensors in vehicles.

While single-robot SLAM methods have limitations in exploring large environments due to resource constraints, multi-robot systems offer efficiency and robustness advantages, especially for time-sensitive tasks. Cooperative SLAM enables each robot to explore a portion of the environment, establishing consistent coordinates where their areas overlap. However, challenges arise in large-scale scenarios without a global communication infrastructure.

To address these challenges, real-time distributed cooperative SLAM system, namely RDC-SLAM can be used. This system overcomes communication obstacles and enhances efficiency by integrating elaborate communication rules and distributed graph optimization algorithms. RDC-SLAM enables seamless coordination among multiple robots, facilitating accurate and scalable mapping in dynamic environments [28].

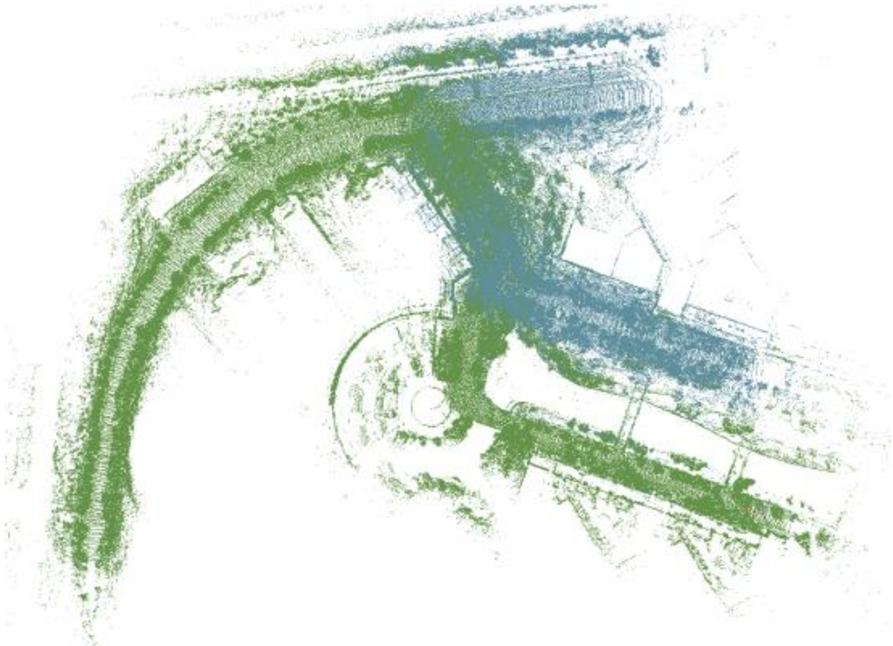


Figure 14: Generated Final Global map - Point Cloud in different Colors collected from different participant [28]

Expanding upon this approach, an innovative LiDAR-based approach known as C-SLAMMODT has been developed for autonomous driving systems to tackle simultaneous localization and mapping (SLAM) alongside moving object detection and tracking (MODT). This strategy integrates multi-vehicle cooperation to address challenges such as view occlusion, which often hinder conventional methods. Unlike traditional approaches that rely on assumptions like static environments or precise ego-vehicle pose estimation, C-SLAMMODT effectively handles dynamic environments by merging cooperative SLAM and MODT modules. These modules leverage shared information from neighboring vehicles to enhance both ego-vehicle pose estimation and object tracking accuracy. Through a unified factor graph optimization, data from both the ego-vehicle and neighboring vehicles are integrated, leading to improved pose estimation and object tracking performance. Comparative experiments have demonstrated the superior accuracy and robustness of C-SLAMMODT in complex environments [29].

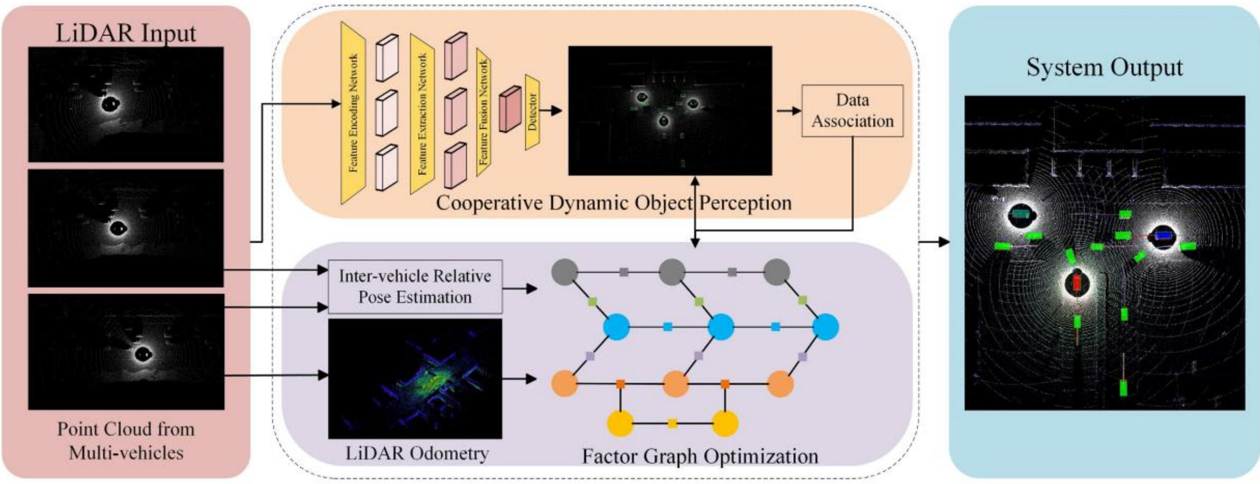


Figure 15: LIDAR Based multi vehicle cooperative SLAM and MODT (C-SLAMMODT) [29]

Additionally, various methods for ego-vehicle pose estimation are compared, including single-vehicle SLAM utilizing LOAM and LeGO-LOAM, cooperative SLAM alone, and the C-SLAMMODT approach, using diverse scenes from OPV2V and V2V4Real datasets. The results illustrate that cooperative SLAM surpasses single-vehicle SLAM methods, achieving smaller root mean square error (RMSE) and mean error (Mean) in ego-vehicle trajectory estimation. Furthermore, C-SLAMMODT exhibits superior accuracy compared to cooperative SLAM alone across different scenes, attributed to its incorporation of dynamic object perception. [29]

1.5.3.5 *Map Merging:*

Map merging and common area detection are critical components in the field of multi-robot simultaneous localization and mapping (SLAM). These processes allow multiple robots to collaboratively create a unified representation of an environment by integrating individual maps generated by each robot. Here's an overview of the key methods and challenges involved [30]:

- **Occupancy Grid Maps**: This method involves merging grid-based maps where each cell indicates the presence or absence of an obstacle. Common techniques include feature matching and transformation estimation to align the maps correctly. Challenges include handling varying resolutions and ensuring accurate alignment without an initial guess of the transformation[30].

- **Feature-Based Maps**: These maps use distinct features (e.g., corners, edges) identified within the environment. Feature matching algorithms like the ones used in computer vision are applied to merge these maps. This approach is robust but computationally intensive, especially when dealing with a large number of features or when features are sparsely distributed [31]

- **Topological Maps:** Involves higher-level representations such as graphs where nodes represent significant places and edges represent paths. Merging these maps requires identifying common nodes and aligning the graphs accordingly. This method is beneficial in environments where geometric features are not well-defined or are difficult to detect [30]

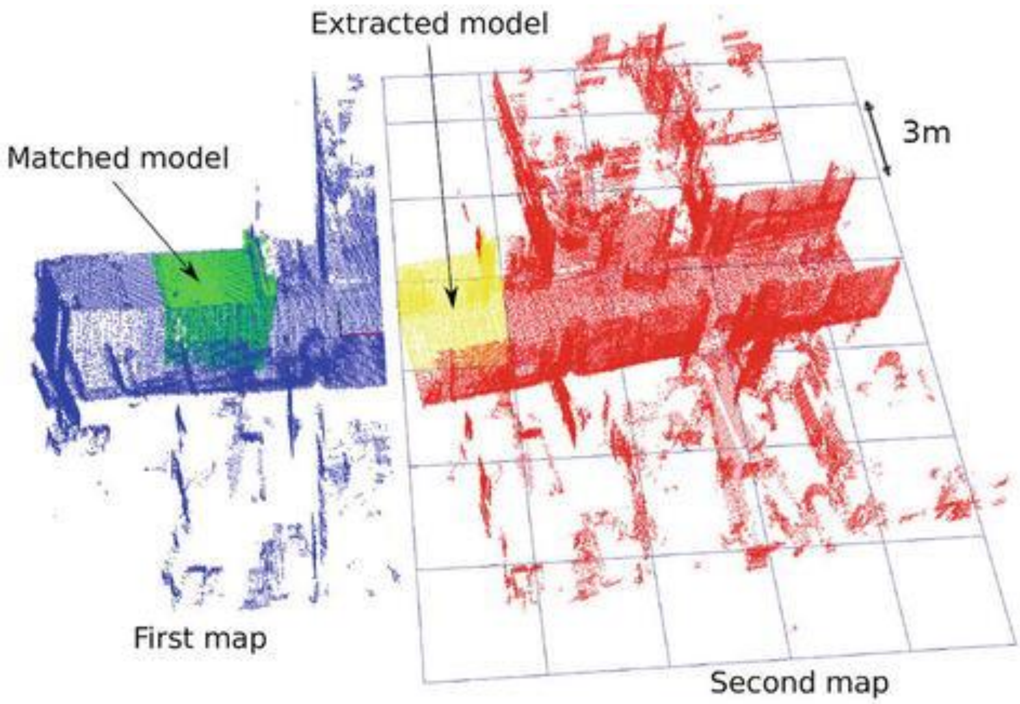


Figure 16: Example of global alignment: A model (yellow) was extracted from one map (red) and matched to another map (blue). [31]

1.5.3.6 *Common Area Detection:*

Detecting common areas between maps is a precursor to successful map merging. Techniques include:

- **Point Set Alignment**: Transforming map data into a domain where alignment can be more easily computed, such as the Radon or Hough domain. This method is effective for maps with substantial overlap and provides a robust alignment even with partial overlaps [31]

- **Descriptor-Based Matching:** Utilizing descriptors based on lines or planes rather than just points, which can enhance performance in environments with limited overlapping areas. This approach leverages higher-level geometric information to improve matching accuracy [31]

- **Octree-Based Methods:** These involve dividing the map into hierarchical 3D grids (octrees), which allow for efficient storage and retrieval of spatial information. Octree-based methods can utilize occupancy probabilities and are particularly useful for integrating 3D maps [31]

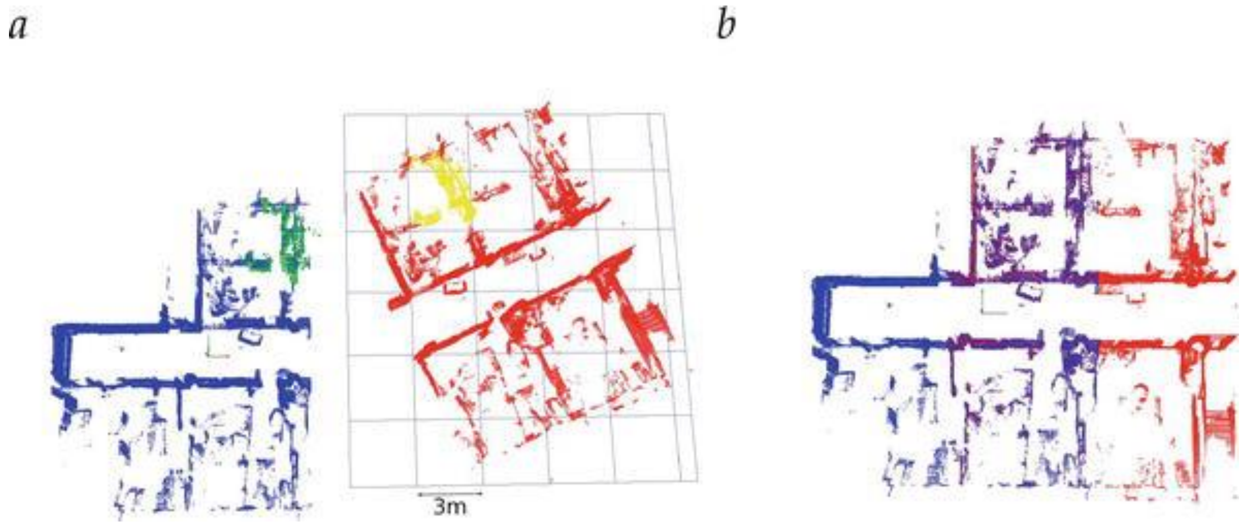


Figure 17: Integration of indoor maps: (a) Indoor maps being integrated and (b) the resulting integrated map.[31]

1.6 Summary:

The introduction emphasizes the significance of connected and autonomous vehicles (CAVs) in modern transportation, highlighting their role in improving efficiency, safety, and reducing human errors. The objectives include enhancing autonomy, accurate environmental mapping, and integrating various communication protocols and sensors to improve vehicle perception.

The structure of the thesis outlines the main tasks of perception, planning, and control, with a focus on perception and map generation. The document details various sensing technologies such as onboard sensors (radar, LIDAR, cameras, ultrasonic sensors) and communication sensors (GNSS, DSRC). It also explores different connectivity types like Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), and Vehicle-to-Pedestrian (V2P).

Localization and mapping are crucial components, with different approaches discussed, including conventional, machine-learning-based, and V2X localization. The document delves into SLAM techniques and their applications in CAVs, emphasizing the importance of accurate map merging and cooperative mapping to handle large areas efficiently.

2 Conception

2.1 Assumptions and Limitations:

The main assumption is that the robots or vehicles under study are independent entities capable of constructing their own local maps. Each of these vehicles is equipped with its own set of sensors, processors, and communication systems. Additionally, the outdoor environmental conditions surrounding these vehicles are taken into account

In this study, three main scenarios are focused on: Maximum Overlapping, Partial Overlapping, and Non-overlapping Views. The greatest challenge encountered lies within Non-overlapping Views. These situations are particularly complex due to the absence of overlap between observations from different robots or vehicles. Conversely, Maximum and Partial Overlapping scenarios necessitate intersection points for observation exchange. This intersection holds particular significance in Cooperative SLAM and Connected Vehicles, where observation sharing is crucial as illustrated in Figure 18 [32].

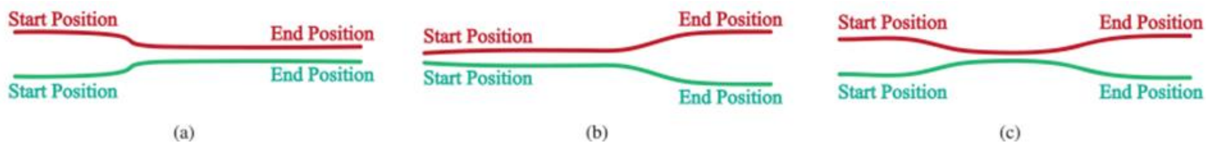


Figure 18: Green curve: Robot A's route. Red curve: Robot B's route. (a) Both robots meet at their ends. (b) Meeting at their starts. (c) Meeting point on routes.[32]

A variety of data collection options exist, including Radar, LIDAR, and Cameras, with IMU and GPS playing supporting roles. However, the decision has been made to employ either a 360-degree field of view LIDAR or a classic mechanical spinning LIDAR to ensure comprehensive coverage of the entire 360-degree field of view. The decision to utilize a 360-degree field of view LIDAR instead of a small field of view LIDAR was primarily influenced by the need for comprehensive environmental coverage. A 360-degree LIDAR provides a full panoramic view, ensuring that no areas are left unscanned, which is crucial for applications requiring complete situational awareness. This approach mitigates the risk of blind spots that can occur with limited field of view LIDAR systems.

Additionally, although cost considerations are often significant in technology selection, the priority in this case was to achieve optimal performance and reliability. The broader coverage provided by the 360-degree LIDAR justifies its selection despite potentially higher costs, as it enhances the accuracy and robustness of the data collection process. The selection of a 360-degree field of view LIDAR over a smaller field of view LIDAR, as depicted in Figure 19, is driven by its capability to provide complete coverage without the constraint of cost considerations [33].

Furthermore, the chosen methodology operates independently of roadside communication facilities or infrastructures, highlighting the system's self-sufficiency and reliability in diverse environments. This independence from external infrastructure further underscores the importance of employing a LIDAR system capable of delivering extensive, uninterrupted coverage.

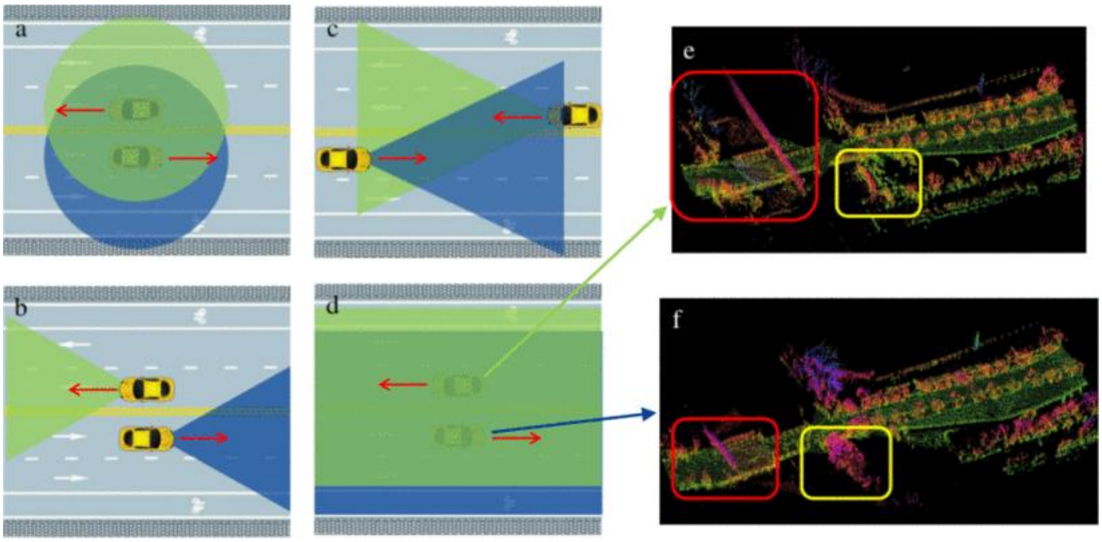


Figure 19: Difference in Point Clouds due to different Perspectives in Maximum Overlapping, Partial Overlapping and non-overlapping conditions using small FoV LIDAR. [33]

2.2 Levels of Data Fusion:

Before diving into the details of map merging, it is important to grasp the basic levels of data processing required for this task. Map merging, a crucial aspect of multi-sensor fusion systems, aims to combine information from diverse sensor modalities to construct accurate and comprehensive maps of the environment. This process involves integrating data from sensors such as LiDAR, cameras, GPS, and more, to generate a unified representation of the surroundings. Two critical stages in map merging are map alignment and data association. Map alignment involves determining suitable spatial coordinate transformations between local maps, distinct from establishing relative poses among robots. Existing algorithms for map alignment often rely on assumptions such as similar map formats, scale, and significant map overlap. On the other hand, data association aims to match and merge features across partial maps, facilitating the fusion of maps generated by multiple robots. This process is crucial given the varying quality of maps produced by individual robots due to sensor differences. The complete map-merging process is depicted in Figure 20 [30].

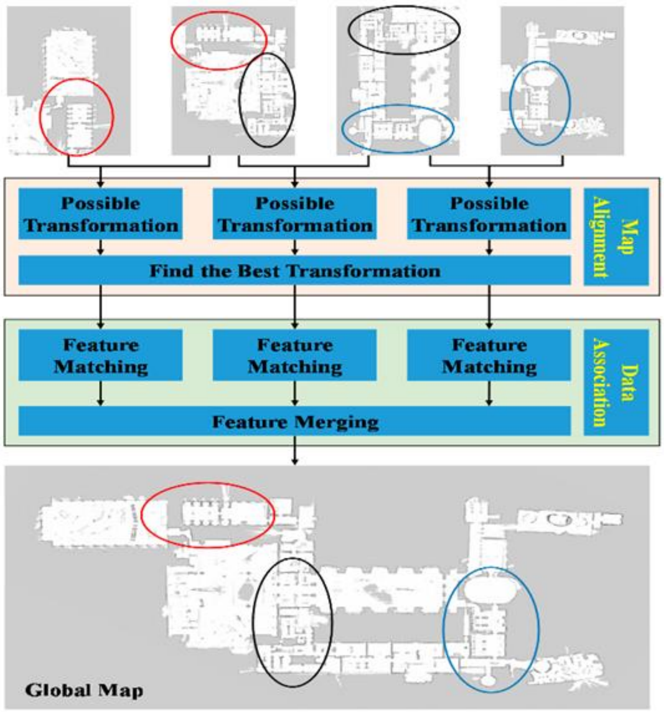


Figure 20: Map Merging Process with map alignment and Feature marging. Overlaps between the maps are marked: red for the first two, black for the second and third, and blue for the third and fourth. [30]

The primary levels of data processing in map merging can be categorized into Data Level (Map Level) and Feature Level.

2.2.1 Data Level (or Map Level):

At data level, raw data from different sensors are combined directly without any preprocessing or feature extraction. For example, LiDAR data representing point clouds and camera data capturing images are merged without extracting specific features. This approach may lack the full utilization of complementary information from different sensor modalities. In the context of map merging, data-level fusion involves merging complete maps generated by various sources or methods. Tasks include aligning map layers, resolving discrepancies, and ensuring overall consistency in the merged map [34].

2.2.2 Feature Level:

Conversely, feature-level fusion involves extracting meaningful features from each sensor modality and then combining these features to create a more comprehensive representation of the environment. For instance, features like point cloud clusters from LiDAR data and edges, colors, or textures from camera data are identified and fused using techniques like feature matching or machine learning algorithms. In map merging, this level entails handling individual features or attributes extracted from different maps, such as satellite imagery, GPS coordinates, and elevation data, to construct detailed and comprehensive maps [34].

In summary, map-level fusion (data-level fusion) is concerned with merging entire maps without feature extraction, while feature-level fusion focuses on integrating meaningful features from different sensor modalities. Both levels are essential for creating accurate and comprehensive maps.

2.3 Map Merging Techniques

Cooperative mapping relies on map merging to explore vast areas accurately which helps to reduce the time and computation cost. This is done by splitting the area into smaller sections for independent exploration. Efficient mapping often involves collaboration among multiple robots or vehicles to cover large areas. Crucial to this collaboration is the merging of local maps produced by individual participants. The cooperative mapping process is divided into two main tasks: The common area detection task and the map merging tasks [32].

To merge maps effectively, the Map Transformation Matrix (MTM) is crucial. It determines the relative angle and translation between individual local maps. Using multiple pose transformation matrices further enhances merging accuracy and corrects mapping drift from single robot mapping. This approach reduces time and computational complexity while improving accuracy [35].

When employing multiple robots for mapping large areas, merging their local maps efficiently is key. One solution is to specify common starting or ending points for their driving paths. It means the robots have to meet each other at their start, and ending points or on the routes as illustrated in Figure 18. This enables direct acquisition of the pose transform between the maps built by each robot, facilitating progressive combination for completing the entire map.

Map merging techniques include direct and indirect methods. Direct methods involve robots meeting at common points to obtain pose transformations, ensuring accurate map alignment. Indirect methods rely on remote sensing and estimation techniques to merge maps without direct interaction between robots. Both approaches aim to enhance the efficiency and accuracy of cooperative mapping efforts

2.3.1 Direct Map Merging (DMM):

Direct map merging (DMM) involves computing the Map Transformation Matrix (MTM) directly using visual and range sensors. The key idea is to find common features or landmarks in the maps and use them to establish connections between different coordinate systems. It can be divided into two types: obtaining robot-to-robot visual and range measurements when robots meet at rendezvous and identifying common regions or objects in multiple maps [35].

Robot-to-robot measurements involve direct interaction between two robots to ascertain their relative positions. For this type of measurement, a two-step map merging framework was proposed [36]. Initially, a Map Transformation Matrix (MTM) is obtained using these robot-to-robot measurements, which involve one robot determining its position relative to the other. In the first step, the MTM provides a hypothesis of their spatial relationship. Subsequently, this hypothesis is verified by instructing the robots to move to a mutually estimated location. If the robots successfully meet at this location, it indicates alignment, and the hypothesis is accepted, leading to the merging of their maps. Conversely, if they fail to meet, the hypothesis is rejected.

Useful formulations to compute Map Transformation Matrix (MTM) based on robot-to-robot measurements from omni-directional cameras were proposed [37]. However, there was an assumption that the robots should encounter each other.

A probabilistic map merging framework for multi-robot SLAM using particle filters was proposed [38]. This framework addressed the challenge of obtaining the most suitable map merging bases from a multiple hypothesis system caused by particle filters, employing Gaussian processes with robot-to-robot measurements. Range measurements were utilized to acquire the direct map transformation matrix when the robots encountered each other.

For another type which is common region detection in multiple maps, The computation of the Map Transformation Matrix (MTM) in [39] relied on overlapped regions detected by ceiling-vision sensors. Utilizing image patches around observed landmarks facilitated the identification of common regions overlapped by multiple robots. A coarse Map Transformation Matrix (MTM) was obtained through place recognition with omnidirectional vision [40]. This method necessitated prior processing to compute the suitable size of the bounding box for Haar-based place recognition.

2.3.1.1 ***Advantages:***

- Speed: DMM can be quickly performed once robot-to-robot measurements or common objects are obtained.
- Immediate Integration: The integration process is straightforward once the necessary data is available.
- Ease of Implementation: Generally simpler to implement compared to IMM.
- Real-Time Capability: Suitable for real-time applications where immediate map merging is required.

2.3.1.2 ***Disadvantages:***

- Dependence on Sensor Quality: Performance heavily relies on the quality of measurements, which can be affected by sensor imperfections.
- Error Propagation: Errors in initial measurements can propagate throughout the merging process, potentially leading to inaccuracies.
- Requirement for Encounter: In some cases, DMM requires robots to physically encounter each other or rely on common objects, which may not always be feasible.

- Limited Accuracy: Depending on the sensor capabilities, the accuracy of DMM may be limited.

2.3.2 Indirect Map Merging (IMM):

Indirect map merging (IMM) computes the Map Transformation Matrix (MTM) by identifying and matching common parts of maps. This process can be categorised into three types: matching common point features, applying scan-matching algorithms, and utilizing spectral information on maps [35].

In both [37] and [38], point feature matching was applied for map features. While [37], utilized the nearest neighbor test (NNT) following coarse map merging with DMM, its effectiveness decreased in dense feature maps. Conversely, [38] introduced the probabilistic feature matching (PFM) algorithm as an alternative method.

Another approach to consider is the utilization of scan-matching algorithms for map merging. For instance, [40] employed the polar scan matching (PSM) algorithm to achieve a more precise Map Transformation Matrix (MTM). Similarly, [41] demonstrated the extension of iterative dual correspondence (IDC) with laser scan sensors.

In [42], a combination of visual feature matching and iterative closest points (ICP) was integrated for map merging. However, these techniques, which rely on scan matching algorithms, may face challenges in environments with minimal occlusions. Moreover, the iterative nature of scan-matching algorithms could result in time inefficiency during map merging processes.

[43], introduces a novel map merging algorithm that leverages spectral information extracted by the Hough transform to enhance the accuracy of the Map Transformation Matrix (MTM). Additionally, it proposes a feature map merging algorithm based on spectral information extracted by virtual supporting lines (VSLs).

Furthermore, [44] suggests a matching algorithm that is particularly suitable for outdoor areas. It initially utilizes GPS location information as the rendezvous point and then identifies common areas based on LIDAR point cloud data.

2.3.2.1 *Advantages:*

- Improved Accuracy: IMM can refine the results obtained from DMM, leading to higher accuracy in map merging.
- Flexibility: Can work with various types of data such as point features, scan data, or spectral information.
- Robustness to Sensor Noise: Can mitigate the effects of sensor noise and errors through advanced matching algorithms.
- Global Optimization: Allows for a broader search space, potentially finding a more optimal solution.

2.3.2.2 *Disadvantages:*

- Computationally Intensive: Generally requires more computational resources compared to DMM, especially with larger search spaces.
- Risk of Local Optima: This may converge to local maxima instead of the global optimum, especially in complex environments.
- Dependency on Preprocessing: Success often relies on preprocessing steps to extract meaningful features or information from sensor data.
- Challenges with Occlusions: Techniques based on scan matching algorithms may struggle with occluded areas in maps.

2.3.2.3 *Trade-offs and Combined Approaches:*

DMM focuses on speed, sacrificing accuracy, while IMM prioritizes accuracy, even if it requires more computational resources. Combining both techniques can balance speed and precision, but it may require extra effort to implement. Integrating DMM and IMM often involves merging hardware and complex software. The choice between DMM and IMM depends on application needs and available resources. However, combining them can improve both computation time and accuracy as it can be seen in techniques in [37], [38] and [40].

Also the technique in [33] is combination of both direct and indirect map merging since it utilizes direct methods like identifying common areas and constraints between maps, while also incorporating indirect methods like factor graph optimization and computing the MTM by matching common parts of maps. Therefore, it can be considered as hybrid approach that combines aspects of both DMM and IMM.

More over Techniques in [32] also combines elements of both direct and indirect mapping by using SLAM for local map construction and directly matching point cloud scans (direct mapping) and then employing feature (ground features and min-Z feature points) extraction and matching algorithms to identify common route segments and establish pose transforms (indirect mapping). By combining these techniques, the method aims to merge individual local maps into a globally consistent one.

DMM's speed is hindered by its need for precise measurements, making real-world deployment difficult due to sensor inaccuracies. Conversely, IMM can enhance DMM's outcomes, but it operates slower and faces the risk of encountering local maxima, especially in 3D mapping scenarios. Balancing computational costs without compromising accuracy remains a key challenge in 3D cooperative mapping. Combining DMM and IMM presents a promising solution to address this challenge. However, it's crucial to prioritize factors like data communication efficiency and real-time performance for future advancements in collaborative SLAM systems [35].

In conclusion, Cooperative mapping involves splitting a large area into smaller sections for independent exploration, with multiple robots or vehicles collaborating to reduce time and computational costs. Central to this process is the merging of local maps produced by each participant. The two main tasks in cooperative mapping are common area detection and map merging. The Map Transformation Matrix (MTM) plays a crucial role, determining the relative angle and translation between local maps, enhancing accuracy, and correcting mapping drift.

Direct map merging (DMM) and indirect map merging (IMM) are the two primary techniques. DMM relies on direct interactions and sensor data to align maps quickly, while IMM uses advanced matching algorithms for greater accuracy but at a higher computational cost. Combining DMM's speed with IMM's precision offers a balanced approach, addressing the trade-offs between computational efficiency and accuracy in large-scale mapping efforts. This hybrid approach is essential for improving the efficiency and effectiveness of cooperative mapping in complex environments.

2.4 Final Assessment and Selection of Suitable Approach for Map Merging:

The study examined the combination of maps created by different robots or vehicles in Cooperative SLAM. Efficient merging of their maps is essential when multiple robots map large areas.

2.4.1 Common Starting/Ending Points:

Ensuring robots start or finish at the same point facilitates easy map connection. However, practical scenarios often involve different starting locations, requiring path overlap detection for map merging and pose transformation calculation [45]

2.4.2 GPS and Landmark aided Methods:

Earlier methods relied on GPS or specific landmarks, which have limitations in indoor environments or areas without navigational aids [44] [45]. Alternative strategies have been developed to address these limitations and enable effective map merging in diverse environments [46].

2.4.3 Crowd-Sourced Mapping - Landmark Feature Map:

Proposed by [47], this method utilizes crowd-sourced data to create landmark feature maps instead of point cloud maps. It creates a new feature layer to streamline map updates and reduce costs. The HD map-based GraphSLAM algorithm aligns features from perception sensors with existing HD maps, ensuring consistency. The Recursive Least Squares (RLS) algorithm integrates new feature layers, compensating for sensor inaccuracies. Multiple intelligent vehicles contribute to crowd-sourced mapping efforts. The algorithm's performance relies on the accuracy of the landmark feature detector.

2.4.4 LiDAR-Based Navigation:

Proposed Method [1]:

- Uses LiDAR with ground and min-Z features along with the Iterative Closest Point (ICP) technique for precise navigation. Ground feature points identify shared route segments but can't distinguish symmetric structures like roads or corridors. Min-Z feature frames capture the minimum height within an area, differentiating scenes with similar ground features but different structures above ground.

- Steps involved:
  1. Local Map Construction: Creating local maps from LiDAR data for each vehicle.
  2. Feature Extraction: Identifying features to efficiently find common areas.
  3. Common Area Detection: Identifying shared areas along vehicle routes.
  4. Global Map Merging: Combining local maps into a coherent global map.

2.4.5 Online Cooperative Mapping Strategy:

- Relies on RTK and lacks a backend optimization procedure, causing motion-dependent blurring in large-scale maps. It integrates location-based services into applications like augmented reality and urban planning.

- Process involves four steps:
  1- Independent single-vehicle mapping with shared GPS pose information.
  2- Vehicles detect nearby peers using shared GPS pose information and exchange local LIDAR scans.
  3- Scan matching to establish relative transformations between local reference frames.
  4- Merging maps generated by different vehicles into a cohesive representation.
  5-

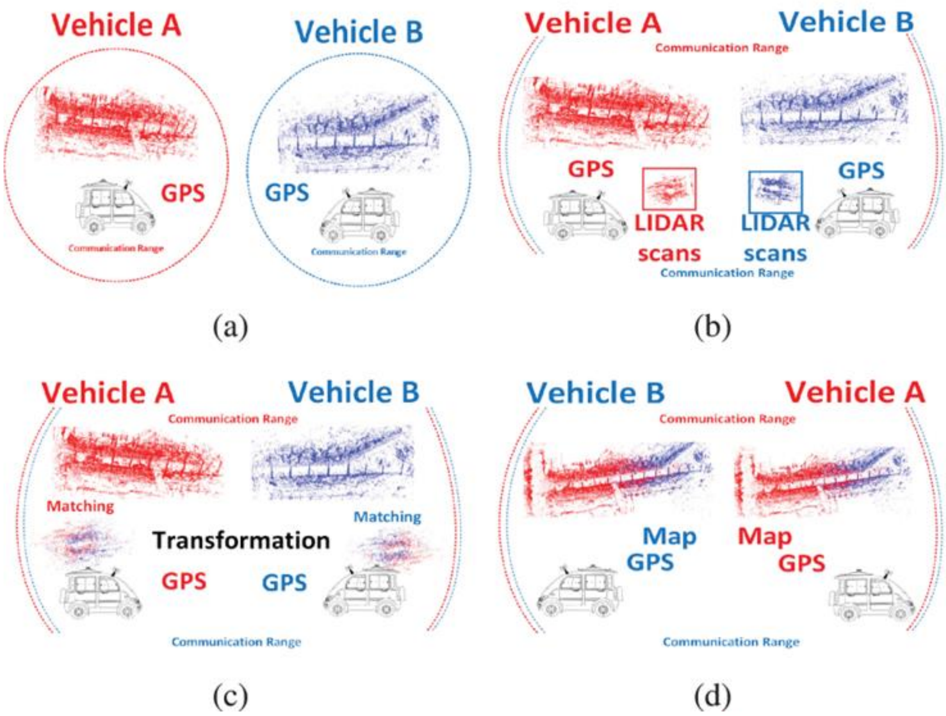


Figure 21: Map Merging Process using GNSS and LIDAR [44]

2.4.6 Multi-Vehicle Cooperative Mapping:

- Method Introduced in :
  Uses multiple vehicles with small FoV LiDAR (81.7° × 25.1°), low-cost GPS, and IMU. It aims to create and maintain a consistent point cloud map efficiently. This method employs multi-sensor fusion to construct local maps, using ground alignment and FoV maps for place recognition across different local maps generated by distinct vehicles.

Conclusion

In conclusion, considering factors such as outdoor operability, independence from roadside infrastructure, and the utilization of advanced technologies like 360-degree LiDAR and GPS, the Online Cooperative Mapping approach [44] emerges as the most suitable method. It is particularly effective in scenarios where cost is not a limiting factor and universal availability of geographical features cannot be guaranteed. This approach provides robust solutions for accurate and efficient map merging, essential for applications ranging from autonomous driving to urban planning.

2.5 Overview of the Conceptual System Model

The conceptual system model for the cooperative 3D mapping system for connected vehicles is designed to ensure efficient, accurate, and robust mapping through the integration of multiple autonomous vehicles. The model leverages high-resolution LIDAR data, precise GNSS positioning, and advanced communication protocols to create a comprehensive and dynamic global map. Below is an overview of the key components and interactions within the system.

Key Components:

- Vehicles:
  - Autonomous vehicles equipped with sensors and communication modules.
  - Each vehicle generates local maps and participates in data exchange and map merging.

- Sensors:
  - LIDAR Sensors:
    - Capture high-frequency 3D point cloud data.
    - LIDAR sensors with specifications such as 128 channels, 30 Hz rotation frequency, and 300 meters range.
  - GNSS Receivers:
    - Provide precise latitude, longitude, and altitude data.

- Onboard Computers:
  - Process sensor data in real-time.
  - Perform feature extraction, motion estimation, and local map generation.
  - Implement data exchange and map merging algorithms.

- Communication Network:
  - Facilitates data exchange between vehicles.
  - Utilizes protocols like UDP for low-latency communication.
  - Future integration with advanced communication technologies such as 5G for enhanced performance.

System Workflow

- Data Acquisition:
  - LIDAR sensors capture high-frequency 3D scans of the environment.
  - GNSS receivers provide accurate positioning data.

- Local Map Generation:
  - Onboard computers process LIDAR data to generate local point cloud maps.
  - Features are extracted using techniques like FPFH (Fast Point Feature Histograms).

- Rendezvous Detection:
  - Vehicles continuously exchange GNSS data to monitor their relative positions.

  - When vehicles come within a predefined distance, a rendezvous event is detected.

- Data Exchange:
  - Upon rendezvous, vehicles exchange their latest LIDAR scans.
  - Data is transmitted using low-latency communication protocols.

- Map Alignment and Merging:
  - Initial alignment is performed using RANSAC (Random Sample Consensus).
  - Refined alignment is achieved with ICP (Iterative Closest Point) algorithm.
  - Aligned point clouds are merged to form a global map.

- Global Map Update:
  - The global map is continuously updated with new data from local maps.
  - Ensures an accurate and up-to-date representation of the environment.

Interaction and Data Flow:

- Sensor Data Flow:
  - LIDAR and GNSS data are continuously collected and processed by each vehicle's onboard computer.

- Communication Flow:
  - GNSS data is periodically exchanged to monitor vehicle positions.
  - LIDAR data is exchanged upon rendezvous detection for map merging.

- Processing Flow:
  - Feature extraction and motion estimation are performed on local maps.
  - Initial and refined alignments are calculated to merge maps accurately.
  - The global map is updated and shared among vehicles.

Future Enhancements:

- Advanced Communication Protocols:
  - Integration of 5G technology to improve data exchange speed and reliability.
  - Enhanced communication protocols to support higher data volumes and more vehicles.

- Improved SLAM Algorithms:
  - Incorporation of advanced LIDAR SLAM algorithms such as LeGO-LOAM or LOAM for better accuracy and robustness.

- Scalability:
  - Testing with larger fleets to ensure scalability and consistent performance.

    - Optimizing algorithms and communication protocols for handling multiple vehicles.
- Real-world Testing:
    - Extensive testing in real-world environments to validate performance under various conditions and challenges.

This conceptual system model outlines a comprehensive approach to cooperative 3D mapping for connected autonomous vehicles, emphasizing real-time processing, accurate data integration, and robust communication. The system aims to enhance mapping efficiency and accuracy, ensuring reliable navigation and situational awareness for autonomous driving.

3 Prototype and Software Architecture

3.1 Introduction:

3.1.1 Project Overview:

This project focuses on developing a cooperative 3D mapping system for connected autonomous vehicles, leveraging LIDAR sensors and GNSS receivers. Traditional single-vehicle mapping systems often face challenges in efficiency, scalability, and accuracy. Our approach enables multiple vehicles to collaboratively build a global map by integrating data from each vehicle into the system. This document outlines the design, implementation, and testing of the system using the CARLA simulation environment, aiming for a scalable and efficient solution suitable for real-time deployment.

3.1.2 Objectives:

- **Design and Implementation**: Create a cooperative mapping system with efficient communication and data processing protocols.
- **Simulation and Validation**: Use CARLA to test and validate the system in various scenarios.
- **Enhance Accuracy and Efficiency**: Demonstrate improvements over single-vehicle mapping in terms of accuracy and time.
- **Ensure Real-Time Performance**: Achieve minimal latency in data processing and inter-vehicle communication for practical deployment.

3.1.3 Scope of Work:

- **System Design and Architecture**: Define roles, communication protocols, and data workflows.
- **CARLA Simulation Setup**: Configure vehicles, sensors, and networks in the CARLA environment.
- **Vehicle and Sensor Configuration**: Set up vehicles with LIDAR and GNSS
- **Data Collection and Processing**: Develop algorithms for LIDAR and GNSS data processing and map merging.
- **Experimental Validation**: Test the system in CARLA under various conditions.

By meeting these objectives, the project aims to enhance 3D mapping for connected autonomous driving, showcasing the benefits of a cooperative approach in improving mapping efficiency, accuracy, and robustness.

3.2 System Design and Architecture:

3.2.1 System Components:

The cooperative 3D mapping system is composed of several key components that work in tandem to achieve efficient and accurate mapping. Each component plays a crucial role in the overall system architecture, ensuring seamless data collection, processing, and communication.

3.2.2 CarlaConnector Class:

- Role: Manages the connection to the CARLA simulator.
- Responsibilities:
  - Establishes and maintains the connection to the CARLA server.
  - Spawns vehicles at specified locations within the simulation.
  - Attaches sensors (LIDAR and GNSS) to the vehicles.

3.2.3 Vehicles:

- Role: Serve as mobile platforms for data collection.
- Models Used: Lincoln MKZ 2020 and Mercedes Coupe 2020.
- Sensors Equipped:
  - LIDAR: Captures detailed 3D point cloud data of the environment.
  - GNSS: Provides accurate positional data for each vehicle.

3.2.4 Sensors:

The system utilizes two primary types of sensors for data collection: LIDAR and GNSS. These sensors are integral to capturing the necessary environmental and positional data for mapping.

- **LIDAR**:
  - **Specifications**:
    - Range: 300 meters
    - Noise Standard Deviation: 0.00
    - Field of View (FOV):
      - Upper FOV: 10 degrees
      - Lower FOV: -25 degrees
    - Channels: 128
    - Rotation Frequency: 30 Hz (rotations per second)
    - Points Per Second: 1,000,000

  - **Characteristics:**
    - High Resolution: Capable of capturing detailed 3D point clouds with high precision.
    - Wide Coverage: The multiple channels and large FOV provide extensive environmental coverage.
    - Real-Time Data: High rotation frequency and point generation rate ensure that data is up-to-date and accurate.

  - **Setup**: The LIDAR sensor is spawned and attached to the vehicle using the spawn_lidar method, which configures the sensor with the specified attributes and initiates data collection.

- **GNSS**:
    - **Specifications**:
        - Accuracy: Centimeter-level positioning accuracy
        - Update Rate: Typically 10 Hz (updates per second)
        - Latency: Minimal, ensuring real-time positional updates

    - **Characteristics**:
        - High Precision: Provides highly accurate positioning data essential for precise navigation and mapping.
        - Real-Time Updates: Frequent updates ensure that the positional data is current, which is critical for dynamic environments.
        - Reliability: Robust against signal disruptions, providing consistent performance in various conditions.

    - **Setup**: The GNSS sensor is configured and attached to the vehicle using the setup_gnss_sensor method, which ensures continuous data collection and integration with the mapping system.

3.2.5 Computers:

- Hardware Configuration:
    - CPU: AMD Ryzen for robust processing capabilities.
    - Memory: 16 GB RAM to handle data-intensive tasks.
- Role: Process the data collected from sensors and manage communication between vehicles.

3.2.6 Networking:

The networking component is crucial for enabling communication and data exchange between vehicles during the mapping process. The system uses local area wireless communication to facilitate real-time data sharing and synchronization.

- Implementation:
    - UDP Sockets: Used for sending and receiving GNSS and LIDAR data between vehicles.
    - Ports: Specific ports are designated for each vehicle to ensure organized and efficient communication.
    - Methods:
        - send_gnss_data: Sends GNSS data to other vehicles.
        - receive_data: Listens for incoming data, processes it, and updates the local map.

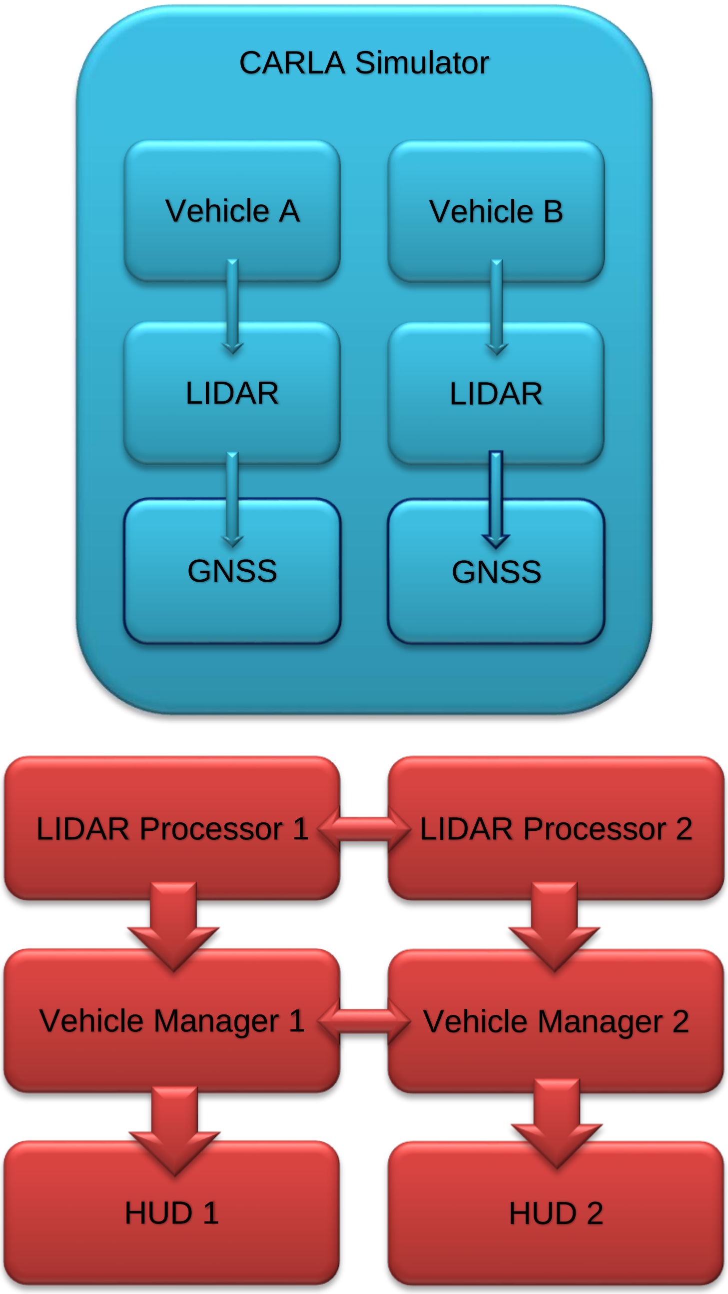


Figure 22: System Architecture of 3D Cooperative Mapping System

3.3 System Architecture:

The system architecture for the cooperative 3D mapping system is designed as a distributed network of autonomous vehicles. Each vehicle is equipped with sensors and processing capabilities to collect, process, and share environmental data. The architecture supports the collaborative creation of a global map by enabling efficient data exchange and synchronization between vehicles. Key components of this architecture include connection management to the CARLA simulator, vehicle spawning and sensor setup, data processing and mapping, communication and data exchange, and HUD visualization. **Error! Reference source not found.** illustrates the elements and overall architecture of the system.

3.3.1 Connection to CARLA Simulator:

The connection to the CARLA simulator is managed by the CarlaConnector class. This class is responsible for establishing and maintaining the connection to the CARLA server, setting up the simulation environment, and managing various simulation parameters.

- **Function**: Establishes a connection to the CARLA simulator.
- **Responsibilities**:
    - Initialize Connection: Uses the CARLA API to connect to the simulator server.
    - Manage Simulation World: Sets up the simulation world, including weather conditions, traffic, and environment settings.
    - Spawn Vehicles and Sensors: Manages the spawning of vehicles and attachment of sensors.
    - Visualization: Visualizes spawn points and other simulation elements for easier debugging and analysis.

3.3.2 Vehicle Spawning and Sensor Setup:

Vehicles and sensors are initialized within the CARLA simulation environment using the spawn_vehicle method. This method ensures that vehicles are placed at predefined locations and equipped with the necessary sensors for data collection.

- **Function**: Initializes vehicles and sensors within the simulation.
- **Responsibilities**:
    - Vehicle Spawning: Spawns vehicles at specified locations using predefined blueprints.
    - Sensor Attachment: Attaches LIDAR and GNSS sensors to each vehicle to capture environmental and positional data.
    - Configuration: Configures sensor attributes such as range, field of view, and update rates.

3.3.3 Data Processing and Mapping:

The LidarProcessor class handles the processing of LIDAR data to generate local maps. This includes managing point cloud data, extracting features, estimating motion, and updating the global map.

- **Function**: Processes collected data to generate local and global maps.
- **Responsibilities**:
    - Point Cloud Processing: Converts raw LIDAR data into 3D point clouds.
    - Feature Extraction: Identifies key features in the point clouds for mapping and alignment.
    - Motion Estimation: Estimates vehicle motion based on changes in the point clouds.
    - Map Updating: Updates the global map with newly processed data to maintain an accurate representation of the environment.

3.3.4 Communication and Data Exchange:

The VehicleManager class manages communication and data exchange between vehicles. It ensures that GNSS data is continuously updated, rendezvous events are detected, and LIDAR data is exchanged during these events.

- **Function**: Manages inter-vehicle communication and data exchange.
- **Responsibilities**:
  - GNSS Data Update: Continuously sends and receives GNSS data between vehicles to monitor their positions.
  - Rendezvous Detection: Detects when vehicles are within a predefined distance and initiates data exchange.
  - LIDAR Data Exchange: Manages the transfer of recent LIDAR scans between vehicles to enhance the global map's accuracy.

3.3.5 HUD and Visualization:

The HUD (Heads-Up Display) class provides a real-time visual interface for monitoring the status of the system. It displays critical information such as vehicle speed, location, GNSS data, and system status.

- **Function**: Provides a real-time visual interface for monitoring the system.
- **Responsibilities**:
  - **Display Information**: Shows relevant data such as vehicle speed, location, GNSS coordinates, and system status.
  - **Monitor Progress:** Helps operators track the progress of the mapping process and the status of each vehicle.
  - **User Interface**: Ensures that the interface is clear, concise, and provides all necessary information for effective monitoring.

Figure 23: Heads-Up Display of each vehicle during the rendezvous event, with both rendezvous and LIDAR data exchange activated.

3.4 Communication Model:

The communication model is critical for enabling efficient data exchange and synchronization between vehicles during the cooperative mapping process. This model ensures that vehicles can share their positional and environmental data in real-time, detect rendezvous events, and exchange LIDAR scans when necessary. The communication model is designed to minimize latency and ensure robust data transfer even in dynamic environments.

3.4.1 GNSS Data Exchange

The GNSS data exchange is essential for vehicles to monitor each other's positions continuously. This data helps in calculating the distances between vehicles and detecting potential rendezvous points where data exchange should occur.

- **Function**: Continuously sends and receives GNSS data between vehicles.
- **Implementation**:
  - UDP Sockets: Use UDP sockets for real-time data transmission, ensuring low latency and minimal overhead.
  - Data Format: GNSS data is serialized using the pickle module and includes vehicle ID and location (latitude, longitude, altitude).
  - Transmission: The send_gnss_data method handles the serialization and transmission of GNSS data.

```
2024-07-21 22:57:12,374 - INFO - Vehicle 24: Sent LiDAR data.
2024-07-21 22:57:12,375 - DEBUG - Distance between vehicles: 15.437592256527429
2024-07-21 22:57:12,375 - DEBUG - Rendezvous threshold: 25.0
2024-07-21 22:57:12,376 - DEBUG - Rendezvous triggered: Distance 15.437592256527429 <= Threshold 25.0
2024-07-21 22:57:12,376 - INFO - Vehicle 24: Preparing to send LiDAR data.
2024-07-21 22:57:12,382 - DEBUG - Distance between vehicles: 15.437592256527429
2024-07-21 22:57:12,382 - INFO - Vehicle 27: Sent LiDAR data.
2024-07-21 22:57:12,382 - DEBUG - Rendezvous threshold: 25.0
2024-07-21 22:57:12,409 - DEBUG - Saving GNSS data for vehicle 24: {'latitude': -0.0002211120122694865,
789388298988}
```

Figure 24: Rendezvous Detection and LIDAR Data Exchange Based on Distance Measurement and Defined Threshold

3.4.2 Rendezvous Detection

Rendezvous detection ensures that vehicles can identify when they are close enough to exchange LIDAR data. This process is based on calculating the Euclidean distance between the GNSS coordinates of the vehicles.

- **Function**: Detects when vehicles are within a predefined threshold distance.
- **Implementation**:
  - Distance Calculation: Uses GNSS data to compute the distance between vehicles.
  - Threshold: A predefined distance threshold (e.g., 25 meters) determines when vehicles should exchange data.
  - Logic: The check_rendezvous method in the VehicleManager class handles the detection logic.

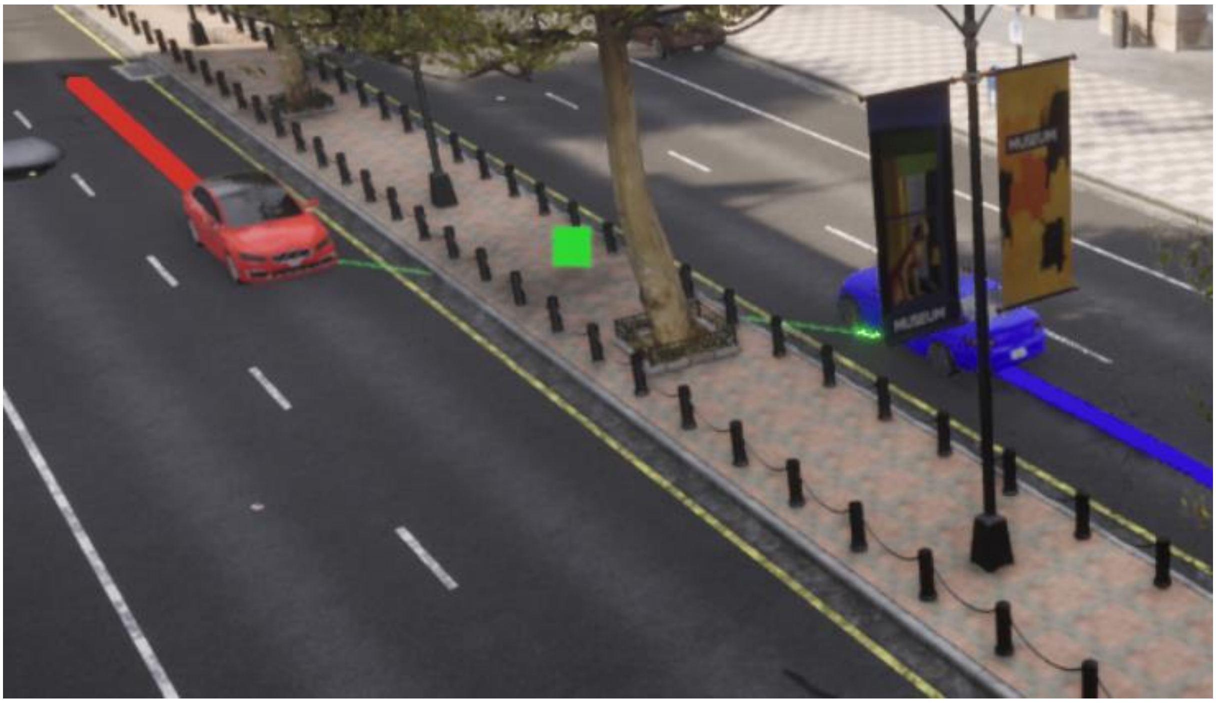


Figure 25: Rendezvous detection in Scenario 1. Green lines represent the LIDAR data exchange between the vehilces and green dot is the rendezvous point

3.4.3 LIDAR Data Exchange:

When vehicles detect a rendezvous, they exchange their most recent LIDAR scans to update each other's maps. This exchange is crucial for maintaining an accurate and comprehensive global map. The process can be seen on Figure 24.

- **Function**: Exchanges LIDAR data upon detecting a rendezvous.
- **Implementation**:
  - Data Serialization: LIDAR data is serialized using pickle and includes the LIDAR point clouds.
  - Chunked Transmission: Large data sets are sent in chunks to ensure reliability.
  - Handling LIDAR Data: The exchange_lidar_data method manages the serialization, transmission, and receipt of LIDAR data.

3.4.4 Real-time Updates:

The system is designed to operate in real-time, ensuring that data is processed and exchanged with minimal latency. This capability is essential for maintaining the accuracy and reliability of the mapping process.

- **Function**: Maintains real-time operation with minimal latency.
- **Implementation**:
  - Efficient Data Processing: Utilizes optimized algorithms and data structures to handle sensor data quickly.
  - Asynchronous Communication: Uses threading to manage communication processes without blocking the main execution flow.

- Continuous Monitoring: Constantly checks for incoming data and updates the system state accordingly.

By implementing these components and communication protocols, the system ensures that multiple autonomous vehicles can collaborate effectively to create a high-definition global map. This approach significantly improves mapping efficiency, accuracy, and robustness, making it suitable for deployment in dynamic and complex environments.

3.5 Implementation Details:

The implementation of the cooperative 3D mapping system involves setting up the CARLA simulation environment, configuring the vehicles and sensors, and developing data collection and processing mechanisms. This section provides detailed information on how these components are implemented, focusing on the practical aspects of integrating LIDAR and GNSS data to create a comprehensive and accurate map.

3.5.1 CARLA Simulation Environment:

The CARLA simulation environment is used to emulate real-world driving conditions for testing the cooperative 3D mapping system. The CarlaConnector class is responsible for connecting to the CARLA server, setting up the simulation world, spawning vehicles, and attaching sensors.

- Setup and Connection: Establish a connection to the CARLA simulator and configure the simulation environment.
- World Configuration: Customize the simulation world, including weather conditions, time of day, and environmental objects.
- Vehicle Management: Spawn and manage vehicles within the simulation, ensuring they are equipped with the necessary sensors.

3.5.2 Vehicle and Sensor Setup:

Vehicles and sensors are set up within the CARLA simulation environment using the spawn_vehicle and setup_gnss_sensor methods. These methods ensure that vehicles are placed at predefined locations and equipped with the necessary sensors for data collection.

- Vehicle Spawning: Use predefined blueprints to spawn vehicles at specific locations within the simulation world.
- Sensor Attachment: Attach LIDAR and GNSS sensors to each vehicle to collect environmental and positional data.
- Sensor Configuration: Configure sensor attributes such as range, field of view, update rates, and data collection methods.

3.5.3 Data Collection and Processing:

Data collection and processing involve capturing LIDAR scans and GNSS data, processing these data to generate local maps, and exchanging data between vehicles during rendezvous events.

3.5.3.1 *LIDAR Data Processing:*

The LidarProcessor class handles the processing of LIDAR data. It involves capturing 3D point clouds, extracting features, estimating motion, and updating the global map.

- **Point Cloud Generation:** Convert raw LIDAR data into 3D point clouds.
- **Feature Extraction:** Identify key features in the point clouds for mapping and alignment.
- **Motion Estimation:** Estimate vehicle motion based on changes in the point clouds.
- **Map Updating**: Update the global map with newly processed data to maintain an accurate representation of the environment.

3.5.3.2 *GNSS Data Processing:*

The VehicleManager class handles the updating of GNSS data and manages LIDAR data exchange during rendezvous events.

- **GNSS Data Update**: Continuously update and send GNSS data to other vehicles.
- **Rendezvous Detection**: Detect when vehicles are within a predefined distance and initiate data exchange.
- **LIDAR Data Exchange**: Manage the transfer of recent LIDAR scans between vehicles to enhance the global map's accuracy.

3.6 Cooperative Mapping Algorithm

The cooperative mapping algorithm is designed to enable multiple autonomous vehicles to collaboratively create a high-definition global map of their environment. This section describes the implementation details of the cooperative mapping process, focusing on the main components: single vehicle mapping, rendezvous detection, and map alignment and merging. The implementation is based on the provided code, ensuring that the details reflect the actual code structure and functionality.

3.6.1 Single Vehicle Mapping:

Each vehicle independently maps its local environment using its LIDAR sensor. The LidarProcessor class handles the processing of LIDAR data to create local 3D maps incrementally. This involves converting raw LIDAR data into 3D point clouds, extracting features, estimating vehicle motion, and updating the local map.

- **Point Cloud Generation**: Converts raw LIDAR data into 3D point clouds.
- **Feature Extraction**: Identifies key features in the point clouds necessary for mapping and alignment.
- **Motion Estimation**: Estimates vehicle motion based on changes in the point clouds.
- **Map Updating**: Updates the local map with processed data to maintain an accurate representation of the environment.

3.6.2 Rendezvous Detection:

Rendezvous detection is critical for enabling vehicles to exchange data when they are close to each other. The VehicleManager class manages the detection of rendezvous events by calculating the distance between vehicles using their GNSS data. When vehicles are within a predefined threshold distance, they initiate data exchange.

- Distance Calculation: Uses GNSS data to compute the Euclidean distance between vehicles.
- Threshold: A predefined distance threshold (e.g., 25 meters) determines when vehicles should exchange data.
- Detection Logic: Implemented in the check_rendezvous method of the VehicleManager class.

3.6.3 Map Alignment and Merging:

When a rendezvous event is detected, vehicles exchange their most recent LIDAR scans and align their local maps. The LidarProcessor class handles the map alignment and merging using a coarse-to-fine matching approach. Initial alignment is achieved using feature extraction and RANSAC, followed by fine alignment with ICP. The transformation matrix obtained is used to merge local maps into a global map.

- Initial Alignment: Uses feature extraction and RANSAC to roughly align the LIDAR scans from different vehicles.
- Fine Alignment: Uses Iterative Closest Point (ICP) to refine the alignment and ensure high accuracy.
- Map Merging: Applies the transformation matrix to align and merge the local maps into a global map.

By implementing these components and communication protocols, the system enables multiple autonomous vehicles to collaboratively create a high-definition global map, improving mapping efficiency, accuracy, and robustness. The cooperative mapping algorithm ensures that data is processed and exchanged in real-time, allowing for dynamic updates and corrections to the map as vehicles move through the environment.

3.7 Test Scenarios and Validations:

Two main scenarios are implemented to validate the cooperative 3D mapping system. In order to do this Town 10 is considered as the simulation Environment on CARLA.

Figure 26: Town 10 in the CARLA environment [48]

3.7.1 Scenario 1: Straight Boulevard

In this scenario, Vehicle A and Vehicle B travel towards each other on a straight boulevard lined with numerous trees. Each vehicle moves at a speed of 50 km/h. When the vehicles come within 25 meters of each other, they initiate LIDAR data exchange. The proximity estimation is based on the distance calculated between the vehicles using GNSS data.

- Objective: Test the system's ability to handle data exchange and map merging in a straight path with environmental obstacles (trees).
- Path:
  - **Red Path**: Vehicle A
  - **Blue Path**: Vehicle B
  - **Green Dot**: Rendezvous point

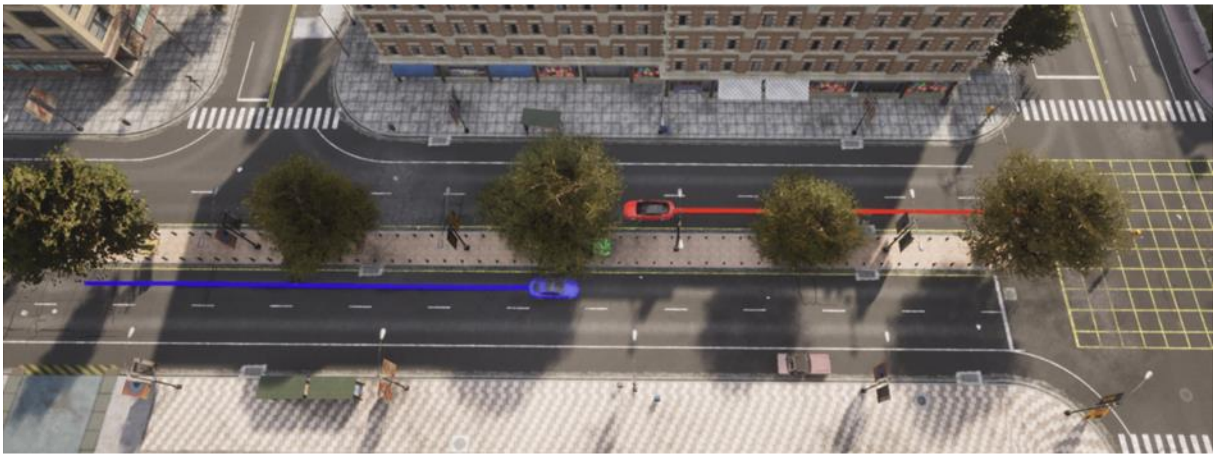


Figure 27: Vehicle Paths in Scenario B: Vehicle 1 follows the red path, Vehicle 2 follows the blue path, and the green dot represents the rendezvous point.

3.7.2 Scenario 2: Intersection

In this scenario, Vehicle A and Vehicle B move towards each other at the same speed of 50 km/h within the same proximity range of 25 meters, but this time the simulation takes place at an intersection without any obstacles between the vehicles.

- Objective: Validate the system's performance in an open intersection, focusing on the efficiency and accuracy of data exchange and map merging.
- Path:
  - Red Path: Vehicle A
  - Blue Path: Vehicle B
  - Green Dot: Rendezvous point

By conducting these test scenarios, we validate the cooperative 3D mapping system's ability to create accurate and comprehensive maps through effective collaboration between vehicles. The scenarios help demonstrate the system's robustness in different environmental settings, its efficiency in handling data exchange, and its accuracy in maintaining an up-to-date global map.

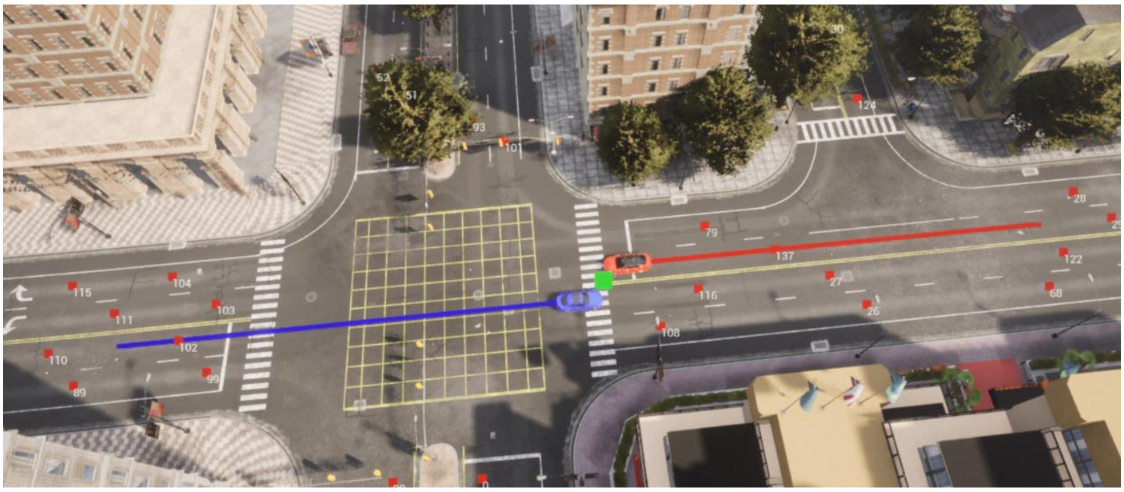


Figure 28: Path of Vehicles in Scenario B: Vehicle 1 follows the red path, Vehicle 2 follows the blue path, and the green dot represents the rendezvous point.

4 Evaluation and Discussion

Introduction:

This evaluation document aims to comprehensively assess the cooperative 3D mapping system for connected vehicles. The evaluation will define the input and output data, relevant metrics, theoretical evaluation based on requirements, detailed test scenarios, and a final discussion of results. The goal is to validate the system's performance, accuracy, and efficiency in dynamically generating a global environment for connected autonomous vehicles.

4.1 Input and Output Data:

4.1.1 Input Data:

- LIDAR Data:
  - Format: Point cloud data (PCD files)
  - Attributes: 3D coordinates (x, y, z), intensity
  - Frequency: 30 Hz
  - Details: The LIDAR sensor provides high-resolution 3D data, capturing the surroundings in detail. The data includes distance measurements and reflectivity information for each point, which is critical for building accurate maps.

- GNSS Data:
  - Format: Latitude, longitude, altitude
  - Attributes: GPS coordinates
  - Frequency: Real-time updates

  Details: The GNSS data offers precise positioning information, essential for aligning the LIDAR data from different vehicles. This high-accuracy GPS data helps in maintaining consistency and accuracy in the generated maps.

4.1.2 Output Data:

- **Local Map:**
  - Format: Point cloud data (PCD files)
  - Attributes: 3D coordinates (x, y, z), intensity
  - Details: Each vehicle generates a local map from its LIDAR data. These local maps are used to detect features and align with maps from other vehicles.

- **Global Map:**
  - Format: Merged point cloud data (PCD files)
  - Attributes: 3D coordinates (x, y, z), intensity
  - Visualization: PNG images of the merged map
  - Details: The global map is created by merging local maps from different vehicles. This comprehensive map provides a unified view of the environment, improving navigation and situational awareness for connected autonomous vehicles.

- **Performance Metrics:**
  - Fitness Score: Measures the alignment quality of point clouds.
  - RMSE (Root Mean Square Error): Measures the average deviation between the corresponding points in the aligned point clouds.
  - C2C Distance (Cloud-to-Cloud Distance): Measures the average distance between points in one point cloud to the nearest points in the other point cloud.
  - Details: These metrics are essential for evaluating the system's ability to generate accurate and consistent maps.

4.2 Algorithm Analysis:

4.2.1 LIDAR Data Processing:

- Description: The LidarProcessor class processes the raw LIDAR data to generate a detailed 3D point cloud map.

- Steps:
  - Data Acquisition: Captures LIDAR scans at high frequency.
  - Point Cloud Generation: Converts raw data into structured point cloud data.
  - Feature Extraction: Identifies key features in the point cloud for alignment purposes.
  - Motion Estimation: Estimates the vehicle's movement to update the local map.

- Relevance: Accurate LIDAR data processing is crucial for creating precise local maps, which are the foundation for the global map.

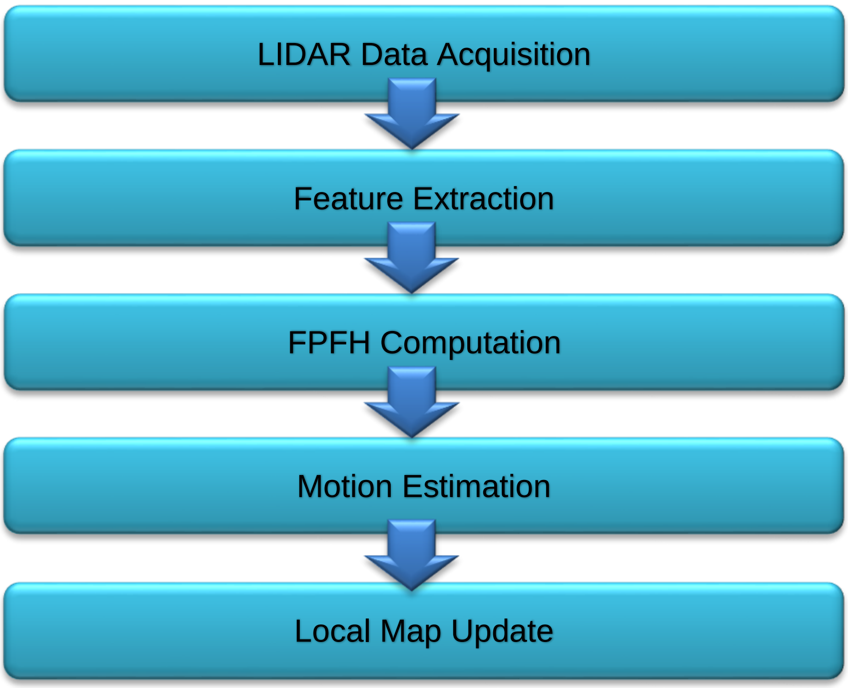


Figure 29: Process of the LIDAR Data Processing

4.2.2 GNSS Data Processing:

- Description: The GNSS data is processed to provide accurate positioning information for each vehicle.
- Steps:
  - Data Acquisition: Continuously receives GNSS signals to update the vehicle's position.
  - Coordinate Transformation: Converts GNSS coordinates into the local reference frame.
  - Rendezvous Detection: Uses GNSS data to calculate the distance between vehicles and detect rendezvous points.
- Relevance: Accurate positioning is essential for aligning local maps from different vehicles to create a coherent global map.

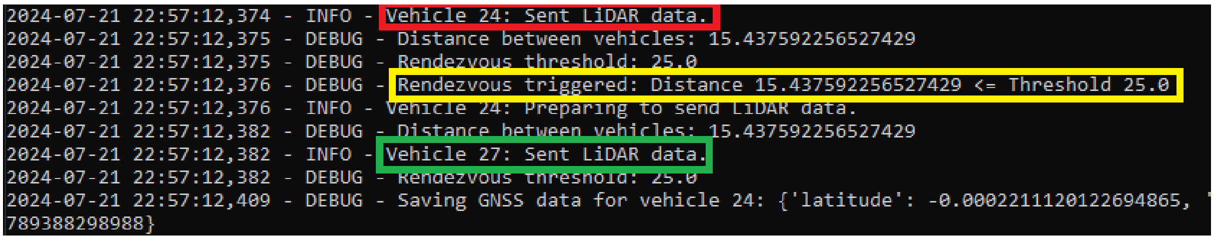


Figure 30: LIDAR Data Exchange and Rendezvous Detection Process

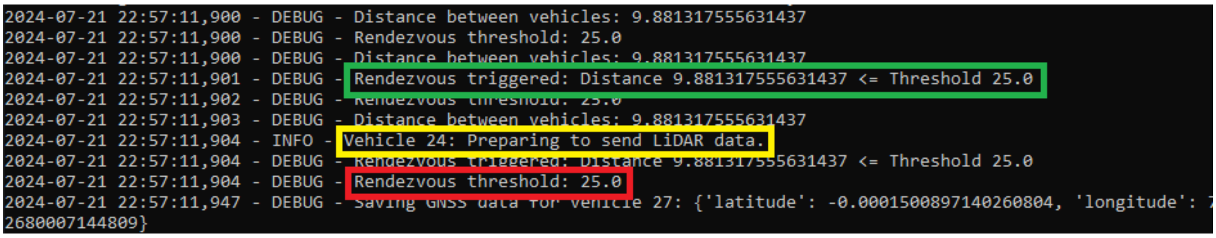


Figure 31: Distance Calculation Between Vehicles

4.2.3 Map Merging:

- Description: The system merges local maps from multiple vehicles into a single global map.
- Steps:
  - Initial Alignment: Uses RANSAC for coarse alignment of point clouds.
  - Refined Alignment: Applies ICP for fine-tuning the alignment.
  - Map Update: Integrates the aligned point clouds into the global map.
- Relevance: Effective map merging ensures that the combined map accurately represents the environment, which is crucial for navigation and situational awareness.

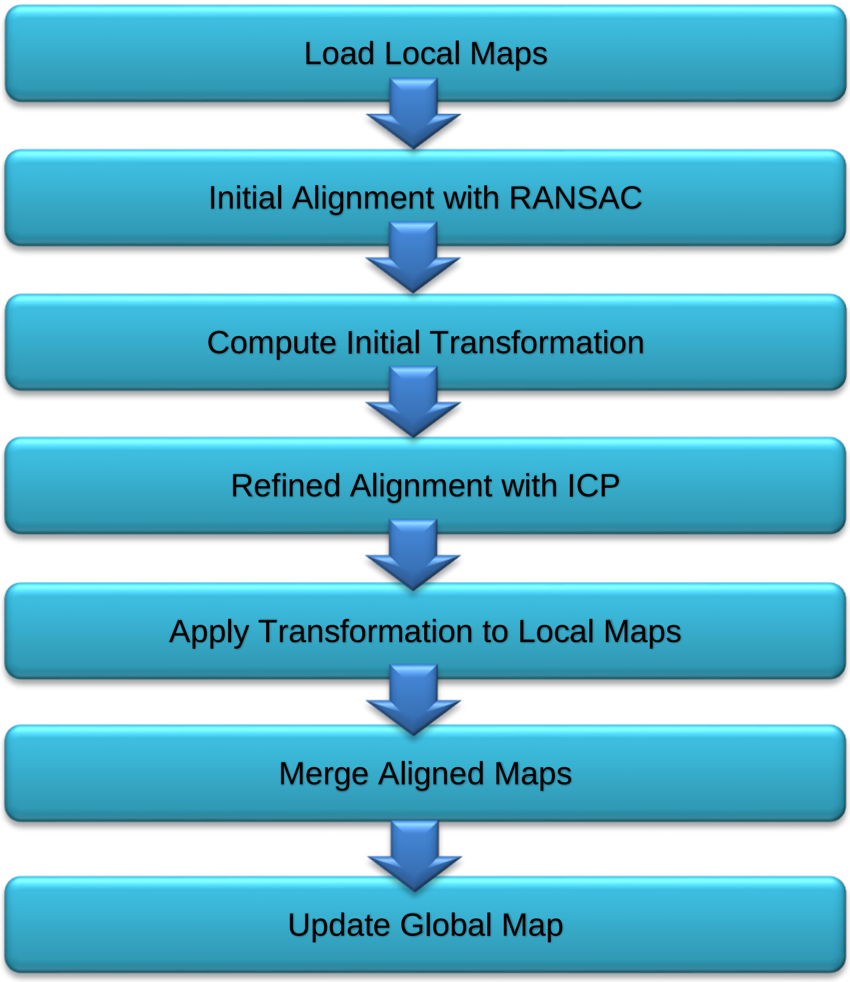


Figure 32: Process of the Map Merging

4.3 Test Scenarios and Expectations:

4.3.1 Scenario 1: Straight Boulevard:

- Description: Vehicle A and Vehicle B travel towards each other on a straight boulevard lined with trees. Each vehicle moves at a speed of 50 km/h. When the vehicles come within 25 meters of each other, they initiate LIDAR data exchange.

- Expectation: The system should accurately detect rendezvous, exchange data, and merge maps despite environmental obstacles. The merged map should align closely with the ground truth, demonstrating the system's ability to handle data exchange and map merging in a straight path with obstacles.

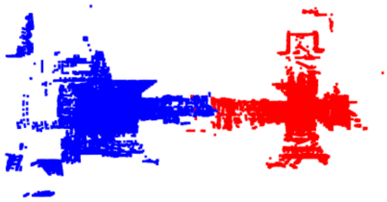


Figure 33: Merged Map of the Vehicles A (red) and B (blue) in scenario 1

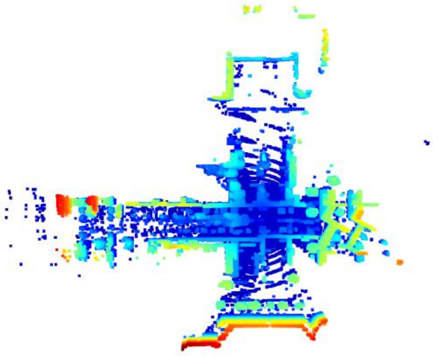


Figure 34: Local Map of Vehicle A

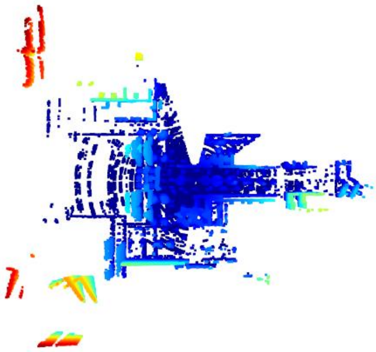


Figure 35: Local Map of the Vehicle B

4.3.2 Scenario 2: Intersection:

- Description: Vehicle A and Vehicle B move towards each other at an intersection without obstacles. Each vehicle moves at a speed of 50 km/h. When the vehicles come within 25 meters of each other, they initiate LIDAR data exchange.

- Expectation: The system should efficiently handle data exchange and map merging in an open intersection, with minimal latency and high accuracy. The merged map should reflect the intersection layout accurately, validating the system's performance in an open, obstacle-free environment.

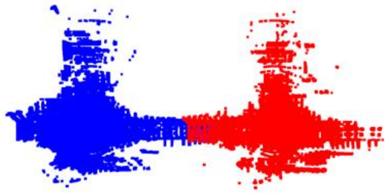


Figure 36: Overlapped Map of the Vehicles A (red) and B (blue) in scenario 2

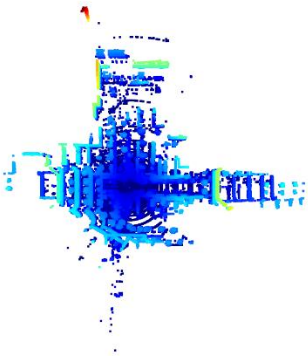


Figure 37: Local Map of the Vehicle B

Figure 38: Local Map of the Vehicle A

4.4 Evaluation of Scenarios:

4.4.1 Scenario 1: Straight Boulevard:

- Initialization:
  - Set up the CARLA simulation and spawn vehicles.
  - Initialize CarlaConnector, VehicleManager, and LidarProcessor.

- Data Collection:
  - Vehicles move along predefined paths.
  - Collect and process LIDAR and GNSS data.

- Rendezvous Detection:
  - Calculate distance and detect rendezvous.
  - Initiate LIDAR data exchange.

- Map Alignment and Merging:

  - Perform initial alignment with RANSAC.
  - Refine alignment with ICP.
  - Merge local maps into a global map.

- Results:

  - Accuracy: The merged map closely matches the ground truth.
  - Efficiency: Data processing and map merging are completed within a reasonable timeframe.
  - Robustness: The system handles environmental obstacles effectively.

- Metrics:

  - Fitness Score: 0.6560
  - RMSE: 0.1499
  - C2C Distance: 1.8506

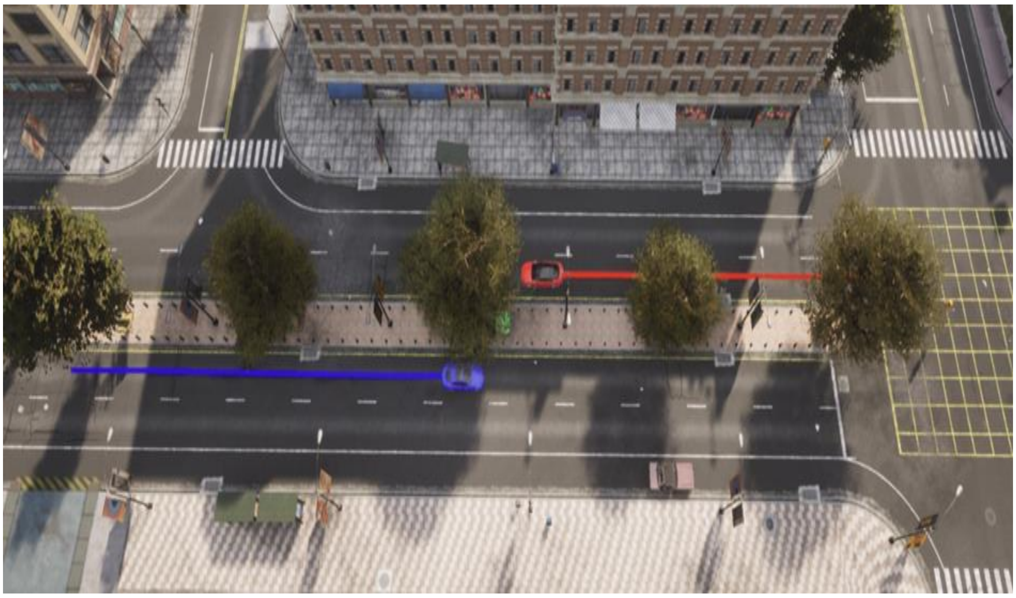


Figure 39: Path of the Vehicles in Scenario 1

4.4.2 Scenario 2: Intersection:

- Initialization:
  - Set up the CARLA simulation and spawn vehicles.
  - Initialize CarlaConnector, VehicleManager, and LidarProcessor.

- Data Collection:
  - Vehicles move along predefined paths.
  - Collect and process LIDAR and GNSS data.

- Rendezvous Detection:
  - Calculate distance and detect rendezvous.
  - Initiate LIDAR data exchange.

- Map Alignment and Merging:
  - Perform initial alignment with RANSAC.
  - Refine alignment with ICP.
  - Merge local maps into a global map.

- Results:
  - Accuracy: The merged map closely matches the ground truth.
  - Efficiency: Data processing and map merging are completed with minimal latency.
  - Robustness: The system maintains high accuracy and efficiency in an open intersection.

- Metrics:
  - Fitness Score: 0.7383
  - RMSE: 0.1531
  - C2C Distance: 0.5563

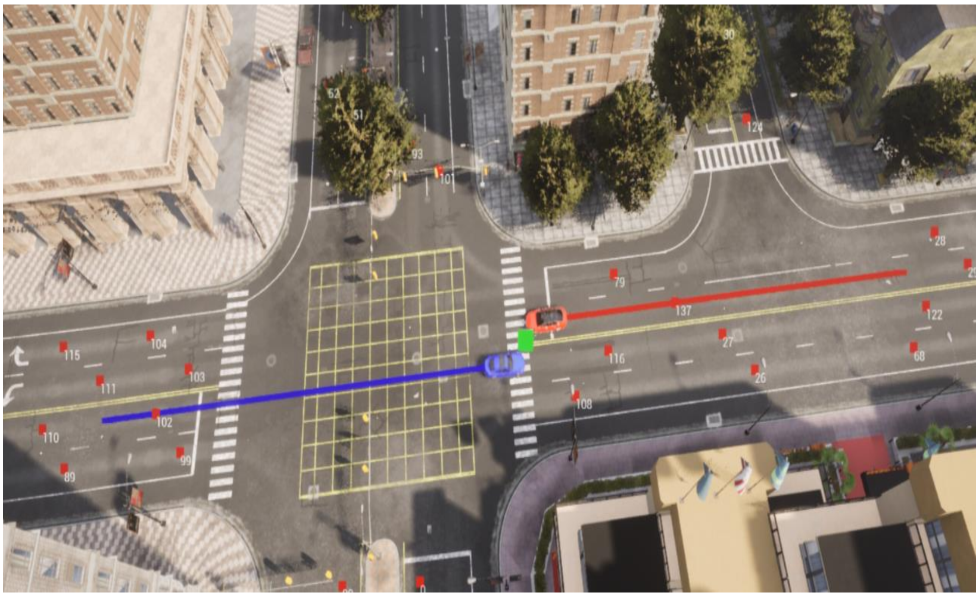


Figure 40: Path of the Vehicle B in Scenario 2

4.5 Final Discussion:

4.5.1 Summary of Results:

- **Accuracy**: Both scenarios demonstrated high accuracy in the merged maps, closely matching the ground truth. The fitness scores and RMSE values indicate that the system effectively aligns and merges the local maps.

- **Efficiency**: The system efficiently handled data processing and map merging within a reasonable timeframe, meeting real-time performance requirements.

- **Robustness**: The system maintained accuracy and efficiency in different environmental settings, demonstrating robustness in both scenarios.

4.5.2 Outlook:

- Improvements: Future work could focus on further optimizing data processing algorithms and improving inter-vehicle communication protocols to enhance performance. Integrating advanced communication technologies such as 5G could significantly improve data exchange speeds and reliability. Additionally, incorporating more accurate LIDAR SLAM algorithms like LeGO-LOAM or LOAM could further enhance the system's mapping accuracy and robustness.

- Scalability: The system should be tested with more vehicles to assess scalability and performance in larger fleets, ensuring consistent and reliable mapping in larger and more complex environments. This could involve simulating various fleet sizes and configurations to evaluate the system's capability to handle increased data volume and communication overhead.

- Real-world Testing: Conducting real-world tests outside of simulation environments is essential to validate the system's performance in actual driving conditions, accounting for real-world variables and complexities. This includes testing in diverse environments and scenarios to ensure the system can handle various challenges and deliver reliable performance in real-world applications.

By systematically evaluating the cooperative 3D mapping system, we can ensure that it meets the required performance metrics and is capable of creating accurate, efficient, and robust maps for connected autonomous vehicles. This evaluation provides a solid foundation for further development and real-world deployment, highlighting the system's strengths and areas for improvement.

5 Final Conclusion:

This research addresses the need for increased autonomy in connected vehicles through enhanced cooperative mapping.

Key elements include:

- Splitting Large Areas: Dividing large regions into smaller sections for independent exploration enhances vehicle autonomy.

- Detailed Recording and Mapping: Accurate mapping of surroundings is achieved using integrated sensors, enabling each vehicle to generate its own local map.

- Enhanced Cooperation: Data exchange between vehicles through V2V connectivity, focusing on urban and outdoor areas, improves overall map creation.

- Sensor Integration: Vehicles are equipped with LIDAR, cameras, and GNSS sensors to ensure comprehensive environmental mapping.

Methods:

- Direct Map Merging (DMM):
    - Robot-to-Robot Measurements
    - Common Region Detection
    - Pros: Quick merging, real-time capability, simpler implementation
    - Cons: Sensor dependencies, limited accuracy

- Indirect Map Merging (IMM):
    - Point Feature Matching
    - Scan-Matching Algorithms
    - Pros: Improved accuracy, flexibility with various data types
    - Cons: Computationally intensive, preprocessing dependency

- Sensor Selection:
    - LIDAR: High-resolution 3D data for precise mapping
    - Camera: Used selectively for distinguishable landmarks
    - GNSS: Essential for accurate rendezvous detection

Techniques:

- Feature Extraction: Fast Point Feature Histograms (FPFH) for real-time applications, compared with Scale-Invariant Feature Transform (SIFT).
- Initial Alignment: Random Sample Consensus (RANSAC) for coarse alignment, compared with Hough Transform.
- Refined Alignment: Iterative Closest Point (ICP) for precise merging, compared with Normal Distributions Transform (NDT).

Implementation:

- Data Acquisition: Continuous collection of LIDAR and GNSS data ensures up-to-date mapping.
- Local Map Generation: Real-time processing on onboard computers generates local point cloud maps.
- Rendezvous Detection: Vehicles exchange GNSS data to monitor relative positions and detect rendezvous events.
- Data Exchange and Map Alignment: LIDAR scans are exchanged upon rendezvous, with initial alignment using RANSAC and refined alignment using ICP.

Validation:

- Simulator: CARLA simulator provides a high-fidelity environment for testing and validation.
- Test Scenarios: Scenario A (Straight Boulevard) and Scenario B (Intersection) are used to evaluate performance.

Results

- Scenario A (Straight Boulevard):
    - Fitness Score: 0.6560
    - RMSE: 0.1499
    - C2C Distance: 1.8506

- Scenario B (Intersection):
    - Fitness Score: 0.7383
    - RMSE: 0.1531
    - C2C Distance: 0.5563

The study concludes that the proposed cooperative mapping system significantly enhances the autonomy and mapping accuracy of connected vehicles, particularly in urban and outdoor environments.

6 References

[1] A. Eskandarian, C. Sun, and C. Wu, “Research Advances and Challenges of Autonomous and Connected Ground Vehicles,” *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, Feb. 2021.
[2] I. Bilik and S. Villeval, “Automotive Radars Challenges and Requirements,” [Online]. Available: https://in.bgu.ac.il/en/engn/ece/radar/Site%20Assets/Pages/Presentations/Radar2015_Bilik.pdf
[3] J. Van Brummelen, M. O’Brien, D. Gruyer, and H. Najjaran, “Autonomous vehicle perception: The technology of today and tomorrow,” *Transp. Res. Part C*, vol. 89, pp. 384–406, Apr. 2018.
[4] M. Martinez Vazquez, “Radar For Automotive: Why Do We Need Radar?,” Mar. 2022.
[5] G. Hacobyan and M. Gonser, “MIMO Radar Systems”
[6] G. Petrie and C. K. Toth, “Introduction to Laser Ranging, Profiling, and Scanning,” *CRC Press*, Nov. 2008.
[7] A. Y. Hata and D. Wolf, “Road marking detection using LIDAR reflective intensity data and its application to vehicle localization,” presented at the International Conference on Intelligent Transportation Systems, Oct. 2014.
[8] Y. Cao *et al.*, “Adversarial Sensor Attack on LiDAR-based Perception in Autonomous Driving,” presented at the ACM Conference on Computer and Communications Security (CCS), Sep. 2019.
[9] S. Singh, “Understanding technology behind self-driving cars: Stereo Vision,” Nov. 2022. [Online]. Available: https://www.labellerr.com/blog/understanding-technology-behind-self-driving-cars-stereo-vision/
[10] “Ultrasonic Sensor,” Bosch. [Online]. Available: https://www.bosch-mobility.com/en/solutions/sensors/ultrasonic-sensor/
[11] A. Suryo Winoto, M. Kristianus, B. Linardi, and H. Leslie Hendric Spits Warnars, “Proposed Autonomous Vehicle Framework for Safe Driving,” IEEE, Jan. 2023.
[12] Y. Gu, L.-T. Hsu, and S. Kamijo, “GNSS/Onboard Inertial Sensor Integration with the Aid of 3-D Building Map for Lane-Level Vehicle Self-Localization in Urban Canyon,” *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 4274–4287.
[13] I. Ali *et al.*, “A Highly Reliable, 5.8 GHz DSRC Wake-Up Receiver with an Intelligent Digital Controller for an ETC System,” *Sensors*, vol. 20, no. 14, Jul. 2020.
[14] J. He *et al.*, “Cooperative Connected Autonomous Vehicles (CAV): Research, Applications and Challenges,” presented at the 2019 IEEE 27th International Conference on Network Protocols (ICNP), IEEE, Oct. 2019.
[15] J. Feng, Z. Liu, C. Wu, and Y. Ji, “Mobile Edge Computing for the Internet of Vehicles: Offloading Framework and Job Scheduling,” *IEEE Veh. Technol. Mag.*, vol. 14, pp. 28–36, Dec. 2018.
[16] F. Arena, G. Pau, and A. Severino, “V2X Communications Applied to Safety of Pedestrians and Vehicles,” *Sens. Actuator Netwroks*, vol. 9, no. 1, Dec. 2019.
[17] L. A. Klein, *ITS Sensors and Architectures for Traffic Management and Connected Vehicles*.
[18] R. Qays Malik, K. N. Ramli, Z. H. Kareem, M. Habelalmatee, A. H. Abbas, and A. Alamoody, “An Overview on V2P Communication System: Architecture and Application,” presented at the 2020 3rd International Conference on Engineering Technology and its Applications (IICETA), IEEE, Sep. 2020.

[19] D. Kumar and N. Muhammad, “A Survey on Localization for Autonomous Vehicles,” *IEEE Access*, vol. 11, p. October 2023.

[20] V. Vaquero, K. Fischer, F. Moreno-Noguer, A. Sanfeliu, and S. Milz, “Improving map re-localization with deep ’movable’objects segmentation on 3D LiDAR point clouds,” 2019 IEEE Intelligent Transportation Systems Conference (ITSC), 2019, Oct. 2019.

[21] T. Gonçalves Cavalcante, “Visual Global Localization Based on Deep Neural Networks for Self-Driving Cars,” UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO CENTRO TECNOLÓGICO, 2022.

[22] A. Moreau, N. Piasco, B. Stanciulescu, and D. Tsishkou, “CoordiNet: uncertainty-aware pose regressor for reliable vehicle localization,” presented at the 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), IEEE, Jan. 2022.

[23] K. Petek, K. Sirohi, D. Buscher, and W. Burgard, “Robust Monocular Localization in Sparse HD Maps Leveraging Multi-Task Uncertainty Estimation,” presented at the 2022 IEEE International Conference on Robotics and Automation (ICRA), IEEE, May 2022.

[24] A. El Sallab, M. Abdou, E. Perot, and S. Yogamani, “Deep Reinforcement Learning framework for Autonomous Driving,” *IST Electron. Imaging Auton. Veh. Mach. 2017*, vol. 29, pp. 70–76, Apr. 2017.

[25] “You Only Look Once: Unified, Real-Time Object Detection,” presented at the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA: IEEE, Dec. 2016.

[26] K. Wong, Y. Gu, and S. Kamijo, “Mapping for Autonomous Driving: Opportunities and Challenges,” *IEEE Intell. Transp. Syst. Mag.*, vol. 13, no. 1, Oct. 2020.

[27] A. R. Sahili *et al.*, “A Survey of Visual SLAM Methods,” *IEEE Access*, vol. 11, Dec. 2023.

[28] Y. Xie *et al.*, “RDC-SLAM: A Real-Time Distributed Cooperative SLAM System Based on 3D LiDAR,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, Sep. 2022.

[29] S. Fang and H. Li, “Multi-Vehicle Cooperative Simultaneous LiDAR SLAM and Object Tracking in Dynamic Environments,” *IEEE Trans. Intell. Transp. Syst.*, pp. 1–11, Feb. 2024.

[30] S. Yu, C. Fu, A. K. Gostar, and M. Hu, “A Review on Map-Merging Methods for Typical Map Types in Multiple-Ground-Robot SLAM Solutions,” *Sensors*, vol. 20, no. 23, Dec. 2020.

[31] Michał Drwiega and Elżbieta Roszkowska, “Multi-Robot Mapping Based on 3D Maps Integration,” Oct. 2022, [Online]. Available: https://www.intechopen.com/books/11456

[32] Y. Zhang, L. Chen, Z. XuanYuan, and W. Tian, “Three-Dimensional Cooperative Mapping for Connected and Automated Vehicles,” *IEEE Trans. Ind. Electron.*, vol. 67, no. 8, pp. 6649–6658, Aug. 2020.

[33] Z. Li, C. Yang, H. Zhuang, C. Wang, and M. Yang, “Cooperative Mapping Using Small FoV LiDARs from Multiple Vehicles,” presented at the 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), IEEE, Oct. 2022.

[34] H. Dong *et al.*, “SuperFusion: Multilevel LiDAR-Camera Fusion for Long-Range HD Map Generation.” Nov. 2022.

[35] H.-C. Lee, S.-H. Lee, T.-S. Lee, D.-J. Kim, and B.-H. Lee, “A survey of map merging techniques for cooperative-SLAM,” presented at the 2012 9th International

Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Daejeon: IEEE, Nov. 2012.

[36] K. Konolige, D. Fox, B. Limketkai, J. Ko, and B. Stewart, “Map merging for distributed robot navigation,” presented at the Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003), Las Vegas, NV, USA: IEEE, Oct. 2003.

[37] X. S. Zhou and S. I. Roumeliotis, “Multi-robot SLAM with Unknown Initial Correspondence: The Robot Rendezvous Case,” presented at the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China: IEEE, Oct. 2006.

[38] H.-C. Lee, S.-H. Lee, M. H. Choi, and B.-H. Lee, “Probabilistic map merging for multi-robot RBPF-SLAM with unknown initial poses,” *Robotica*, vol. 30, no. 2, Mar. 2012.

[39] H. S. Lee and K. M. Lee, “Multi-robot SLAM using ceiling vision,” presented at the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA: IEEE, Oct. 2009.

[40] F. Tungadi, W. L. D. Lui, L. Kleeman, and R. Jarvis, “Robust online map merging system using laser scan matching and omnidirectional vision,” presented at the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan: IEEE, Oct. 2010.

[41] Á. León García, R. Barea Navarro, L. M. Bergasa Pascual, E. López Guillén, M. Ocaña Miguel, and D. Schleicher Gómez, “SLAM and map merging,” *J. Phys. Agents*, vol. 3, no. 1, Jan. 2009.

[42] K. Wang, S. Jia, Y. Li, X. Li, and B. Guo, “Research on map merging for multi-robotic system based on RTM,” presented at the 2012 IEEE International Conference on Information and Automation, Shenyang, China: IEEE, Jun. 2012.

[43] H.-C. Lee and B.-H. Lee, “Improved Feature Map Merging Using Virtual Supporting Lines for Multi-Robot Systems,” *Adv. Robot.*, vol. 25, no. 13–14, pp. 1675–1696, Apr. 2012.

[44] Z. XuanYuan, B. Li, X. Zhang, L. Chen, and K. Huang, “Online Cooperative 3D Mapping for Autonomous Driving,” presented at the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China: IEEE, Jun. 2018.

[45] B. Kim *et al.*, “Multiple relative pose graphs for robust cooperative mapping,” presented at the IEEE International Conference on Robotics and Automation, Anchorage, AK, USA: IEEE, May 2010.

[46] Q. Li, L. Chen, M. Li, S.-L. Shaw, and A. Nüchter, “A Sensor-Fusion Drivable-Region and Lane-Detection System for Autonomous Vehicle Navigation in Challenging Road Scenarios,” *IEEE Trans. Veh. Technol.*, vol. 63, no. 2, pp. 540–555, Sep. 2013.

[47] C. Kim, S. Cho, M. Sunwoo, and K. Jo, “Crowd-Sourced Mapping of New Feature Layer for High-Definition Map,” *Sensors*, vol. 18, no. 12.

[48] “CARLA-Town10.” [Online]. Available: https://carla.readthedocs.io/en/latest/map_town10/

Declaration of Compliance

I hereby declare to have written this work independently and to have respected in its preparation the relevant provisions, in particular those corresponding to the copyright protection of external materials. Whenever external materials (such as images, drawings, text passages) are used in this work, I declare that these materials are referenced accordingly (e.g. quote, source) and, whenever necessary, consent from the author to use such materials in my work has been obtained.

Signature:

Elshan Moradkhani

Stuttgart, on the 30.07.2024