SELF-DRIVING VEHICLES AS HIGH-PERFORMANCE COMPUTING SYSTEMS – A HYBRID END-DEVICE TO CLOUD APPROACH

by

AHMAD FAYEZ KHALED ABDELHAFIZ

A dissertation submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY IN ELECTRICAL AND COMPUTER ENGINEERING

2025

Oakland University
Rochester, Michigan

Doctoral Advisory Committee:

Subramaniam Ganesan, Ph.D., Chair
Shadi Alawneh, Ph.D.
Nasim Nezamoddini, Ph.D.
Darrell Schmidt, Ph.D.

This work is dedicated to the memory of my beloved father, Fayez Abdelhafiz. Your love, guidance, and support are deeply missed. I ask Allah (SWT) to bless your soul and grant you Jannah Al-Firdous.

To my dear mother Asiya, my brothers, and sisters, and to my loving wife Wafa and our wonderful children Joury, Shahem, and Asiya, your unwavering love and encouragement have been my greatest motivation.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my committee chair, Professor Subramaniam Ganesan, for his invaluable guidance, support, and encouragement throughout my doctoral journey. His insight, patience, and expertise have been instrumental in shaping the direction and quality of this research.

I would also like to extend my sincere appreciation to my committee members, Dr. Shadi Alawneh, Dr. Nasim Nezamoddini, and Dr. Darrell Schmidt for their engagement, continuous support, and motivation through the process.

My heartfelt thanks go to my mother, Asiya Aleassa, whose unwavering love, prayers, and sacrifices made this achievement possible. I am deeply grateful to my brothers and sisters for their constant encouragement and belief in me throughout this long journey.

To my beloved wife, Wafa Bdeir, I owe endless thanks for her patience, love, and support. Her understanding and strength gave me the motivation to persevere during the most challenging times. Finally, to my children, whose smiles and joy have been my greatest source of inspiration. This achievement is as much yours as it is mine.

Ahmad Fayez Khaled Abdelhafiz

ABSTRACT

SELF-DRIVING VEHICLES AS HIGH-PERFORMANCE COMPUTING SYSTEMS – A HYBRID END-DEVICE TO CLOUD APPROACH

by

Ahmad Fayez Khaled Abdelhafiz

Adviser: Subramaniam Ganesan, Ph.D.

The rapid evolution of autonomous-vehicle (AV) technology has defined a new era of intelligent transportation systems, promising improved safety, efficiency, and scalability. However, the exponential increase in the number of onboard sensors and the corresponding growth in machine-generated data have created challenges in real-time processing, cloud dependency, and sustainable data management. Traditional centralized cloud infrastructures struggle to process this volume efficiently, motivating the need for distributed, edge–cloud hybrid computing paradigms.

This research introduces a hybrid edge-to-cloud framework in which autonomous vehicles utilize their onboard supercomputers when underused or idle as distributed processing nodes.

These vehicles collectively perform computational tasks such as lane feature extraction, harvesting data, HD-map generation, and update, operating as a parallel network of edge machines that seamlessly integrates with the cloud for aggregation and scalability.

The proposed system demonstrates this concept through a crowdsourced HD-map creation pipeline that fuses lane level data from multiple vehicles and integrates it with open geographic information from OpenStreetMap (OSM). The system architecture includes edge-based preprocessing, cloud-based multi-vehicle fusion, and map redistribution.

Each vehicle locally extracts lane features from camera and GPS data and transmits compact representations to the cloud. The cloud aligns and fuses these extractions using geometric registration, probabilistic averaging, and version-controlled map management.

Experimental results on a self-collected three lane highway dataset achieved absolute accuracy of 0.98–1.15m RMSE ($2\sigma \leq 1.35$m) and relative inter-lane accuracy of 0.11–0.13m mean $|\Delta W|$ ($2\sigma \leq 0.18$m), satisfying commercial HD-map standards.

Beyond HD mapping, this study illustrates the broader potential of autonomous vehicle data harvesting, where the collective computational and sensing power of a fleet can be leveraged for large scale tasks.

The findings demonstrate that a camera-only, hybrid edge–cloud approach can achieve LiDAR comparable mapping precision on highway while significantly reducing the cost and cloud dependency, paving the way toward higher levels of autonomy within smarter and more sustainable transportation networks.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABS	Anti-lock Braking System
ADAS	Advanced Driver Assistance System
AI	Artificial Intelligence
AV	Autonomous Vehicle
AWS	Amazon Web Services
BSM	Blind Spot Monitoring
CLRNet	Cross Layer Refinement Network
ENU	East North Up
FoV	Field of View
FPS	Frame Per Second
FSD	Full Self Driving
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GT	Ground Truth
HD	High Definition
ICP	Iterative Closest Point
INS	Inertial Navigation System
IoU	Intersection over Union
JPDA	Joint Probabilistic Data Association
LiDAR	Light Detection Ranging
LRR	Long Range Radar

MAE	Mean Absolute Error
MCU	Microcontroller Unit
MMS	Mobile Mapping System
MRR	Medium Range Radar
NHTSA	National Highway and Transportation Safety Administration
NN	Neural Network
OEM	Original Equipment Manufacturer
OSM	OpenStreetMap
P95	95th Percentile
RANSAC	Random Sample Consensus
RMSE	Root Mean Square Error
ROI	Region Of Interest
RTK	Real Time Kinematic
SDV	Software Defined Vehicles
SoC	System on Chip
SOTIF	Safety of the Intended Functionality
SRR	Short Range Radar

CHAPTER ONE
OVERVIEW

Self-Driving vehicles aim to improve transportation by enhancing safety, efficiency & accessibility. These vehicles can carry more than 21 different sensors (i.e., cameras, radars & lidars). Based on a recent study by Lucid Motors [1], it shows that these autonomous vehicles can generate closer to 40 Gbit/s and approximately 19 terabytes per hour of data. According to the AAA American Driving Survey [2], the average American spends 60.7 minutes driving every day which is equivalent of almost 15 full days annually. Thus, at the data rates for a single vehicle could generate up to 6,650 TB of data for higher-level autonomy, in just one year. Most of the collected data usually is offloaded without any filtration to the cloud for Neural Networks (NN) training purposes that helps in improving the perception capabilities, increasing the amount of the comprehended HD map features, and in training motion planning algorithms for better decision making. Processing this amount of data requires a huge amount of memory, powerful processors, and reliable internet connection to offload the collected data to the could for processing purposes.

The availability of comprehensive and up to date HD maps is the one of the most critical systems in AV stack that is necessary to the development and implementation of autonomous vehicles. These maps provide the needed information about the vehicle's surroundings, enabling accurate localization, path planning, and decision-making [3].

Traditionally, HD maps have been created using dedicated mapping vehicles equipped with advanced sensors, such as LiDAR-based mapping which require specialized equipment, extensive manual labor, and significant resources. As a result, these methods face challenges in storage, processing, scalability, cost-effectiveness, and real-time updates [4].

Recent advancements in Autonomous Vehicles' (AVs) computer vision and computation power have created a great potential for the generation of HD maps from camera data. Feature extraction from a single front-facing camera, coupled with the availability of data in OpenStreetMap (OSM) and the onboard computation power, offer a promising avenue for generating HD maps without the need for expensive hardware. This collaborative mapping platform provides a valuable resource for enhancing map accuracy and completeness.

Motivated by the potential of leveraging the computation power of onboard supercomputers, camera data and OSM for HD map creation, this research aims to develop a novel methodology that utilizes idle autonomous vehicles as map-generating platforms. By fusing the data captured by other vehicles driving on unmapped roads and integrating it with OSM, the research seeks to overcome the limitations of traditional mapping methods and enable scalable, cost-effective, and real-time HD map creation.

This research aims to leverage onboard supercomputers in autonomous vehicles to create a distributed computing system that complements the centralized cloud resources.

1.1 Problem Statement

Industrial HD maps rely heavily on specialized mapping fleets equipped with survey-grade lidar, high-precision GNSS/INS, and extensive manual post-processing. Although this approach can reach centimeter accuracy, it remains expensive to scale and slow to refresh, especially in urban networks that change frequently. Recent reviews consistently identify update cadence, maintenance cost, and scalability as the central bottlenecks for widespread deployment.

At the same time, cloud-only paradigm struggles to ingest and process continuous, multi-gigabytes data from large fleets, especially where connectivity is intermittent or costly. A hybrid (end-device ↔ cloud) processing and compressing relevant features on-vehicle, then fusing across vehicles in the cloud, offers a promising way to reduce bandwidth, contain cost, and increase the availability.

1.2 Motivation

In recent years, the technology growth in Autonomous Vehicles (AVs) allowed these vehicles to be more capable of navigating in complex environments autonomously. With these capabilities, there are increasing needs for accurate and up-to-date HD maps that provide detailed information about the surrounding environment. However, Automotive manufacturers (OEMs) investing in AVs development, face significant expenses in getting HD maps from third-party suppliers (i.e., Here, TomTom, Ushr) and in maintaining these maps for their fleets.

Moreover, many OEMs have large fleets of vehicles equipped with the necessary sensors and computing capabilities to create HD maps autonomously. These vehicles spend most of their time idle or in low-demand situations, which unlocks new opportunity to leverage their onboard sensors and computers for map creation without additional investment in dedicated mapping infrastructure. This enables OEMs to utilize their existing fleet resources for map creation.

Overall, this research seeks to advance the state-of-the-art in HD map creation by developing a scalable, cost-effective, and collaborative approach that leverages idle autonomous vehicles by integrating visual data with OSM data. Through a combination of innovative algorithms, sensor fusion techniques, and cloud-based processing, the goal is to define a new approach for distributed and crowd-sourced HD map creation that supports the autonomous driving technology.

1.3 Research Objectives

The proposed work is aimed to enhance my expertise in HD map generation techniques and cloud computing. It also aims to help me acquiring hands-on experience with implementing HD map generation algorithms, visual feature extraction, and map integration with a deep dive into distributing these tasks over multiple vehicles. In addition, the results of this work will provide a new technique to utilize the idle AVs by using them to generate a cost-effective HD map that can be adopted by OEMs to use in their vehicles for localization and navigation purposes.

This Thesis will cover the following main points:

- Conduct an in-depth literature review of existing research in cloud computing, HD map generation techniques, ground truthing, computer vision, and map integration with (OSM).
- Preprocess the collected datasets to ensure compatibility, synchronization, and alignment.
- Use algorithms for visual feature extraction from camera images, including key points, edges, and road surface paintings relevant to map creation.
- Correlate GPS data with road segments in the OSM database and develop algorithms for aligning visual features extracted from camera data with OSM road networks and infrastructure, then fuse and integrate the extracted features into the generated HD map.
- Demonstrate the collaborative map generation by fusing the collected data from end devices.

1.4 Thesis Contributions

The contributions of this research work can be summarized in the following:

- Demonstrates that lane-line extraction with CLRNet, followed by OSM map alignment, can generate HD useful map layers suitable for AV localization and navigation without relying on survey fleets while improving update cadence at reasonable cost. This approach retains the semantics that AV stacks need (e.g., lane boundaries, lane counts, and road attributes).
- Implement a hybrid end-device to cloud architecture that leverages idle onboard compute to pre-process lane features before upload and then use the cloud for multi-vehicle data aggregation. This division of tasks reflects edge-computing design principles for autonomous vehicles and addresses concerns about bandwidth, latency, and robustness in cloud-only pipelines.

The expected contribution for industry is a cost-effective pipeline that OEMs could adopt.

1.5 Thesis Organization

This thesis is organized into five chapters as follows:

- Chapter 1 – Overview: Introduces the background, motivation, research problem, objectives, and contributions of the study.
- Chapter 2 – Literature Review: Reviews existing work in HD map creation, distributed computing for autonomous vehicles, and OSM-based map fusion.
- Chapter 3 – Methodology: Describes the proposed hybrid end-device-to-cloud architecture, the data collection process, feature extraction techniques, and the algorithms used for OSM alignment and fusion. It also explains the dataset, hardware and software configurations, system design, and parameter settings used for implementation.
- Chapter 4 – Results and Discussion: Presents and analyzes the experimental results, evaluating the proposed system in terms of accuracy and performance compared to traditional approaches.
- Chapter 5 – Conclusion and Future Work: Summarizes the main findings, discusses limitations, and outlines directions for future research.

1.6 Assumptions

The following assumptions are made to define the operational and technical feasibility of this research:

- Each vehicle is equipped with at least one front-facing camera, and industrial GNSS sensor capable of generating synchronized and timestamped data. Vehicles have onboard computing units (SoC or MCU) capable of performing local preprocessing tasks when not in use by the ADAS stack (i.e., idle).
- Camera calibration parameters and sensor alignments are known, fixed, and remain stable throughout data collection.
- Sufficient network connectivity is available to allow data synchronization between the vehicle and the cloud.
- The OpenStreetMap (OSM) base data is assumed to be topologically correct and rich enough to support feature alignment and semantic labeling.

1.7 Scope and Limitations

1.7.1 Scope

This research focuses on the development and evaluation of a camera-based HD map generation methodology built within a hybrid edge–cloud computing architecture. The scope is limited to using vision-based data (camera imagery, GPS) combined with OpenStreetMap as the external reference. The system is evaluated primarily in terms of mapping accuracy and completeness, demonstrating feasibility for distributed HD map creation without dependence on LiDAR sensors.

1.7.2 Limitations

While the proposed framework offers a promising low-cost alternative, several limitations exist:

- Lane detection accuracy may decline in adverse lighting or occluded environments (i.e., rain, shadows, night).
- GPS drift and camera calibration errors may introduce localization offsets that affect the transformation of perceived lane markers into global coordinate system.
- The current implementation simulates a hybrid processing but does not include real-time vehicle-to-cloud communication or fully distributed networking.
- Experiments are conducted using a limited dataset collected under specific conditions; generalization to all geographic regions or road types is outside the present scope.

CHAPTER TWO
LITERTAURE REVIEW

This chapter reviews the foundations of ADAS and autonomous vehicle technologies, highlighting the growing role of HD maps as a crucial enabler for safety, localization, and decision-making. It examines traditional HD map generation methods, emerging camera-based and crowdsourced approaches, and the evolution of hybrid edge–cloud computing paradigms for large scale data processing.

The literature demonstrates that while significant advances have been made, challenges remain in cost, scalability, and real-time update mechanisms. These challenges define the motivation and context for the methodology proposed in this research.

2.1 Advanced ADAS and Autonomous Vehicle Systems

2.1.1 ADAS vs. Full Autonomy and the Role of HD Maps

Advanced Driver Assistance Systems (ADAS) and Autonomous Vehicles (AVs) represent the evolution of modern transportation technologies aimed at improving safety, efficiency, and accessibility [5]. According to National Highway and Transportation Safety Administration (NHTSA), there are 5 levels of driving automation: Level 0 - No driving automation. Although, the vehicle may have assistive or warning features like ABS and Blind Spot Monitoring (BSM), Level 1 - Driver assistance, Level 2 - Partial automation, Level 3 - Conditional automation, Level 4 - High automation, and Level 5 - Full automation. Figure 1 below shows the details of these levels based on SAE.

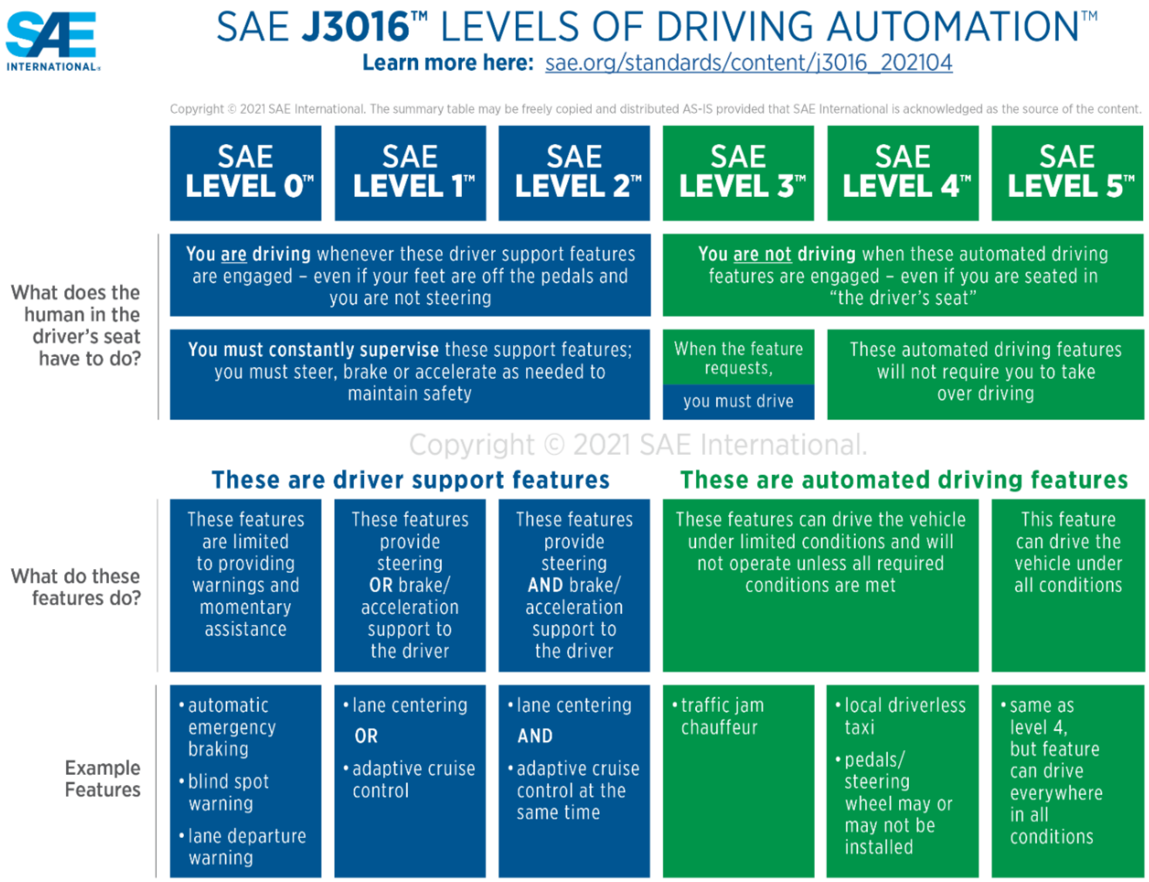

Figure 1 SAE Levels of Driving Automation

ADAS typically operates at SAE Levels 1-2, providing limited driver assistance such as adaptive cruise control, lane keep assist or collision avoidance, whereas full autonomy (SAE Levels 4-5) requires the vehicle to perform all driving functions under defined conditions without human intervention as. As vehicles advance toward higher levels of autonomy, the dependency on environmental perception and mapping grows significantly.

Autonomous vehicles (AVs) utilize a combination of sensors, cameras, radars (i.e., SRR, LRR) surrounding the host vehicle, LiDAR and GPS that are used to perceive the surroundings and localize the host vehicle. Then a powerful onboard computing hardware (MCUs and SoCs) is used to make real-time decisions to navigate safely in any environment.

The key components of self-driving cars include as can be seen in Figure 2:

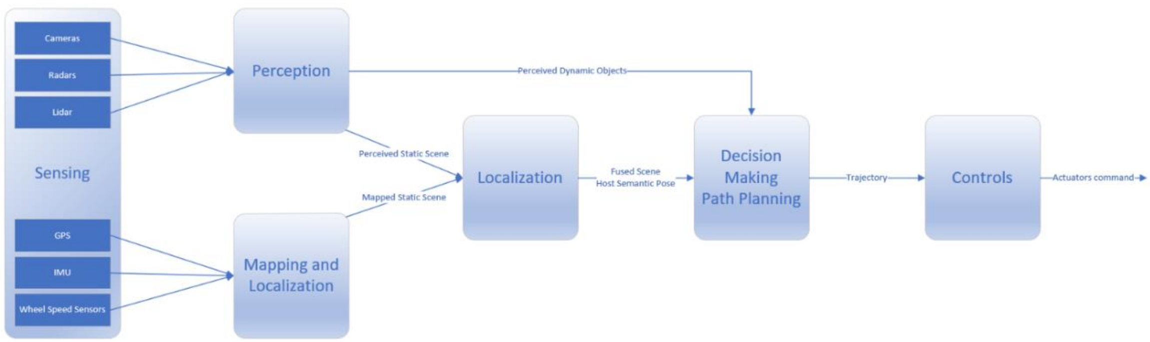

Figure 2 High level components of AVs stack

1- Sensing System: Self-driving cars are equipped with different types of sensors to perceive the environment, including:

   a. Cameras: Fisheye, Narrow FoV and Wide FoV cameras are used to capture visual information about the surroundings, including the static scene (i.e., lane markings, road edges and traffic signs) and dynamic scene (i.e., surrounding vehicles, pedestrians, and cyclists).

   b. Light Detection and Ranging (LiDAR): transmits low-power, eye-safe laser pulses to measure distances to objects of interest and create detailed 3D map of the environment.

c. Radars: Long-Range Radars (LRR), Mid-Range Radars (MRR) and Short-Range Radars (SRR) in Figure 3 use radio waves to determine the speed and distance of the detected objects, including vehicles, pedestrians, and obstacles.

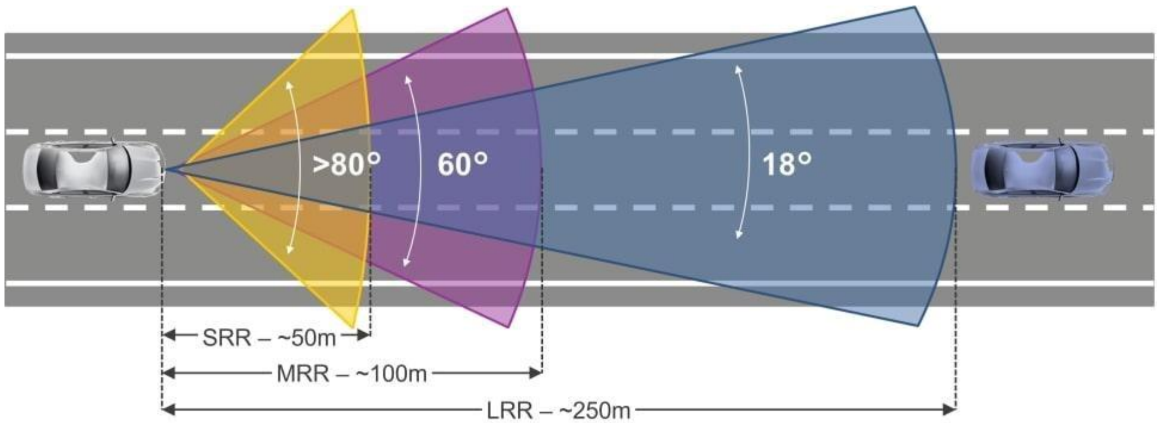

Figure 3 Automotive Radar Categories

d. Global Positioning System (GPS): Provides precise location and navigation information.

2- Onboard Computing hardware: Powerful onboard computers (SoCs and MCUs) that process all the sensor data, run complex algorithms to fuse the sensors data, and make real-time decisions to control the vehicle's actuators.

3- Perception: Responsible for fusing the different sensors data to generate a perceived dynamic and static scene, track multiple objects, and predict their location.

4- Mapping and Localization: Self-driving cars are heavily dependent on high-definition (HD) maps to accurately determine their semantic pose (position and orientation) relative to the surrounding environment. HD maps provide detailed information about road geometry, lane markings, traffic signs/lights, and landmarks, enabling precise localization. In localization stack, the perceived static scene is fused with the mapped scene to generate more accurate and detailed environment which is helpful in minimizing the error generated by outdated HD maps or inaccurate sensor detections.

5- Decision Making and Path Planning: A critical system in self-driving cars. These processes involve determining the optimal trajectory and making real-time decisions to navigate safely and efficiently in complex environments.

6- Control Systems: Control vehicle motion consumes the trajectory generated by decision making system and blindly manage vehicle dynamics, including acceleration, braking, and steering. These systems ensure smooth control of the vehicle's movements by applying filter and limiters to eliminate lateral and/or longitudinal jerk.

In ADAS, maps are mainly used for route planning and route information. However, in higher autonomy levels, high-definition (HD) maps become a critical sensor layer, supporting precise localization and decision-making. These maps provide centimeter level geometric and semantic details of the road network, including lane boundaries, curbs, crosswalks, and traffic signs. They supplement real-time data by offering prior knowledge of the road environment, hence improving safety, reducing computational load, and enabling predictive control in complex scenarios [4].

The advancements in hardware technology and software techniques added more driving capabilities to lower levels of autonomy. Over the past decade, the size of software growth is a factor of 10 every 5 – 7 years. This growth mandates the manufacturers in the automotive industry to become more software-oriented [6] and opened the door for the Software Defined Vehicles (SDV) concept.

2.1.2 Mapping as a Sensor and Enabler for Safety

Recent literature defines HD maps function as virtual sensors, providing information unavailable to onboard sensors [7]. They store static environmental elements such as road geometry, lane topology, traffic light positions, and regulatory information.

In recent years, companies such as Tesla have introduced hands-free driving systems like Full Self-Driving (FSD), which are designed to operate without relying on HD maps. Instead, Tesla's end-to-end AI system depends solely on advanced vision sensors capable of detecting objects up to 300 meters away. However, based on an independent study using a Tesla Model 3 for benchmarking, it was observed that the system struggles with complex scenarios such as intersections and roundabouts due to the absence of prior lane connectivity information, the data typically provided by HD maps. This limitation not only reduces driving efficiency and electric vehicle range (a major source of driver anxiety) but also causes a noticeable decline in FSD performance under poor weather or low-visibility conditions.

Figure 4 below shows how the FSD could not traverse the intersection because it could not see the far end of the intersection.

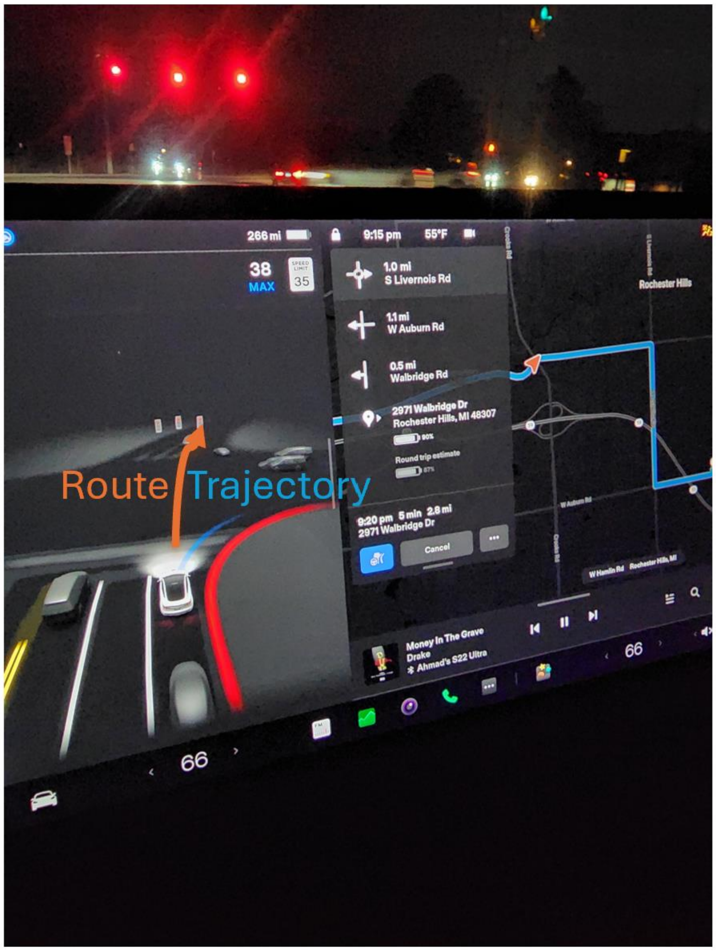

Figure 4 Tesla Model3 Mismatch between route and trajectory

The prior knowledge in HD map enhances the reliability of perception, especially under challenging conditions such as occlusion, low light, or sensor noise. Studies have demonstrated that fusing map data with sensor observations can significantly reduce localization error and improve redundancy, supporting functional safety requirements in automated driving systems [7] , [8].

Figure 5 below shows the timeline of the major hands-free ADAS system in the US market and the role of HD maps in these systems.

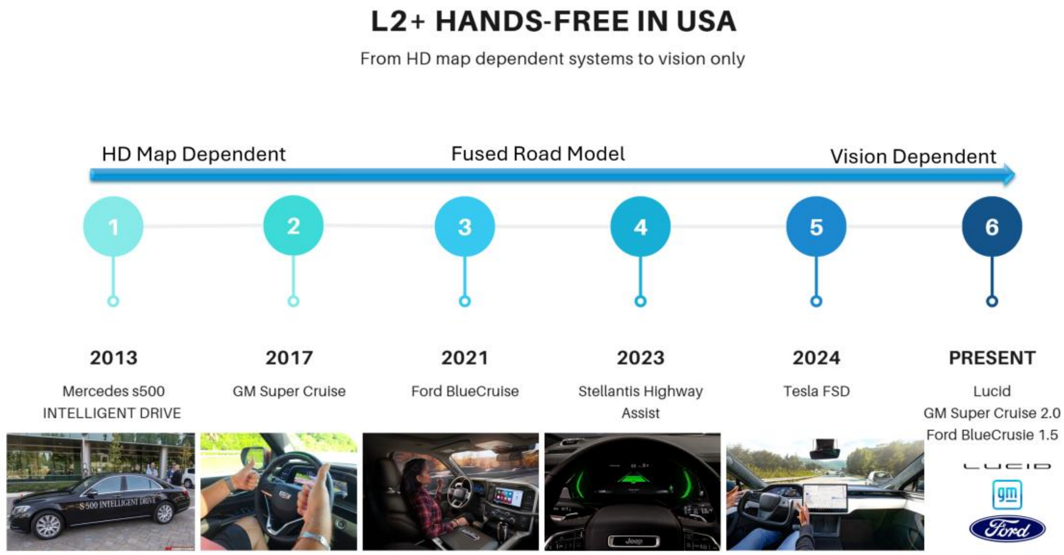

Figure 5 Major L2+ Hands-Free systems and HD map dependency

Functional safety frameworks such as ISO 26262 and SOTIF (Safety of the Intended Functionality) recognize map accuracy and freshness as essential components of system integrity, especially for higher autonomy levels (i.e., Level 3 and Level 4) [9]. Inaccurate or outdated maps can lead to degraded localization, unsafe lane changes, or misinterpretation of traffic controls. Thus, HD map quality directly affects overall vehicle safety performance [4].

2.1.3 Challenges in Modern ADAS and AV Data Pipelines

Despite their benefits, AV systems face multiple challenges in managing the volume, velocity, and variety of data generated by onboard sensors. A single vehicle equipped with multiple cameras, radar, and LiDAR sensors can produce tens of gigabits per second of raw data, leading to terabytes per hour of recordings which challenges processing and storage infrastructures [10].

Traditional centralized cloud systems struggle to process such massive and continuous data streams due to latency, bandwidth constraints, and cost [11]. This motivates the shift toward distributed or hybrid architectures where preprocessing is conducted on the vehicle (edge) before data is transmitted to the cloud for aggregation and learning [12]. Additionally, scalability and update cadence remains as an open challenge where HD maps must be refreshed frequently to reflect construction, lane changes, or new infrastructure, but traditional pipelines may take weeks or months to update [13].

2.2 High-Definition (HD) Map Creation: Methods and Challenges

Map creation suppliers like TomTom and Ushr have created HD maps for different operating domains and purposes. However, these maps are not open-sourced and only used for commercialization and for internal research purposes. That explains the reason behind the lack of datasets and research articles availability. Due to the reasons above, this section will be summarizing the state-of-the-art techniques for HD map creation.

2.2.1 HD Map Definition and Layered Structure

HD maps differ from navigation maps or standard definition maps in both resolution and attributes. While navigation maps provide road centerlines and attributes adequate for turn-by-turn navigation, HD maps contain centimeter-level accuracy and detailed semantic layers describing road geometry, lane topology, markings, traffic signs, and roadside objects [14]. Figure 6 below shows some of the HD map attributes.

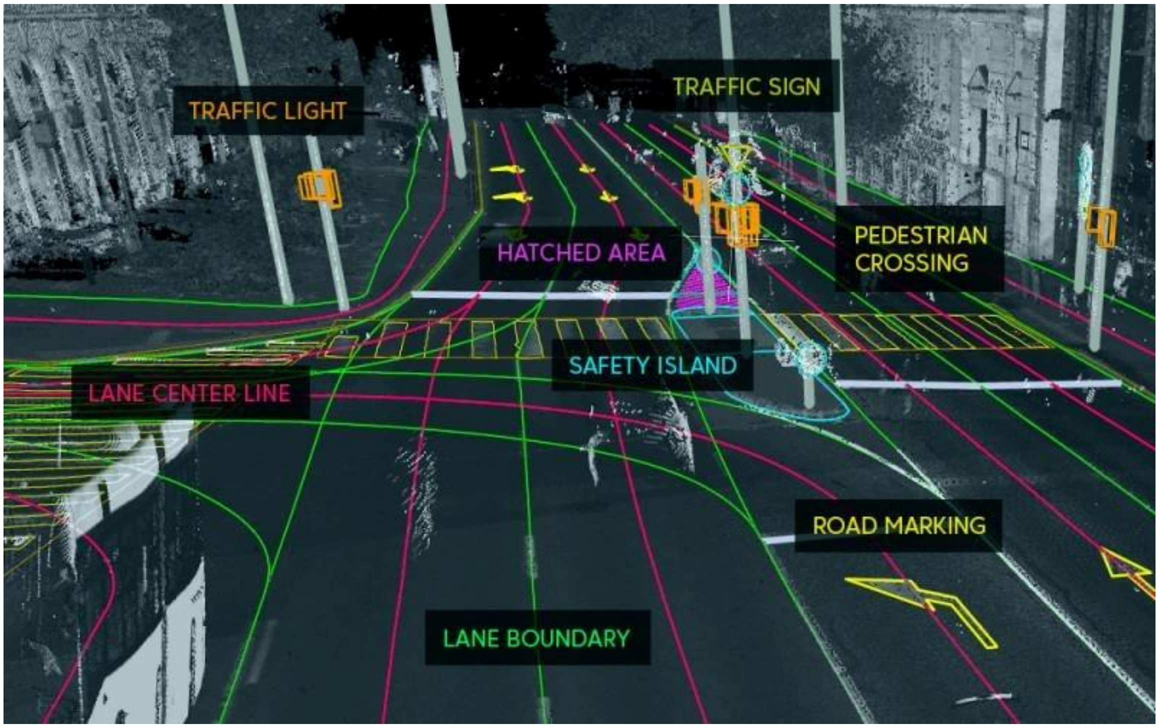

Figure 6 HD map generated by aiMotive

In 2010, Mercedes-Benz introduced the concept of HD maps that was used in the 2013 historic "Bertha Benz Memorial Route" event, where a 2013 Mercedes s500 INTELLIGENT DRIVE with production sensors, was able to drive from Mannheim to Pforzheim, Germany in fully autonomous mode by utilizing a combination of vision, radars and "digital road maps". The Bertha Benz Memorial Route has around two thousand lane segments which were annotated manually [15].

Figure 7 below shows the evolution timeline of maps and their specs and applications.

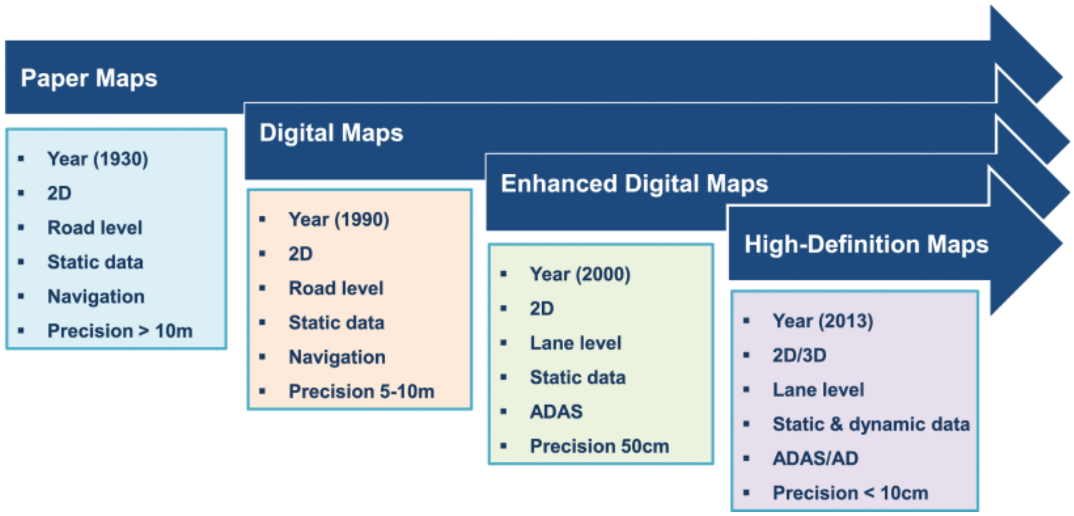

Figure 7 Evolution timeline of maps

HD maps are typically structured in hierarchical layers. Breaking down HD maps into layers allows to have a more structured data representation of the road environment, it makes the HD map easier to build, store, retrieve and update [16]. In the automotive industry, there are multiple standards in constructing HD maps, the most common adopted structure contains three layers as can be seen in Figure 8, this structure assures centimeter-level localization accuracy, lane-level geometric and semantic information availability [7].

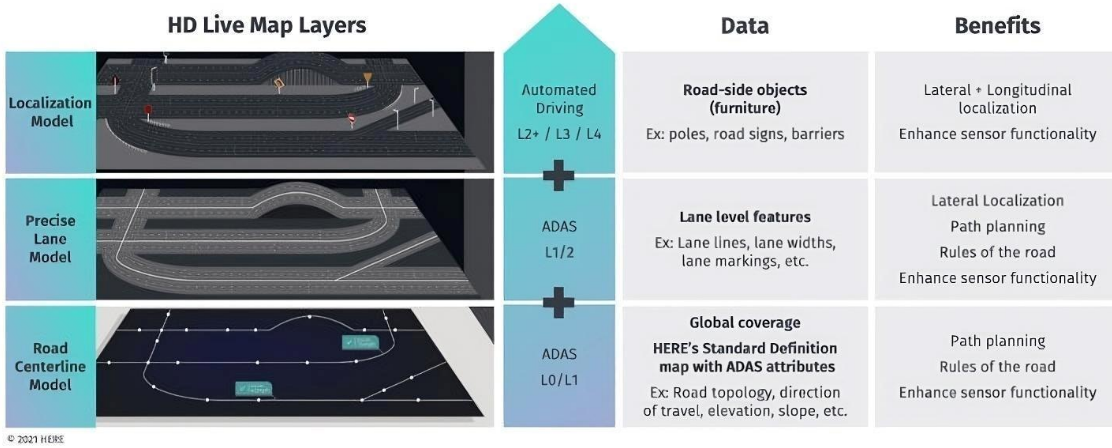

Figure 8 HD map layers by HERE

- Road Model Layer: Like google maps, it defines the topology, direction of travel, intersection information, elevation, and speed limits per road. Usually, this layer is used for path planning (navigation) and turn to turn directions.
- Precise Lane Model Layer: This layer defines the features per lane, such as the lane width, lane lines, lane adjacency, speed limit per lane and lane connectivity in complex intersections/roundabouts. It is the main layer for lateral localization and automatic lane changes, it is also the only input for the autonomous vehicle to traverse an intersection.
- Localization model layer: A rich semantic information layer. This layer contains traffic control devices information such as traffic lights and traffic signs. This layer plays a key role in localizing the ego vehicle laterally and longitudinally, especially in the urban area, which is rich in features.

The representation of HD maps often takes the form of vector-based lane graphs or semantic topological networks, supporting efficient indexing and map-matching [16]. Key attributes such as accuracy (≤ 10 cm), completeness, and freshness determine their usability for autonomous driving applications [17].

2.2.2 Traditional HD Map Creation Workflows

Building HD maps is a complicated process that can be separated into multiple steps. The first step in the procedure of building an HD map is “data collection”, where a specialized vehicles equipped with a suite of high-end and well-calibrated sensors to collect data about the environment. These vehicles are usually equipped with a very accurate GPS as Real-Time Kinematic (RTK) positioning system with a very high accuracy (few centimeters of error), high-resolution LiDARs and cameras to collect 2D and 3D data.

This is a very critical step since the point cloud map generation and feature extraction methods heavily rely on the quality of the collected data [6]. Defining the sensors parameters and calibrating the cameras for precise data collection can be very challenging [18], to overcome this issue, several manufacturers provide the whole setup that contains a farm of sensors in one package, called mobile mapping system (MMS) which is easier to install and use. Figure 9 below shows the sensor farm of MMS from HERE maps.

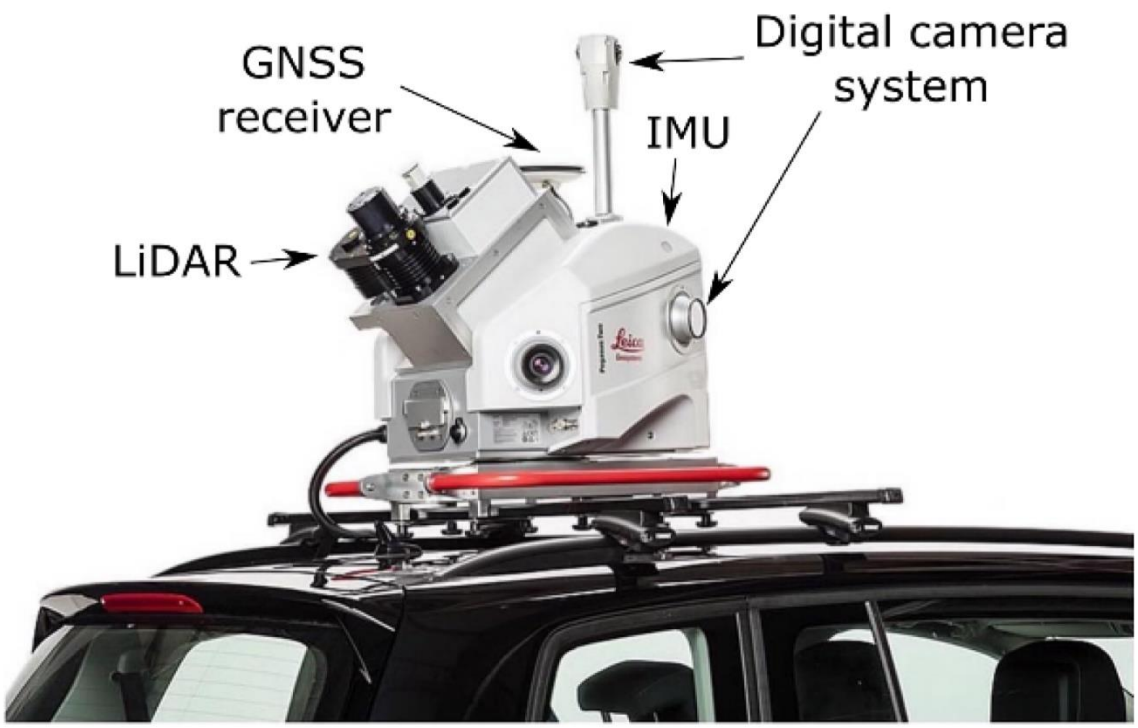

Figure 9 The sensor suite of MMS

In [19], Elhashash et al. discussed the variant of sensors used to build MMS and reviewed their uses and applications. Data collection methods are summarized in table 2.1 below:

Table 2.1: HD-Map Data Collection Methods

	Crowdsourcing	**Open-sourced dataset**	**Self-collected dataset**
Advantages	Low cost, Efficient, Up-to-date data, Different environments, Data size.	High quality, Timesaving in data collection, Data is organized and does not require post-processing	Customizable data collection (specific scene/environment, sensor types, data size)
Disadvantages	Requires multiple drives to reach the required coverage and accuracy	Data may not be up-to-date, and limited types of environments or scenarios	Data collection for large-scale maps is time-consuming, data may not be up-to-date, and small mapping coverage, Hardware can be expensive, Sensor calibration.

The second step in building HD maps is “point cloud map generation”, where in this step a collected data is pre-processed (down sampled, noise reduction) then fuse (LiDAR, Cameras, GPS) using Kalman filter to generate an initial HD map.

This step can be completed online or offline. In online approach, the data is collected and processed on the fly to build the HD map, this approach requires very powerful storage and computation resources. The vehicle then offloads the completed HD to the cloud. On the other hand, in offline approach, the data is collected for the whole trip without any filtration, then the vehicle with MMS offloads all the data to the cloud for processing purposes.

Once the point cloud map is generated, the third step is to “extract the features for HD map”, in [20] L. Ma reviewed the different methods and algorithms used to build HD maps and extract road features from the point clouds. In this step, the HD map becomes more informative by extracting the lane information, and landmarks. Either manual extraction or machine learning extraction is used in this step. Figure 10 illustrates the complete process to build HD maps.

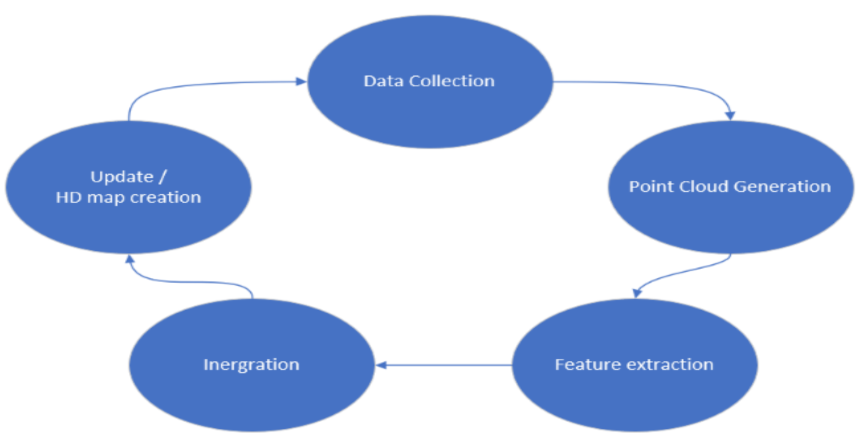

Figure 10 HD map building process

While ensuring high accuracy, this approach is expensive and time-consuming. It requires specialized hardware, manual post-processing, and frequent recalibration. The scalability is limited by the number of mapping vehicles, and the update frequency is constrained by logistics and weather conditions. In rapidly changing environments such as urban roads or construction zones, maps may quickly become outdated [21]. Recent studies highlight the HD map update cadence problem as one of the largest barriers to widespread HD map deployment [22].

2.2.3 Vision-Based and Crowd-sourced HD Map Generation

The increasing computational power of automotive-grade processors and advances in computer vision have enabled camera-only HD mapping. Deep learning models such as CLRNet [23], LaneATT [24], and Gen-LaneNet [25] extract lane boundaries and semantic road features directly from pre-collected video data. When fused with vehicle pose estimates (from GNSS/IMU), these features can be aggregated to reconstruct lane-level geometry.

Crowdsourced mapping leverages data collected from a fleet of vehicles operating under normal driving conditions. This paradigm transforms every equipped vehicle into a mapping node, continuously contributing observations. As multiple vehicles traverse the same road, the fused data improves coverage, accuracy, and completeness through statistical convergence. The inclusion of open geographic databases such as OpenStreetMap (OSM) provides an additional source of prior geometry and metadata, facilitating alignment and validation of extracted features [26].

In [27], W. Jang proposed a new method to construct an HD-map using monocular camera without further human labors. However, their approach requires manual alignment to the global positioning. They used an odometry information from the vehicle to generate the map which introduced a significant error in the generated map due to sensor drift.

Y. Zhue in [28] proposed an automatic lane-level mapping approach using semantic-particle filter, achieving accurate and robust reconstruction of HD-maps in urban scenes. They used a LiDAR to achieve the required accuracy which is not always available in production vehicles.

Crowdsourced mapping has its own challenges in ensuring geometric consistency, handling variable camera calibrations, and mitigating environmental effects such as lighting, shadows, or weather. Furthermore, synchronization between vehicles and temporal coherence of map updates remain active research areas.

2.2.4 HD Map Maintenance and Scalability

HD-maps can be easily created in one visit by sensor-rich vehicles using RTK, LiDAR and high-resolution cameras. Commercial mapping suppliers use limited number of vehicles to generate HD maps, which limits the updates to maintain any changes in the environment including detours, direction of travel, and placement/removal of semantic features, which is a major challenge for a large-scale map. This limited number of the specialized vehicles used for creating HD maps limits the frequency of visits for each road and changes cannot be incorporated into the map in time. Usually, map suppliers usually update their database every 6 to 12 months. Wrong decisions can be taken by the self-driving vehicles due to errors in HD maps which can lead to severe damages [29].

J. S. Berrio in [30], reviewed the 2 standard approaches to implement the map maintenance processes: the cumulative adjustment and the transformative adjustments. The cumulative adjustment preserves the original features and keeps adding new features to the map causing it to grow without a limit. On the other hand, the transformative adjustment modifies the initial map structure by adding new features and deleting features that no longer exist.

Identifying the differences between the initial map and the newly collected data, and then integrating these differences to update the different layers of the map can be very challenging [31]. Change detection algorithms can be adapted to identify the variations between the collected data and the base layer of the initial map (point cloud).

An alternative and efficient way to this, is crowdsourcing methods by OEMs, where regular vehicles can continuously and frequently contribute to update changes to mapped areas [32]. This method introduces a new layer to the HD map that is called crowdsourcing layer. However, OEMs are still not clear how to take advantage of this layer by integrating it with the existing HD maps provided by the suppliers. Figure 11 below shows a summary for HD Map change detection from crowdsourced data.

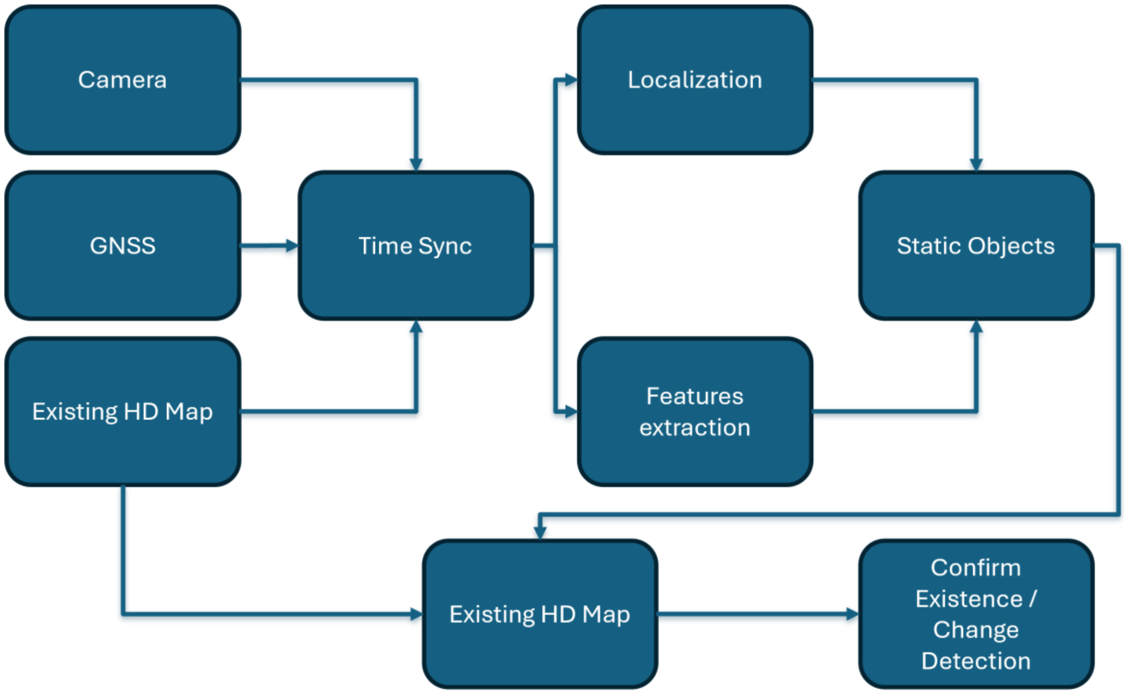

Figure 11 Map change detection from crowdsourced data

Existing map providers often adopt a cloud-centric model, where data is uploaded to centralized servers for processing. While effective, this approach suffers from bandwidth limitations, high operational costs, and latency. For OEMs managing large fleets, such reliance on centralized infrastructure becomes financially and technically restrictive. These limitations have enabled the emergence of hybrid edge–cloud architectures that balance computational load between onboard devices and the cloud [33].

2.3 Edge–Cloud (Hybrid) Architectures for Autonomous Systems

2.3.1 Edge Computing in the Autonomous Vehicle Context

Edge computing refers to the practice of performing computation closer to the data source rather than in distant cloud servers. In the context of AVs, this means utilizing onboard processors or roadside units to perform low-latency tasks such as perception, preprocessing, and decision making [31].

In a mapping context, edge nodes (vehicles themselves) can perform feature extraction, and data compression before transmitting essential information to the cloud. This offloading strategy conserves communication resources and protects data privacy, as raw sensor data (which may include personal information) remains local [34].

2.3.2 Cloud Computing for Large-Scale HD Mapping

Cloud computing provides unlimited storage and processing capabilities, making it essential for large-scale map aggregation and analysis. The cloud layer typically performs tasks such as global data fusion, semantic consistency checking, model training, and map versioning. Distributed cloud platforms (e.g., AWS, Azure, GCP) support scalable pipelines capable of ingesting petabytes of data from fleets of connected vehicles [35].

However, fully cloud-centric architectures introduce significant latency and depend heavily on continuous connectivity. In scenarios with poor network coverage, data upload delays can slow map updates and degrade responsiveness. Moreover, data transfer costs become prohibitive when dealing with high-volume sensor streams. These limitations motivate a hybrid integration of cloud and edge resources [36].

2.3.3 Hybrid Edge–Cloud Architecture for HD Map Generation

A hybrid architecture combines the strengths of both paradigms: computation at the edge for immediate processing and in the cloud for large-scale aggregation. In this model, vehicles act as intelligent nodes capable of local feature extraction. The cloud serves as the coordination layer, integrating information from multiple vehicles to generate a unified, validated HD map [37].

Typical workflows involve three stages:

- Edge Stage: Data collection, feature extraction (lanes, road markings), and compression performed locally on the vehicle's onboard computer.
- Cloud Stage: Fusion of edge-processed data from multiple vehicles, alignment with OSM or existing HD map layers, and consistency verification.
- Distribution Stage: Dissemination of updated map tiles or deltas back to vehicles for real-time usage.

Figure 12 below show a high-level diagram of the complete workflow.

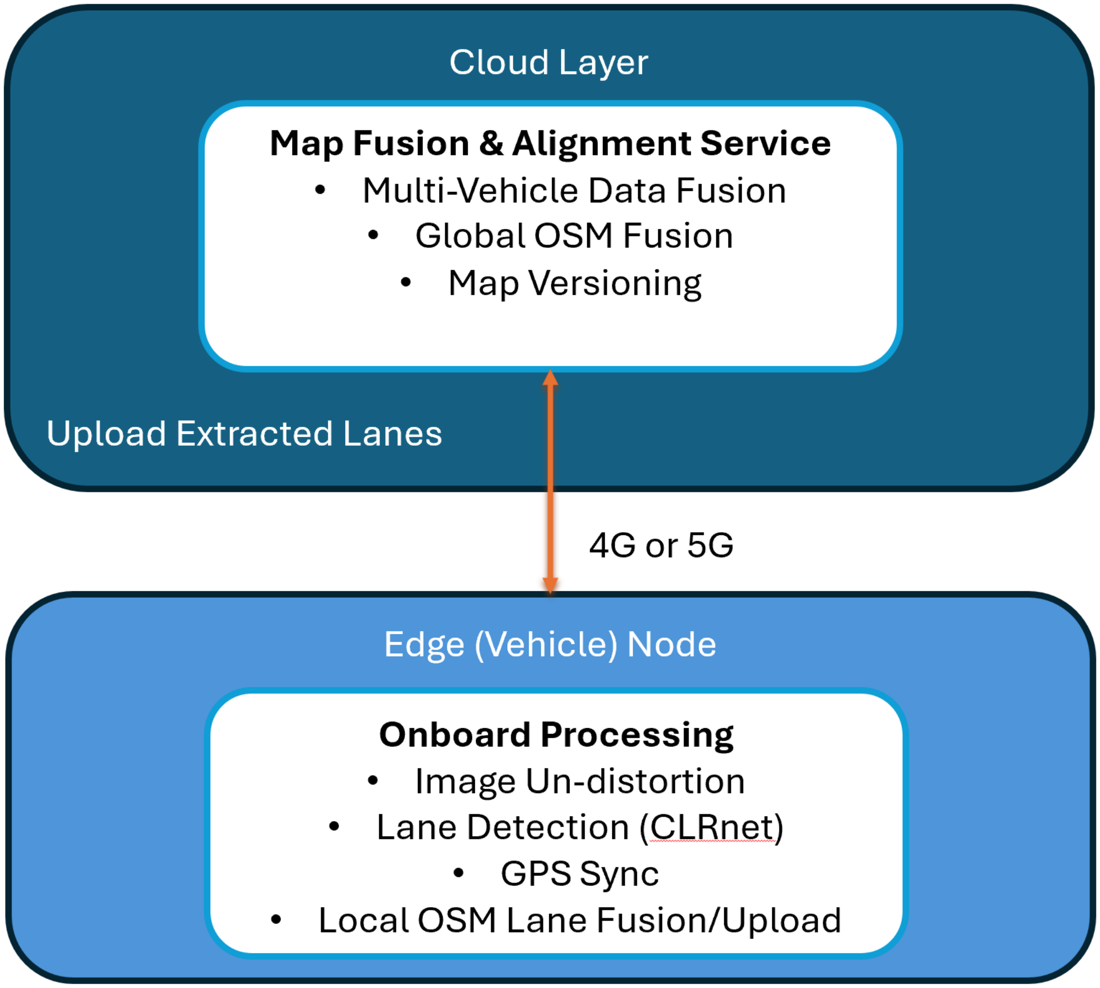

Figure 12 High-level diagram of the complete workflow

This hierarchical structure reduces bandwidth consumption, enables near-real-time updates, and allows the system to scale across fleets. It also enhances robustness: if connectivity is temporarily lost, vehicles can continue collecting data and cache it locally until the connectivity is restored [31].

2.3.4 Technical and Operational Challenges

Despite the potential of the hybrid edge – cloud approach, it introduces new challenges. Synchronization and coordination between thousands of vehicles require robust communication protocols and version control mechanisms. Security and privacy concerns arise from sharing sensor data that may include personal or location-sensitive information [38]. Resource management which determines which tasks execute locally versus in the cloud remains a complex optimization problem dependent on network conditions and computational capacity [37].

Furthermore, standardization across manufacturers is still lacking, leading to non-standard implementations. Reliable evaluation frameworks are also needed to assess accuracy, bandwidth savings, and latency across real-world deployments [39]. Nevertheless, the hybrid paradigm is widely viewed as a foundation for scalable and cost-efficient autonomous driving ecosystems.

CHAPTER THREE
METHODOLOGY

This chapter details a hybrid edge–cloud pipeline that produces HD-map lanes from camera-only fleet data, aligns/fuses them with OpenStreetMap (OSM), and distributes updates back to vehicles. At the edge (vehicle), we run lightweight preprocessing and lane-feature extraction using CLRNet then combine the extracted lane lines with OSM map attributes (e.g., lane count, current lane ID based on host trajectory). The cloud layer performs multi-vehicle fusion, Global OSM alignment, versioning, and tile distribution. Figure 9 below shows the hybrid edge – cloud pipeline. The approach is consistent with recent connected vehicle edge–cloud frameworks emphasizing bandwidth reduction, low latency, and scalability.

Section 3.1 presents the system architecture. Section 3.2 details the data collection and preprocessing. Section 3.3 covers map alignment and fusion with local OSM. Section 3.4 describes the implementation of the hybrid edge–cloud pipeline to aggregate multi-vehicle observations with the global HD map. Section 3.5 summarizes the system implementation, and finally, section 3.6 covers the chapter summary.

3.1 System Architecture

The proposed architecture (Figure 13) consists of three primary components:

- Edge-vehicle nodes: Each autonomous vehicle in the fleet serves as a local compute node when idle (or during low-demand periods). Onboard sensors (cameras, GPS) capture environmental data. A lightweight preprocessing module runs onboard to process images from the front facing camera, synchronize the timestamps between camera frames and GPS frames and then extract the lane features. Only compact features (e.g., lane polylines with confidence, current lane ID and host trajectory) are queued for upload to reduce bandwidth. This follows edge-computing best practices for Avs where the compute happens close to data, transmit summaries, not raw streams.
- Cloud aggregation layer: Preprocessed features from multiple vehicles are uploaded to the cloud. Global fusion, OSM alignment (e.g., Map Matching) [40], version control happens here.

  $$Mglobal = f(Mlocal1, Mlocal2, \ldots, MlocalN) \quad (3.1)$$

  Where $Mlocal$ denotes the local HD map fragment uploaded by vehicle i, and f represents the fusion and alignment process.
- Distribution and update layer: The resulting HD map tiles are pushed back to the vehicle for localization purposes. The update cadence is managed to support real time refresh.

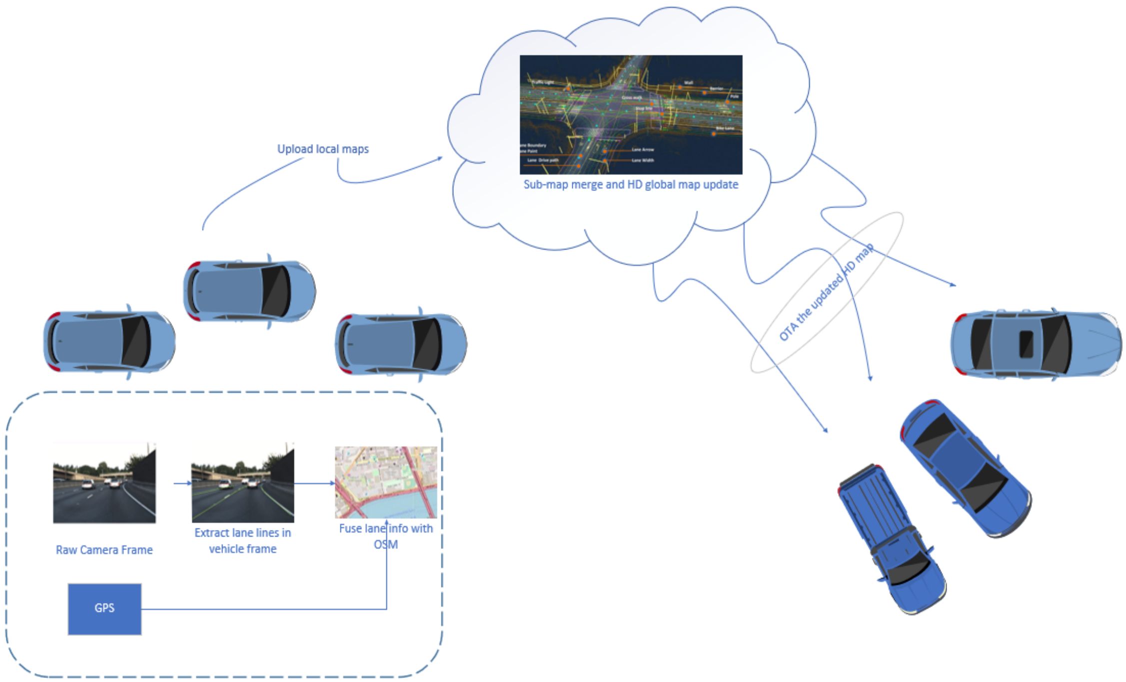

Figure 13 Complete framework design showing the tasks on crowdsourcing vehicles and on the cloud.

3.2 Data Collection and Preprocessing on the Edge

The data capture platform consists of a GoPro Hero9 camera (narrow field of view) [41] equipped with an integrated GPS sensor. Video recordings were captured at a resolution of 1920 × 1440 pixels and a frame rate of 30 FPS, while the GPS module logged position data (latitude, longitude, altitude) at 18 Hz.

3.2.1 Camera Calibration

Camera parameters include intrinsic, extrinsic, and distortion coefficients. These parameters are crucial for correcting lens distortion, measuring object dimensions in real-world units, and determining the camera's position and orientation within a scene [42].

The pinhole camera model relates 3D world coordinates (X,Y,Z) to 2D image coordinates (u,v) as follows:

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K[R\ t]\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.2}$$

where:

- s is the scaling factor,
- K is the intrinsic matrix,
- R & t represent rotation and translation (extrinsic).

The intrinsic matrix K is defined as:

$$K = \begin{bmatrix} fx & 0 & cx \\ 0 & fy & cy \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

where fx , fy are focal lengths in pixels, and cx , cy denote the principal point.

To estimate these parameters, a set of known 3D world points and their corresponding 2D image points is required. These correspondences can be obtained by capturing multiple images of a calibration pattern such as a checkerboard. Once the point pairs are established, the parameters can be computed, enabling accurate alignment between camera projections and real-world geometry [43]. Figure 14 illustrates the main elements involved in the calibration process.

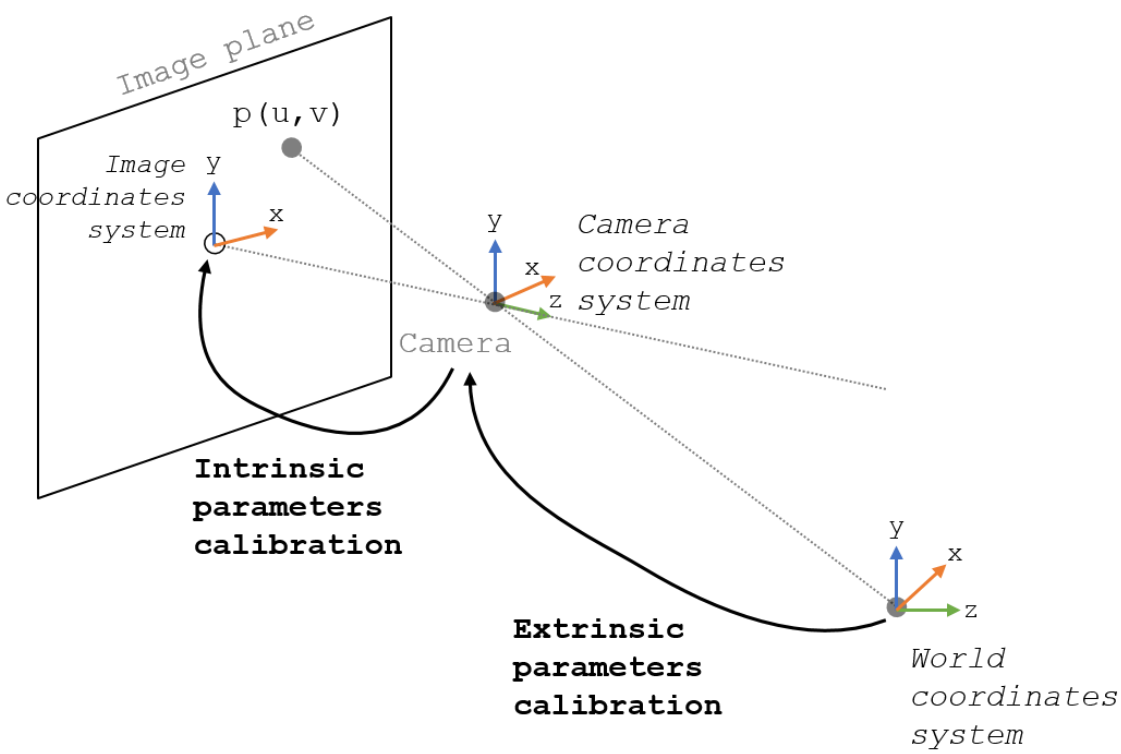

Figure 14 Elements involved in a camera calibration

In this work, the **estimateMonoCameraFromScene** function in MATLAB [44] was used to determine the camera's intrinsic parameters based on scene geometry. The function requires the pixel coordinates of a trapezoid within the image (as shown in Figure 15), along with the real-world dimensions of the corresponding rectangle: a lane width of 3.65 m and a lane marker length of 4.0 m (Figure 16).

Figure 15 Pixel coordinates of a trapezoid in the scene image for calibration

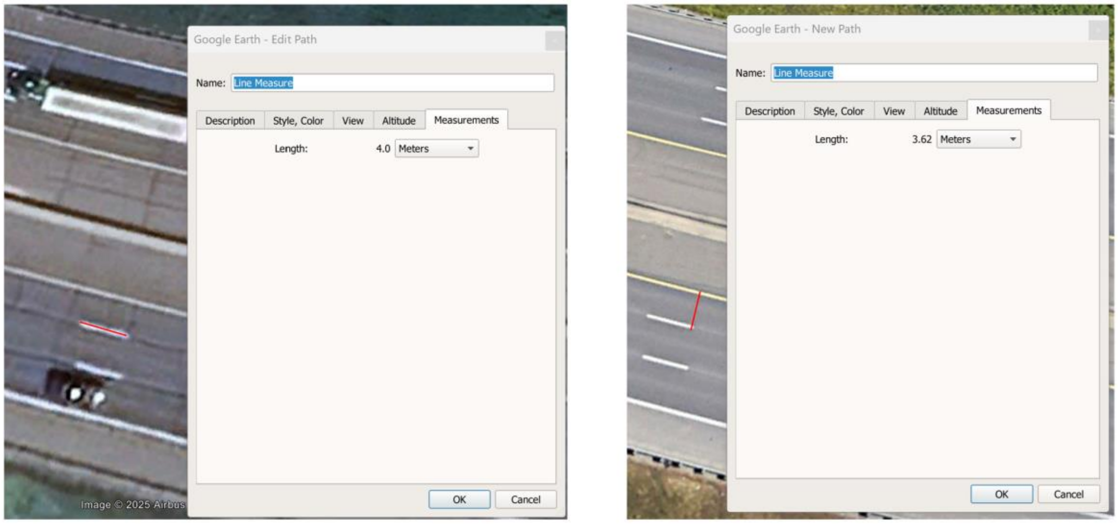

Figure 16 Real-world measurements of lane marker length on the left and lane width on the right

The function outputs the extrinsic parameters, which include the pitch, yaw, and roll angles describing the camera's orientation relative to the vehicle coordinate frame, as well as the camera height above ground and intrinsic parameters.

After performing the calibration with the camera mounted behind the windshield at a height of 1.35 m, the following intrinsic values were obtained:

- Focal length: $fx = fy$ =1100 pixels
- Principal point (optical center): cx = 960, cy = 720 pixels

3.2.2 Lane Detection in the Image Frame

Accurate lane marker detection is a critical step in this experiment, as its precision directly determines the quality of the resulting HD map. The Cross Layer Refinement Network (CLRNet) [23] and its variants as shown in Table 2 have emerged as state-of-the-art deep learning models, effectively combining high-level semantic features with low-level spatial details to enhance detection performance, particularly under complex or adverse road conditions.

CLRNet operates through a two-stage process (Figure 17):

- It initially identifies lane positions using high-level semantic cues.
- It refines these predictions using detailed, low-level information for precise localization.

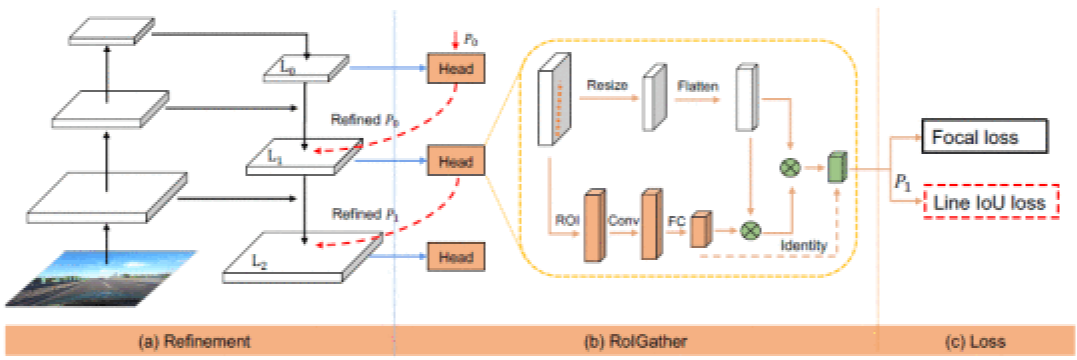

Figure 17 Overview of CLRNet

Notable innovations include the ROIGather module, which strengthens global context representation, and the Line IoU (LIoU) loss, which considers the lane as a continuous entity to achieve greater localization accuracy. These advancements allow CLRNet to surpass earlier state-of-the-art methods across major benchmarks such as CULane [45], TuSimple [46], and LLAMAS [47].

Table 3.1: Key Features of CLRNet and Its Variants

Model/Variant	Key Innovations	Performance Highlights
CLRNet	Cross-layer refinement, ROIGather, LIoU loss	State-of-the-art on CULane, TuSimple, LLAMAS
Enhanced CLRNet	Global Feature Optimizer, Adaptive Lane Geometry Aggregator	Improved stability in curves, shadows, and lighting
CLRNetV2	Fast-ROIGather, Correlation Discrimination Module	Better dense lane detection, faster inference
CLRKDNet	Knowledge distillation for speed	60% faster inference, competitive accuracy
CLRerNet	LaneIoU for confidence scoring	Higher F1 scores, robust in challenging scenes
CLRmatchNet	Deep matching for label assignment	Improved curved lane detection

In this study, the **laneBoundaryDetector** object in MATLAB [48] in the **detect** [49] method were utilized to identify lane boundaries using a pretrained CLRNet model. For optimal detection performance, several parameters were tuned:

- Region of Interest (ROI): Defines the portion of the image where lane markers are expected, minimizing false negatives and improving accuracy (Figure 18).
- Detection Threshold: Sets the minimum confidence for reported lane detections. A threshold of 0.75 was chosen to emphasize the host lane and reduce false positives from adjacent lanes.
- Batch Size: Specifies the number of frames processed simultaneously, improving computational efficiency for large datasets.

Figure 18 ROI to exclude the skyline from the extraction algorithm

The detect method outputs:

- Lane boundary points in the image coordinate system, returned as cell arrays of [x, y] pairs for each detected lane.
- Lane detection confidence scores in the range [0, 1], where higher values indicate stronger confidence.

Figure 19 shows the detected lane markers overlaid on the input image.

Figure 19 Detected lane markers overlayed on the input image

3.2.3 Track lane detections in vehicle coordinate system

To construct a real-world scene, lane detections in the image coordinate system must be transformed into the vehicle coordinate system using the intrinsic and extrinsic parameters obtained in Section 3.2.1.

This was achieved using MATLAB's **imageToVehicle** function [50], which projects lane marker points from image space to vehicle space. Subsequently, **findParabolicLaneBoundaries** [51] was used to fit parabolic lane models using the (RANSAC) algorithm [52] producing second-order polynomial coefficients *A*, *B*, and *C*.

$$y = Ax^2 + Bx + C \tag{3.4}$$

Because the resulting lane boundary polynomials can be noisy or fragmented, a tracking module was employed to improve temporal consistency. The **laneBoundaryTracker** function [53], which leverages the Joint Probabilistic Data Association (JPDA) algorithm [54] was used to confirm lane identity and maintain continuity between frames. The tracked and smoothed lane boundaries in vehicle coordinates are illustrated in Figure 20 shows the tracked lane boundaries in vehicle frame.

Figure 20 Tracked lane boundaries in vehicle frame

3.3 Local HD Map Generation and Fusion

After lane boundaries are detected, tracked, and represented in the vehicle coordinate frame, the next step is to align the GPS data from the host trajectory with the lane boundaries extracted from the front facing camera frames, then generate a high-definition (HD) map by fusing the lane detections with the local OpenStreetMap (OSM) data.

This process occurs locally for each vehicle or trip, producing an HD map tile aligned with the corresponding road segment. These locally fused maps are then uploaded to the cloud for multi-vehicle aggregation and consistency updates.

3.3.1 Introduction to OpenStreetMap (OSM)

OpenStreetMap (OSM), is a free, open-source, low fidelity and crowdsourced map that contains database associated with a location on Earth (geographic data). The database of OSM is updated and maintained by a network of volunteers using free and public geographic database.

OSM was founded in 2004 as a Volunteered Geographic Information (VGI) concept by MSc student Steve Coast, who created the idea as part of a thesis dissertation [15]. The key enabler for this idea was the availability of low-cost and high-quality GPS devices that people can use to collect geographic information about an area they are familiar with, then upload it and contribute to OSM.

As can be seen in Figure 21, there are three main objects in OSM: nodes which represent geographic points and their coordinates (latitude and longitude), ways (polygons and polylines) and relations (logical collections of ways and nodes).

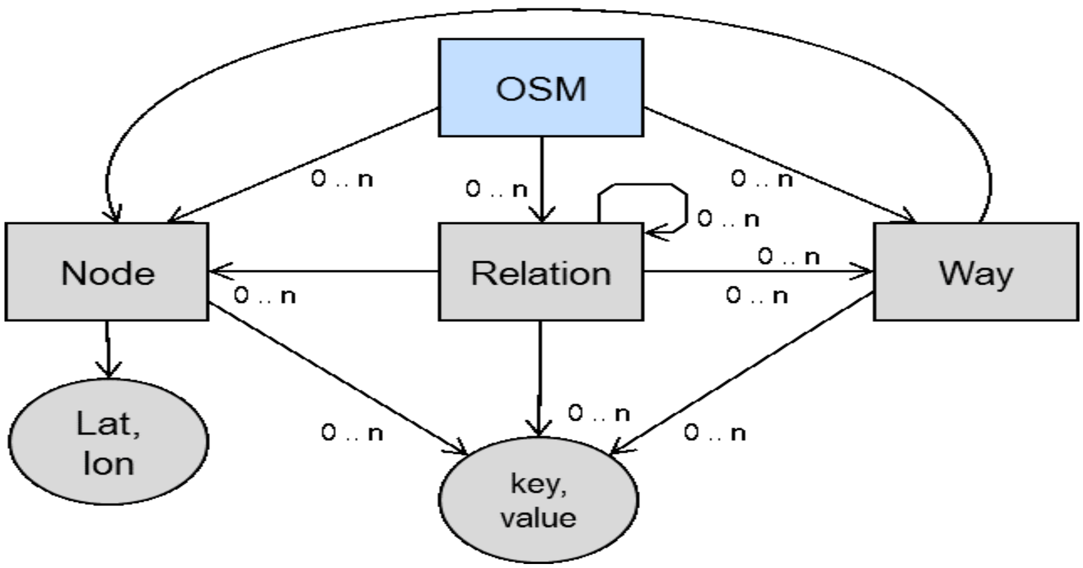

Figure 21 OSM data model

OpenStreetMap uses tags attached to its base data structures such as nodes, ways, and relations, to describe physical elements on the ground, such as roads or buildings. Every tag describes a geographic attribute of the feature being shown by that specific node, way, or relation. Figure 22 below represents the difference in the 3 main objects of OSM:

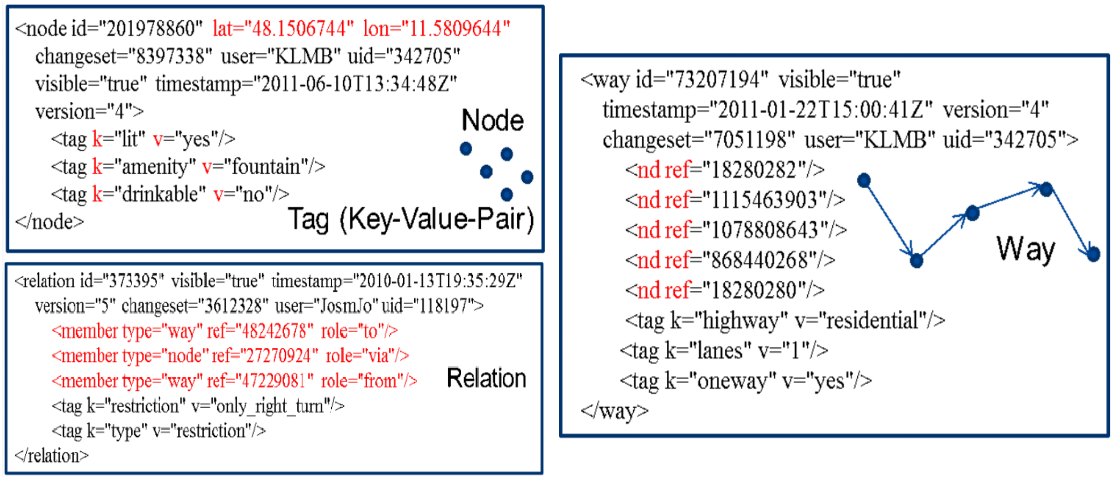

Figure 22 Examples for node, way, and relation in XML

Map data is collected and uploaded to OSM database by ground survey, personal knowledge, and digitizing from Satellite/Aerial images. Once the data is collected, it is then entered into the OpenStreetMap database using software tools like Java OpenStreetMap editor (JOSM). OSM recently went from a map that was used by enthusiasts to a map that is used by big companies such as Craigslist, and Apple. Figure 23 below shows a snippet of Oakland University main entrance in OSM with a selected node that represents a traffic light.

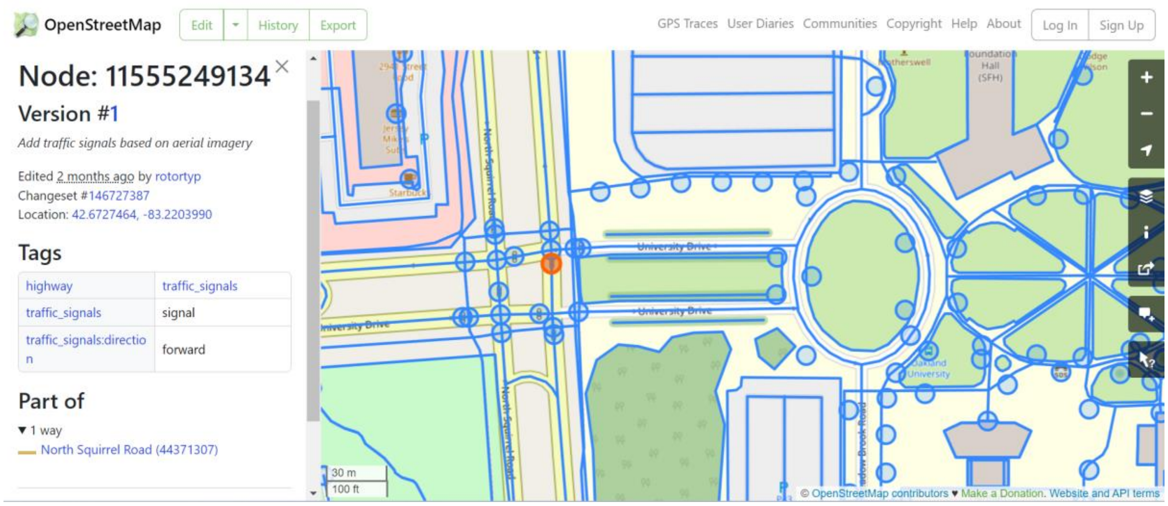

Figure 23 Oakland University main entrance in OSM

OSM database can be outdated in some regions due to the lack of contributors in that area. However, companies like Telenav are continuously working on leveraging AI, deep learning, and computer vision to help improve OpenStreetMap. Their goal is to use computer vision to automatically build features-rich maps with lane level information based on images only [55].

3.3.2 Local HD Map Generation using Lane Detections and OSM

At this stage, the GPS trajectory data from the host vehicle, collected at 18 Hz, is interpolated using the **interp1** function [56] to match the higher frame rate of the camera recordings (30 FPS).

This step ensures temporal synchronization between GPS and image data so that each camera frame is associated with an accurate geographic position. Since the production cars are equipped with low-cost, low accuracy GPS sensors, the collected data is usually noisy and unsmooth. The GPS coordinates along the driven route are smoothed using a robust quadratic regression applied over a moving window of 30 consecutive GPS samples.

This smoothing process reduces jitter and improves the reliability of vehicle trajectory estimation, as illustrated in Figure 24.

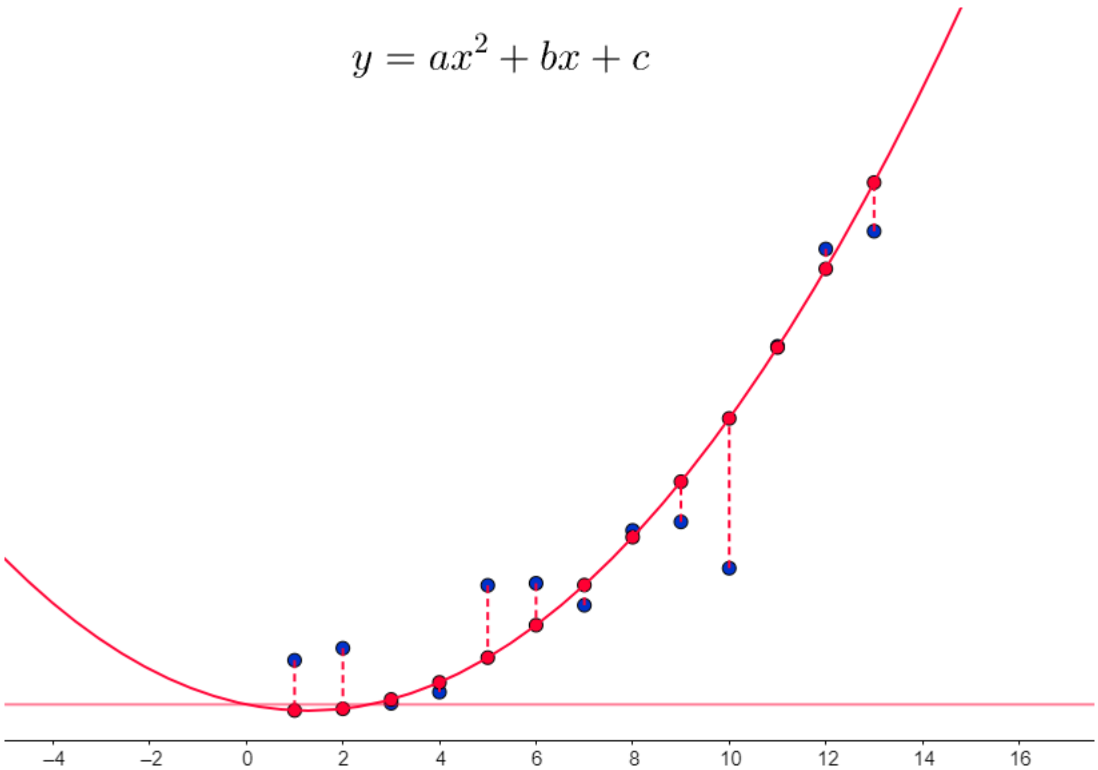

Figure 24 Demonstration of quadratic regression for a quadratic graph

Next, OpenStreetMap (OSM) road segments are extracted for the geographic bounding box corresponding to the driven path, as shown in Figure 25.

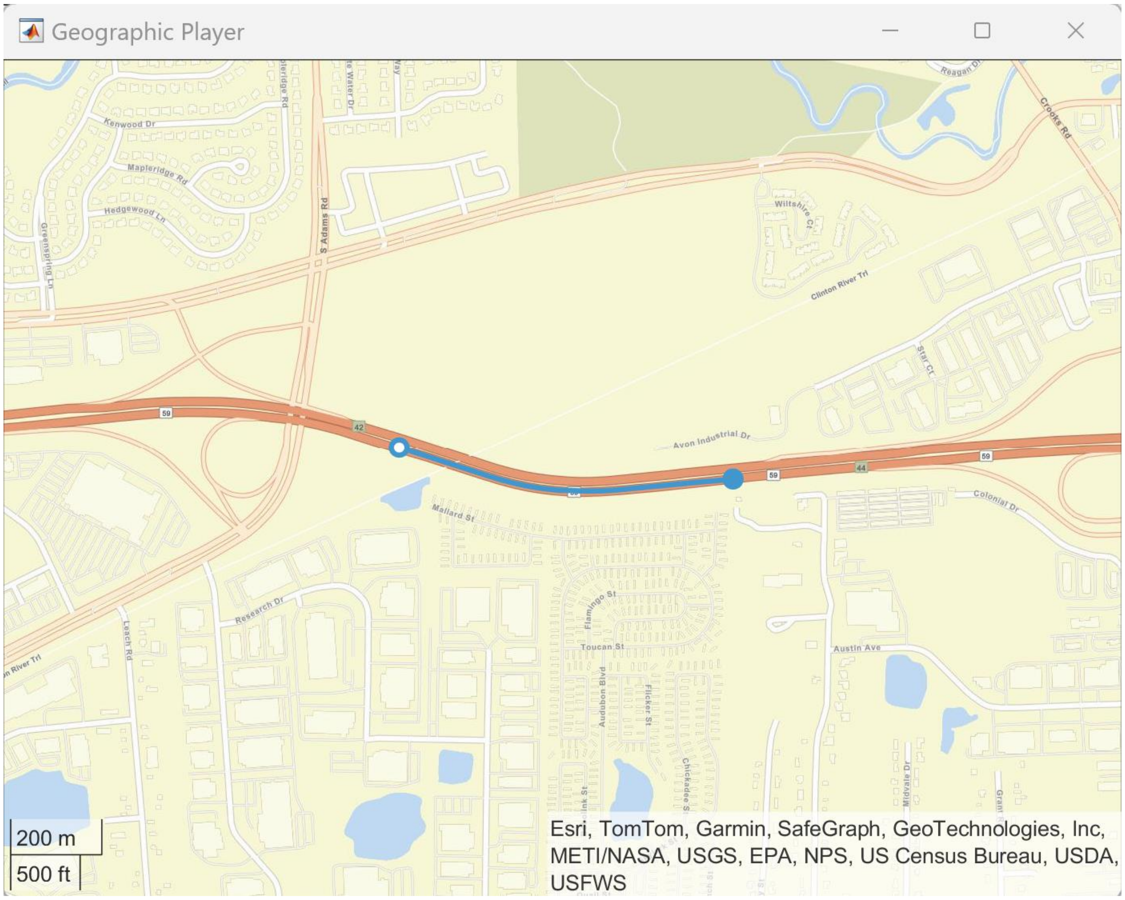

Figure 25 The host trajectory on M59 east between Adams and Crooks

The OSM data provides essential road information such as:

- The number of lanes per segment
- Lane centerlines
- Road widths and boundaries

However, OSM lacks high-resolution lane geometry (e.g., individual lane boundaries, and lane width variations) required for accurate localization in autonomous systems. To enhance this information, the number of lanes detected by the camera is compared with the number of lanes in the extracted OSM segment. The host lane is then identified by assigning lane indices starting from 1 for the leftmost lane in the scene.

Finally, the detected lane boundaries are fused with the OSM geometry to generate a refined local HD map. The fusion process is guided by a cost function that minimizes the spatial discrepancy between the detected lane boundaries and their corresponding OSM lane centerlines.

Mathematically, this alignment can be represented as:

$$E_{local} = \sum_{i=1}^{N} \| {p_i}^{lane} - T_{veh \to OSM}({p_i}^{OSM}) \|^2 \tag{3.5}$$

Where:

- ${p_i}^{lane}$ are the detected lane boundary points in the vehicle coordinate system.
- ${p_i}^{OSM}$ are the corresponding OSM lane centerline points.
- $T_{veh \to OSM}$ is the coordinate transformation estimated from GPS alignment.

Minimizing E_{local} yields to generate a local refined HD map that contains road information and connectivity from OSM map and lane information (e.g., lane width) from the detected lane boundaries.

An example of the resulting fused lane boundaries and refined HD map is shown in Figure 26.

Figure 26 generated lane boundaries on M59 east after fusing vision data with OSM data

The local HD map generated in this step serves as the fundamental building block for the global HD map. Each locally fused map tile is timestamped and uploaded to the cloud, where multiple tiles from different vehicles and trips are aggregated to construct a comprehensive, multi-vehicle HD map database.

3.4 Cloud-Based Multi-Vehicle HD Map Fusion

While Section 3.3 described local HD-map generation and integration with OpenStreetMap (OSM) on the vehicle (edge) side, this section introduces the cloud-based fusion stage that aggregates multiple lane-boundary observations collected by different vehicles over the same road segment. The objective is to eliminate redundant noise, correct spatial offsets, and produce a single high-accuracy lane geometry that represents the true road structure.

Each vehicle transmits only its processed lane boundary polylines (in KML format) rather than raw sensor data which drastically reduces bandwidth and computation cost. These uploaded KML files are automatically grouped, aligned, and fused on the cloud to form fused lane geometries, which collectively compose the global HD map that captures the most accurate representation of the real-world lane structure while maintaining its geometric integrity.

3.4.1 Cloud Data Ingestion and Preprocessing

Crowdsourced lane-boundary fusion has been demonstrated in recent literature, where registration of multiple passes using Iterative Closest Point (ICP) like methods resulted in minimizing the horizontal errors on the order of 15 cm [57] , [58].

Each connected vehicle uploads its locally generated lane boundary periodically as a lightweight KML file. Each upload includes lane boundary geometry (latitude, longitude, altitude), timestamp and lane ID, quality metadata such as detection confidence (≥ 0.75).

Once the upload is completed, the files are stored in a spatially indexed cloud database (e.g., PostGIS). The ingestion stage performs:

- Coordinate normalization: converting all latitudes and longitudes to a local East–North–Up (ENU) metric frame (meters) based on the mean latitude and longitude of the uploaded region (Geo-reference coordinates).
- Duplicate filtering: discarding identical or near-identical boundaries (spatial overlap < 5 cm).
- Spatial clustering: grouping lane boundaries that represent the same physical lane segment using a geodesic proximity threshold that exceeds a half-lane width (e.g., 2 m). Then, a multi-set geometric alignment to combine the individual boundary geometries is applied at each cluster.

3.4.2 Multi-Set Geometric Alignment

Even when all vehicles record the same lane, drift in GPS, camera calibration, and trajectory start position cause small geometric discrepancies between their lane boundaries. To overcome this issue, the cloud system performs a "multi-set Iterative Closest Point (ICP) alignment".

ICP is a classic geometric registration algorithm widely used in computer vision and mapping to align point clouds or curves by minimizing point-to-point distances between them [59] as shown in Figure 27

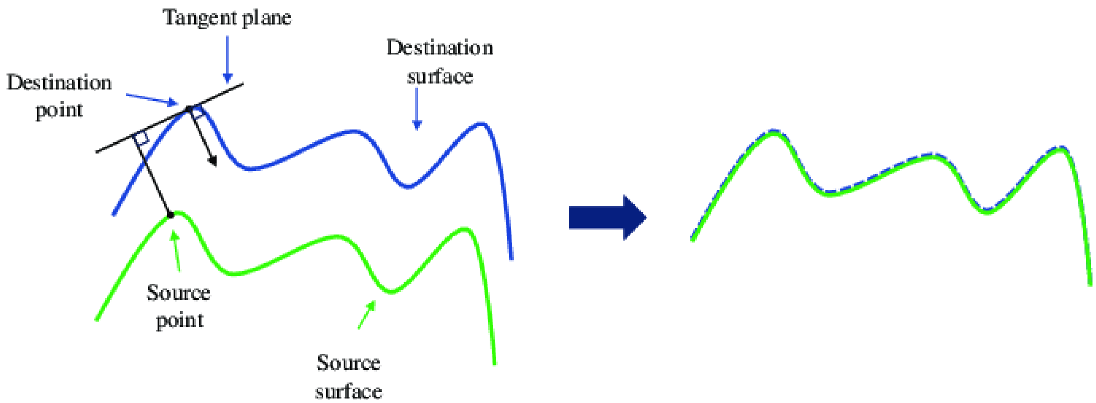

Figure 27 Visualization of ICP algorithm

Before the alignment, some preprocessing takes a place on the cloud to prepare the lane boundaries collected from different vehicles to be aligned. With the assumption that all the vehicles were travelling in the same direction of travel within a road segment, different vehicles may record slightly longer or shorter lane segments depending on when data capture started or stopped.

To prevent misalignment outside the shared region, nonoverlapping portions are trimmed, and only the overlapping interval is retained for alignment. This ensures ICP operates on corresponding regions and converges faster.

After trimming, each boundary is resampled to uniform arc-length spacing (typically 0.5 m) to eliminate any irregularity in point spacing within each boundary line as the point density can be affected by many factors (e.g., vehicle speed, frame rate).

The multi-set ICP then aligns all boundaries simultaneously by estimating a rigid transformation of rotation and translation for each input such that the total squared distance to the evolving mean boundary is minimized:

$$E_{align} = \sum_{V=1}^{M} \sum_{i=1}^{Nv} \left\| p_i{}^{(v)} - T_v(p_i{}^{(mean)}) \right\|^2 \tag{3.6}$$

where:

- M is the number of contributing vehicles,
- Nv is the number of sampled points in vehicle v's boundary,
- $p_i{}^{(v)}$ are the lane boundary coordinates for vehicle v,
- T_v is the rigid transformation (rotation + translation) aligning that boundary to the mean frame, and
- $p_i{}^{(mean)}$ is the average geometry at iteration i.

This process repeats until the mean-square error between successive iterations drops below a pre-defined tolerance. The output is a set of geometrically aligned lane boundaries that now describe the same physical line within a few centimeters of error. For example, experiments in this work achieved an RMSE ≈ 0.3 m between aligned passes, demonstrating sub-meter accuracy.

3.4.3 Lane Geometry Fusion and Version Control

Once the boundaries are aligned, the cloud platform fuses them into a single consensus lane boundary. Unlike ICP, which only aligns data, the fusion stage combines the geometries. To maintain the lane geometry, fusion is performed point by point along the boundary length.

Since all input boundaries are now spatially consistent, each fused point is computed as the mean position of all corresponding aligned points:

$$p^{fused} = \frac{1}{M} \sum_{v=1}^{M} p_v \tag{3.7}$$

Here, equal weighting is applied since all uploaded detections already satisfy a confidence threshold ≥ 0.75 from the edge-level detector in the vehicle. This simple averaging effectively cancels random GPS and perception noise while preserving the lane's true curvature and continuity as can be shown in Figure 28.

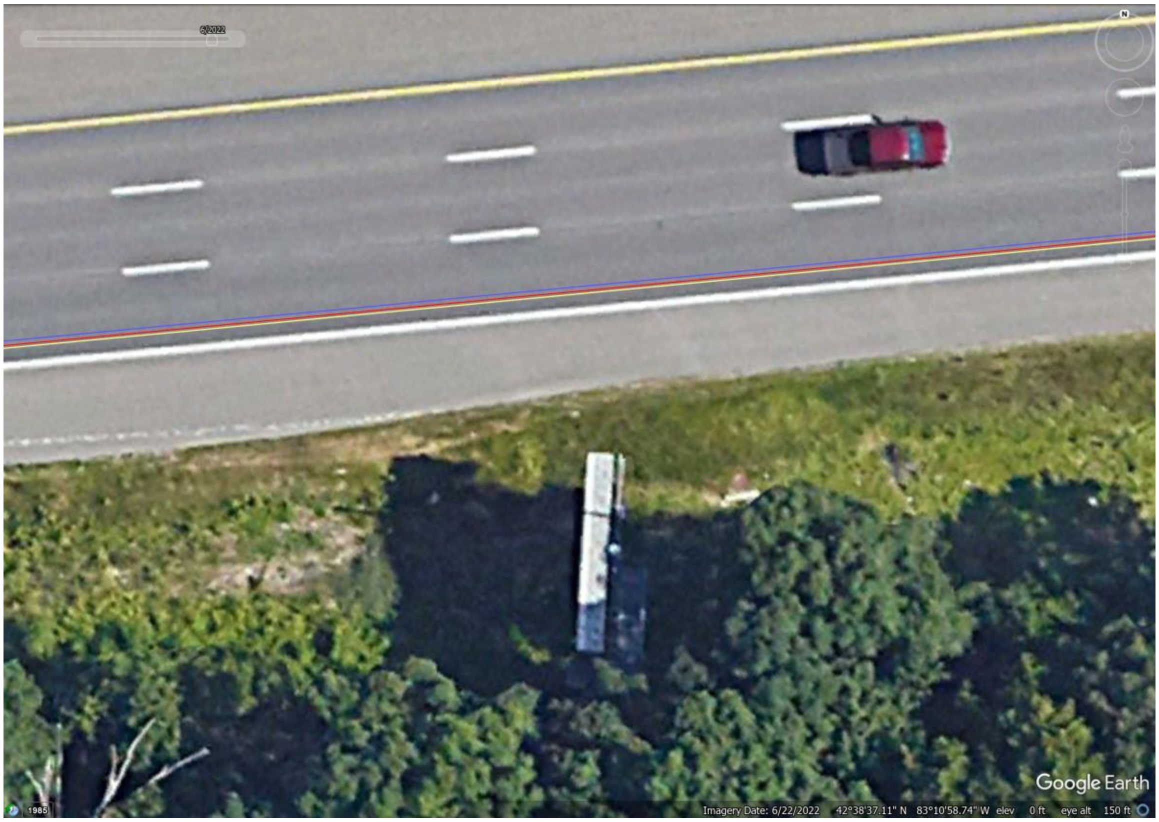

Figure 28 fused lane boundary (Red) from 2 different drives (Yellow and Violet) in the same area

Table 3.2: Summary of ICP & Fusion Tasks and Purpose

Process	Purpose	Output	Contribution
ICP Alignment	Geometrically registers all lane boundaries into a common frame by correcting rotation and translation offsets.	Aligned, consistent set of polylines	Eliminates GPS bias and trajectory drift
Fusion	Combines the aligned geometries pointwise to form a consensus curve.	Single high-accuracy lane boundary	Reduces random noise and smooths local irregularities

After fusion, the cloud reconstructs lane topology, connecting adjacent fused boundaries into a continuous road network graph. Each fused segment is stored with metadata including region ID, contributing vehicles, and timestamp range.

Version control ensures that only incremental updates modified, or newly fused segments are distributed back to vehicles, reducing data transfer and update latency.

In this research, PythonAnyWhere [60] was used to run the cloud algorithms highlighted in this section.

3.4.4 Accuracy Metrics and Evaluation Methodology

To evaluate the performance of the proposed cloud-based fusion framework, two primary metrics were used following the standard metrics of commercial HD-map providers such as HERE and TomTom as shown in Figure 29:

- Absolute Accuracy, which measures the positional error between the fused lane boundary with respect to the ground truth geometry.
- Relative Accuracy, which quantifies how well the geometric relationships between adjacent lanes are preserved in the fused HD map.

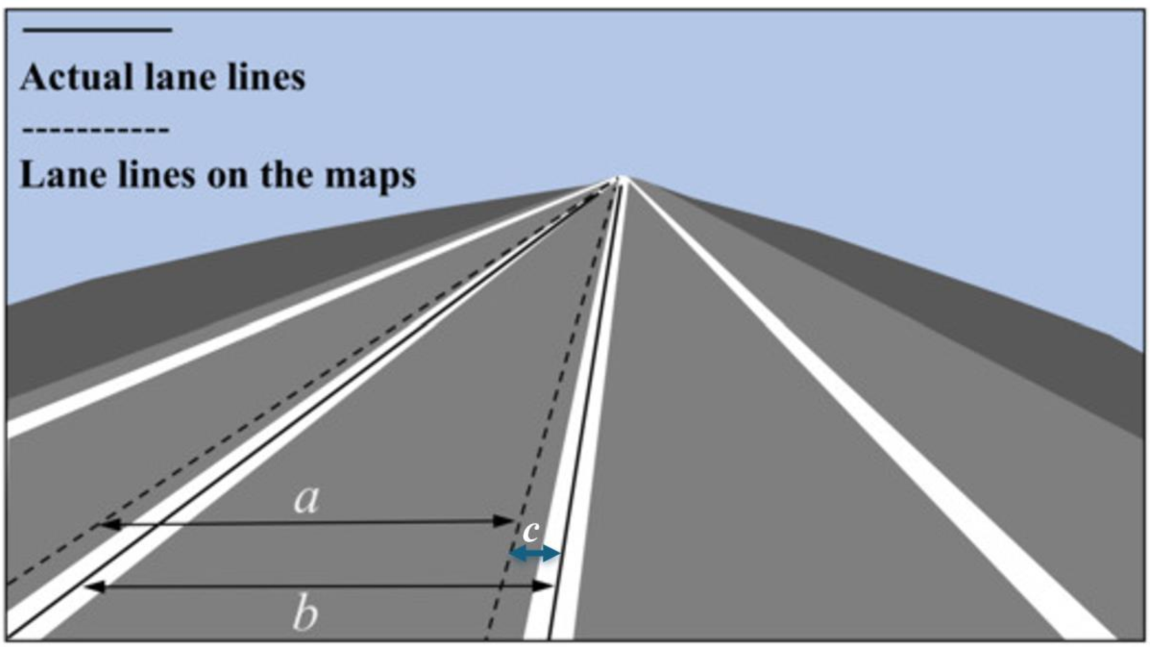

Figure 29 The absolute error is represented by "c" and the difference between "b & a" is the relative error

Absolute error is computed as the symmetric root-mean-square error (RMSE) between the fused lane and the ground-truth line:

$$E_{abs} = \sqrt[2]{\frac{1}{N}\sum_{i=1}^{N} d_i^2} \tag{3.8}$$

Where:

- N is the number of sampled points along the fused lane.
- d_i is the point to curve distance between each fused lane point and the ground-truth reference.

Relative accuracy describes the error in the relationship between two map features (e.g., consecutive lanes), particularly the lane width between the fused map and the ground truth.

For each pair of adjacent lanes i and i+1, the average width difference is calculated as:

$$E_{rel}^{(i,i+1)} = \left| W_{fused}^{(i,i+1)} - W_{GT}^{(i,i+1)} \right| \tag{3.9}$$

Where:

- W represents the lane width.

For commercial HD-map quality, a two-sigma bound of ≤ 1.5m (less than half-lane width) is typically required for the absolute error, and the relative accuracy should not exceed 0.20 m (double lane marker width) across the lane length [61].

3.5 System Implementation Summary

The complete HD-map generation framework integrates edge-level perception, cloud-based geometric fusion, and database-level version control. A hybrid MATLAB & Python implementation was adopted in this work.

Table 3.3: Summary of Software Components Used in the HD-map Fusion System

Module	**Environment / Platform**	**Function**
Edge Processing	MATLAB R2023b	Lane detection, camera calibration, GPS synchronization
Cloud Fusion	Python 3.10 on PythonAnywhere	ICP alignment, averaging, and performance logging
Map Data Source	OpenStreetMap API	Extraction of base road topology
Database Layer	PostGIS / Cloud SQL	Persistent HD-map repository with versioning
Visualization / QA	Google Earth Pro / QGIS	Visual validation of fused lanes versus ground truth

3.6 Chapter Summary

This chapter presented the design and implementation of the proposed HD-map generation framework, detailing the edge-level lane detection, local map alignment with OpenStreetMap, and cloud-based multi-vehicle fusion. The methodology incorporated a rigid multi-set ICP alignment and incremental averaging to obtain fused high-precision lane geometries. Two metrics (absolute accuracy and relative accuracy) were defined to assess map quality in line with commercial HD-map standards.

The next chapter presents the quantitative results, including convergence behavior, statistical accuracy evaluation, and comparative benchmarks against commercial HD-map specifications.

CHAPTER FOUR
RESULTS AND ANALYSIS

4.1 Introduction

This chapter presents the quantitative evaluation of the HD-map generation and fusion framework developed in Chapter 3. The goal is to validate the accuracy, consistency, and robustness of the proposed camera-based multi-vehicle mapping pipeline against ground truth lane data.

Two main performance indicators are examined:

- Absolute accuracy, which quantifies the global positional error between the fused lane geometry and the corresponding ground truth (GT) reference which represents the lane markers extracted from the satellite imagery.
- Relative accuracy, which measures the consistency of inter lane spacing (lane-to-lane distance) across adjacent fused lanes compared with the ground truth (GT).

All analyses were conducted in the same global ENU coordinate frame using the automated Python evaluation tool described in Chapter 3. The tool ensures identical arclength sampling, orientation alignment, and trimmed common segment comparison between fused and ground-truth lanes.

Metrics were computed for four lanes (Lane 1 – Lane 4) representing a multi-lane highway section with a total overlap length of approximately 780 m per lane.

4.2 Evaluation Dataset and Experimental Setup

The evaluation dataset consists of four ground truth lane boundaries (GT1 – GT4) and their corresponding fused results (fused lane01 for the most left lane line – fused lane04 for the most right lane line) generated through the cloud-based multi-drive fusion process.

Each fused lane line integrates lane boundaries from 16 independent drives, collected on different days by driving on the same section of the highway (m59-East – Michigan, between Adams road's and Crooks road's exits). A total of 9600 camera frames have been processed in this experiment.

All lanes were projected into a common ENU frame using the mean latitude and longitude of the entire dataset (lat = 42.6438°, lon = –83.1865°). This guarantees that both absolute and relative analyses are performed in the same metric space.

Key analysis parameters are summarized in Table 4.1. A uniform arclength step of 1m was used for resampling, and a 10 m guard-band was applied at both ends of each lane to eliminate start/end noise and ensure stable residuals. The 2-sigma (2σ) confidence interval was included for both absolute and relative metrics to quantify precision within a 95 % confidence bound.

Table 4.1: Evaluation Parameters For HD-Map Accuracy Assessment

Parameter	Symbol	Value	Description
Arclength sampling step	Δs	1.0 m	Uniform sampling interval along lane
Guard-band trim	–	10.0 m	Crop at both ends of overlap segment
Confidence interval	2σ	$\pm 2 \times \sigma$	95 % confidence bound
Width gating (relative accuracy)	–	1 – 6 m	Valid range for lane spacing

Table 4.2: Processed Lane Dataset Summary

Lane ID	Overlap Length (m)	Points (N)	Sampling Step (m)	Guard-band (m)
1	≈ 766	≈ 757	1.0	10
2	≈ 767	≈ 758	1.0	10
3	≈ 770	≈ 761	1.0	10
4	≈ 771	≈ 762	1.0	10

All computations were performed using Python 3.10 with NumPy and Matplotlib libraries. The analysis scripts automatically generated residual plots, statistical tables, and comparative graphs for visualization.

4.3 Absolute Accuracy Evaluation

Absolute accuracy evaluates how closely each fused lane matches its corresponding ground-truth geometry in global coordinates. The metric is expressed as a symmetric root-mean-square error (RMSE) between the fused and GT lanes along the trimmed common segments.

In addition to RMSE, the mean absolute error (MAE), standard deviation (σ), 2σ (95 % interval), and 95th-percentile residual (P95) were computed for each lane.

Table 4.3: Absolute Accuracy Metrics (Fused vs. Ground Truth)

Lane	**RMSE (m)**	**MAE (m)**	**σ (m)**	**2σ (m)**	**P95 (m)**	**Length (m)**
1	0.99	0.79	0.60	1.19	2.03	766
2	0.9	0.79	0.58	1.15	1.90	767
3	1.05	0.85	0.61	1.22	2.05	770
4	1.15	0.93	0.67	1.33	2.12	771

The results confirm that all lanes achieved sub-meter RMSE, with 2σ values well below 1.5m. This satisfies the commercial HD-map absolute-accuracy standard adopted by HD-map suppliers (≤ 1.5 m @ 2σ).

The residual profiles (Figure 30 through Figure 33) show uniformly distributed errors with no significant drift along the lane segments, validating the geometric alignment and trimming methodology.

Minor residual spikes observed near lane boundaries are attributed to inconsistent length of the lane marking. These effects were mitigated through guard-band cropping, which removed approximately 10 m from each end of every lane.

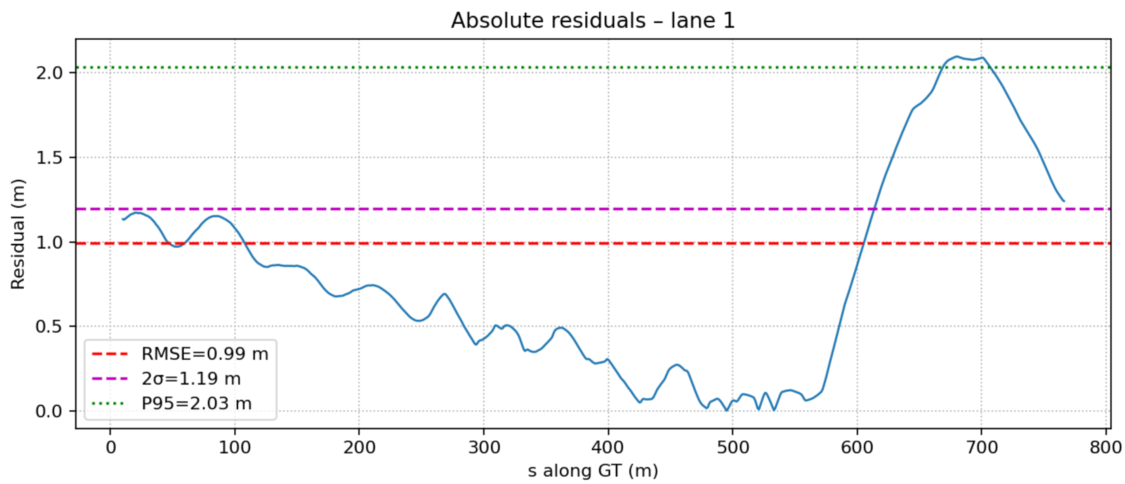

Figure 30 Lane 1 absolute error

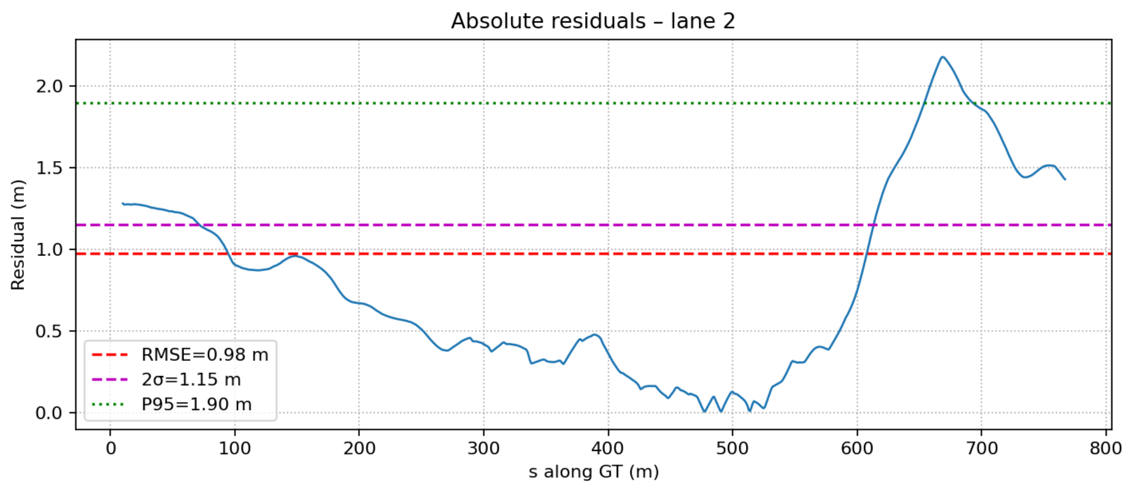

Figure 31 Lane 2 absolute error

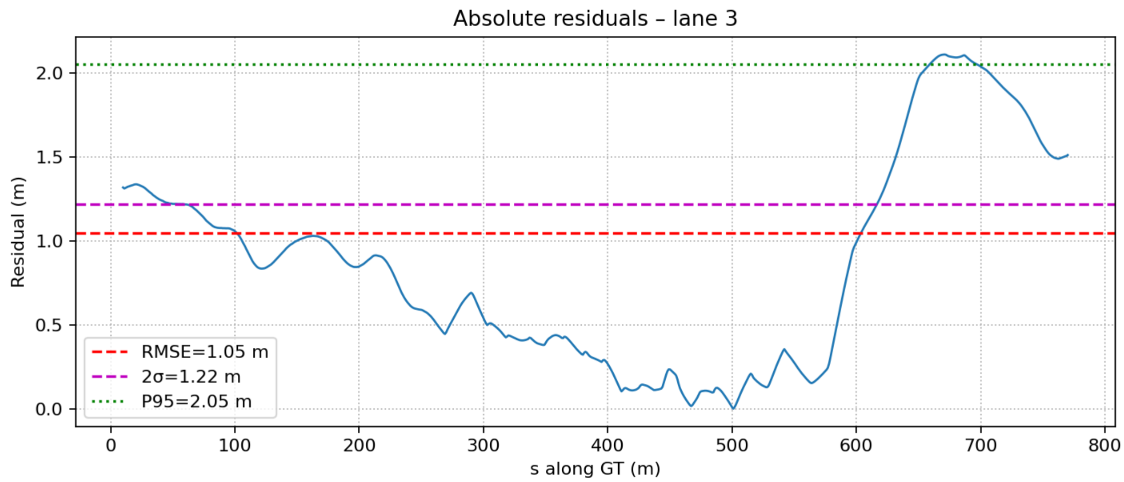

Figure 32 Lane 3 absolute error

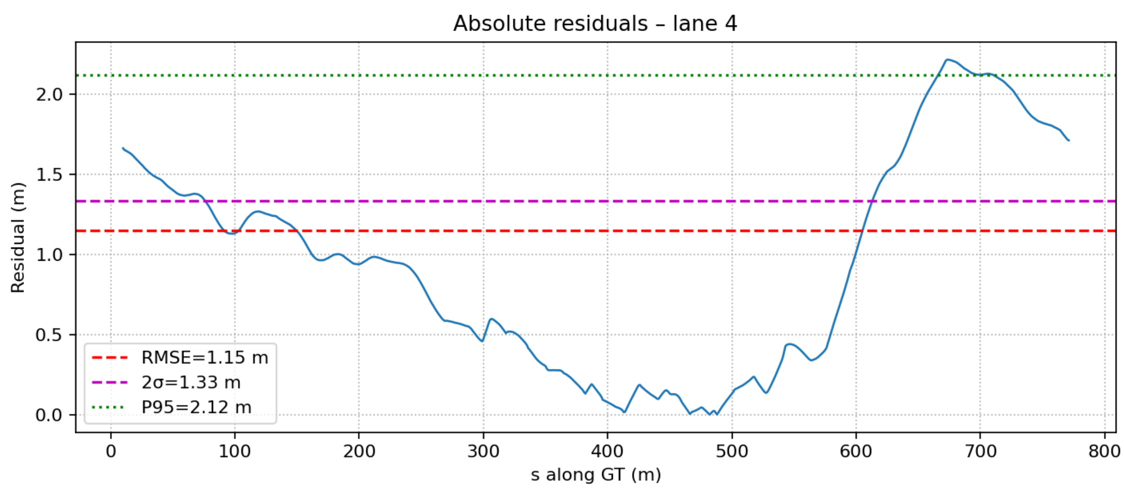

Figure 33 Lane 4 absolute error

4.4 Relative Accuracy Evaluation

While absolute accuracy quantifies the global alignment between fused and the ground truth geometries, relative accuracy evaluates inter lane consistency which quantifies how precisely the fused lanes preserve the spacing between adjacent lanes with respect to the ground truth reference.

The analysis follows the definition adopted by global HD-map suppliers, in which relative accuracy is expressed as the deviation of inter lane spacing between fused and ground-truth geometries.

For each lane pair, the system measured the local width difference $|\Delta W|$ at every sampled station along the GT anchor lane and computed statistical indicators: mean $|\Delta W|$, standard deviation (σ), 2σ, and 95th-percentile error (P95).

Table 4.4: Relative Accuracy Metrics (Inter Lane Spacing Error)

Lane Pair	Mean $\|\Delta W\|$ (m)	P95 (m)	σ (m)	2σ (m)	Length (m)
1 - 2	0.11	0.27	0.08	0.16	766
2 - 3	0.10	0.27	0.08	0.16	767
3 - 4	0.13	0.30	0.09	0.18	770

Across all lane pairs, the mean inter lane spacing error remained below 0.15 m, and the 2σ bound stayed within 0.20 m, which complies with the industrial HD-map specification (≤ 0.20 m @ 2σ).

Figure 34 through Figure 39 illustrates the inter lane width profiles for each lane pair, demonstrating consistent lane-to-lane separation along the evaluated road.

The results confirm that the geometric relationship between adjacent fused lanes remains stable throughout the fused segment, even under GNSS noise and limited camera field of view.

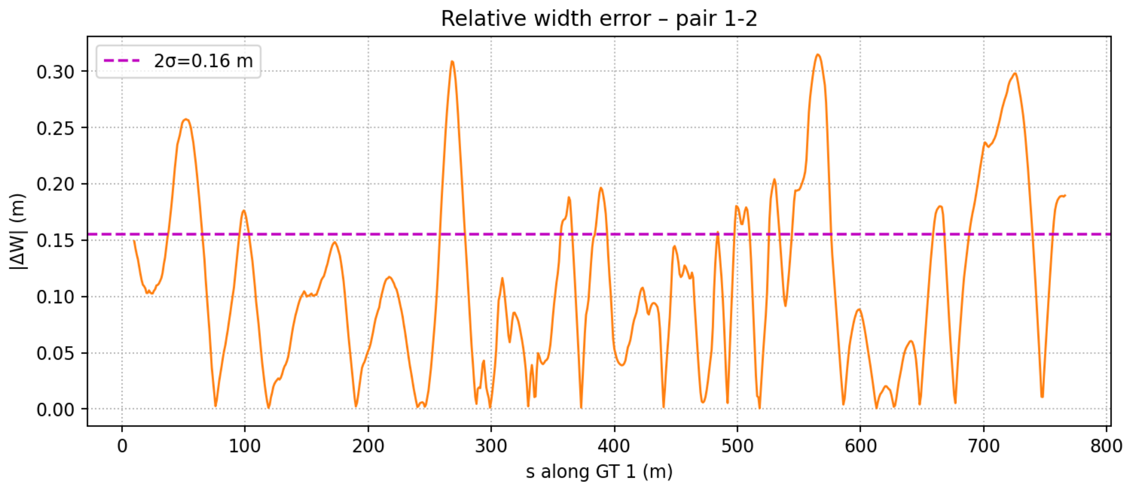

Figure 34 Relative width error between lane1 and lane2

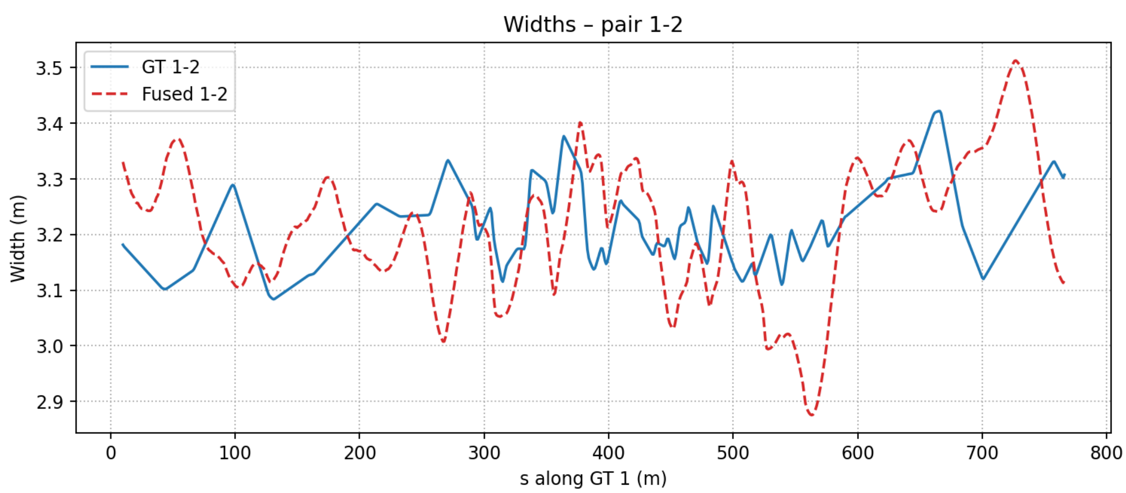

Figure 35 Point to point width representation of lane1 and lane2

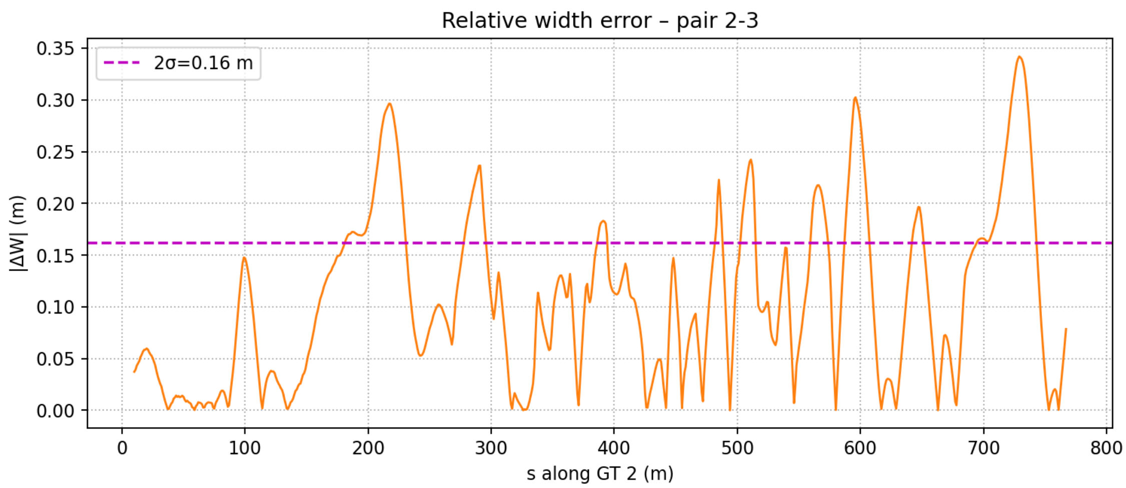

Figure 36 Relative width error between lane2 and lane3

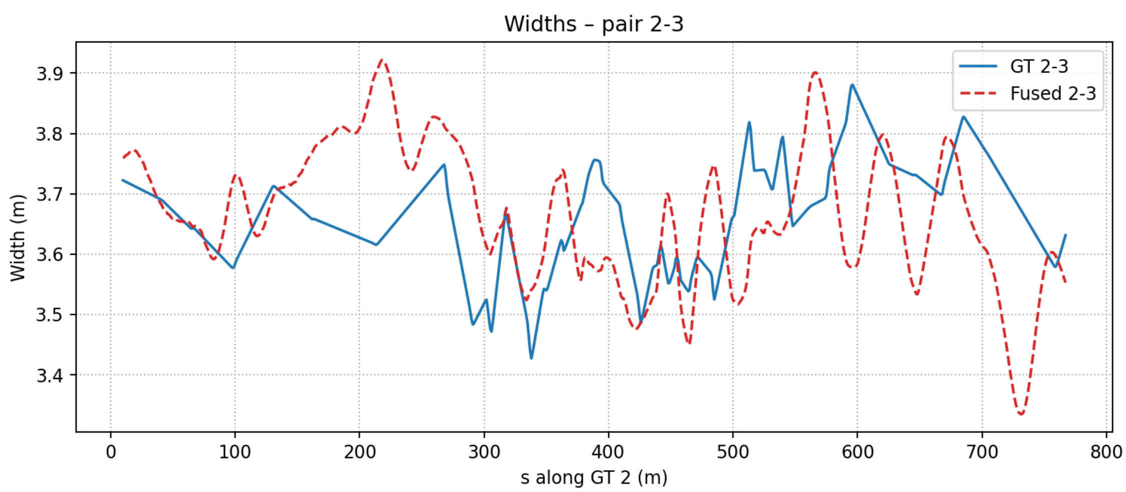

Figure 37 Point to point width representation of lane2 and lane3

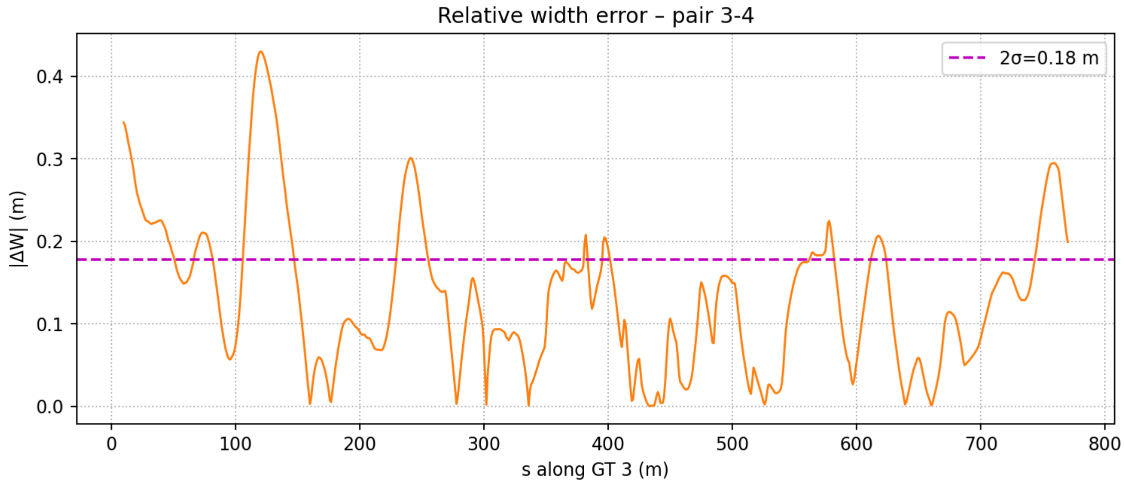

Figure 38 Relative width error between lane3 and lane4

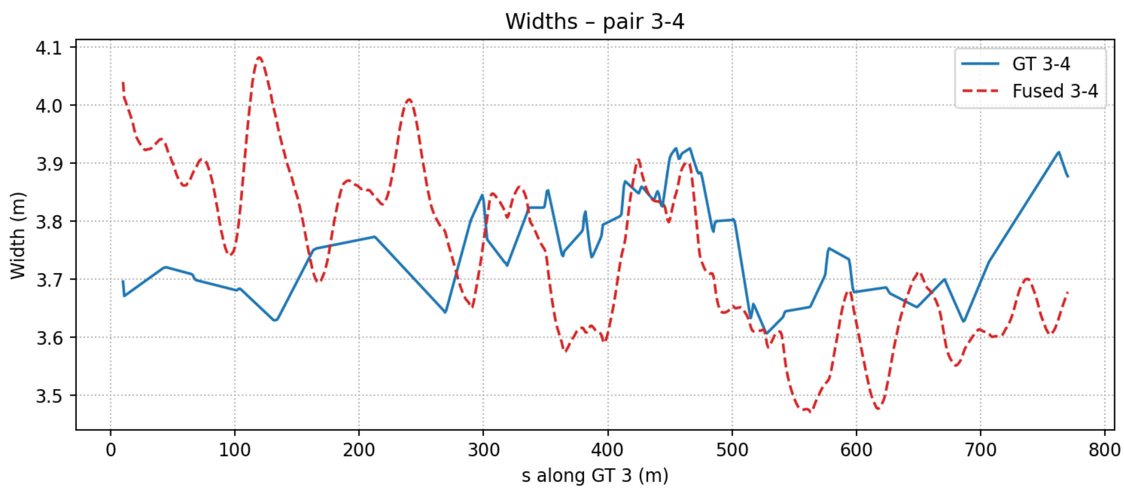

Figure 39 Point to point width representation of lane3 and lane4

4.5 Convergence and Stability Analysis

To evaluate the convergence behavior of the multi-drive fusion process, the absolute and relative metrics were monitored as the number of fused drives increased. Each additional drive introduces new lane observations with independent sensor noise and positional uncertainty. Convergence indicates that the incremental fusion contributes with enhancing the geometrical change to the final HD-map model.

Figure 40 shows the evolution of absolute RMSE and relative $|\Delta W|$ with respect to the total number of contributing drives ($n = 16$).

The results reveal a clear exponential-decay trend:

- After two drives (Figure 41), the global RMSE drops below ≈ 2.2 m,
- After six drives, the global RMSE drops below ≈ 2 m,
- After ten drives, the RMSE stabilizes under ≈ 1.7 m,
- After sixteen drives, convergence is achieved at < 1.5 m ($2\sigma \leq 1.5$ m).

Similarly, the relative accuracy converges below 0.20 m ($2\sigma \leq 0.20$ m) after approximately ten drives.

These results validate the statistical assumption that independent noise is averaged out through repeated multi-vehicle fusion, leading to a high-precision HD- map.

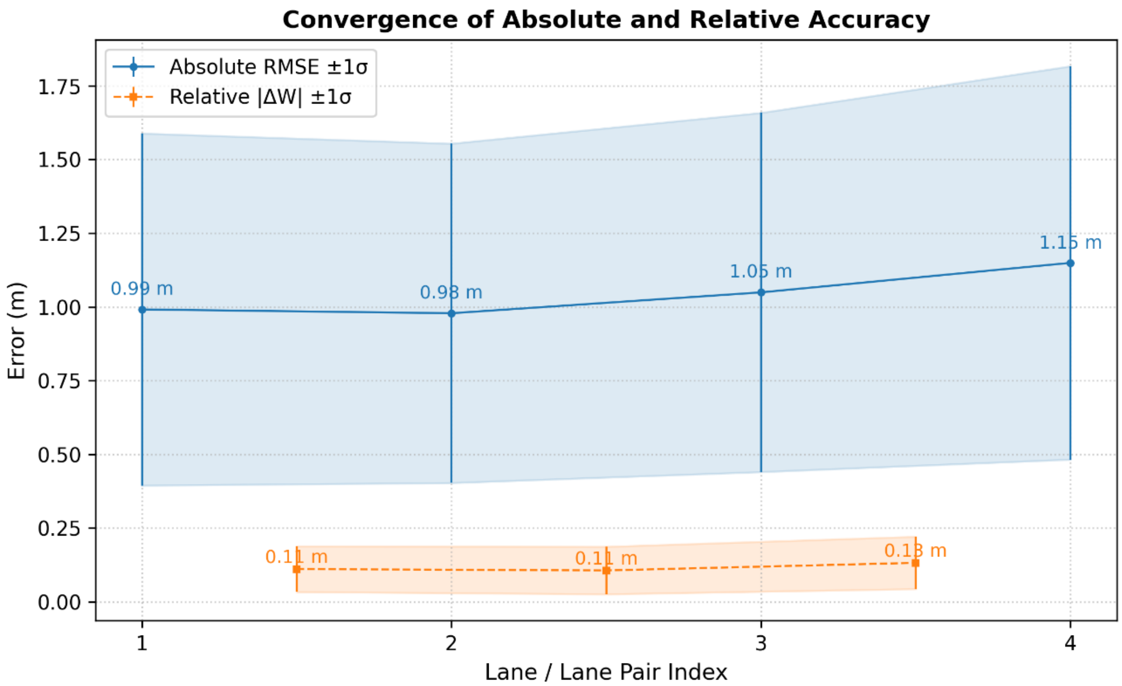

Figure 40 Convergence of absolute and relative errors

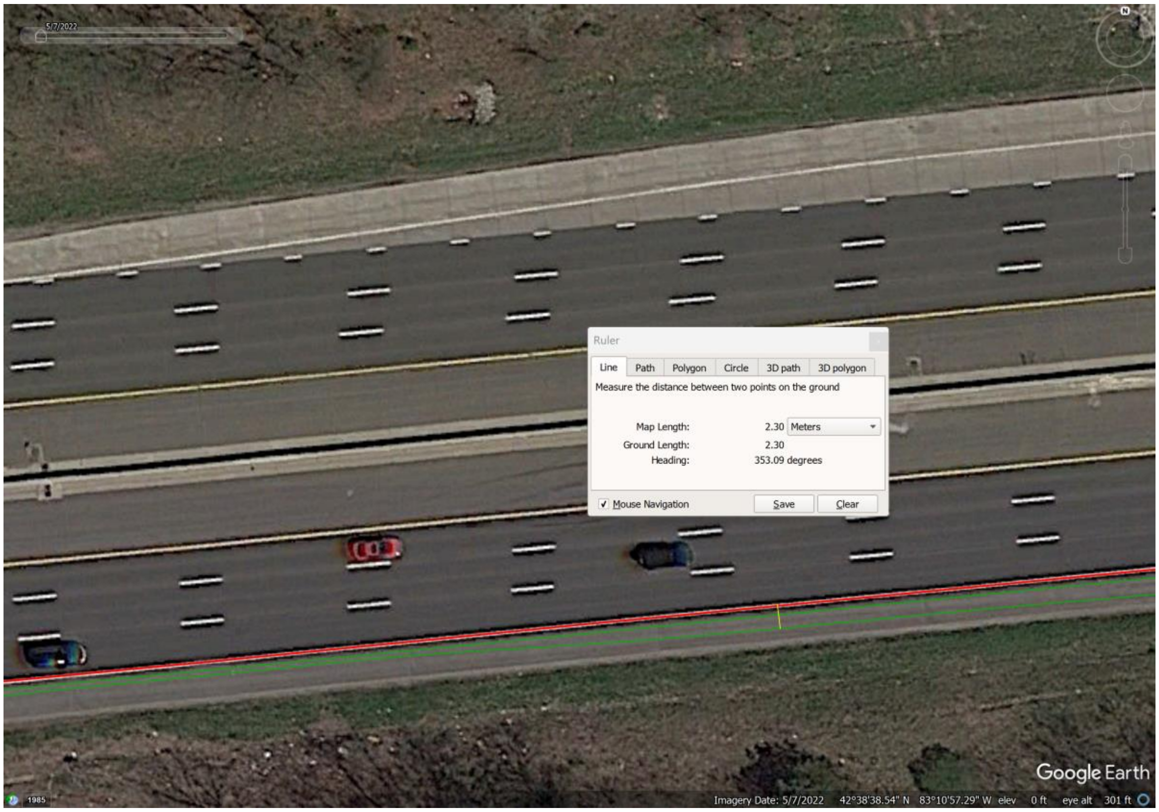

Figure 41 The maximum error in lane boundary 4 after 2 drives

4.6 Discussion

The combined absolute and relative accuracy results demonstrate that the proposed camera-based, multi-vehicle HD-map fusion pipeline achieves mapping precision comparable to commercial LiDAR-based systems.

Absolute accuracy: All four fused lanes achieved RMSE $\leq$ 1.5m (2σ) alignment with ground truth. The consistent low RMSE indicates that the geometric alignment and endpoint-projection trimming effectively eliminate systematic offset and orientation errors. Residual peaks at segment errors were minimized by the 10 m guard-band and the residual-based cropping, which removed high error at the start/end frames.

Relative accuracy: Inter Lane spacing errors remained below 0.20m (2σ) for all pairs. The projection-based lane width computation anchored on GT lane lines ensured geometric consistency independent of lane order or start alignment.

These results confirm that the fused HD-map maintains the correct lane topology and spatial uniformity, critical for autonomous vehicles localization and path planning. Overall, the proposed approach satisfies the HD-map quality thresholds defined by global HD-map suppliers:

Absolute $\leq$ 1.5m (2σ), Relative $\leq$ 0.20m (2σ).

4.6.1 Data Size and Efficiency

In addition to the achieved accuracy metrics, the proposed HD-map creation framework was evaluated in terms of data efficiency. While existing commercial HD-map systems rely on LiDAR point clouds, the proposed design focuses on lightweight lane level representations suitable for real time crowdsourced updates.

Lane boundaries are represented as a structured object that contains metadata and spatial information required for map fusion and localization. The lane boundary data structure is defined as follows:

```
struct LaneBoundary {
    uint32 time_stamp;            // Unix or UTC timestamp (ms)
    ID laneID;                    // Lane identifier (from OSM)
    ID roadID;                    // Road identifier (from OSM)
    float confidence;             // Detection confidence [0.0-1.0]
    LanePoints leftBoundary;      // Array of (lat, lon, alt) points
    LanePoints rightBoundary;     // Array of (lat, lon, alt) points
    LanePoints trajectory;        // Vehicle path (lat, lon, alt)
};
```

Each detected lane boundary is represented by a compact structured object designed for efficient fusion. The structure captures information using OpenStreetMap (OSM) identifiers and KML compatible coordinate arrays. Each lane object occupies an average of 22 KB for a 780 m segment, Hence, a three lane 1 km road segment (4 lane boundaries) occupies less than 100 KB in total, which is much smaller than LiDAR-based or dense-vision HD maps, which can exceed 1–2 GB per km as can be seen in table 4.5.

The compact data footprint significantly reduces both communication and storage overhead while maintaining sufficient geometric fidelity for lane-level localization. This efficiency confirms that the proposed approach is supporting the vision of scalable real time map updates for large fleets of autonomous vehicles.

Table 4.5: Comparison of HD-Map Data Representations

Method	**Sensor Type**	**Typical Data Size (per km)**	**Absolute Accuracy**	**Relative Accuracy**	**Remarks**
LiDAR-based (HERE HDLM [62], TomTom HD Map [63])	LiDAR + INS + RTK	1–2 GB	≤ 1.0 m (spec dependent)	≤ 0.20 m	Extremely accurate but data heavy and costly to maintain
Vision-based (Mobileye REM [64])	Cameras + GNSS	0.10 MB of semantic objects	-	≤ 0.10 m	Proprietary cloud-mapping network
Proposed Framework	Cameras + GNSS (Edge–Cloud)	< 0.10 MB (≈ 22 KB/lane)	≤ 1.5 m (2σ)	≤ 0.20 m (2σ)	Lightweight, scalable crowdsourced HD map

4.6.2 Practical Implications for AV Localization

Although an absolute error exceeding 1m may seem high for a high-definition map, modern autonomous vehicle (AV) stacks can effectively compensate for this error through lane-vision correction techniques (Figure 42).

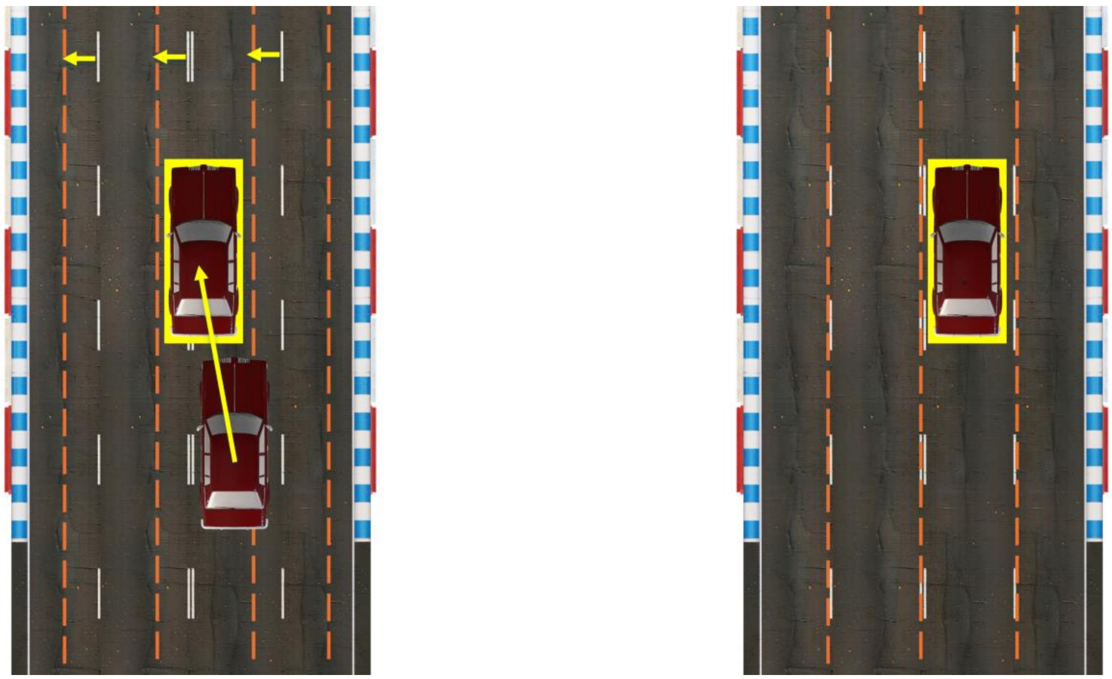

Figure 42 Vision-lane correction – left represent the global error in the map compared to GT & right shows the shifted pose after applying correction

Advances in vision hardware and software allow the perception system to accurately detect lane boundaries and use them to refine the vehicle's localization.

As long as the HD-map absolute error remains within half a lane width, the host vehicle can register to the correct lane in the map. The localization module then compares the host pose relative to the perceived lane boundaries with its global pose in the HD map and minimizes the discrepancy by adjusting the global host pose accordingly. This closed loop correction ensures reliable localization even when the map geometry has sub meter level residual errors.

This demonstrates that low-cost, camera-based sensing combined with geometric fusion can produce commercially viable HD-map outputs without the expense of using Mobile Mapping System (MMS) for data acquisition.

4.7 Chapter Summary

This chapter presented a comprehensive quantitative evaluation of the HD-map generation framework proposed in Chapter 3. Absolute, relative accuracy and map efficiency metrics were derived by comparing fused multi-vehicle lane data to ground-truth references over a ≈ 780 m highway segment.

Key findings include:

- Absolute accuracy: all lanes achieved RMSE and $2\sigma \leq 1.5$m, meeting industry standards.
- Relative accuracy: inter lane spacing error remained ≤ 0.20 m (2σ) across all lane pairs.
- Convergence: absolute error decreased below 1.5 m after ≈ 10 drives, confirming the stability of the multi-vehicle fusion process.
- Consistency: orientation alignment, shared ENU projection, and trimming strategies ensured geometrically consistent comparisons.
- Efficiency: a three lane 1 km road segment (4 lane boundaries) occupies less than 100 KB in total.

These outcomes verify that the proposed camera-based HD-map fusion system can achieve the positional requirements needed by autonomous driving stack and advanced driver assistance applications, offering a scalable and cost-effective alternative to LiDAR-centric mapping systems.

CHAPTER FIVE
CONCLUSION AND FUTURE WORK

This research proposed and validated a vision-based, multi-vehicle HD-map generation and fusion framework capable of achieving lane-level precision using only camera and GPS data. The primary goal was to demonstrate that reliable, cost-effective, and scalable HD-map creation can be achieved without dependence on LiDAR sensing by utilizing the onboard sensors when they are idle or on low demand.

The system architecture consisted of three core layers:

- Edge-vehicle preprocessing, where raw camera and GPS data are synchronized and filtered to extract lane boundary features.
- Cloud-based fusion, where multiple vehicle observations are geometrically aligned and merged using a Kabsch-based transformation and probabilistic aggregation.
- Distribution and update layer, which provides real-time HD-map tiles back to vehicles for localization and navigation.

To ensure quantitative rigor, the framework was tested on self-collected multi lane highway data, with each fused lane boundary aligned to an independent ground truth (GT) reference. The evaluation quantified both absolute and relative accuracy using symmetric point-to-curve distance metrics and projection-based inter-lane width comparisons.

5.1 Key Findings

The experimental results confirmed that the proposed system meets commercial HD-map accuracy standards.

Absolute Accuracy: Each fused lane achieved an RMSE between 0.98 m and 1.15 m, with 2σ values under 1.35 m and P95 below 2.1 m.

This demonstrates sub-meter mean alignment with GT and satisfies the global HD-map specifications (≤ 1.5 m @ 2σ).

Relative Accuracy: Inter lane spacing errors remained below 0.14 m (mean |ΔW|) and 0.18 m (2σ) across all lane pairs, confirming stable geometry and lane-to-lane consistency.

Convergence: Multi-drive fusion experiments showed that absolute RMSE converges below 1 m after approximately 10 drives, while relative accuracy stabilizes below 0.2 m (2σ).

Scalability and Efficiency: The edge–cloud design minimized bandwidth by uploading only lane features (~25KB/km) per lane, not raw imagery. The method supports incremental updates and large-scale deployment across fleets.

These findings collectively prove that a camera-only HD-map system can deliver lane-level fidelity comparable to LiDAR-based mapping at a fraction of the cost.

5.2 Limitations

Although promising, the framework has several limitations that define the boundary of its current performance:

1. Dependence on lane-mark visibility: performance of CLRNet degrades in heavy rain, night conditions, or when road paint is worn.
2. Localization reliance on GNSS: fusion accuracy assumes moderate GNSS stability, large drifts could propagate geometric offset and cause fusion problems.
3. Limited geographic scope: the evaluation was limited to a straight and slightly curved three-lane highway section, tight-curves and urban intersections remain to be tested.
4. Feature-level granularity: only lane boundaries (markers) were fused, additional semantic layers (traffic signs, signals, crosswalks) were not included.
5. Real-time constraints: the current implementation performs fusion offline, latency performance under live streaming conditions has not yet been validated.

5.3 Future Work

Building on these results, several research extensions are proposed:

1. Real-time incremental fusion**:** deploy the cloud aggregation process to operate continuously, enabling live HD-map updates from connected fleets.
2. Urban and curved-road validation: expand experiments to multi-curvature, intersection, and ramp environments to evaluate scalability of geometric alignment.
3. Sensor fusion enhancement: integrate IMU and radars to improve robustness under poor vision conditions.
4. Semantic map enrichment: enrich the lane-level map with additional features such as road edges, signs and stop-line geometry for full HD-map layers.
5. End-to-end integration with AV stack: embed the mapping module within a localization loop to evaluate real-time performance with lane-vision correction.

5.4 Closing Remark

This research demonstrated that constructing a high-definition map with high accuracy is achievable using low-cost sensors on vehicles with ADAS hardware like front facing camera combined with industrial GPS and geometric fusion. The developed framework achieved lane-level precision, sub-meter global alignment, and centimeter scale inter-lane consistency without LiDAR dependency.

The combination of edge preprocessing, cloud fusion, and automated evaluation creates a scalable solution for fleet-wide HD-map creation and maintenance. These findings contribute to the advancement of affordable, real-time HD-mapping technology, building a safer and more accessible autonomous-driving systems.

REFERENCES

[1] F. Götz, "The Data Deluge: What do we do with the data generated by AVs?," 22 1 2021. [Online]. Available: https://blogs.sw.siemens.com/polarion/the-data-deluge-what-do-we-do-with-the-data-generated-by-avs/.

[2] W. Kim, V. Añorve and B. Tefft, *American Driving Survey, 2014 – 2017 (Research Brief),* Washington, D.C: AAA Foundation for Traffic Safety., 2019.

[3] K. T. Asrat and H.-J. Cho, "A Comprehensive Survey on High-Definition Map Generation and Maintenance," *ISPRS Int. J. Geo-Inf.*, p. 232, 2024.

[4] M. Yang, K. Jiang, B. Wijaya, T. Wen, J. Miao, J. Huang, C. Zhong, W. Zhang, H. Chen and D. Yang, "Review and challenge: High definition map technology for intelligent connected vehicle,," 2024.

[5] M. Lu, K. Wevers and R. Van Der Heijden, "Technical Feasibility of Advanced Driver Assistance Systems (ADAS) for Road Traffic Safety," *Transportation Planning and Technology,* vol. 28, no. 3, p. 167–187, 2005.

[6] P. Mallozzi, P. Pelliccione, A. Knauss, C. Berger and N. Mohammadiha, "Autonomous Vehicles: State of the Art, Future Trends, and Challenges," *Automotive Systems and Software Engineering,* p. 347–367, 2019.

[7] G. Elghazaly, R. Frank, S. Harvey and S. Safko, "High-Definition Maps: Comprehensive Survey, Challenges, and Future Perspectives," *IEEE Open Journal of Intelligent Transportation Systems,* vol. 4, pp. 527-550, 2023.

[8] F. Li, C. Fu, D. Sun, H. Marzbani and M. Hu, "Reducing Redundancy in Maps without Lowering Accuracy: A Geometric Feature Fusion Approach for Simultaneous Localization and Mapping," *ISPRS International Journal of Geo-Information,* vol. 12, no. 6, p. 235, 2023.

[9] V. N. Pai and I. Barosan, "Maps and their impact on the functional safety of automated driving,," *Journal of Safety Engineering and Applied Sciences,* vol. 10, pp. 256-270, 2023.

[10] D. Katare, D. Perino, J. Nurmi, M. Warnier, M. Janssen and A. Y. Ding, "A Survey on Approximate Edge AI for Energy Efficient Autonomous Driving Services," *IEEE Communications Surveys & Tutorials,* vol. 25, no. 4, 2023.

[11] W. Y. Zhaolong Gao, "The real-time data processing framework for blockchain and edge computing,," *Alexandria Engineering Journal,,* vol. 120, pp. 50-61, 2025.

[12] J. Xie, X. Zhou and L. Cheng, "Edge computing for real-time decision making in autonomous driving: Review of challenges, solutions and future trends,," *International Journal of Advanced Computer Science and Applications,* vol. 15, pp. 59-69, 2024.

[13] H. Zhang, D. Paz, Y. Guo, A. Das, X. Huang, K. Haug, H. I. Christensen and L. Ren, "Enhancing Online Road Network Perception and Reasoning with Standard Definition Maps," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* Abu Dhabi, 2024.

[14] X. Zhou, Y. Li and D. Doria, "A comprehensive survey on high-definition map generation and updating for autonomous driving," *ISPRS International Journal of Geo-Information*, vol. 13, no. 7, 2024.

[15] P. Mooney and M. Minghini, "Review of OpenStreetMap Data," in *Mapping and the Citizen Sensor*, Ubiquity Press, 2017.

[16] Z. Bao, S. Hossain, H. Lang and X. Lin, "A review of high-definition map creation methods for autonomous driving," *Engineering Applications of Artificial Intelligence*, 2023.

[17] H. Lyu, J. S. B. Perez, Y. Huang, K. Li, M. Shan and S. Worrall, "Online High-Definition Map Construction for Autonomous Vehicles: A Comprehensive Survey," *Journal of Sensor and Actuator Networks*, vol. 14, no. 1, 2025.

[18] K. E, C. MB, T. AMG, V. N, M. T, K. H, K. A and H. E, "Accurate Calibration Scheme for a Multi-Camera Mobile Mapping System," *Remote Sensing*, vol. 11, no. 23, p. 2778, 2019.

[19] M. Elhashash, H. Albanwan and R. Qin, "A review of mobile mapping systems: From sensors to applications," *Sensors*, vol. 22, no. 11, p. 4262, 2022.

[20] L. Ma, Y. Li, J. Li, C. Wang, R. Wang and M. Chapman, "Mobile Laser Scanned Point-Clouds for Road Object Detection and Extraction: A Review," *Remote Sensing*, vol. 10, no. 10, p. 1531, 2018.

[21] F. Tsushima, N. Kishimoto, Y. Okada and W. Che, "CREATION OF HIGH DEFINITION MAP FOR AUTONOMOUS DRIVING," *The International*

Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vols. XLIII-B4-2020, pp. 415-420, 2020.

[22] S. M. Bateman, H. Ning Xu, C. Zhao, Y. B. Shalom, V. Gong, G. Long and W. Maddern, "Exploring Real World Map Change Generalization of Prior-Informed HD Map Prediction Models," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Seattle, 2024.

[23] T. Zheng, Y. Huang, Y. Liu, W. Tang, Z. Yang, D. Cai and X. He, "CLRNet: Cross Layer Refinement Network for Lane Detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, 2022.

[24] L. Tabelini, R. Berriel, T. M. Paixão, C. Badue, A. F. D. Souza and T. Oliveira-Santos, "Keep your Eyes on the Lane: Real-time Attention-guided Lane Detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[25] Y. Guo, G. Chen, P. Zhao, W. Zhang, J. Miao, J. Wang and T. E. Choe, "Gen-LaneNet: A Generalized and Scalable Approach for 3D Lane Detection," in *European Conference on Computer Vision*, 2020.

[26] F. Zhang, W. Shi, M. Chen, W. Huang and X. Liu, "Open HD map service model: an interoperable high-Definition map data model for autonomous driving," *International Journal of Digital Earth*, vol. 16, no. 1, p. 2089–2110, 2023.

[27] W. Jang, J. An, S. Lee, M. Cho, M. Sun and E. Kim, "Road Lane Semantic Segmentation for High Definition Map," in *IEEE Intelligent Vehicles Symposium (IV)*, Changshu, 2018.

[28] Y. Zhou, Y. Takeda, M. Tomizuka and W. Zhan, "Automatic Construction of Lane-level HD Maps for Urban Scene," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Prague, 2021.

[29] S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, F. Mccullough and A. Mouzakitis, "A Survey of the State-of-the-Art Localization Techniques and Their Potentials for Autonomous Vehicle Applications," *in IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 829-846, 2018.

[30] J. S. Berrio, S. Worrall, M. Shan and E. Nebot, "Long-Term Map Maintenance Pipeline for Autonomous Vehicles.," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, pp. 10427-10440, 2020.

[31] S. Liu, L. Liu, J. Tang, B. Yu, Y. Wang and W. Shi, "Edge Computing for Autonomous Driving: Opportunities and Challenges," *in Proceedings of the IEEE,* vol. 107, no. 8, pp. 1697-1716, 2019.

[32] B. Wijaya, K. Jiang, M. Yang, T. Wen, X. Tang and D. Yang, "Crowdsourced Road Semantics Mapping Based on Pixel-Wise Confidence Level," *Automotive Innovation*, vol. 5, p. 43–56, 2022.

[33] L. Gillam, K. Katsaros, M. Dianati and A. Mouzakitis, "Exploring edges for connected and autonomous driving," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Honolulu, 2018.

[34] A. MFA, Y. S, A. R. SF, A. A, M. A. AH and S. M., "Edge computing for Vehicle to Everything: a short review," *F1000Research*, vol. 10, 2023.

[35] Dziyauddin, R. Akmam, D. T. Niyato, N. C. Luong, M. A. M. Izhar, M. Hadhari and S. M. Daud, "Computation Offloading and Content Caching Delivery in Vehicular Edge Computing: A Survey," *Computer Networks,* 2019.

[36] A. Biswas and H.-C. Wang, "Autonomous Vehicles Enabled by the Integration of IoT, Edge Intelligence, 5G, and Blockchain," *Sensors,* vol. 23, no. 4, 2023.

[37] Q. Zhu, B. Yu, Z. Wang, J. Tang, Q. A. Chen, Z. Li, X. Liu, Y. Luo and L. Tu, Cloud and Edge Computing for Connected and Automated Vehicles, Now Foundations and Trends, 2024.

[38] L. Bai, J. Cao, M. Zhang and B. Li, "Collaborative Edge Intelligence for Autonomous Vehicles: Opportunities and Challenges," *IEEE Network,* vol. 19, no. 2, pp. 52-60, 2025.

[39] C. Sandu and I. Susnea, "Edge computing for autonomous vehicles - A scoping review," in *20th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, Iasi, 2021.

[40] P. A. Longley, M. F. Goodchild, D. J. Maguire and D. W. Rhind, in *Geographic Information Systems and Science*, Wiley and Sons, 2001.

[41] "GoPro Hero9 Black Specs," GoPro, [Online]. Available: https://www.dpreview.com/products/gopro/actioncams/gopro_hero9_black/specifications.

[42] Z. Zhang, "A Flexible New Technique for Camera Calibration.," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 22, no. 11, pp. 1330-1334, 2000.

[43] Marhworks, "What Is Camera Calibration?," Mathworks, [Online]. Available: https://www.mathworks.com/help/vision/ug/camera-calibration.html.

[44] Mathworks, "estimateMonoCameraFromScene," Mathworks, [Online]. Available: https://www.mathworks.com/help/driving/ref/estimatemonocamerafromscene.html. [Accessed 2025].

[45] X. Pan, J. Shi, P. Luo, X. Wang and X. Tang, "Spatial as deep: Spatial cnn for traffic scene understanding," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[46] TuSimple, "Tusimple benchmark," [Online]. Available: https://github.com/TuSimple/tusimple-benchmark/.

[47] K. Behrendt and R. Soussan, "Unsupervised labeled lane marker dataset generation using maps," in *International Conference on Computer Vision (ICCV)*, 2019.

[48] Mathworks, "laneBoundaryDetector," [Online]. Available: https://www.mathworks.com/help/driving/ref/laneboundarydetector.html?s_tid=srchtitle_support_results_1_laneBoundaryDetector.

[49] Mathworks, "detect," Mathworks, [Online]. Available: https://www.mathworks.com/help/driving/ref/laneboundarydetector.detect.html. [Accessed 2025].

[50] Mathworks, "imageToVehicle," Mathworks, [Online]. Available: https://www.mathworks.com/help/driving/ref/monocamera.imagetovehicle.html?s_tid=srchtitle_support_results_2_imageToVehicle+. [Accessed 2025].

[51] Mathworks, "findParabolicLaneBoundaries," Mathworks, [Online]. Available: https://www.mathworks.com/help/driving/ref/findparaboliclaneboundaries.html?s_tid=srchtitle_support_results_1_+findParabolicLaneBoundaries. [Accessed 2025].

[52] P. Torr and A. Zisserman, "MLESAC: A New Robust Estimator with Application to Estimating Image Geometry," *Computer Vision and Image Understanding,,* vol. 71, no. 1, pp. 138-156, 2000.

[53] Mathworks, "laneBoundaryTracker," Mathworks, [Online]. Available: https://www.mathworks.com/help/driving/ref/laneboundarytracker-system-object.html?s_tid=srchtitle_support_results_1_+laneBoundaryTracker. [Accessed 2025].

[54] D. Musicki and R. Evans, "Joint Integrated Probabilistic Data Association - JIPDA," *FUSION 2002,* vol. 2, no. 2, pp. 1120-1125, 2002.

[55] P. Kandal, *A glimpse into the future of Mapmaking with OSM,* Today Software Magazine.

[56] Mathworks, "interp1," Mathworks, [Online]. Available: https://www.mathworks.com/help/matlab/ref/double.interp1.html. [Accessed 2025].

[57] S. H, H. B, H. Q, Z. Y and S. J., "A Lightweight High Definition Mapping Method Based on Multi-Source Data Fusion Perception," *Applied Sciences,* vol. 13, no. 5, p. 3264, 2023.

[58] A. Shi, Y. Cai, X. Chen, J. Pu, Z. Fu and H. Lu., "GlobalMapNet: An Online Framework for Vectorized Global HD Map Construction," *ArXiv,* 2024.

[59] D. Shin, K.-m. Park and M. Park, "High Definition Map-Based Localization Using ADAS Environment Sensors for Application to Automated Driving Vehicles," *Applied Sciences,* vol. 10, no. 14, 2020.

[60] Anaconda, Anaconda, [Online]. Available: https://www.pythonanywhere.com/. [Accessed 2025].

[61] T. Yu, H. Huang, N. Jiang and T. D. Acharya, "Study on Relative Accuracy and Verification Method of High-Definition Maps for Autonomous Driving," *ISPRS International Journal of Geo-Information,* vol. 10, no. 11, p. 761, 2021.

[62] HERE, "HERE HD Live Map," [Online]. Available: https://www.here.com/platform/hd-live-map. [Accessed 2025].

[63] TomTom, "Orbis Maps for Automation," [Online]. Available: https://www.tomtom.com/products/orbis-maps-for-automation/. [Accessed 2025].

[64] Mobileye, "Road Experience Management," [Online]. Available: https://www.mobileye.com/technology/rem/. [Accessed 2025].