A Feasible Solution of the Local Dynamic Map Based on Binary Decision Diagrams and Spatial-Behavioral Granularity

A Thesis
submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

by

Arvind Kumar
(Student Number: 19899005)

Kyutech
Kyushu Institute of Technology

Graduate School of Life Science and Systems Engineering
Kyushu Institute of Technology
Japan

September, 2022

A Feasible Solution of the Local Dynamic Map Based on Binary Decision Diagrams and Spatial-Behavioral Granularity

by

Arvind Kumar

Submitted to the Graduate School of Life Science and Systems Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Author ....
Graduate School of Life Science and Systems Engineering
September, 2022

Certified by ....
Hiroaki Wagatsuma
Professor
Thesis Supervisor

Accepted by ....
WADA Chikamune
Professor
Chairman, Department Committee on Graduate Thesis

at the

Kyushu Institute of Technology

September 2022

Following Committee members examined this doctoral Thesis:

Professor ....
Chairman, Thesis Committee
Professor of Human Intelligence and Machines

Professor Keiichi Ueda ....
External Member, Thesis Committee
Department of Mathematics, University of Toyama, Japan

Professor Hiroaki Wagatsuma ....
Thesis Supervisor, Thesis Committee
Department of Human Intelligence Systems

Professor Keiichi Horio ....
Member, Thesis Committee
Department of Human Intelligence Systems

Professor Hakaru Tamukoh ....
Member, Thesis Committee
Department of Human Intelligence Systems

A Feasible Solution of the Local Dynamic Map Based on Binary Decision Diagrams and Spatial-Behavioral Granularity

by

Arvind Kumar

Submitted to the Graduate School of Life Science and Systems Engineering
on September, 2022, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Autonomous vehicles (AVs) have been increasing rapidly on the road in recent years. However, the safety of AVs is of significant concern, which we must ensure. AVs use sensor information to achieve autonomy, but sensors such as cameras and lidar have limitations, and vehicles cannot rely on them entirely for safe navigation. To assist AVs with static information, high-definition maps (HD maps) can facilitate the complex static details of the surrounding for safe autonomy. However, we can model complex static information using HD maps for navigation; detecting and maintaining the traffic participant's dynamic information using sensors of the ego vehicle alone is still a significant concern for safe navigation. In such a situation of sensing limitations, Cooperative Intelligent Transport Systems (C-ITS) is one approach to facilitate vehicle navigation through sharing information between the traffic participants.

The C-ITS approach has various Intelligent transportation system (ITS) station units, namely Personal, Vehicle, Road-side and Central ITS station units. A Local Dynamic Map (LDM) is a critical component in any ITS station's facilities layer. LDM is one way to maintain static and dynamic information of the traffic participants in a consistent geometrical way. It is a necessary facility in C-ITS to share sensor information between participating traffic agents. Moreover, it maintains information about the objects that are either part of the traffic or influenced by it.

The International Organization for Standardization (ISO) and European Telecommunications Standards Institute (ETSI) have also made standardization efforts. Since its inception in the SAFESPOT project, implementations of LDM have been mostly four-layer data organizations. Where Layer 1 and Layer 2 maintain static information and transient static information. Then, Layer 3 and Layer 4 contain transient dynamic and highly dynamic data. Depending upon the requirement, the LDM community realized memory-based or database-based LDM. We utilized the decision diagram to enhance the safety aspect of the traffic participants in the memory/ database-based LDM setup. We utilized Shared Binary Decision Diagram (SBDD) and Geohash

granular properties to detect the near-miss situation, i.e. when vehicles come very close.

However, besides DynaMap, there is also a common understanding since the SAFESPOT project introduced LDM to use the database and supported query language to retrieve data from the LDM. Hence, most implementations use different databases and query languages to execute it. Although, the LDM community has explored LDM depending on the database variants. Nevertheless, remarkably less emphasis has been given to the type of data stored in the LDM. This thesis attempted to fill this gap in the LDM to enhance the moving vehicle's safety aspect. We proposed a novel method of data representation for vehicle future geographical occupancy information using a binary decision diagram (BDD). We show that sharing BDD-based information is consistent with the C-ITS nature of the data sharing since the algebraic operation between the exchanged BDDs can confirm the possibility of future interaction. We calculated potential future occupancy using Kamm's circle, shown in the ROS-based simulator and modified the mid-point circle generation algorithm to find the BDD representing the set of Geohash enclosing the Kamm's circle. We also reported data insertion and collision avoidance check time of the linked list-based BDD on PostgreSQL database-based LDM.

Keyword

Local Dyanmic Map, Database, Cooperative Intelligent Transport Systems, Binary Decision Diagrams, Kamm's Circle, Autonomous Vehicles, Collision avoidance.

Acknowledgments

There have been many people who helped during this exciting journey of PhD. I appreciate their guidance and thank them sincerely.

First of all, I would like to thank my PhD supervisor Professor Hiroaki Wagatsuma for giving me the chance to work in his laboratory. I learned a lot from him; his lessons helped me in research and grew as a professional.

Further, big thanks to staff members in the Wagatsuma laboratory: Kaoru Ono, Chie Osa and Kurumi Nakamura, san, for their support whenever facing language/ documentation issues. I also express gratitude to KYUTECH, The New Energy and Industrial Technology Development Organization(NEDO) and Makino Milling Machine Co., Ltd.

My colleagues in the laboratory: Jisha Maniamma, Ankur Dixit, Dondogjamts Batbaatar, Pancho Dachkinov, Amarbold Purev, and Manu Shrivastava, made this PhD full of fun. Also, I sincerely appreciate Natsuki Shirasawa, Keita Kawano, Satoru Mishima, Huynh Le Phu Trung and Wataru Oshiumi's help during my stay on campus.

Special thanks to Manipal University Jaipur, Jaipur, for their encouragement to go for PhD during my stay there. At last, I would like to thank my mother, Smt. Prem Kumari, brother Pradeep kumar, and sister Anjali pal for their encouragement and belief in me. Arigato to my wife, Vibha pali, for her tremendous support throughout the PhD journey.

Arvind Kumar

Contents

List of Figures

List of Tables

Acronyms

AD Autonomous Driving

AVs Autonomous vehicles

BDD binary decision diagram

C-ITS Cooperative Intelligent Transport Systems

CAMs Cooperative Awareness Messages

DDM Dynamic Distributed Maps

DENMs Decentralized Environmental Notification Messages

DPM Dynamic Public Map

ETSI European Telecommunications Standards Institute

HD maps high-definition maps

ISO International Organization for Standardization

ITS Intelligent transportation system

LDM Local Dynamic Map

MTBDD Multi-Terminal Binary Decision Diagram

OBDD Ordered Binary Decision Diagram

OSM OpenStreetMap

R-LDM Relational Local Dynamic Map

ROBDD Reduced Ordered Binary Decision Diagram

ROS Robot Operating System

SBDD Shared Binary Decision Diagram

V2I Vehicle to Infrastructure

V2V Vehicle to Vehicle

V2X Vehicle to Everything

VANets Vehicular Ad hoc Networks

ZDD Zero suppressed Decision Diagram

Chapter 1

Introduction

1.1 Background

Industry 5.0 and Industry 4.0 talk about the connection between the physical and virtual worlds. Industry 4.0 is considered a more technological-driven transformation. Unlike previous industrial revolutions, Industry 5.0 focuses on achieving societal goals beyond jobs and growth and is more human-centric, resilient and sustainable. It is a more value-driven initiative, not a technology-driven revolution but will lead to technological transformations, leading to more value generation in the economy, ecology, and Society. In Japan Council for Science, Technology and Innovation made an initiative of Society 5.0 in which every person can lead an active and enjoyable life with the help of more human-centric technologies like in Industry 5.0 [3]. Today both industrial revolutions are considered to exist side-by-side. The terminologies used in 5.0 and 4.0 industries may vary, but there is a cross-over between the technologies [4]. So, many technologies in Industry 4.0 will benefit in achieving Industry 5.0.

In Society 5.0, mobility will play a vital role by making movement pleasant, without congestion and accident-free through autonomous driving. Nevertheless, it has become crucial to use Industry 4.0 technology for moving objects, e.g., vehicles, to share data to make mobility smooth and safe since AVs cannot depend solely on the sensors because of their limitations [5]. For example, LIDAR and cameras may suffer from limited vision during the rain [6]. Therefore, the connection between them and sharing data may enhance vision and the ability for safe, cooperative planning of the vehicles. C-ITS is one such approach for sharing data between vehicles.

Also, According to SAE (Society of Automotive Engineers) International in the United States, the international standard SAE J3016 was published in 2014 in the first place, revised three times and released in the form of the latest version in 2021 [7], which defines ADAS (Advanced Driving Assistance System) and ADS (Automated Driving System) clearly. Simultaneously, European Commission has strongly promoted industry-government academia research group projects on intelligent transport systems in the framework of Horizon 2020, which focused on domains of CCAM (Cooperative, connected and automated mobility) and C-ITS (Cooperative Intelligent Transport Systems) [8, 9] with white papers and reports in 2017-2018. The aim of the establishment of sustainable mobility extended from technological aspects; they have encouraged a developmental process from three evolutions as C-ITS, CV (Connected Vehicles) and AV (Automated Vehicles) to a practical CCAM realization in Society.

C-ITS main aim is to improve transport in terms of safety (e.g., crash avoidance, obstacle detection), efficiency (e.g., navigation, lane access control) and comfort (e.g., parking) using information and communication technologies [10]. Therefore, it exchanges information with its surrounding Vehicles, Infrastructure (roadside/ urban), service providers (map providers), pedestrians, and more. Since there is a large amount of information exchange between traffic participants, we need an efficient way to handle information exchange between the concerned objects. To handle the above critical information exchange, LDM plays a vital role, which is a critical component in C-ITS. As we know, autonomous Intelligent Transport Systems use only sensor information of the ego vehicle for navigation. In contrast, C-ITS uses sensor information from the ego vehicle and sensor information from nearby vehicles or infrastructure for navigation or planning tasks. Therefore C-ITS makes use of cooperation and exchanges information with nearby infrastructure or vehicles using Vehicle to Infrastructure (V2I)/ Vehicle to Vehicle (V2V) or both Vehicle to Everything (V2X) for ego vehicles operation. Thus the above message exchange between the vehicle and infrastructure is one of the vital components of the cooperative operation of the traf-

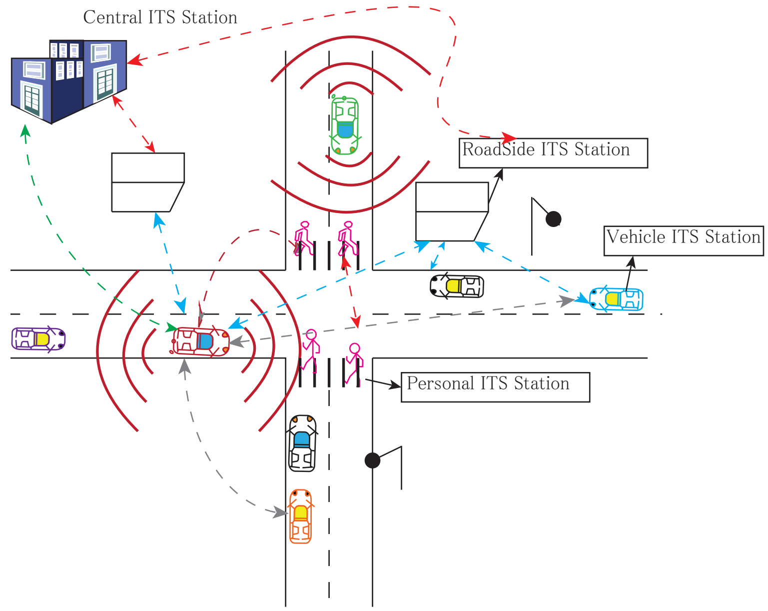


Figure 1-1: Different types of ITS stations.

fic participants. Various types of ITS stations are involved in the traffic situation as follows [11] (please refer to figure: 1-1).

1. Personal ITS station - Personal ITS subsystem (ITS equipment used by an individual) provides communication and application functionality in hand-held devices such as PDAs or mobile phones. It contains a personal ITS station. The devices used as an ITS station can connect/ interact with other ITS stations.

2. Vehicle ITS station - The Vehicle ITS subsystem, i.e. ITS equipment used in the Vehicle, contains the Vehicle ITS station.

3. Roadside ITS station - Roadside subsystem contains Roadside ITS stations. Mostly these subsystems are mounted near the road, for example, on the gantries and poles.

4. Central ITS station - Central ITS subsystem contains the Central ITS station, which is part of an ITS central system.

Every ITS stations consist of 7 layer architecture as follows (please refer figure 1-2):

1. Applications.

2. Management.

3. Communications.

4. Facilities.

5. Networking and Transport.

6. Accesss.

7. Security.

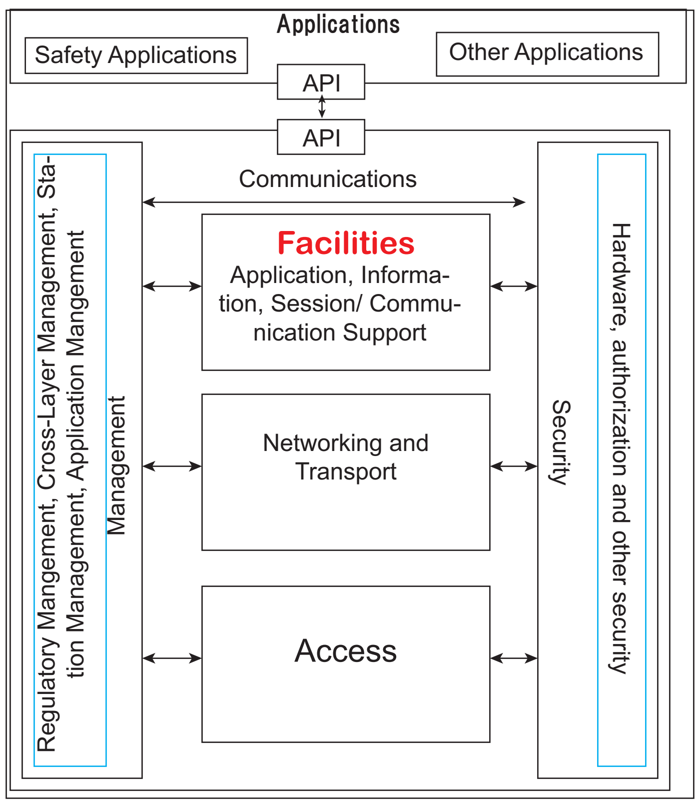


Figure 1-2: Seven layer architecture in an ITS station.

In ITS stations, the facilities layer is responsible for providing various facilities to support cooperative traffic operations. Furthermore, information support is one of the functions in this layer we focus on in our Thesis. The information support of the

ITS station contains the LDM as shown in figure 1-3, which is responsible for storing the static and dynamic components of the traffic in a geometrically consistent way.

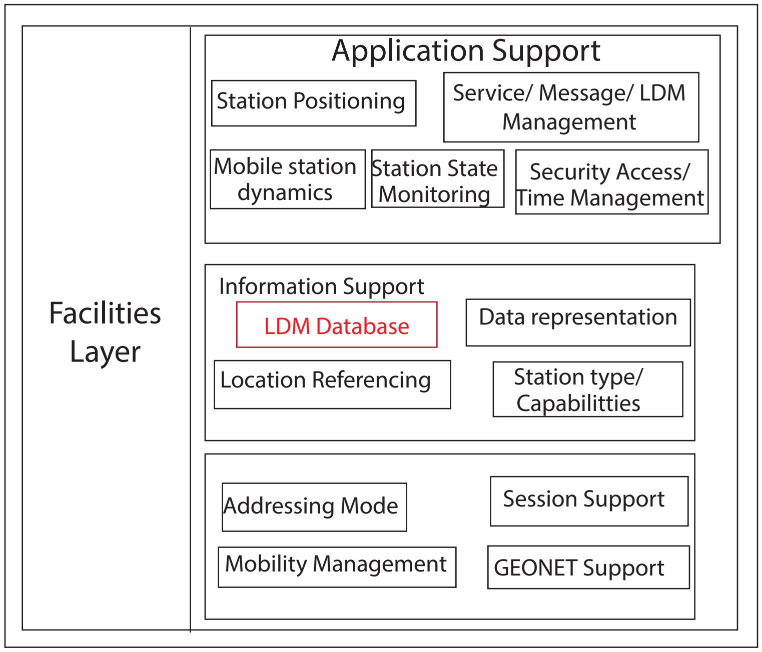


Figure 1-3: LDM in facilities layer of ITS station.

To handle or accommodate the message/status exchange in the traffic, LDM plays a significant role. LDM is a crucial facility in C-ITS. Please refer figure: 1-3. It maintains information about the objects that are either part of the traffic or influenced by it. LDM is divided into layers depending upon the dynamicity of data stored. It contains four layers having:

1. Layer 1: Contains permanent static information. It contains detailed information of a road map with application to ADAS. e.g., Map Data.

2. Layer 2: This layer is an extension of layer 1. It includes quasi-static information. e.g., Traffic signs, trees, buildings.

3. Layer 3: This layer contains temporary information for a particular region. e.g., Traffic jams, weather conditions, traffic signals.

4. Layer 4: This layer contains temporary information about dynamic or highly dynamic objects, e.g., moving vehicles and pedestrians.

LDM maintains static (geographical) and dynamic (traffic participants) information. The type of ITS station (See Figure 1-1) mainly decides the size of the geographical area and the number of traffic participants will be needed to manage; the requirement of Infrastructure based LDM vary from the vehicle-based LDM [12].

Hence, LDM is a data store, which traffic participants may utilize to know information about the static, e.g. road geometry information and dynamic information like vehicle position information. However, although the LDM community has focused on realizing the LDM using different databases [13, 14, 15], minimal emphasis has been given to the type of data stored itself. Therefore, in this Thesis, we focused on the data stored to improve the safety aspect of traffic participants in the LDM setup. The objective of this Thesis is discussed in the following section.

1.2 Objectives

The initial standard given by ETSI is TR 102 863 (V1.1.1) [2], which described LDM as an embedded conceptual data store in an ITS station that maintains the topographical, positional, and status information related to the ITS station within the host stations geographic area. It identified LDM as a key facility function in the facilities layer. Essential data sources of LDM are Cooperative Awareness Messages (CAMs) and Decentralized Environmental Notification Messages (DENMs). Standard discussed various applications of the LDM, 'Cooperative navigation Location-based services' are one of them, which can provide location-based information for cooperative navigation. In addition standard mentioned various types of data, namely, Type1 (static), Type2 (transient static), Type3 (transient dynamic) and Type4 (Highly dynamic). Also, the standard discussed highly dynamic data (Type 4) information content for nearby vehicles in which our work is an extension in an LDM to support the vehicle occupancy field. See table 1.1.

Table 1.1: Nearby vehicle information content for Layer 4 data. [2]

Information content	Type	Status
Vehicle identifier	Pseudonymous identity	
Vehicle position	• Latitude • Longitude • Position confidence (%) • Elevation • Elevation confidence (%) • Heading • Heading confidence (%)	
Vehicle type	One of the following: • Car • lobby • ...	
...	...	...
Route navigation advice	• Direction of next routed turn • Distance to next routed turn • Distance to next stop line	
Vehicle occupancy (%)		
...	...	...

The document also highlighted the requirement of the mechanism to update the LDM by storing processed information on the required objects back into the LDM to make it available for other applications. ITS applications analysis : functionality portion of the standard mentioned use cases from the LDM, out of which we believe our approach may benefit the following use cases UC_CA_03 (Across traffic turn collision risk warning), UC_CA_04 (Merging Traffic Turn Collision Risk Warning), UC_CA_05 (Cooperative merging assistance), UC_CA_06 (Intersection collision warning), UC_CA_07 (Cooperative forward collision warning).

This thesis aims to enhance the safety aspect of mobility in Society 5.0 by improving the information content of the LDM in the C-ITS setup. In our case, we achieve this requirement by improving the vehicle occupancy field to store current and reachable positions in the near future to facilitate collision avoidance scenarios and improve the traffic objects safety.

Mainly two problems are addressed:-

(a) Detection of the vehicles presents near the ego vehicle to facilitate the near-miss detection.

(b) The future geographical occupancy information of the participating vehicles consistent with the C-ITS nature of data sharing was introduced in the LDM to detect potential interaction of the vehicles in the near future.

To achieve the above objectives, we used binary decision diagrams. Shared Binary Decision Diagram (SBDD) was used to facilitate near-miss detection, and we used algebra over Reduced Ordered Binary Decision Diagrams (ROBDDs) to verify the collision avoidance. Furthermore, Geohash was used to represent the concerned geographical space because Geohash efficient partitioning of geographical locations as a boolean string helped us to treat geographical problems as a Boolean string manipulation. Also, partitioning the earths surface using a set of bounding boxes as Geohash allows us to solve the representation problem as there are infinite numbers of points to be mapped otherwise. A set of Geohash was encoded into Decision Diagrams as a representation for the bigger geographical space. Finally, we used Kamm's circle method to estimate the geographical space a vehicle can reach soon and express them in the ROS-based simulator CoincarSIM. We reported geographical occupancy data insertion and collision avoidance check time of the linked list-based BDD on PostgreSQL database-based LDM.

1.3 Key Features

This Thesis has the following main contributions listed below:

(a) Successfully utilized the features of decision diagrams in the LDM setup to improve the safety aspect of the traffic participants.

(b) SBDD, along with Geohash, was used to detect near-miss situations of the vehicles in the LDM setup.

(c) We demonstrated the Vehicle's future geographical occupancy over time as a feature in the LDM.

(d) Using a BDD, we proposed a novel data representation method for a vehicle's geographical occupancy information.

(e) We established that algebraic operations between the exchanged BDDs can confirm the possibility of future interaction, which is consistent with the C-ITS nature of data sharing.

(f) We presented the data insertion and collision avoidance check time of the linked-list-based BDD on the PostgreSQL database-based LDM.

(g) We modified the mid-point circle generation algorithm to develop a BDD for the Geohash set enclosing the Kamm's circle of a given radius.

Hence, an information processing approach was adopted, i.e. information stored in the LDM is not raw sensor data but processed information to make it practical in the safety aspect of the moving vehicles. Furthermore, Geohash made it suitable to be used by any traffic participants without any coordinate transformation. At last, JSON format was adopted to store encoded decision diagrams, making it suitable for widely available databases or LDM.

1.4 Organization

Chapter 2, Literature Review: In this chapter, we reviewed the state-of-the-art for LDM approach. Also, it examines the Geohash, HD maps, reachability and Decision diagrams. Finally, we compared the essential studies published on LDM, which are vital for the current study.

Chapter 3, Methodology: In this chapter, we described the methodologies we used to achieve our objective in this Dissertation. We discussed static and dynamic maps suitable from an autonomous driving perspective. Also, the chapter examined the construction of the Lanelet map using the JOSM tool and a scenario in the ROS-based CoinCar-SIM simulator. Next, we highlighted the importance of LDM in this domain. We illustrated the mesh and its application in various areas and then emphasized Geohash as a mesh and its significance. After that, we presented the ROS framework we used to create a scenario in our case. Then we discussed reachability analysis, Kamms circle and their applications in the domain of ADs.

Furthermore, We explained the Decision Diagrams (mainly ROBDD and SBDD) and algebra supported by ROBDDs. Thereon, we presented Geohash and Kamms circle and the usefulness of BDDs. At last, we ascertained LDM implementations using Relational and Graph databases.

Chapter 4, Results: First, in this chapter, the Dissertation discusses the risky area around the moving Vehicle. We divided the road segment (a lanelet in this case) into the Voronoi region and guessed that a dangerous place for any other non-ego vehicle is the region where the car is about to enter next; we presented the results achieved. Then, we highlighted the issue with the above approach and solved the above issue in following of the chapter. Furthermore, in **Section 4.1**, in another experiment, we proposed to use Geohash for the localization of participating moving agents in the experiment as a region for collision avoidance. We reason to encode the Geohash into a decision diagram (here, SBDD) makes comparison faster to check that vehicles are present at nearby locations than Geohash string matching or calculating the geographical distance between the two vehicles using floating point calculations when queried. In **Section 4.2**, we used binary decision diagram to maintain spatial location reachable by the Vehicle over time. We showed how to find neighbouring Geohash and Geohashes inside the reachable Kamm's circle. Encoded overapproximated vehicle position over time into the BDD displayed that collision check operations over BDDs are helpful for safety check operations like collision avoidance. We used the algebraic operation over BDDs for collision avoidance checks.

Chapter 5, Discussion: This chapter discusses the limitation which may arise while using our approach. First, we mentioned the challenges we may face while using SBDD. Also, we discussed the limitation due to Kamm's circle and ROBDD data structure and possible solutions which may be helpful. Further, we presented how we could reduce the overapproximation in our approach by incorporating lane restriction in the LDM setup and also presented the approach that may help handle the uncertainty due to communication delay/ data loss cases in the C-ITS scenario.

Chapter 6, Summary: At last, in this chapter, we summarized our overall approach and results. Then, we discussed the problem solved and how the approach discussed can be improved in the future, and finally, we put the main points of the Thesis and concluded our work .

Chapter 2

Literature Review

This chapter reviewed the state-of-the-art LDM approach, HD maps, Geohash, occupancy prediction and the decision diagrams. At last, we summarize the LDM approach relevant to the current research.

2.1 Local Dynamic Map

The initial standard given by ETSI is TR 102 863 (V1.1.1) [2]; it describes LDM as an embedded conceptual data store in an ITS station that maintains the topographical, positional, and status information related to the ITS station within the host station's geographic area. Along with ETSI, the ISO has also made standardization efforts. Since LDM's introduction in the SAFESPOT [16] project, it has been common to implement it using a database and query language to query information. SAFESPOT project has made a pivotal effort to introduce the concept of LDM to improve road safety in the cooperative scenario. The project ended in 2010, but part of its project report and published papers by its members are available at [17, 18, 19, 20, 21]. Depending upon the type of databases available, mainly relational/graph and streaming databases type of LDM varies [13, 14, 15]. To store and monitor the data for the ITS station to handle them for various dynamicity involved in the traffic scenarios based on the world model. Out of which graph database provides good performance for a large amount of data due to its underlying technology [14, 22]. However, there is no published result comparing the performance of the different databases available, as per best of our knowledge.

In [22] the standard data format OpenLABEL was discussed for all the sensor information before storing it in the database. It is concerned in the above paper that, unfortunately, the research community primarily ignored the standard data format for an LDM. However, it has been a critical component of the C-ITS. It used a standard OpenLABEL JSON data format for data annotation in autonomous driving cases.

ETSI EN 302 895 (V1.1.0) [23] extended the previous report and added new functionalities like compositional data structures and LDM Data Providers/ Customers. International standards are ISO/TS 17931:2013 [24] , and ISO/TS 18750:2015 [25] report defines a comparable standard to ETSI. Eggert et al. [14] proposed Relational Local Dynamic Map (R-LDM), which is a fully interconnected graph-based approach instead of layered. The authors claimed to represent a consistent world model with this approach. It used the Neo4j database and CYPHER query language to implement the LDM and used it for camera-to-map alignment and risk-based behaviour evaluation. Eiter et al. [15] used semantic web technologies, here ontologies combined with spatial stream database. LDM ontology with expressive spatial-stream query language helped to infer new information over streams. The authors showed the integration of semantic web technologies with LDM and V2X. The experiment involved the PostgreSQL extension PIPELINEDB database and PTV VISSIM simulation environment. Netten et al. introduced DynaMap. The authors emphasized that the dynamic map requirement for roadside units is different from the dynamic map for vehicles. It is a dynamic map for Roadside or Central ITS Stations. It proposed a novel architecture for world models, world objects, and data sinks. Koenders et al. [26] utilized the fact that LDM cannot store the data of all things all the time. Hence, the author introduced a streamed filtering technique to delete irrelevant data. Moreover, it used their relational schema, which has tables for areas, roads and objects. Zoghby et al. [27] built distributed LDM in the context of Vehicular Ad hoc Networks (VANets). Vehicles cooperate to increase their field of view. It created an extended map called Dynamic Public Map (DPM), depending upon Dynamic Distributed Maps (DDM). Simulation having many vehicles validates the distributed dynamic map. Shimada et al. [13] implemented the LDM using the specification given by the SAFESPOT project to evaluate the performance of the LDM while changing the number of vehicles and the computer environment for the collision detection task. Authors implemented LDM using Postgres database with PostGIS extension and loaded map in the database using. osm2pgsql tool for data in static layer concerning tables. PreScan and Simulink created a simulation environment to populate the dynamic layer tables.

2.2 HD maps/ Geohash

In [28, 29, 30] discussed AV's navigation across cities facing various traffic scenarios and obstacles. The above publication mentioned the importance of HD maps modelling the static details of the traffic. An OpenStreetMap (OSM) is a widely available free mapping project available online. OpenDrive [31] is the map format that is useful for the detailed mapping of the static scenario available in the traffic and is suitable for the AVs navigation support. Lanelet [32, 33] discussed a different format for the detailed mapping of the static scenario in the traffic arena. Lanelets are the extension of the OSM format suitable to handle complex road scenarios at a lane level accuracy. Geohash [34], GeoSOT [35] and GeoSOT-3D [36] are some of the methods to divide the geographic coordinates into a grid in a hierarchical mesh and assign code. To represent any position on the earth, it uses a specific grid and has a corresponding code.

2.3 Occupancy Prediction

Barth et. al. [37] and Eidehall et. al. in [38] worked with single future vehicle occupancy of the vehicle soon later. Barth et. al. [37] detected the full-motion state of the vehicle, including velocity, acceleration and yaw rate. Eidehall et. al. [38] worked upon the auto brake system and predicted the path which would be optimal for the ego vehicle based on the predicted positions of all the objects present in the scene. Multiple studies based on a countable set of future predicted paths are available at [39, 40, 41], and predictions with the associated probability distribution are in [42, 43, 44]. Although the above work results have contributed to the future position prediction of the traffic participants for our case in this Thesis, we used the findings from work of Althoff et. al. in [45][46] to predict possible worst-case occupancy.

2.4 Decision Diagrams

Sheldon B. Akers [47] first introduced the concept of representing Boolean function in terms of a diagram. Later, Randal E. Bryant [68] introduced Reduced Ordered Binary Decision Diagram (ROBDD). Due to the evolution of Decision Diagrams over the years, BDDs have many variants like ROBDD; Zero suppressed Decision Diagram Zero suppressed Decision Diagram (ZDD) [48], Shared Binary Decision Diagram Shared Binary Decision Diagram (SBDD) [48], Multi-Terminal Binary Decision Diagram Multi-Terminal Binary Decision Diagram (MTBDD) [49] and many more. This Thesis used SBDD and ROBDD for the functional enhancement of the LDM.

Following is the summary of the research on the LDM described above:-

Table 2.1: Summary state-of-the-art of the LDM.

Papers	Description
[17, 18, 19, 20, 21]	The Local Dynamic Map was first introduced in the SAFESPOT project. SAFESPOT project has made a pivotal effort to introduce the concept of LDM to improve road safety in the cooperative scenario. The project ended in 2010, but part of its project report and published papers by its members are available. SAFESPOT project modelled LDM as a four layer model.
[2, 23, 24, 25]	The International Organization for Standardization (ISO) and European Telecommunications Standards Institute (ETSI) and have also made standardization efforts. Standard mentioned about data which are needed to be stored in the database as well as its complete role in the Intelligent Transport Systems.
[13]	First, to implement the LDM and publish its performance results. The database used was Postgres and Postgis (Same as the SAFESPOT project). The paper shows that as the number of vehicles increased, the LDM internal processing experienced a high load.
[14]	First, to implement LDM using a Graph database. The database used was Neo4j. The paper shows that a real-world model of a traffic scenario can be best modelled using a graph database.
[15]	Implemented LDM using streaming database. Ontology was used to query the database.
[12]	Dynamap: Emphasized that the dynamic map requirement for roadside units is different from the dynamic map for vehicles. It is a dynamic map for Roadside or Central ITS Stations.
[22]	Implemented LDM using Graph database and converted sensor data to OpenLABEL format, a standard JSON file for all the sensor data. It increases the interoperability of the data as JSON files can easily be transferred over the network and stored in the database. Also, the OpenLABEL format acts as a standard so every other vehicle can understand it.

Chapter 3

Methodology

This chapter provides a background of the technologies that support an autonomous driving project and is needed to understand this Thesis. First, in section 3.1, we introduced the type of maps used to manage static and dynamic information of the traffic participants. Subsequently, we provided insights into the domain of the Mesh in section 3.2, Occupancy Prediction in section 3.3, after that in section 3.4 described Decision Diagrams and at last, we discussed the database used to realize LDM in this Thesis.

3.1 Maps in Autonomous Driving

3.1.1 Static HD maps

HD maps are essential for autonomous driving [28, 29, 30]. The purpose of HD maps is to provide correct information about the vehicles surroundings. They carry high importance because sensors have their limitations due to various conditions (e.g. Rain is terrible for LIDAR), occlusions, and sensor range; also, traffic encompasses rules and regulations, which are very difficult for any sensor to detect always. Therefore, maps help in reducing the uncertainty arised due to sensor information. Nevertheless, to support functions in autonomous driving, maps should be able to describe complex traffic scenarios on the highway and the city situations. Therefore, HD maps act as knowledge for the navigating vehicle to know the rules and regulations for its lane and environment.

Thus due to complex requirements in the AD situation, we need detailed information on the situation. Therefore research community on ADs has made various attempts. Although the widely available free mapping project OSM [50] is available; it cannot model complex city scenarios, requiring lane-level accurate information with various rules. In this direction, various companies use maps by commercial map providers like Here [51, 52] and TomTom [53, 54]. In this direction, OpenDrive [31] and Lanelet [32, 33] are the two freely available versions of HD maps, and AD domains are widely using these map formats. Although OpenDrive is comprehensive enough to create the details of the complex traffic scenarios, there is no freely available library to interpret and process the data. We can easily modify the lanelet map using the voluntarily available tool JOSM [55] after adding 'lanelets.mapcss' and 'lines.mapcss' style files in the JOSM editor style and 'laneletpresets.xml' for as tagging presets. We choose Lanelet as a mapping platform to model static information for our case for the above reason. In particular, we model Lanelet information in the database by storing corresponding information to create a static layer of the LDM. Lanelet uses OSM based XML representation of the data. It consists of three layers:

1. Layer 1 (physical layer): consists of visual elements on the road (points and linestrings).

2. Layer 2 (relational layer): The physical layer elements are connected to create the lanes, areas and traffic rules (lanelet, area and regulatory elements).

3. Layer 3 (topological layer): Deal with context and neighbourhood relationships of the relational layer.

Points are the fundamental element of the Lanelet map. It can represent a point on a road, poles or trees etc. Two or more Points combine to form LineStrings (e.g. road markings, curbs). These linestrings combine to form the lanelets. Non-differentiable cases may arise, but a possible solution exists [56]. Lanelets are the atomic section of the map where the directed motion occurs, i.e. within lanelet, traffic rules do not change, and their topological relationship with other lanelets will also not change. Therefore, we can combine these lanelets to form a complex physical map layer. See figures 3-1, 3-2 and 3-3.

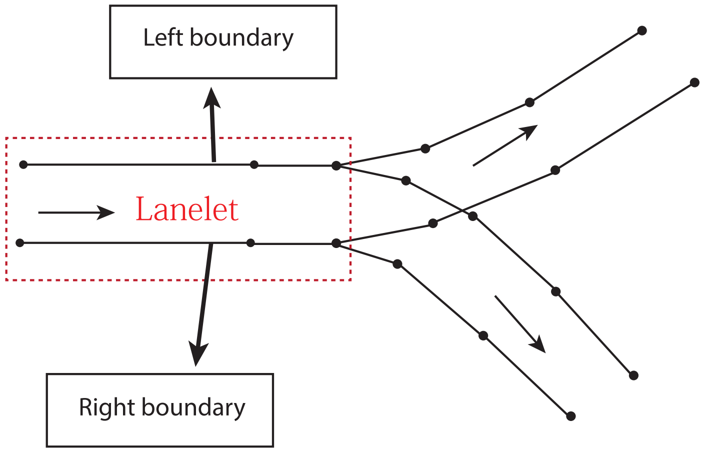


Figure 3-1: Lanelet.

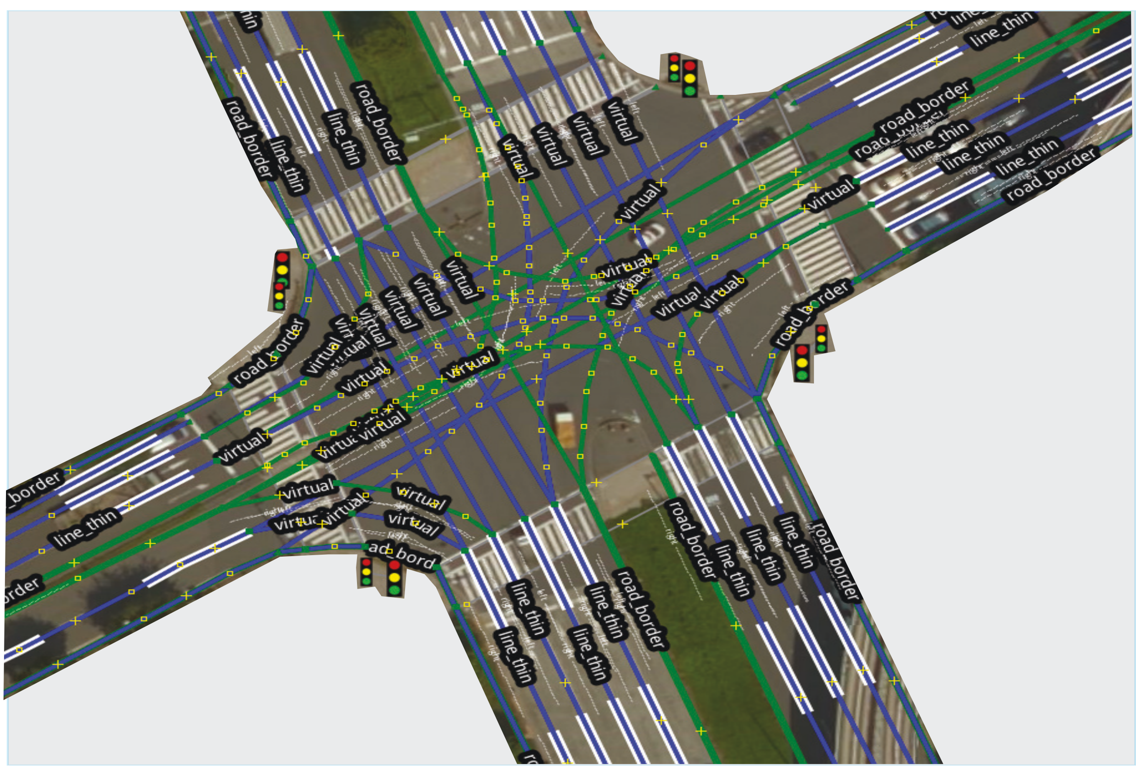


Figure 3-2: Lanelet map for an intersection scenario.

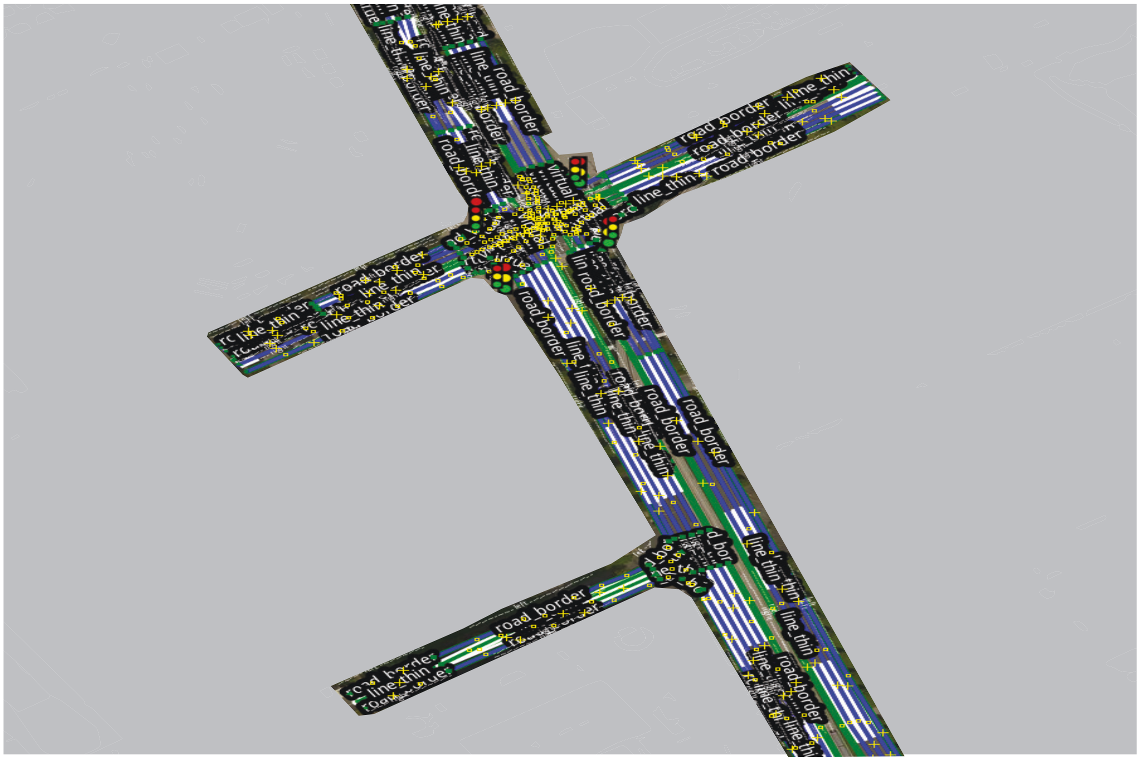


Figure 3-3: Lanelet map for a city Road.

3.1.2 Dynamic maps

As static maps are essential for developing complex traffic scenarios which do not move, on the other hand, dynamic functionality in the map is also a fundamental requirement to keep track of the moving entities in the traffic. Furthermore, we need dynamic information about the traffic participants since safe predictive driving is the most significant challenge in addition to the precise vehicle localization [57]. Most dynamic map implementations utilize a four-layer model to organize data, as mentioned in LDM. In these layers, one and layer two contain static and transient static information, plus layers three and four have transient dynamic and highly dynamic data where update time is less than a month, an hour, a minute and a second for the layers 1, 2, 3 and 4 respectively. Although six-layer model is proposed in [58, 59].

3.2 Geograhical Mesh and Coding

Geographical data can be divided broadly into three types [60]:

1. Vector
2. Raster
3. Mesh

Raster is the simplest form of a mesh. However, vector or raster is not always suitable for modelling the natural world features, such as to model hydrology, and metrology data mesh data are more suitable than vector or raster. Moreover, the mesh can be in many forms like unstructured, structured or mixed meshes. However, a mesh is highly suitable for many rendering applications [61]. Nevertheless, for this Thesis, we needed a geographic mesh with associated code to support our application's localization.

Geographic mesh plays a vital role in organizing spatial data and assigning corresponding code. It divides the earth's surface into a multi-level grid without spatial overlap. Mesh model and associated grid coding system have added advantages in many aspects, like increasing spatial retrieval efficiency through dimensional reduction or data management of massive distributive datasets. Geohash [34], GeoSOT [35] and GeoSOT-3D [36] are some of the methods to divide the geographic coordinates into a grid in a hierarchical mesh and assign code [62]. Among these, we are using Geohash as it has simple rules for coding. Although, Geohash has some disadvantages like it does not have a clear rule for encoding multi-dimensional objects like lines or polygons [63] [64]. Moreover, Geohash encoding depends upon the z-order curve, so its spatial locality is not good. Therefore, there may be significant differences between Geohash encoding of nearby space [65]. However, we considered it because of its simple structure. It divides the whole geographical space into a binary grid where alternate bits cross bit by bit and represent longitude and latitude. See figure 3-4.

3.2.1 Geohash

Geohash represents geographical locations using a sequence of letters and digits. Geohash consists of English characters except a, i, l, o and contain digits 0-9 at every level of the representation. The string length corresponds to the size of the geographical area designated by the Geohash, as shown in Table 3.1. It is a hierarchical spatial data structure subdividing the space into smaller subspaces depending on the Geohash length. For example, the first character divides the space into 4 x 8 (four rows and eight columns); after that division of regions alternates between 8 x 4 and 4 x 8. A space-filling curve decides the sequence number of the areas. When alternate characters binary representation are combined in Geohash, two strings for determining row X (latitude bits) and column Y (longitude bits) cross bit by bit.

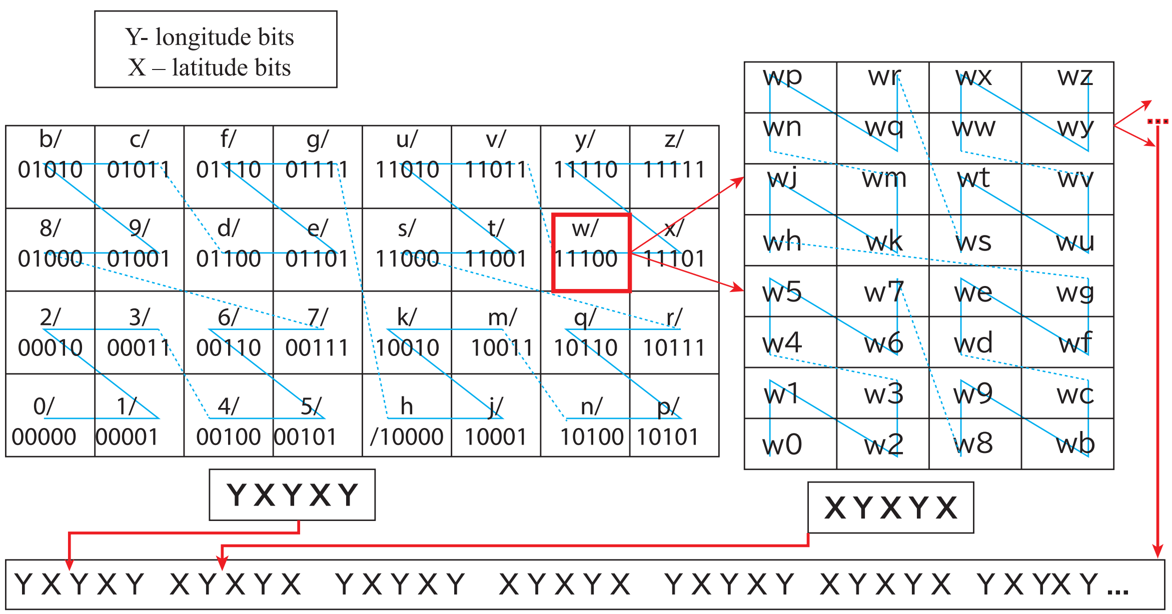


Figure 3-4: Geohash follow an alternate sequence of space filling curves. Alternate characters binary representation determining latitude X bits and longitude Y bits cross bit by bit.

Table 3.1: Geographical size of Geohash encoding

#Label in Geohash	Distance in north and south [m]	Distance in east and west [m]	An Geohash example
1	4989600	4050000	w
2	623700	1012500	wy
3	155925	126562.5	wyh
4	19490.625	31640.625	wyhb
5	4872.65625	3955.07813	wyhby
6	609.082031	988.769531	wyhby3
7	152.270508	123.596191	wyhby3k
8	19.0338135	30.8990479	wyhby3kf
9	4.75845337	3.86238098	wyhby3kf5
10	0.59480667	0.96559525	wyhby3kf5f
11	0.14870167	0.12069941	wyhby3kf5fs
12	0.01858771 ($\approx$ 1.86 [cm])	0.03017485 ($\approx$ 3.02 [cm])	wyhby3kf5fst

3.3 Occupancy Prediction

In Society 5.0, to facilitate safe mobility, it should be accident-free. Current Autonomous Vehicles use sensor information-based algorithms for safe navigation, but sensors have their limitations. In C-ITS, vehicles cooperate between them to facilitate safe navigation. Local Dynamic Map plays a significant role in traffic participants sharing information in the C-ITS setup. One of the main challenges for ensuring safety in Autonomous Driving (AD) is the uncertain behaviour of the traffic participants. Therefore, to ensure safety in the traffic scenario, we need the current and future occupancy of the traffic participants. See figure 3-5. Consequently, we found the participants' future occupancy in the C-ITS procedure. We stored them in the LDM. We predicted the worst-case occupancy using Kamm's circle for a given scenario.

Moreover, If the vehicle's intention is known, then the expected occupancy will be a subset of the worst-case occupancy. So first, we tackled the problem for the worst-case occupancies and later discussed in the Discussion chapter that the occupancies with traffic restrictions would be subsets of worst-case occupancy. In this Thesis, We approached the problem of occupancy prediction from an abstraction of reachability analysis point of view. Using the above approach, we overapproximated the occupancy of the participating vehicles.

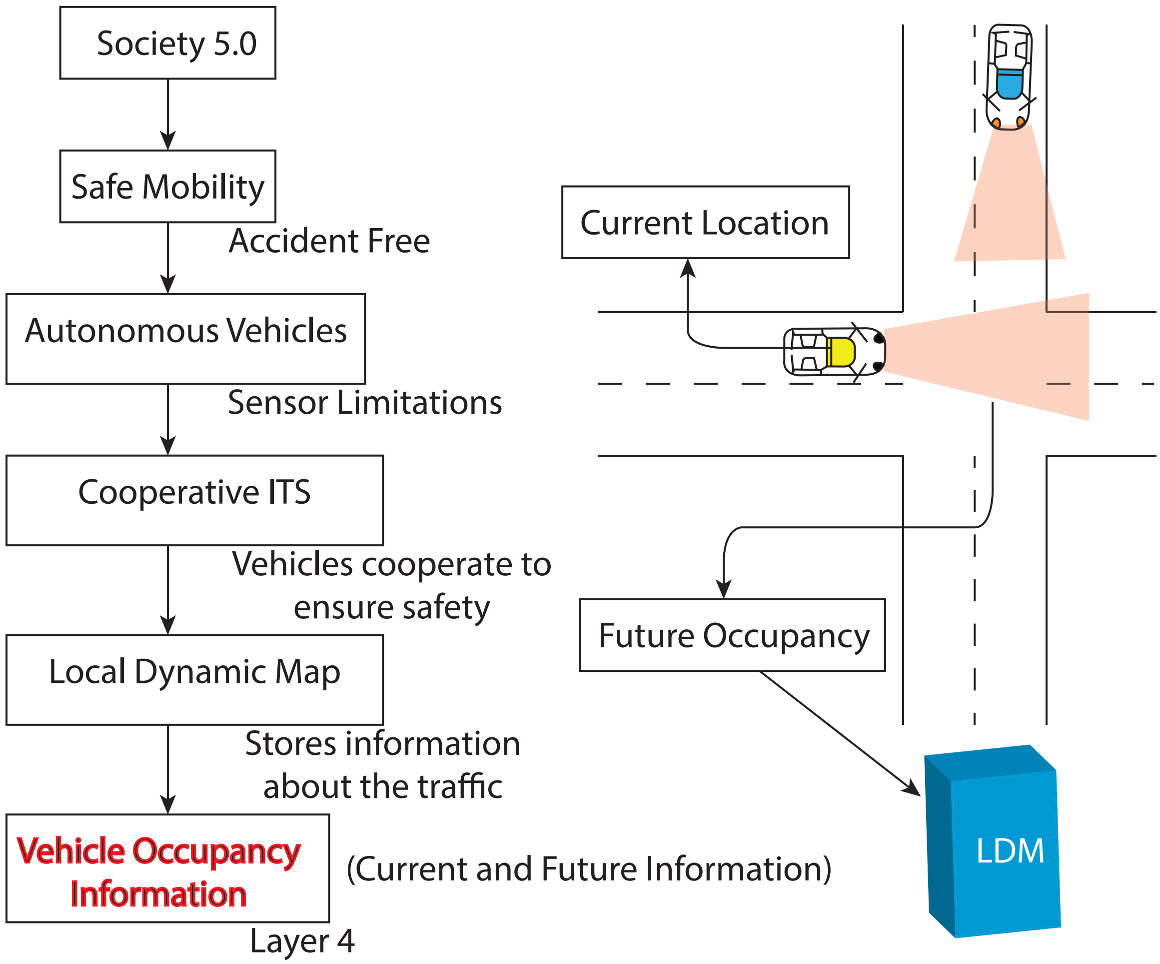


Figure 3-5: Store future occupancy in the LDM.

Below we describe the concept of reachability analysis, Abstraction and the use of Kamm's circle in Abstraction to find occupancy based on [66][46].

The mathematical model of non-ego vehicle considered in [45, 46] are as follows:

1. $C1$: Positive acceleration becomes Nil after vehicle speed reaches the maximum speed (V_{max}).

2. $C2$: To a speed above the parameterized speed, positive longitudinal acceleration is inversely proportional to the speed.

3. $C3$: Driving backwards is not allowed.

4. $C4$: a_{max} is the maximum absolute acceleration.

5. $C5$: Leaving road/lane/crosswalk/sidewalk boundary is not allowed. Crossing a lane is allowed when lane marking or traffic rules do not restrict it.

Out of which we consider $C4$. Nevertheless, decreasing the number of constraints increases uncertainty and leads to higher occupancy regions for a vehicle in the future. So it will not affect our future geographical occupancy representation since more information is available regarding the traffic participants. So we can introduce them. To reduce the uncertainty, we can include more conditions and, hence, future geographical occupancy prediction will be more precise and conservative. Before proceeding further, we will explain the concept of Reach set, Reachable set on finite state machine to understand the idea. Later we define the model for a dynamical system and reachable set for a given model. Thereon we define Abstraction and Occupancy prediction using Kamm's circle. For more details please refer [67][46].

3.3.1 Reach Set and Reachability set for a finite state machine

Let $S = (X, U, T)$ is a finite state machine. Where X is the finite set of states, U is the finite set of control inputs and $T : X \times U \to X$ is the transition function. X_0 is the set of initial states.

Reach Set

The set of states x at time t for which sequence of control inputs $u_0, u_1, ..., u_{t-1}$ exists from the initial states $x_0 \in X_0$ are known as Reach Set $R(X_0, t)$. Reach sets for discrete case are shown below, Refer figure: 3-6, 3-7 and 3-8.
In the following figures, control input set U has the values to go East (E), West (W),

North (N) and South (S). The red and blue bins represent the initial and reach states at time t, respectively.

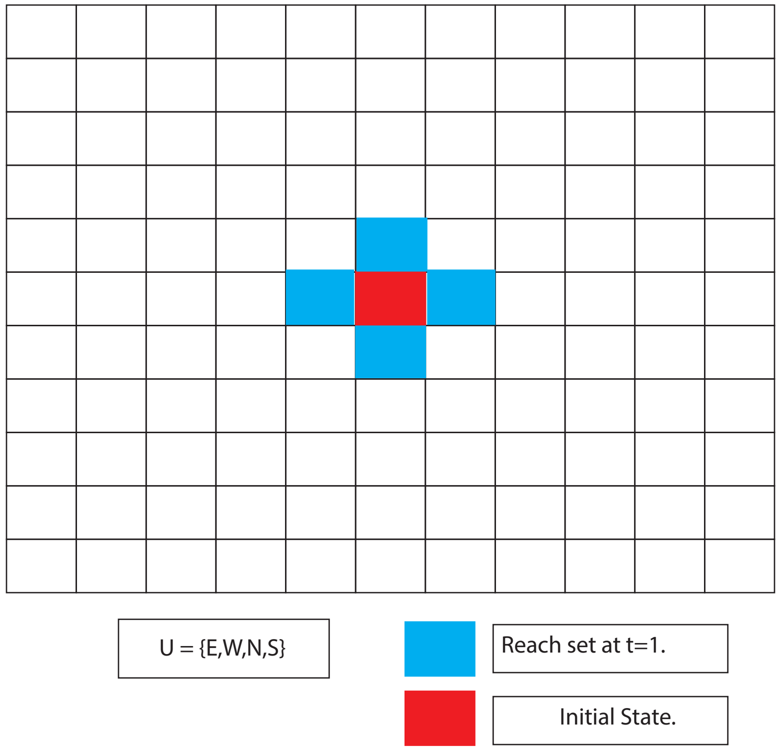


Figure 3-6: Reach set in a finite grid at time t=1.

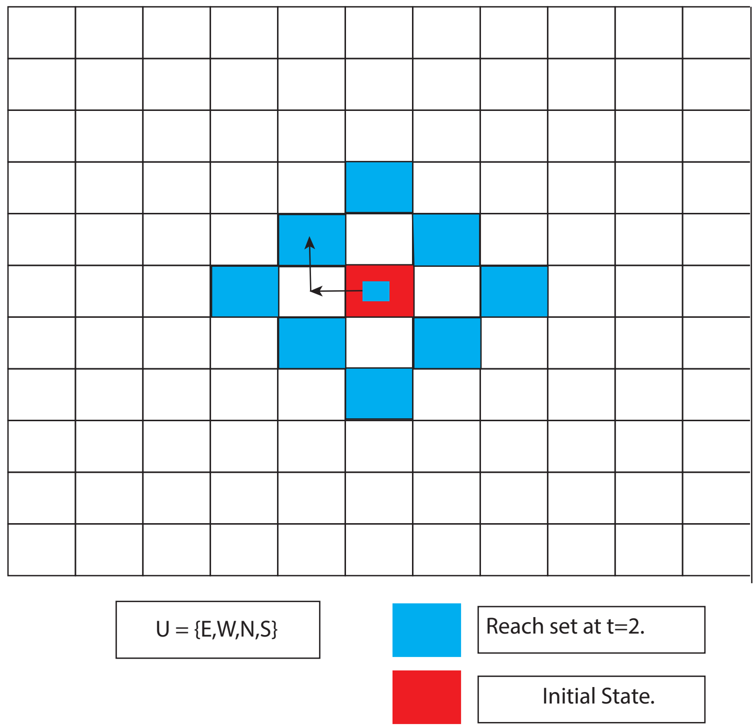


Figure 3-7: Reach set in a finite grid at time t=2.

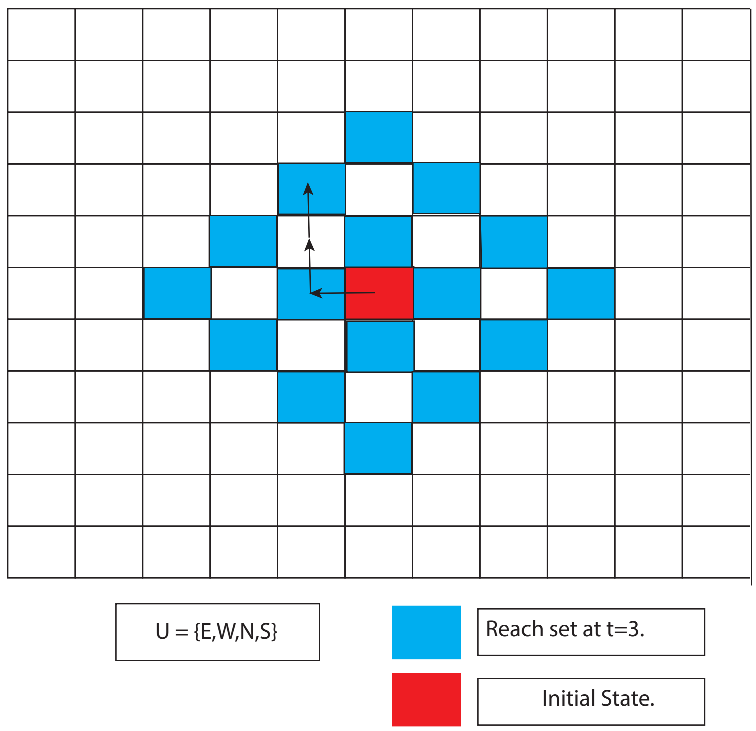


Figure 3-8: Reach set in a finite grid at time t=3.

Reachable Set

Reachable set at time t is the union of all the Reach set $\leq t$ i.e (Refer equation 3.1 and figure: 3-9).

$$\overline{\mathrm{R}}(X_0, t) = \cup_{s \leq t} R(X_0, t) \tag{3.1}$$

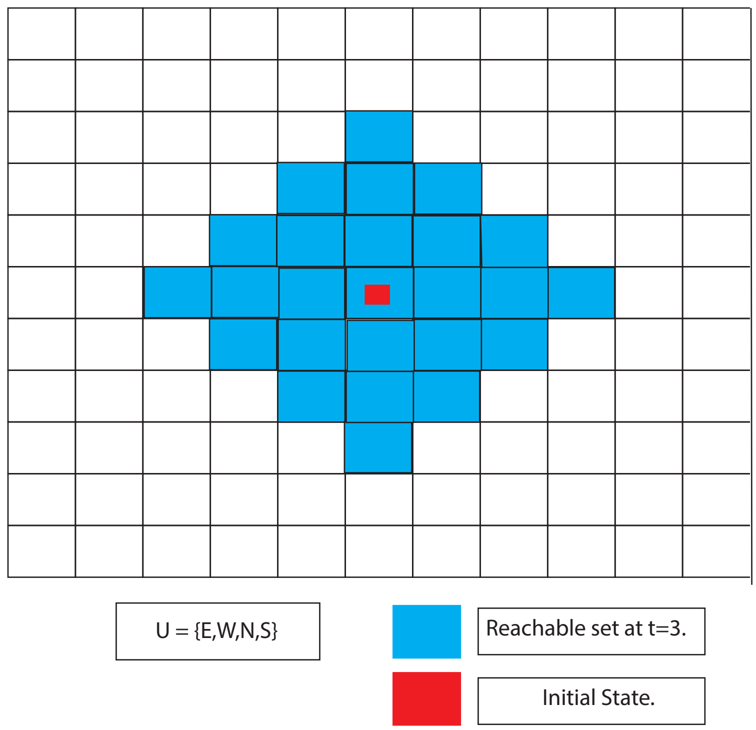


Figure 3-9: Reachable set at time for finite grid t=3.

3.3.2 Reachable set for a dynamical system

First, we define a dynamical system model used and then we define the reachable set for a given model.

Model

In this thesis, we considered a vehicle as a point mass model [68]. Represented as:

$$\ddot{S}_x(t) = a_x(t), \;\; \ddot{S}_y = a_y(t), \;\; \sqrt{a_x^2 + a_y^2} \leq a_{max}. \tag{3.2}$$

The point mass model abstracts the vehicle as a point, and it can only accelerate within the bounds of Kamm's circle. In addition, the point mass model ignores the minimum turning radius of the vehicle. We can use the Kinematic single-track model to include a minimum turning radius. Higher order differential equation in 3.2 can be written into the set of linear differential equations using state space model. State space model of a car is given as:

$$\dot{x} = Ax + Bu \tag{3.3}$$

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{3.4}$$

Where in 3.4 has state variables as:

$$x_1 = s_x,\ x_2 = s_y,\ x_3 = \dot{s_x} = v_x,\ x_4 = \dot{s_y} = v_y \tag{3.5}$$

$$u_1 = a_x,\ u_2 = a_y \tag{3.6}$$

The model is defined as:

$$M = (f, \mathcal{X}_0, U) \tag{3.7}$$

Where f belongs to the dynamical system:

$$\dot{x} = f(x(t), u(t)) \tag{3.8}$$

Using equations 3.4 and 3.8 f represents the set of linear differential equations representing the dynamics of the model. Where x and u represents the state of the system and input respectively at any time t. Possible initial states and inputs are bounded sets belonging in $x(0) \in \mathcal{X}_0, \forall t : u(t) \in U$.

Reachability

For a given model M reachability for a time interval $t \in [0, r]$ is defined as:

$$\overline{\mathrm{R}}(M, [0, r]) = \bigcup_{t \in [0,r]} R(M, t) \tag{3.9}$$

Where $R(M, t)$ is reach set at time t.

Hence, for a discrete case $t \in \{t_1, ..., t_n\}$. Reachable set at time interval $t \in \{t_1, ..., t_n\}$ is as follows:

$$\overline{\mathrm{R}}(M, [t_1, ..., t_n]) = \bigcup_{t \in \{t_1, t_2, ..., t_n\}} R(M, t) \tag{3.10}$$

The above equations with conditions $C1$ - $C5$ form a hybrid automaton; however, the reachability analysis of a hybrid automaton is time-consuming. Hence observing [46], we pursued Abstraction.

Abstraction

For a model M of a given dynamical system, Abstraction is the model M_i if the reachable set of the M_i includes all the reachable sets of the M.i.e (See figure:3-10)

$$\forall t > 0 : R(M, t) \subseteq R(M_i, t) \tag{3.11}$$

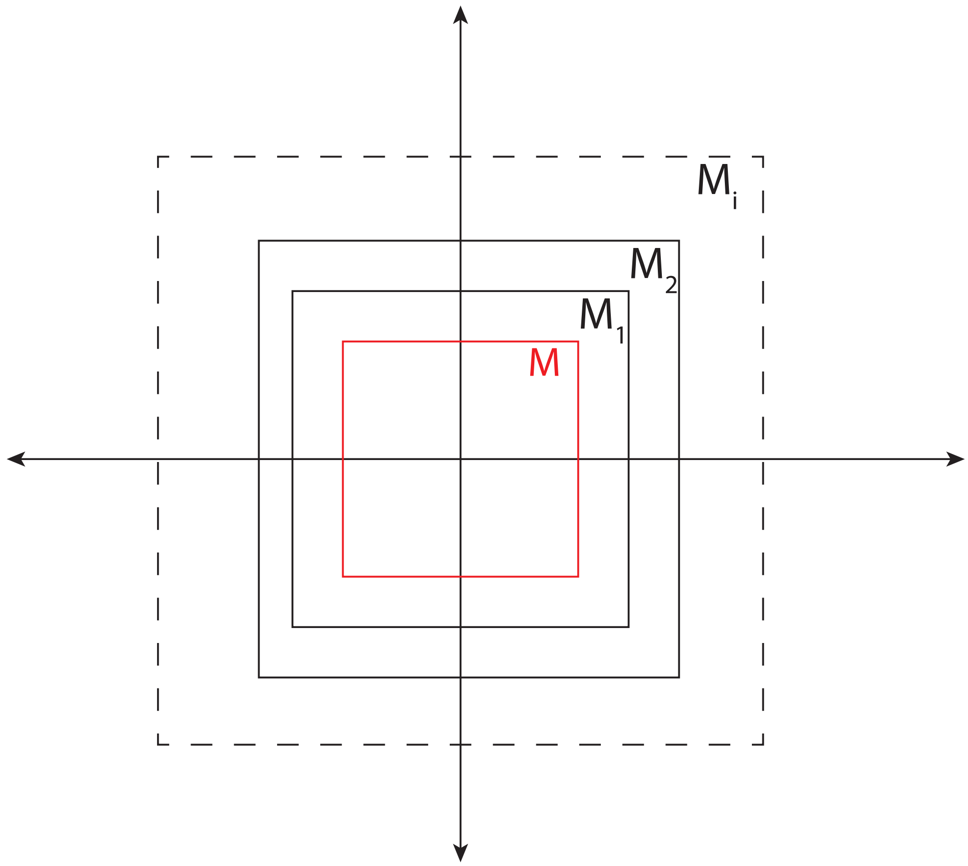


Figure 3-10: Abstraction of a model contains all reachable states which are reachable by the original model.

For considering occupancy [46],:

$$proj(x) = [x_1, x_2, x_3]^T \tag{3.12}$$

Where; state vector $x \in \mathbb{R}$ for a given model and x_1, x_2, x_3 are x-position, y-position and orientation in the 2-D space. i.e for a given state variables of the model $proj(x)$ returns the corresponding location (x,y positions) and orientation of the vehicle.

Hence equation,

$$proj(R(M,t)) = \{proj(x) \mid x \in R(M,t)\} \tag{3.13}$$

gives all positions and orientation. For the given abstractions of model M_0 as models $M_i, i = 1, ..., m$.

Therefore for the Abstraction of the model, M_0 as models $M_i, i = 1, ..., m$ follows the following property:

$$\forall t > 0 : R(M_0, t) \subseteq R(M_i, t) \tag{3.14}$$

and over-approximated occupancy is given by equation:

$$\forall t > 0 : proj(R(M_0, t)) \subseteq \bigcap_{i=1}^{m} proj(R(M_i, t)) \tag{3.15}$$

As per the above approach, the accuracy and computation time will increase as the number of abstract models increases. In our case, we used an Abstraction model considering condition $C4$. Therefore, occupancy only based on $C4$ will overapproximate the occupancy based on $C1$ - $C5$. Furthermore, Kamm's circle was used as an abstraction model to define occupancy based on $C4$.

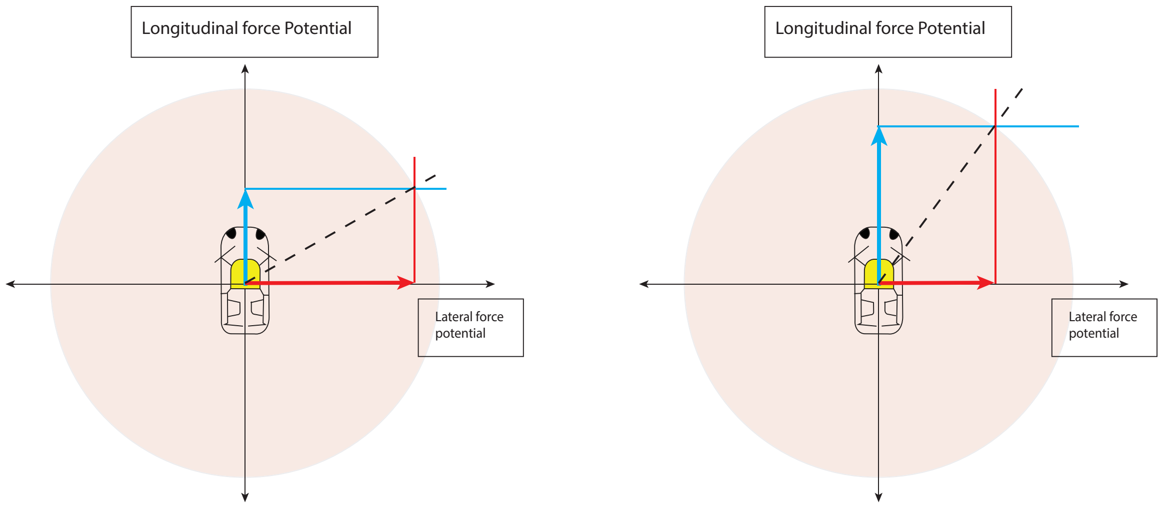
Kamm's Circle

It is challenging to consider the trajectory that is possible by vehicle over time. In [46][69] described the overapproximated occupancy (Reach Set) at time t with centre c(t) and radius r(t) as (See figure 3-14):

$$c(t) = \begin{bmatrix} s_x(0) \\ s_x(0) \end{bmatrix} + \begin{bmatrix} v_x(0) \\ v_y(0) \end{bmatrix} t \; ; \; r(t) = \frac{1}{2} a_{max} t^2 \tag{3.16}$$

Where;

- $c(t)$ is a position of a vehicle at time t.
- $s_x(0) \, and \, s_y(0)$ is the position of the vehicle at time t=0.
- $v_x(0) \, and \, v_y(0)$ is the velocity in x and y directions of vehicle at time t=0.
- $r(t)$ is radius of a Kamm's/ Traction circle at time t.
- a_{max} is the maximum acceleration possible of a given vehicle.



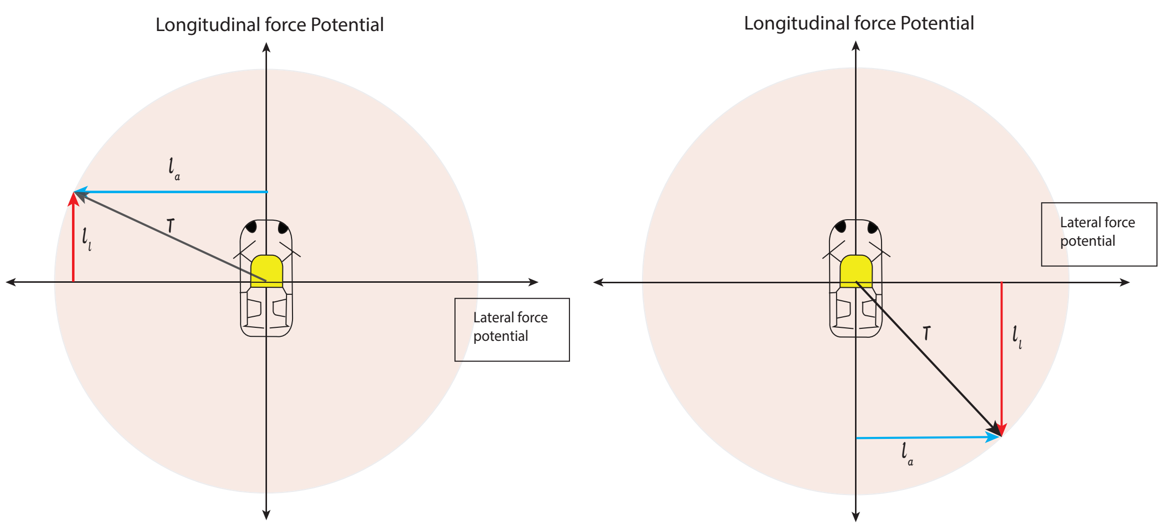
(a) High lateral force limit the longitudenal force.

(b) High longitudenal force limit the lateral force.

Figure 3-11: Longitudenal and Lateral forces limit inside a Kamm's circle .

Kamm's/ Traction circle limits the maximum forces applicable between tires and the road. See figure: 3-11.

So, a_{lo} longitudenal acceleration and a_{la} lateral acceleration satisfies eq.3.17 without losing the grip. See figure 3-12.



(a) Acceleration with Left lateral force.

(b) Deacceleration with Right lateral force.

Figure 3-12: Longitudenal force (Acceleration/Deacceleration) and Lateral force (Left/ Right).

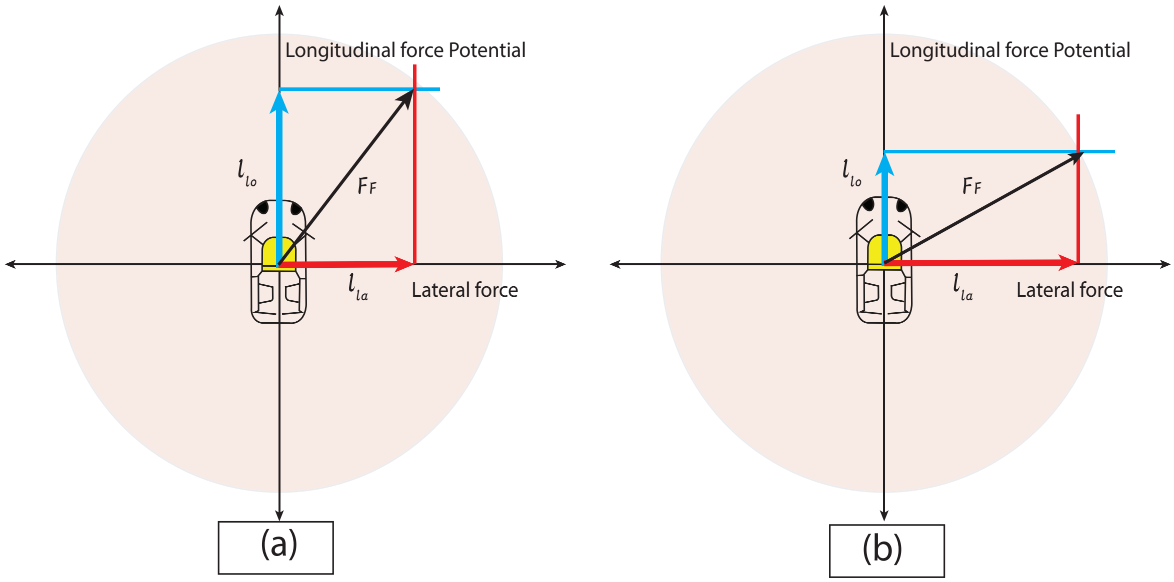


Figure 3-13: Reach sets/ Reachable set for the center of gravity of a moving car.

$$
\begin{aligned}
a_{lo}^2 + a_{la}^2 &\leqslant F_F \\
a_{lo}^2 + a_{la}^2 &\leqslant \mu_r^2 g^2
\end{aligned}
\tag{3.17}
$$

Where μ_r and g represents the friction coefficient and gravitational acceleration respectively. eq. 3.17 forms and circle of radius $\frac{1}{2}\mu_r g t^2$ and using eq. 3.16, we get; $a_{max} = \mu_r g$.

Hence, the Reach set for a vehicle at time t is given by Kamm's circle. Furthermore, the Reachable set will be a collection of all such Reach sets in the future. So, we can use the Reachable set as occupancy for a given vehicle in the future.

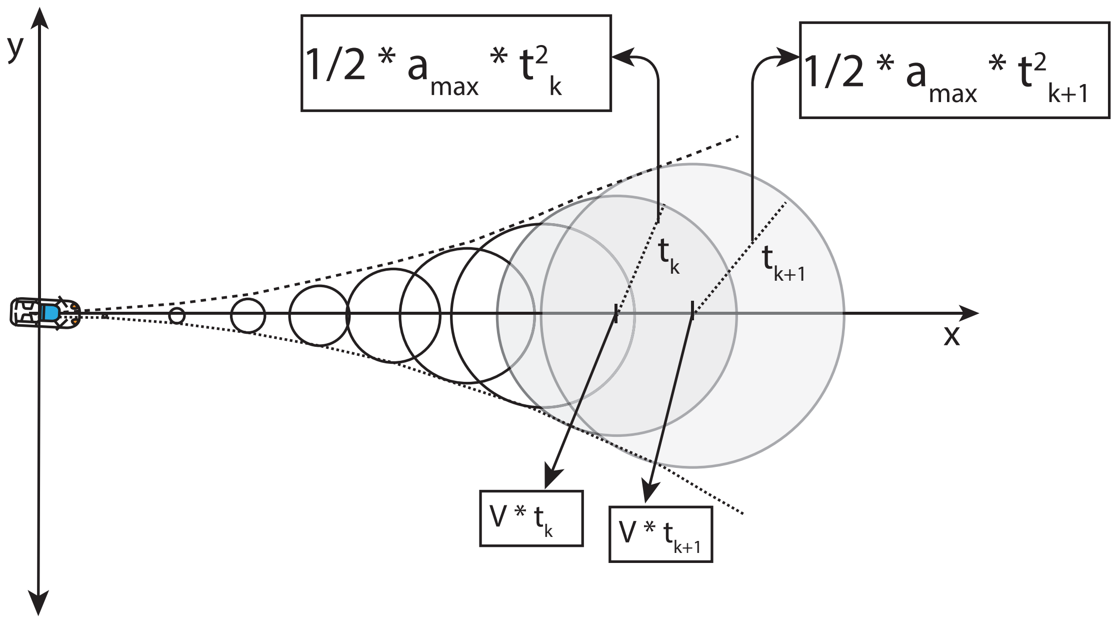


Figure 3-14: Reach sets/ Reachable set for the center of gravity of a moving car.

3.4 Decision Diagrams

3.4.1 Boolean Function

A Boolean function is of the form $f : \{0,1\}^k \rightarrow \{0,1\}$, where k-tuples of Boolean variables takes values to Boolean values 0 (false) or 1 (True). Suppose valuation V means the total combination of values that k-tuple boolean variables can take, then each k-tuple assignment in V can be written as $\Gamma : v \rightarrow [0,1]$ from value in fixed set V to a boolean value. Where $v \epsilon V$. The Boolean function can also be represented using Boolean variables and Boolean operations (and, or, not), also known as literals. e.g. $x_1 x_2 \overline{x_3} + x_4$. Where concatnation, + and $\overline{x}$ represent and, or, not operations over variables.

3.4.2 Reduced Ordered Binary Decision Diagrams

The BDD is a graph representation of the boolean functions. The Basic idea behind the BDD is Divide and Conquer. More specifically, BDD is a rooted directed acyclic graph (DAG), where non-leaf nodes have labels with Boolean variables and leaf nodes have labels 0 (zero) or 1 (True), which correspond to boolean function output. BDD can represent most of the boolean functions in feasible size compared to the truth table or binary tree for boolean functions that always takes 2^n space. Decision Diagram in which the relative ordering of variables on each path from the root to the leaf is fixed (also known as Ordered Binary Decision Diagram (OBDD)), and it combines the isomorphic subgraphs present in the graph to create Reduced Ordered Binary Decision Diagram (ROBDD).

Each OBDD has the following components [70]-

$G = ((Q, v_0, E), V \cup \{0, 1\}, <, L)$

- (Q, v_0, E) is a rooted directed acyclic graph. Q is a finite set of nodes. v_0 is the root node and $E \subset Q \times Q$. Each non-leaf node has its successors, namely low and high.
- V is a finite set of Boolean variables.
- $<$ is a total order on $V \cup \{0, 1\}$
- L is a mapping satisfying the following conditions:-
  - Leaf are mapped to 0 and 1 and non-leaf nodes are mapped to V.
  - If (v,v') $\in E$ then $L(v) < L(v')$.

Thus a Graph G over boolean variables V represents a boolean function. The interpretation of BDD is based on the Shannon Expansion.

$$f = \overline{x}f[x] + xf[\overline{x}] \tag{3.18}$$

Thus, according to the Shannon expansion, each node of the graph has low and high and ROBDD can be obtained from OBDD by minimizing the redundancy in the representation using the following rules:

- Merge all zero and one nodes to a single unit of zero and one node.

- Merge any isomorphic nodes. i.e. if $l(x) = l(y)$ and $h(x) = h(y)$ then merge these nodes into one and point all incoming node to any one of them. Here l and h represents low and high child of any given node of a graph.

- Eliminate any node that have two children nodes as isomorphic.

The size of the ROBDD depends upon the represented function and the variable order we choose. For a given variable order, ROBDD representation for the Boolean function is the canonical representation, i.e. function has a unique representation.

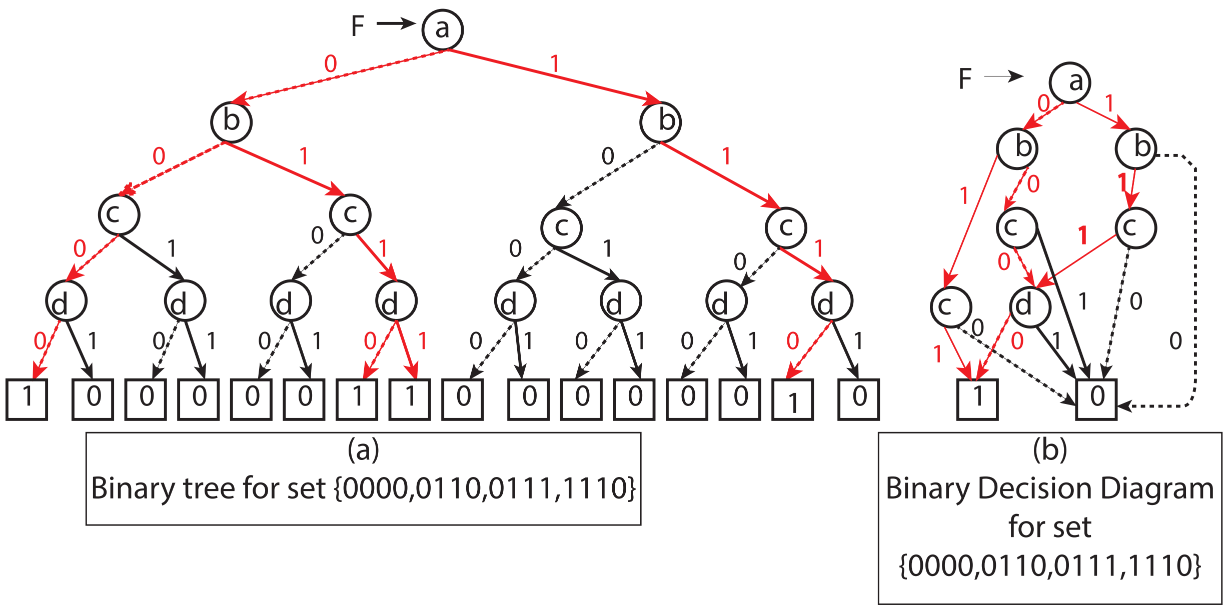


Figure 3-15: (a) Binary Decision Tree representation for a given set has fixed size and large as compare to BDD representation. (b) binary decision diagram representation for a given function has compact representation.

3.5 Database

Since the introduction of the LDM in the SAFESPOT, using a database to implement has become a common standard. Although the non-database implementation of the LDM also exists [27]. The LDM community, to a great extent, explored the type of database performance in the implementation of the LDM [13] [15] [22] [26] . Also, supporting query languages for the concerned databases were utilized to execute the query in the database to insert or extract the data. In general, the following databases are available in the literature in the majority.

1. Relational database
2. Graph database

In the current Thesis, we used both non-database/ databases implementation without emphasizing the particular kind of database for any specific task since our primary goal was to facilitate LDM for the safety enhancement of the participating vehicles.

In particular, in this Thesis, we utilized memory-based LDM and conducted a relational database (PostgreSQL database + SQL query language) based implementation for storing future occupancy information in the LDM[22]. Although the type of database used significantly affects the performance of the LDM, we used it independently since our main focus will be on the information processing approach of the data before storing it in the LDM to facilitate the safety verification task in the C-ITS setup.

Chapter 4

Results

4.1 Potential Risk Region Representation and Near Miss Detection

Autonomous Driving use Lanelets for physical layer representation of roads. It is a polyline representation of road boundaries, an extension of the OSM data model. An atomic Lanelet consists of left and right polyline boundaries. A group of several Lanelets forms a road network (as shown in figure 3-1). Vehicle localizes by the Lanelet in which it is present, but they often can have extended length and wide breadth; therefore, controlling and ensuring the safe vehicle movement within Lanelet is a critical issue we address here in this subsection. We propose to divide the Lanelet into several regions and consider occupancy within areas/regions in a Lanelet.

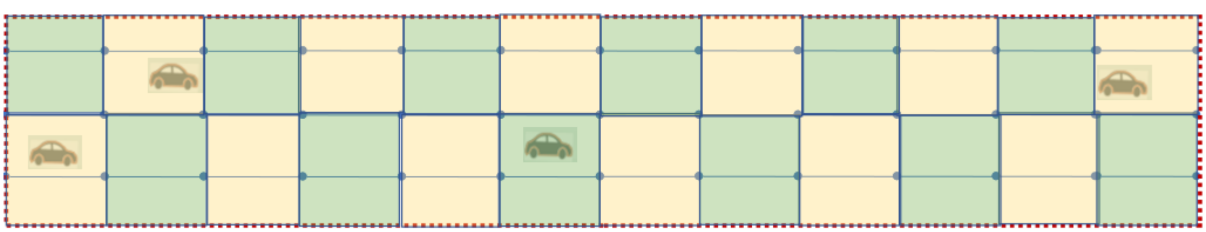


Figure 4-1: Division of Lanelets into several regions.

Also, graph-based LDM (Figure: 4-2, 4-3) represent the areas within the road/ Lanelets in this subsection.

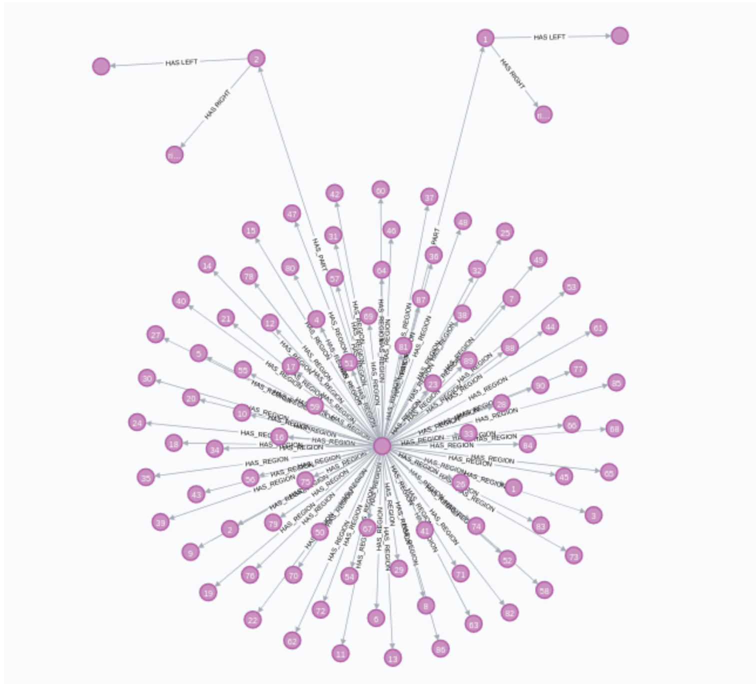


Figure 4-2: Graph based representation of the LDM. Neo4j Command:- Match(n) Return n;.

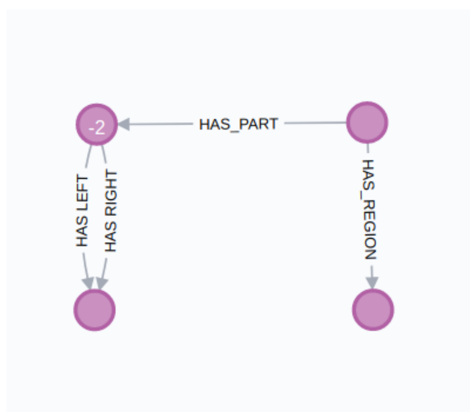


Figure 4-3: LDM Schema. Neo4j Command:- CALL db.schema.visualization().

4.1.1 Approach and Outcome

CommonRoad [71] was used to evaluate our method. CommonRoad scenarios contain the road network using Lanelets. The first two lanelets were utilized, shown in Fig. 4-4. (a), represented by yellow filling and dotted red in the figure. Next, we interpolated in-between values of the polylines and found Voronoi regions using them. Voronoi regions divide the lane into small sub-divisions. These divisions and regions in road networks were further represented using the graph. Then depending upon the area in which the vehicle belongs, we can predict that its forward region and the neighbouring region to which the vehicle may go shortly are the risky regions for other vehicles. Thus, using the above analogy, predictions of dangerous areas within the lane were introduced and depending upon the accuracy we need, the size of region grids can be changed before the experiment.

In Fig. 4-4 a) red rectangle contains the vehicle under analysis in CommonRoad Scenario: ZAM_ZIP_1_2_T-1. Fig. 4-4. b). red filling denotes risky regions just before the turn of the vehicle under analysis. Fig. 4-4. c). red filling denotes the risky regions after the turn of the vehicle under analysis. Blue points in Fig. 4-4. b) and c) denotes lanelet boundary points and orange points denotes the Voronoi points computed by the algorithm created Voronoi region.

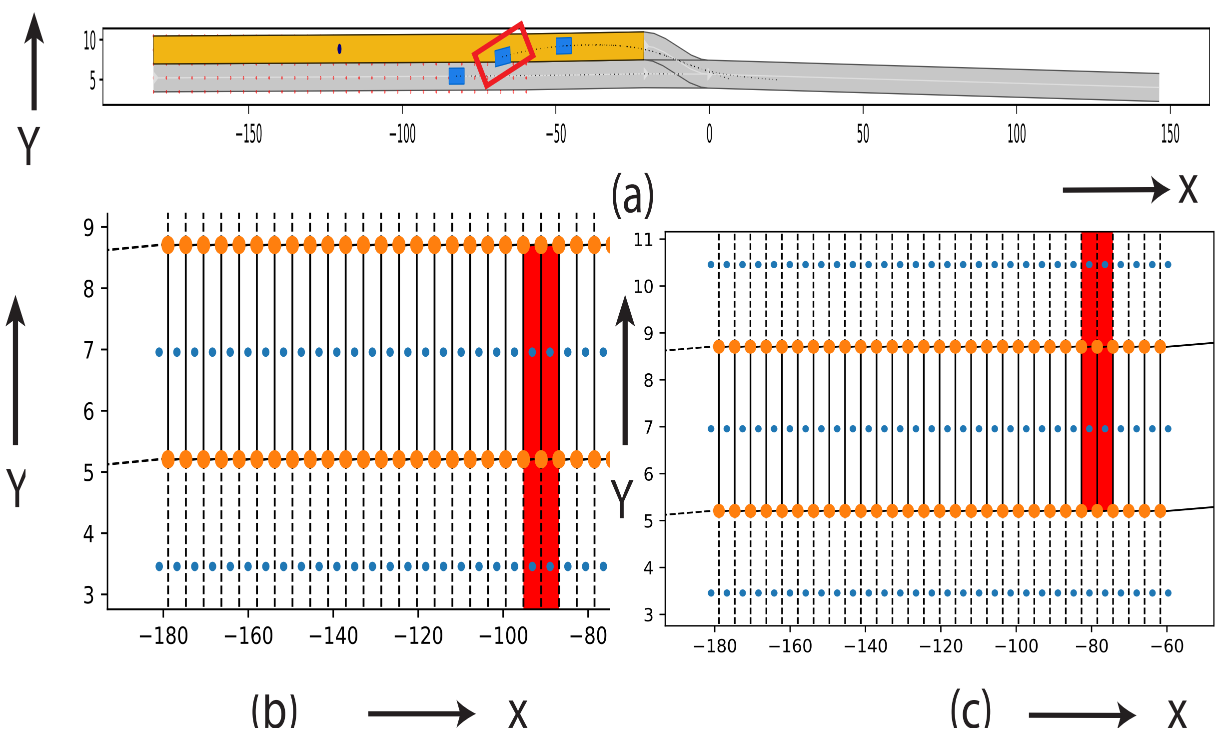


Figure 4-4: (a). Scenario: ZAM_ZIP_1_2_T-1. b) and c): Voronoi regions and risky regions

Dividing Lanelets into regions can help for detecting more accurate localization and risky areas prediction within Lanelets. But, the above method suffers from serious drawbacks such as:-

- Voronoi based region's information is not based on an established standard that all the participants can understand and share.
- When we store vehicle localized regions in the LDM, it should support easy checking to avoid risky situations.

Therefore to tackle the above drawbacks, we used the following approaches:

- We used Geohash to divide the Lanelet into regions. Geohash is the well-established standard that all the traffic participants can understand. (See figure 4-5)
- We encoded the Geohash into Shared Binary Decision Diagram to check if traffic participants share the exact Geohash location. Sharing the Geohash locations signifies vehicles are very close, furthur which can lead to a risky situation.

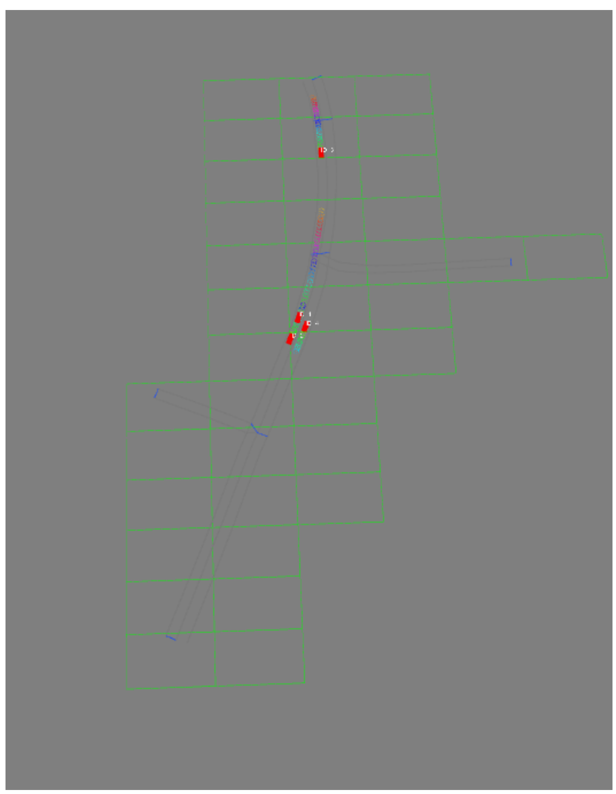


Figure 4-5: Geohash regions in a map.

LDM stores the information about the traffic participants. Hence, we need to avoid near accident cases using the information in the LDM to ensure safe mobility. LDM stores information about all the participating vehicles. Accordingly, to confirm safe mobility, we need to check or compare the localization information of all the vehicles. Since vehicles are moving and future occupancy information of the vehicles are not available in this setting, thus depending upon the current localization information of all the traffic participants, we need a method to check nearby vehicles as soon as possible to avoid a potential collision.

Therefore, we proposed storing traffic participants' localization information as a shared binary decision diagram encoding its geohash Boolean representation to detect the near miss situation. In SBDD, Decision diagram representation of equivalent Boolean function share node. Therefore, we can check quick equivalence between two Boolean functions since equivalent Boolean functions share the same node means the same location in the memory (See figure 4-6). Hence checking for equivalent Boolean function has been varied to conform to the identical memory location, and this operation is quick. Therefore SBDD can perform a quick equivalence check for the Boolean functions [1].

We used the above method for a quick equivalence check between the SBDDs of the vehicle's geohash. In the consideration of minimization of computation time, a SBDD can be used. It is a representation of the target multiple-output function, the nodes are shared among BDDs representing the various outputs and a partitioned SBDD consists of two or more SBDDs that share nodes, which can be obtained systematically in an optimized way [72].

In consideration to reduce computational cost in the risk detection, we focused on two points as:

1. Area-based detection of nearby objects, instead of point-based positions.

2. Equivalence of Boolean functions, instead of operations with the floating-point arithmetic processing.

This section focused on its realization and the comparative analysis of computa-

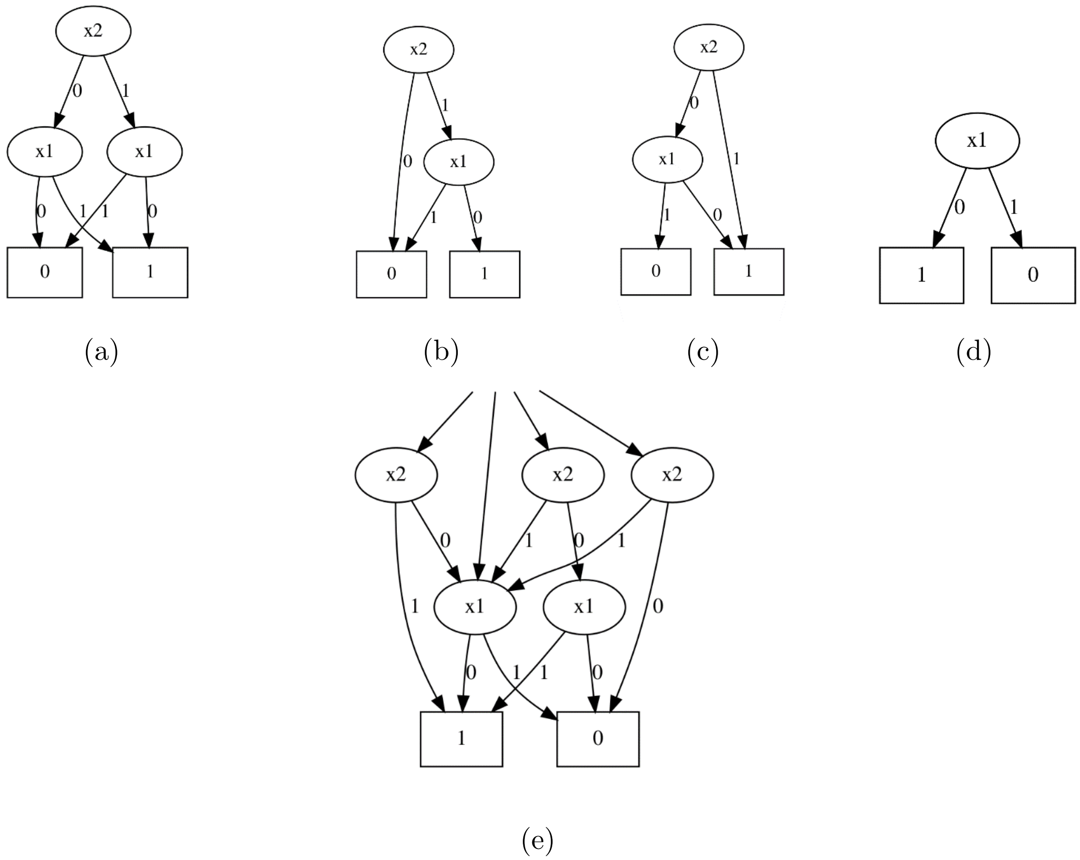


Figure 4-6: Equivalent Boolean function share the node [1].

tion time between cases of the traditional way for representing geographical vehicle positions for the nearby vehicle detection and the proposed way.

There is a possibility to reduce a computational cost if the shared binary decision diagram can be introduced for the detection of nearby objects. In this sense, it can be hypothesized that the encoding of geographical positions into Boolean values and a calculation in the form of Boolean functions are beneficial to reduce the computational time [73][70].

The warning algorithm calculates the stopping distance derived from vehicle speeds, acceleration/deceleration and vehicle-to-vehicle local distance measured by on-vehicle sensors, which presumably require a floating-point arithmetic processing of sensor real values if it is implemented in a V2V system. Thus, the computational cost needs to be evaluated in comparison with cases of the floating-point real value representation of geographical positions in the world model.

4.1.2 Integration of Geohash and BDD

In the proposed method, a SBDD was used for the integration of the Geohash-based geographical representation.

For the integration in the proposed method, a Geohash code was converted from characters (a-z) and digits (0-9) to the binary representation by using a conversion table as 0:(00 000), . . ., 9:(01 001), b:(01 010), . . ., z:(11 111), which eliminates specific characters as a, i, l, o according to the Geohash definition. Thus, five Boolean variables were applied to representing each character of the target Geohash code, and then the converted Boolean representation represents a geographical position in the world model consistently. For the utilization to the LDM vehicle management in the proposed method, procedures were designed as follows.

1. Geographical position conversion: (latitude, longitude) to Geohash code, e.g., (33.88919521551, 130.71065559849) is represented as Geohash of wyhby3kdbeyd.

2. Boolean expression conversion of Geohash code for the BDD procedure as hashing, e.g., Boolean expression for Geohash wy as a part of the given code is $(x1 \wedge x2 \wedge x3 \wedge \neg x4 \wedge \neg x5) \vee (x1 \wedge x2 \wedge x3 \wedge x4 \wedge \neg x5)$.

3. Construction of the shared BDD for a given Boolean expression.

4. The (latitude, longitude), Geohash and SBDD encoding for a given point are stored and used for the comparison.

4.1.3 Computer Experiment and Results

ROS based implementation of the LDM framework

Lanelet road network framework [33][74] was modified for the current purpose to integrate the CoInCar-Sim simulator [75] to manage road scenarios, and a geographical map for vehicles was derived from the open-source map project known as OSM. The OSM data format was transformed to our customized Lanelet system working on the Robot Operating System (ROS) as shown in Figure 4-7.

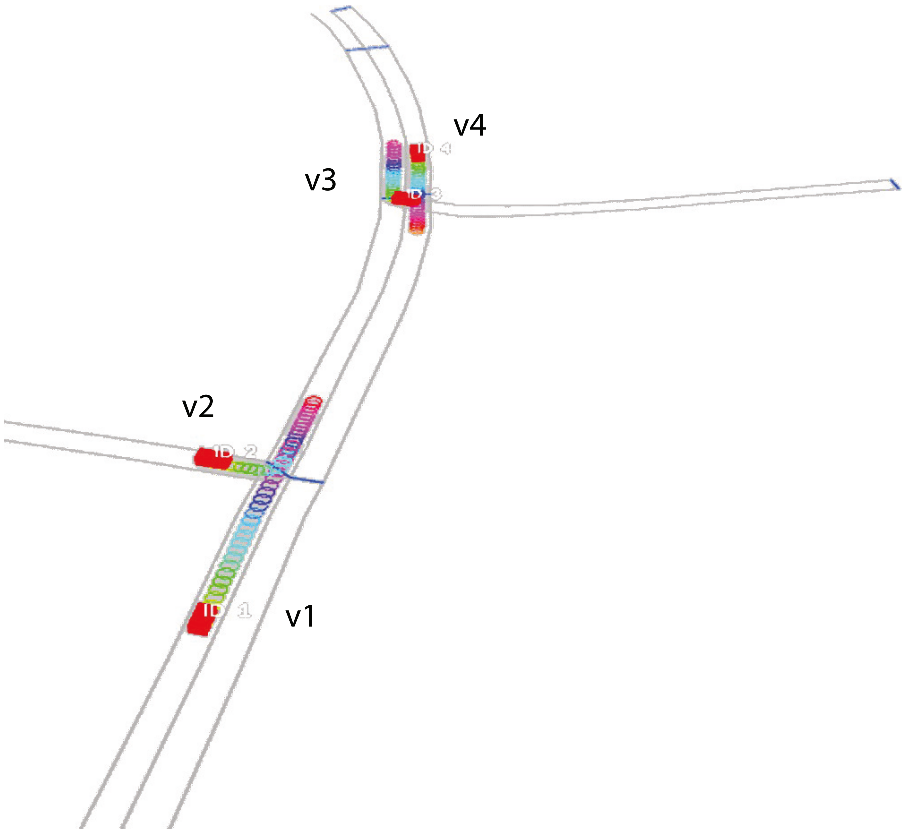
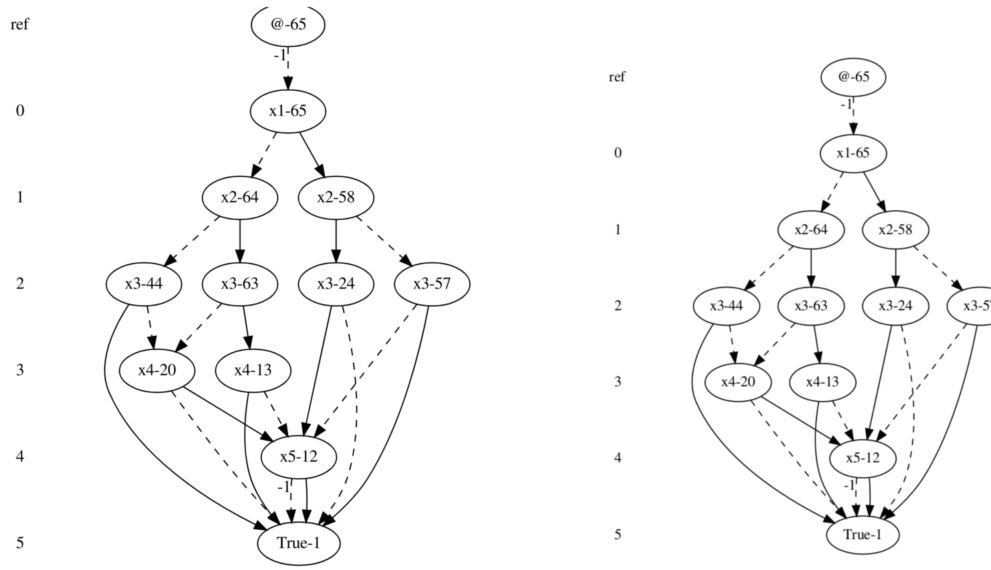


Figure 4-7: An ROS-based demonstration of the LDM system with four vehicles moving on the road, which was implemented in the modified Lanelet associated with the CoincarSIM. It was used for the validation framework of the proposed method. v1, v2, v3 and v4 represent respectively vehicle1, vehicle2, vehicle3 and vehicle4.

Numerical comparisons.

For the validation, we compared computational costs in multiple conditions. The experiments were done in the computer with Intel(R) Core(TM) i9-9900K CPU (3.60GHz) having 64 GB RAM. Different calculation methods were applied to the validation of computational costs.

For the validated comparison, at least, three conditions are necessary, such as a condition equivalent to the traditional implementation as demonstrated by Shimada et al. [13] (floating-point real value representation: F), discrete spatial representation as Geohash (Matching Geohash only: G) and the proposed method as the integration of BDD and Geohash (proposed method: P) in which set equivalent SBDD are checked for equivalence (See figure: 5-9) on success equivalence we perform Geohash matching 'G'. The comparison between F and G indicates the effectiveness of discrete representations of spatial locations with respect to the continuous representations. The comparison between G and P indicates the effectiveness of the SBDD implementation with respect to the calculation without SBDD.



(a) V1: SBDD for the wyhby3kdb.

(b) V2: SBDD for the wyhby3kbd.

Figure 4-8: SBDD for the set {w,y,h,b,3,k,d}. Set Equivalent Boolean function share the node [1].

Figure 4-9, 4-10 showed the average time to compute from 100 trials in each condition. In the condition of the floating-point number comparison (F in the figure panel), geographical positions of vehicles were represented by the floating-point number, and the ordinary arithmetic calculation was used for the detection of nearby vehicles. In the Geohash condition without any BDD scheme (G in the figure panel), geographical positions of vehicles were represented by the Geohash encoding as a string and the string operation was used for the detection of nearby vehicles. The proposed method (P in the figure panel) was implemented as described above. Interestingly, computation time in conditions of F and G was almost three times larger than the proposed method in the case of two-vehicle interactions (Figure 4-9, 4-10). The tendency was consistent in any combinations of vehicles (Figure 4-9). In the comparison with various numbers of vehicles, the proposed method (P) took the computation time as 12.1

[s] (two vehicles), 15.23 [s] (three vehicles), and 21.43 [s] (four vehicles). This increase rate was significantly low in comparison with other methods as shown in Figure 4-10. This result clearly proved that our hypothesis was valid and the proposed method effectively reduces the necessary computation time even with an increase of the number of vehicles.

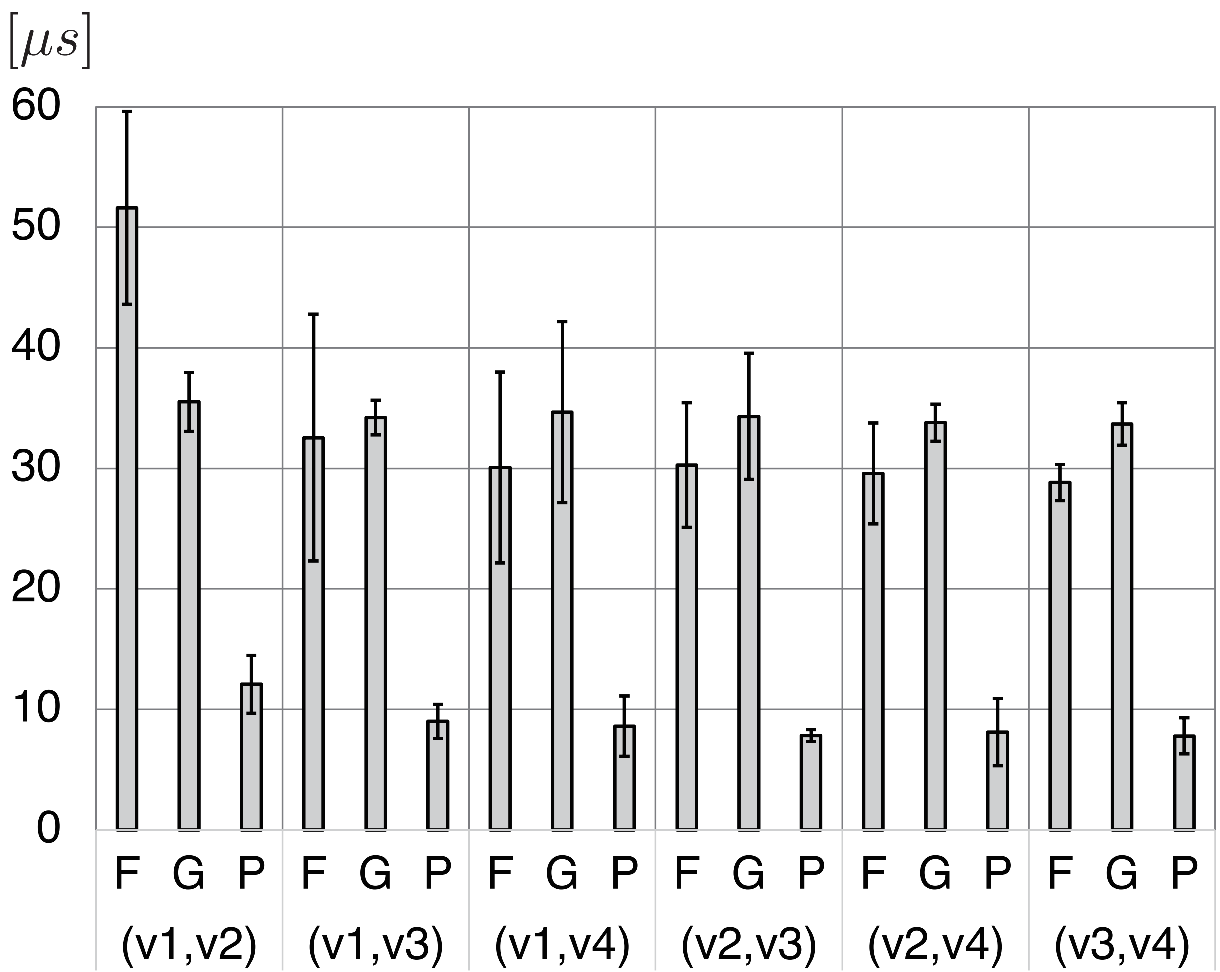


Figure 4-9: Two vehicle Interaction comparison among different calculation methods in computational costs. Position verifications of multiple vehicles by using the floatingpoint number (F), Geohash code string without BDDs (G) and the proposed method (P) were shown in each panel. Each average elapsed time was obtained from 100 trials in each condition and the error denotes the standard deviation.

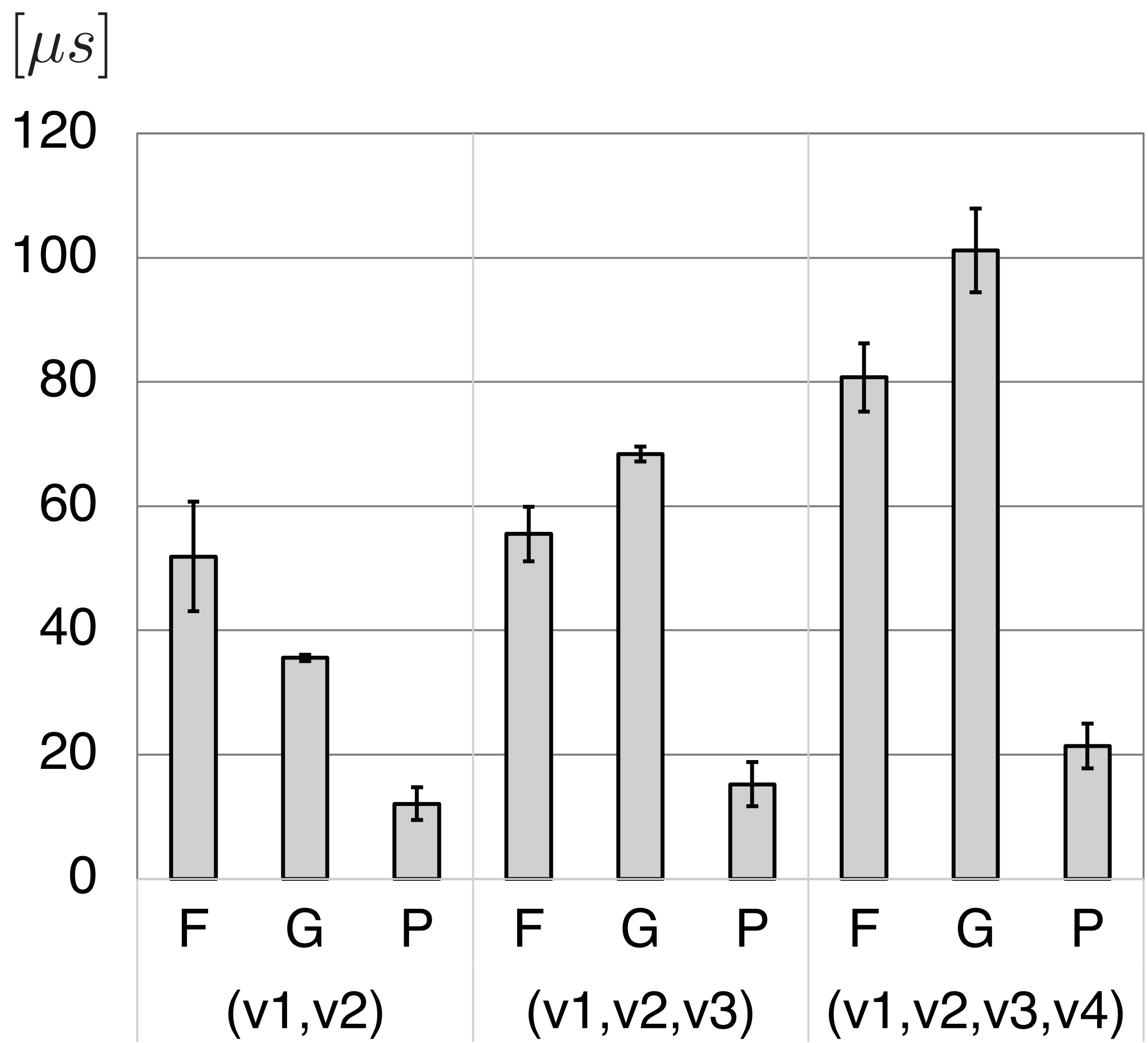


Figure 4-10: N vehicle interactions (N = 2, 3, 4) comparison among different calculation methods in computational costs. Position verifications of multiple vehicles by using the floatingpoint number (F), Geohash code string without BDDs (G) and the proposed method (P) were shown in each panel. Each average elapsed time was obtained from 100 trials in each condition and the error denotes the standard deviation.

Analysis

We hypothesized that the integration of Geohash encoding of geographical positions of vehicles and the shared BDD minimizes the computation time in the verification of multiple vehicle positions, and successfully established the testable framework based on ROS with modified Lanelet system and CoInCar-SIM-based scenario manager . Results of computer experiments clearly demonstrated the effectiveness to reduce the necessary computational time in comparison with other conventional conditions.

Equivalence check of Geohash SBDD representations of the vehicle localization significantly reduces the computing time, but the above methods have the following limitations:

1. We need a central computing setup to perform an SBDD-based Boolean function equivalence check.

2. We have only considered the current geohash location of a vehicle, but in practical vehicles that are moving, we need to consider near future occupancy.

Hence, we approached the challenges with the following setup to improve the above limitations:

1. We used the BDD (ROBDD) in the distributed setup instead of SBDD.

2. We included the neighbouring geohash locations, which a vehicle could occupy soon (using Kamm's circle).

3. We increased the depth of the BDD to determine each geohash uniquely.

Next, the section discusses the novel method of data representation for vehicle future geographical occupancy information using a binary decision diagram (BDD). We show that sharing BDD-based occupancy information is consistent with the C-ITS nature of data sharing since algebraic operations between the exchanged BDDs can confirm the possibility of future interaction. We calculated potential future occupancy using Kamm's circle, shown in the ROS-based simulator. We reported data insertion and collision avoidance check time of the linked list-based BDD on PostgreSQL database-based LDM.

4.2 Potential Risk Estimation Scheme in the Local Dynamic Map using Kamms Circle and Binary Decision Diagrams

Knowing the vehicle's current location is not sufficient to enhance the safety aspect. In addition, since nearby vehicles may not interact in the future, vehicles far away may interact in the near future (See figure 4-11). Thus knowing the future occupancy of the vehicles is of great importance. Therefore, in the following subsection, we attempt to include the future occupancy information of the traffic participants in the LDM.

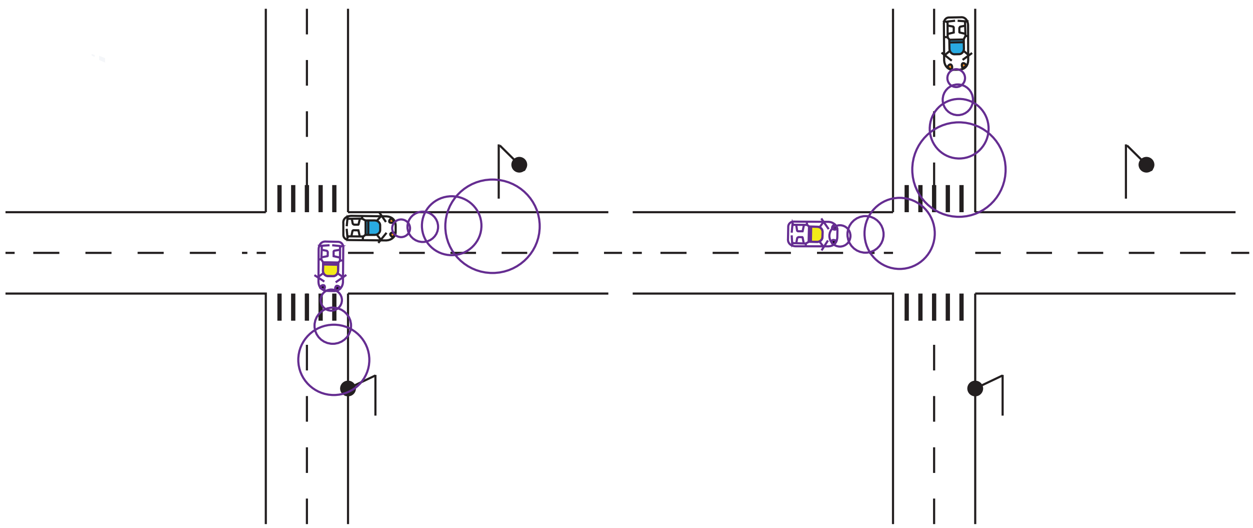


Figure 4-11: Vehicles near may not interact in the future whereas vehicles away may interact.

From above chapters, it is clear that each Geohash has its unique binary representation. This binary representation for locating a region in space motivated us to use BDD since BDDs are reasonably small for many Boolean functions as compared to corresponding binary tree representation. See Figure 3-15. Also, it supports algebraic operations on BDDs, which correspond to equivalent set-theoretic operations. Moreover, Computer-Aided Design, formal verification, and other related fields extensively used BDDs already for Boolean function manipulation successfully.

4.2.1 Geohash set as a BDD

To represent the space available for navigation, enhance space representation in LDM, and deal with an infinite number of available points on the road. Therefore, we considered Geohash as a primary unit space. Furthermore, since Geohash represents a geographical area, its size varies depending on the number of characters/levels Geohash has.

This chapter considers a Geohash of ten levels/characters. It has a distance of approx 0.59 meters from north to south and 0.96 meters from east to west. See Table 3.1. Moreover, to represent the collection of Geohashes, we encoded it using BDD.

1. *BDD representation of a unit Geohash:* A Geohash is a unique symbolic representation of all the points available within the given area on earth. For each character in Geohash, we can have 32 possible values (English alphabets except 'a', 'i', 'l', 'o' and decimal system digits 0-9) and can be represented using five boolean variables ($2^5 = 32$) . See Figure 3-15. Therefore, we used five nodes in a BDD to represent the corresponding Boolean variables for a binary representation of a given character in a Geohash. Thus, in a given Geohash, each character has its five corresponding nodes in the BDD. Since we considered Geohash of 10 characters/levels, we needed 50 nodes for corresponding bits, plus two extra nodes to represent zero and one node in a BDD. (For experiments, we assumed the vehicle will be within a Geohash, having a distance of 4872 meters (north to south) and 3955 meters (east to west). Hence, 5 level BDD with 25 nodes served the purpose) i.e first 5 level of Geohash doesnot change in our setting. Every corresponding node, low or high, has its values depending upon the boolean function represented. To represent a single Geohash using BDD corresponding binary string ends at one (1) node of a BDD, and all other binary string ends in zero. See figure 4-12.

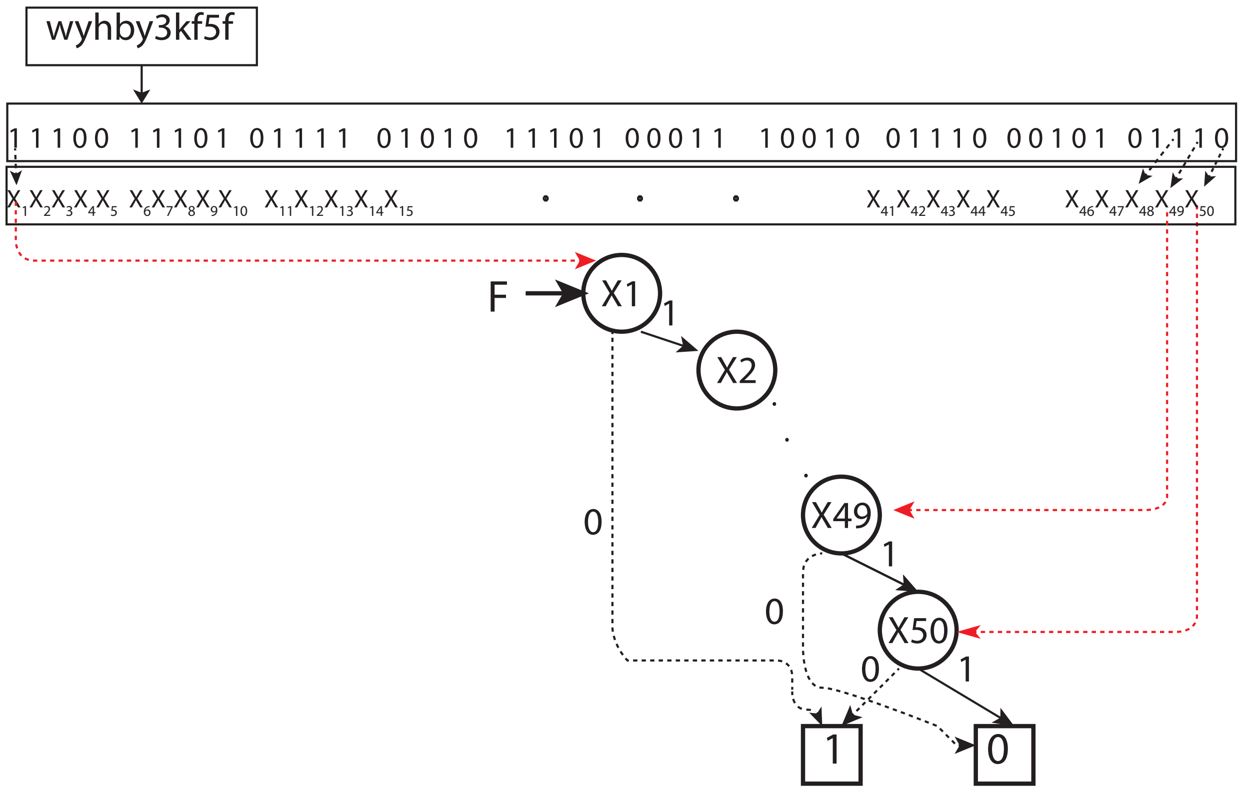


Figure 4-12: BDD representation for a unit Geohash.

2. *BDD representation of a set of Geohash:* We utilized BDDs synthesis (borrowed term ' synthesis' from [76]) to represent a set of Geohashes in a single BDD. We can build BDD's for complex sets/functions using BDD synthesis. E.g., BDD for function f can combine with function g to represent BDD for $f\,AND\,g$, $f\,OR\,g$, $NOT\,f$, $f\,XOR\,g$. Following are corresponding set interpretations for a given BDDs representing f and g sets (here Geohash sets) of the above synthesis operations. We used *apply* method in [77] to achieve following operations :

   (a) $f\,OR\,g$ is the set union operation. $f \cup g = \{\alpha \mid \alpha \in f \; or \; \alpha \in g\}$

   (b) $f\,AND\,g$ is the set intersection operation. $f \cap g = \{\alpha \mid \alpha \in f \; and \; \alpha \in g\}$

   (c) $f\,XOR\,g$ is the set symmetric difference operation. $f \oplus g = (f \backslash g) \cup (f \backslash g)$

   Thus from above, to add a Geohash in a given BDD representation of a set of Geohashes, we performed OR operation between two corresponding BDDs representations, See figure 4-14. Encoded BDD for set of 701 BDDs is shown in figure 4-13. Intersection of two BDDs represent a set of Geohashes can be seen in the figure 4-15.

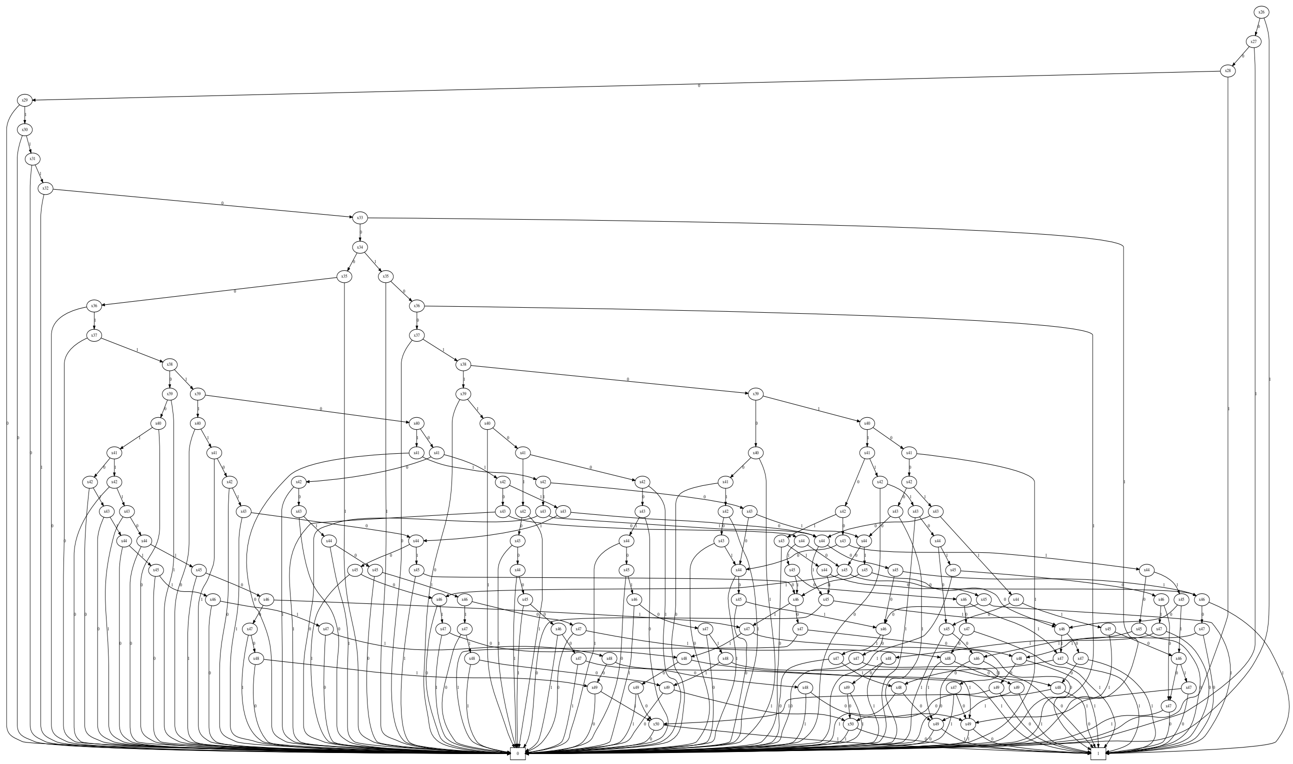


Figure 4-13: BDD for set of 701 Geohashes.(Interconnection between 25th - 50th node is shown for clarity.)

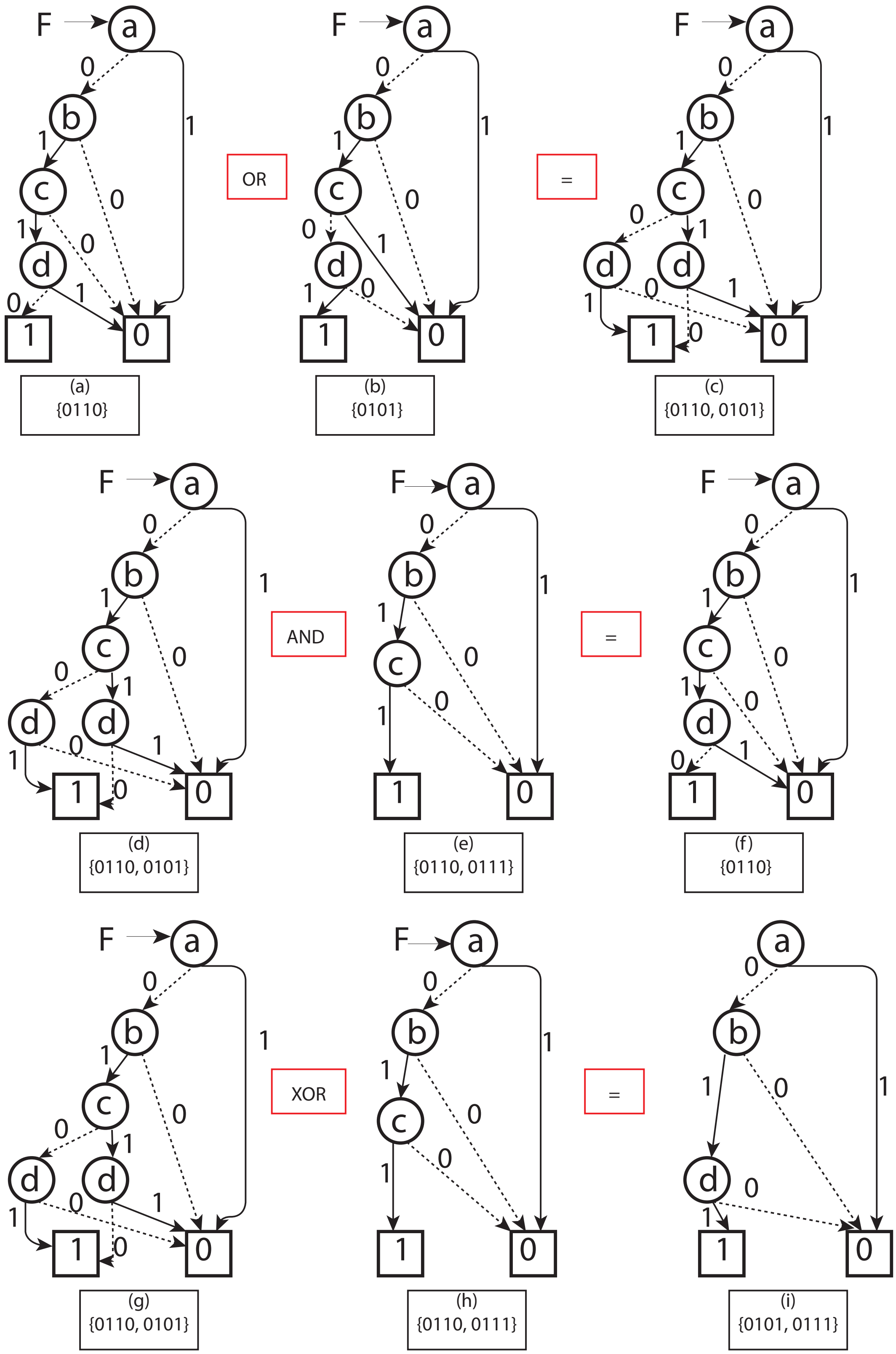


Figure 4-14: (a,b,c) BDD OR operation is equivalent to set union operation. (d,e,f) BDD AND operation is equivalent to set intersection operation. (g,h,i) BDD XOR operation is equivalent to set symmetric difference operation.

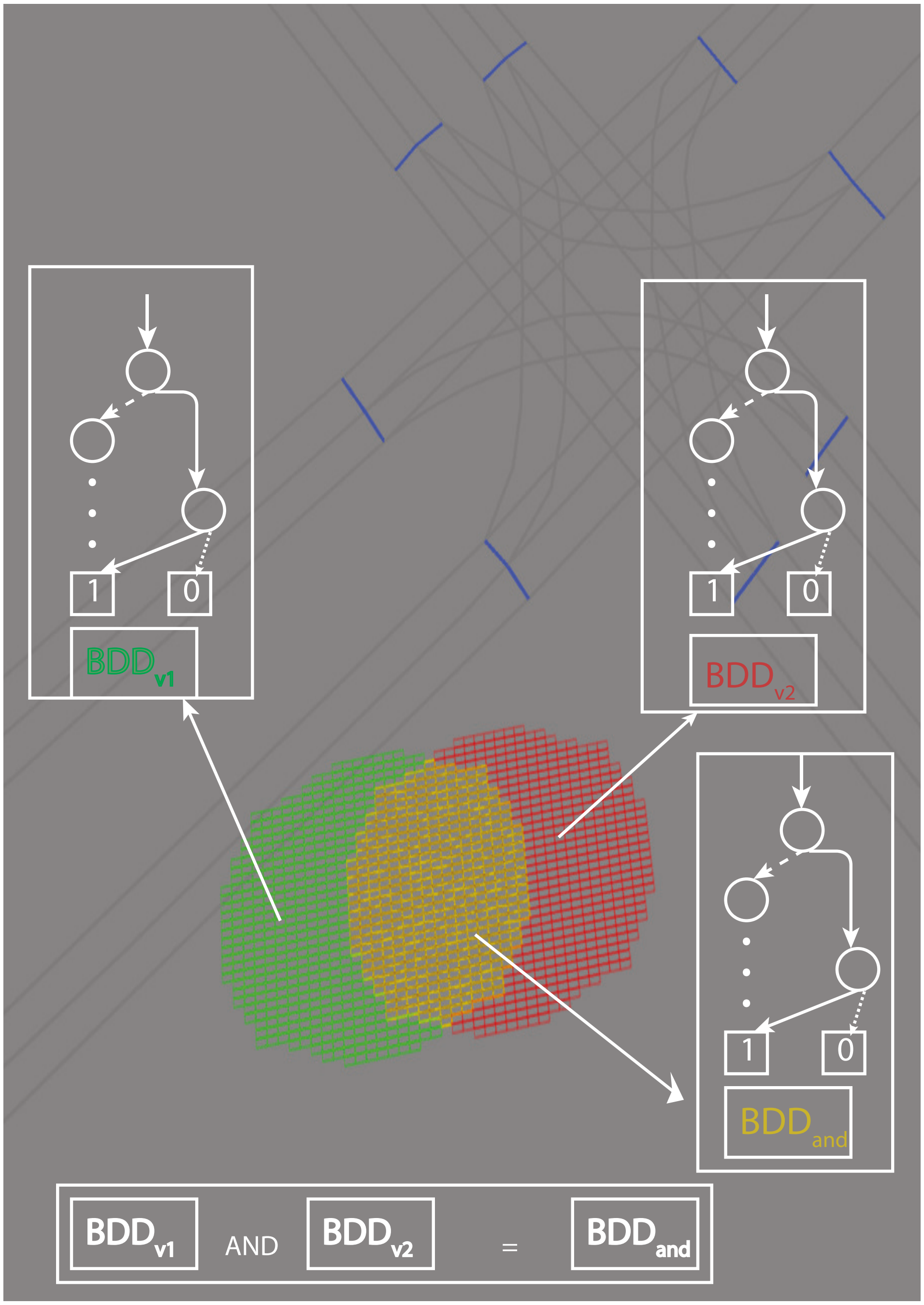


Figure 4-15: BDD intersection for set of geohashes.

4.2.2 Algorithms

Following operations, from [77] were used for Geohash based BDD manipulation in this work.

1. *Reduce:* Give reduced BDD in it's canonical form.

2. *Apply:* Perform synthesis operation between two BDD's. $f_1 < op > f_2$

3. *Satisfy-One:* Returns any one element in S_f. Where S_f is the set of all Geohash represented by a given BDD.

4. *Satisfy-All:* Output S_f. All Geohashes, a given BDD, satisfy.

Eight-Neighbour Geohash generation algorithm for a given BDD

In the previous, subsection we mentioned vehicle occupancy possible over time t using Kamm's/Traction circle. This subsection discusses the algorithms used to generate the BDD for such vehicle occupancy over time t. We used Algorithm 1, Algorithm 2 and Algorithm 3 to find (east, west), (north, south), (north-east, north-west, south-east, south-west) neighbours BDD for a given Geohash encoded BDD, respectively, along with Algorithm 5 to generate the concerned Kamm's/Traction circle BDD.

First, algorithms 1, 2 and 3 used the Satisfy-One [77] method to find an input that satisfies the BDD. Then, T calculates the transition bits [78], it computes value 1 for the needed bits flip needed to calculate the neighbour Geohash. Finally, we generated the bit string for the neighbour after the XOR operation between the satisfying input with the transition string.

Modified Midpoint Circle Generation Algorithm

Moreover, the mid-point circle generation algorithm is modified (see algorithm 4) to find the BDD for all the Geohash present inside a given circle of radius r. Mid-point circle generation algorithm is used in computer graphics to rasterize the circle. The mid-point circle generation algorithm uses the 8-way symmetry present in the circle. Therefore, if we can calculate the points in one octant, we can generate the points in all other seven octants. Assuming the centre is (0,0) Mid-point circle generation algorithm in step, I calculate the first square/pixel at $(x_0, y_0) = (0, r)$. After that, to generate the next squares/pixels in the first quadrant, p a decision parameter finds its use. Step II calculates the p decision parameter initial value $p_0 = \frac{5}{4} - r$. Then, in Step III, depending upon the weight of decision parameter p, the successive value of 'p' and squares/pixels takes their value as follows:

If $p_k < 0$ then:

$(x_k, y_k) = (x_k + 1, y_k)$ and new p_k is calculated as $p_{k+1} = p_k + 2x_{k+1} + 1$

else:

$(x_k, y_k) = (x_k + 1, y_k - 1)$ and new p_k is calculated as $p_{k+1} = p_k + 2x_{k+1} + 1 - 2y_{k+1}$

Next, in Step IV, Algorithm determines symmetry points in the other seven octants and repeat steps III to IV until $x <= y$.

The modified mid-point circle generation Algorithm 5 generated the BDD of all the Geohashes contained in the circle of radius r. See figure 4-12. Step I is the initialization step. In step II, the BDD for all the Geohashes with a Black arrow are generated as shown in the figure and merged with the BDD (circle_BDD) to represent all Geohashes within the circle by using or operation, as or operation on BDD is equivalent set union operation. Then, in Step III algorithm initialized the decision parameter p with $p = INTEGER(ROUND(5/4) - r)$. Step IV, depending upon the value of p, generated successive Geohashes available in the first quadrant, successive p values and more parameters of the circle as follows:

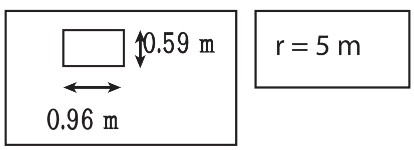
if $p <= 0$:

Generate east BDD and union it with circle_BDD. Also update the value $x\_k = x\_k + 0.96$, $e\_count = e\_count + 1$ and record the north limit of this BDD from the center. Finally, update the value of decision parameter as $p = p + 2 * x\_k + 1$.

else:

Generate south east BDD and union it with circle_BDD. Also update the value $x\_k = x\_k + 0.96; y\_k = y\_k - 0.59$ and record the north and east limit of this BDD from the center. Finally, update the value of decision parameter as $p = p + 2 * x\_k + 1 - 2 * y\_k$.

In Step V, the algorithm generated BDD3 and BDD4 in the east and west direction of the origin, as shown in figure 4-12 and depending upon the value of BDD3 (east) and BDD4 (west) Geohash; the algorithm generated BDD's in north and south directions taken the first quadrant north limit as a limit (green arrow). All caused BDDs are taken union with circle BDD (circle_BDD). Finally, in Step VI, depending upon the value of the east limit of the first quadrant new limit a is calculated and generated BDDs upon added with BDD representing the circle.



i	Pi	Xi+1, Yi+1	2Xi+1	2Yi+1
0	-4	(0,5.3)	1.8	10.8
1	-1.2	(0.9,5.3)	3.6	10.8
2	3.4	(1.9,4.7)	5.4	9.6
3	0.2	(2.8,4.1)	7.2	8.4
4	0	(3.8,3.6)		

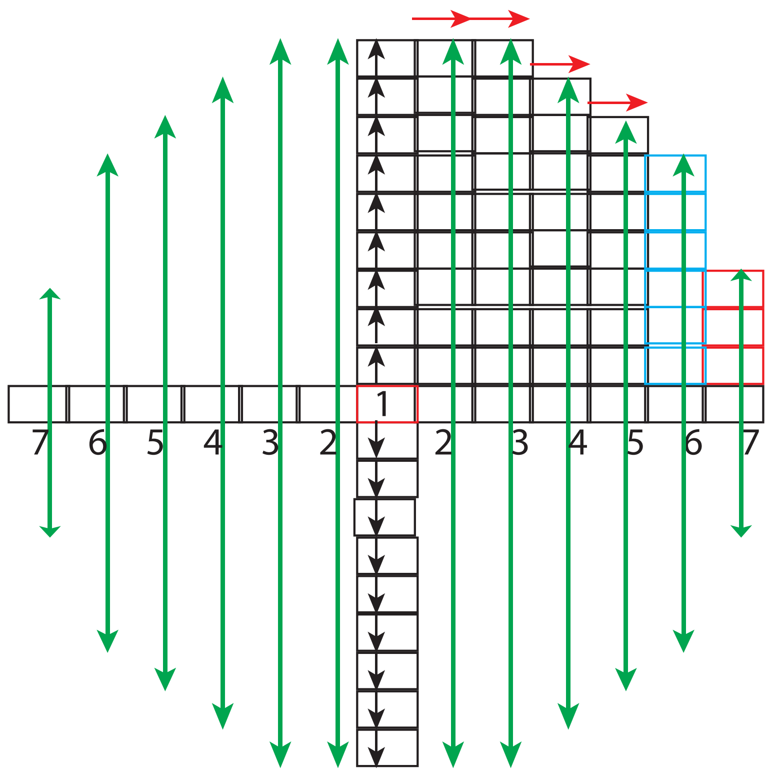


Figure 4-16: Modified Midpoint Circle Algorithm

Algorithm 1: Algorithm to find west/east Neighbour BDD for a given Geohash BDD.

```
Input: inpGeo -Geohash BDD. h in {east, west}.
Output: West/East Neighbour Geohash BDD.
S = Satisfy-One(inpGeo)
T = [0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0]
If h == west then
   for i = 1 to T.length increment 2:
      for j = 1 to i increment 2:
          T[j] = T[j] and not(S[j])
else:
   for i = 1 to T.length increment 2:
      for j = 1 to i increment 2:
          T[j] = T[j] and S[j]
T[T.length-2] = 1
for i = 1 to T.length increment 2:
   S[i] = S[i] xor T[i]
return createStringtoBDD(S)
```

4.2.3 Experiment

To experiment, we created the lanelet map [33] for scenario 1 (see figure:4-17) and scenario 2 (see figure:4-19) using JavaOpenStreetMap (JOSM) and loaded them into ROS based simulator CoInCar-Sim with multiple vehicles. Also, the vehicle data is generated in scenario '2' and stored as a CSV file. Data fed from CSV files into the LDM at every interval of 50ms and ego vehicle query the LDM to get information for collision detection task at every 100ms, same as experiment setup in [13]. To check the performance of our approach and compare it with previous results, we created a schema of LDM tables as mentioned in shimada et al.[13] for their safety driving system setup. We build LDM above Postgres database. Furthur constructed a 'roadelement' table to store the lanelets corresponding to scenarios '1' and '2' static layers. Also, we build an 'egomotorvehicle' and 'motorvehicle' layer four tables to keep the ego vehicle and non-ego vehicle information, respectively. we built an 'alongroadelement' table to link-layer one and four tables. As per the setup mentioned in [13].

Algorithm 2: Algorithm to find north/south Neighbour BDD for a given Geohash BDD.

```
Input: inpGeo -Geohash BDD. v in {south, north}.
Output: South, North Neighbour Geohash BDD.
S = Satisfy-One(inpGeo)
T = [1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1]
If v == south then
   for i = 0 to T.length increment 2:
      for j = 0 to i increment 2:
         T[j] = T[j] and not(S[j])
else:
   for i = 0 to T.length increment 2:
      for j = 0 to i increment 2:
         T[j] = T[j] and S[j]
T[T.length-1] = 1
for i = 0 to T.length increment 2:
   S[i] = S[i] xor T[i]
return createStringtoBDD(S)
```

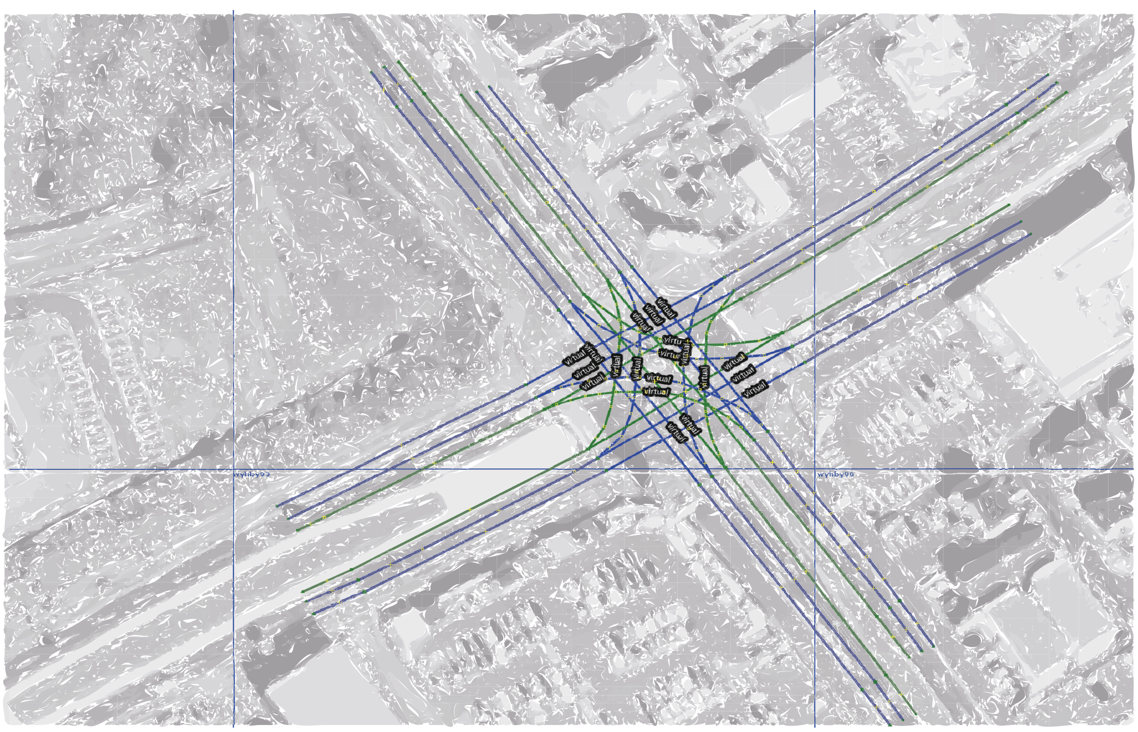


Figure 4-17: (Scenario-1) Intersection Scenario

Algorithm 3: Algorithm to find diagonal Neighbour BDD for a given Geohash BDD.

```
Input: inpGeo -Geohash BDD. h in {west, east}, v in {south, north}.
Output: Output: Diagonal Neighbour in north-west,north-east, south-east,
  south-west Geohash BDD.
S = Satisfy-One(inpGeo)
T = [1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1]
for i = 0 to T.length:
   for j = 0 to i:
      If j%2 == 0 then
         If h == west then
            T[j] = T[j] and not(S[j])
         else:
            T[j] = T[j] and S[j]
      else:
         If v == south then
            T[j] = T[j] and not(S[j])
         else:
            T[j] = T[j] and S[j]
T[T.length-1] = 1
T[T.length-2] = 1
for i = 0 to T.length:
   S[i] = S[i] xor T[i]
return createStringtoBDD(S)
```

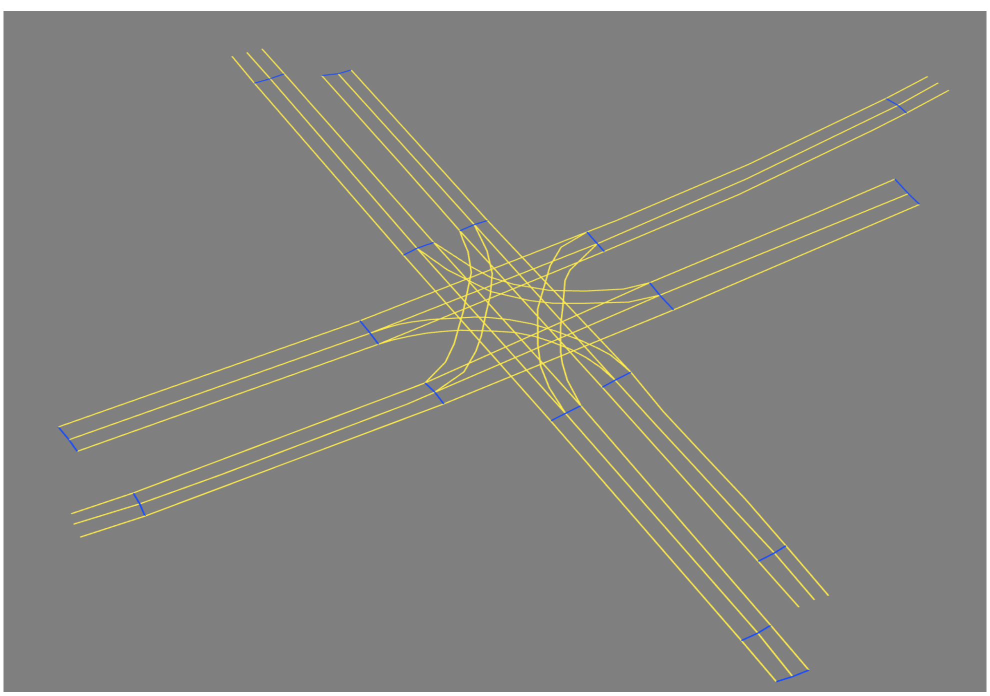


Figure 4-18: Loaded lanelet map in CoincarSIM simulator.

Algorithm 4: Midpoint Circle Generation Algorithm.

Input: **r** - radius of a circle., (x_c, y_c) center of the circle.

Output: Output: Squares to include on a square grid to form a circle of radius r.

I. First square to include $((x_0, y_0) = (0, r))$

II. Calculate the initial value for the decision parameter.

$$p_0 = \frac{5}{4} - r$$

III. For successive value of k. (x_k, y_k) is determined as follows.
If $p_k < 0$ then:
$(x_k, y_k) = (x_k + 1, y_k)$ and new p_k is calculated as $p_{k+1} = p_k + 2x_{k+1} + 1$
else:
$(x_k, y_k) = (x_k + 1, y_k - 1)$ and new p_k is calculated as
$p_{k+1} = p_k + 2x_{k+1} + 1 - 2y_{k+1}$

IV. Determine the symmetry points in other seven octants.

V. Repeat the step III to IV until $x \leq y$.

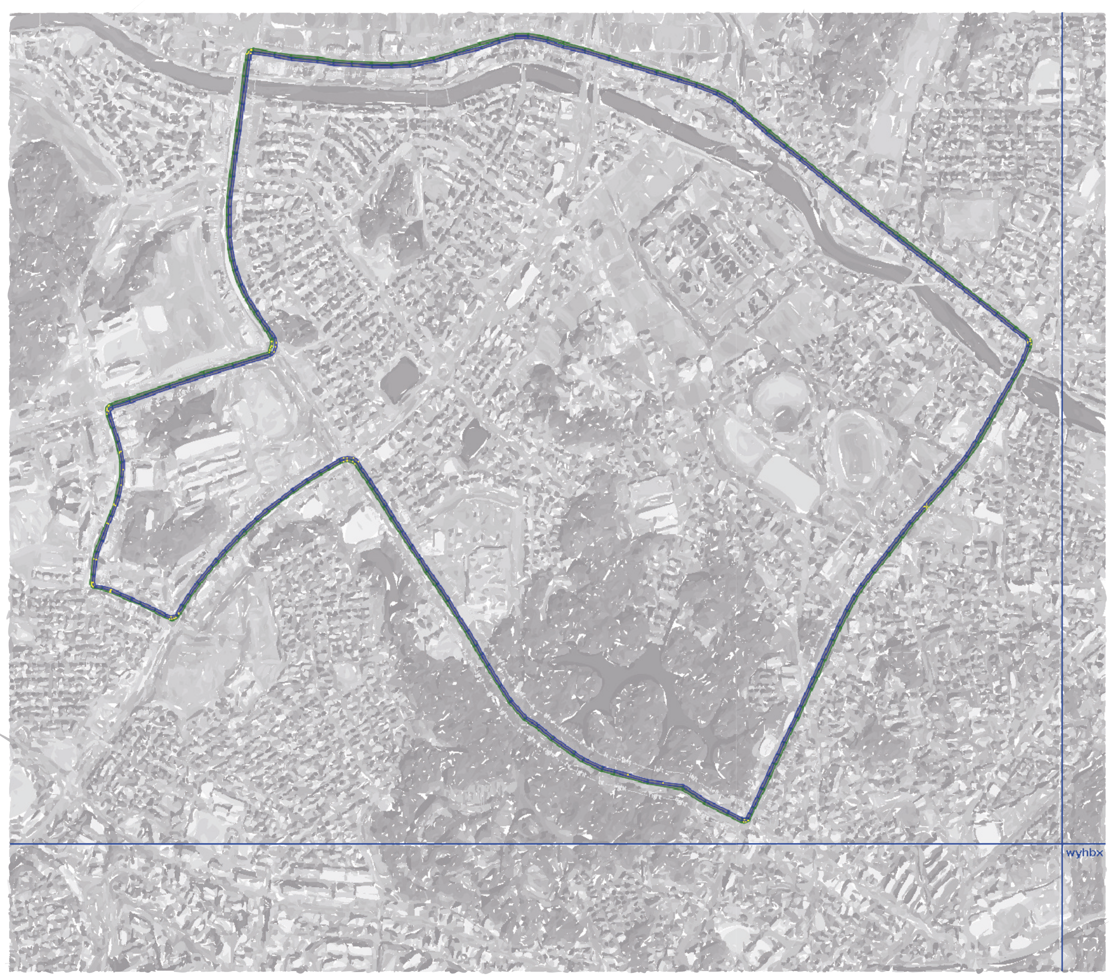


Figure 4-19: (Scenario-2) City Road Scenario

Algorithm 5: Modified Midpoint Circle Generation Algorithm.

```
Input: inpGeo - Center Geohash BDD, r - radius in meters unit.
Output: BDD for a collection of Geohashes containing a circle of the given
  radius.
Begin
    Step I.
    up_count = ⌈radius/0.59⌉
    quad1_north_limit = quad1_east_limit = []
    circle_BDD = inpGeo
    BDD1 = BDD2 = BDD3 = BDD4 = inpGeo
    x_k = y_k = 0;  n_count = e_count = 0
    Step II.
    for k = 0 to up_count:
      a. BDD1 = Generate north BDD of BDDs.
      b. BDD2 = Generate south BDD of BDDs.
      c. y_k = y_k + 0.59; d.n_count = n_count + 1
      e. BDD1 ∪ BDD2 ∪ circle_BDD. /*Apply union of BDD1 and BDD2
     BDD with circle_BDD*/
    Step III.
    p = INT(ROUND(5/4) - r)
    Step IV.
    while x_k <= y_k:
      if p <= 0:
        a. BDD1 = Generate east BDD of BDD1.
        b. x_k = x_k + 0.96; c. e_count = e_count + 1
        d. quad1_north_limit.append(n_count)
        e. BDD1 ∪ circle_BDD.
        f. p = p + 2 * x_k + 1
      else:
        a. BDD1 = Generate south east BDD of BDD1.
        b. BDD1 ∪ circle_BDD.
        c. x_k = x_k + 0.96; d. y_k = y_k - 0.59
        e. quad1_east_limit.append(e_count)
        f. e_count = e_count + 1; g. n_count = n_count - 1
        h. quad1_north_limit.append(n_count)
        j. p = p + 2 * x_k + 1 - 2 * y_k
    quad1_east_limit.append(x_count)
    /* Till this point quad1_north_limit contains distance(in no. of Geohash)
     of all Geohash in first quadrant of the circle in north direction and
     quad1_east_limit distance (in no. of Geohash) of Geohash in first
     quadrant of the circle in east direction w.r.t inpGeo.*/
```

Step V.
for w = 0 to quad1_east_limit.*length*-1:
  a. Generate BDD3 and BDD4 east and west of BDD3 respectively.
  b. $circle\_BDD \cup BDD3 \cup BDD4$
  for k = 0 to quad1_north_limit[w]:
    a. Generate BDD5 and BDD6 north and south of BDD3 respectively.
    b. Generate BDD7 and BDD8 north and south of BDD4 respectively.
    c. $circle\_BDD \cup BDD5 \cup BDD6 \cup BDD7 \cup BDD8$
Step VI.
for w = quad1_east_limit.*length*-1 to 0:
  a. Generate BDD3 and BDD4 east and west of BDD3 respectively.
  b. $circle\_BDD \cup BDD3 \cup BDD4$
  c. a = $\lceil x\_count[w] * (1.6) \rceil$ /*1.6 is ratio Geohash (10 level) breadth to height*/
  if $a >= quad1\_north\_limit[w]$ then:
    a. a = quad1_north_limit[w]
  for k = 0 to a:
    a. Generate BDD5 and BDD6 north and south of BDD3 respectively.
    b. Generate BDD7 and BDD8 north and south of BDD4 respectively.
    c. $circle\_BDD \cup BDD5 \cup BDD6 \cup BDD7 \cup BDD8$
return circle_BDD

All experiments are performed in Ubuntu 18.04 environment on a computer with Intel(R) Core(TM) i9-9900K CPU (3.60GHz) having 64 GB RAM. For simplicity, we considered $a_{max} = 10m/s^2$ value corresponding to friction coefficient $\mu = 1.02$ and $g = 9.81m/s^2$. For the generation of Kamm's/Traction circles, we took a time step size $\Delta t = t_{i+1} - t_i$ of 0.4 seconds and up to a time horizon of $t_h = 1.2$ seconds. We computed the BDD of all the Geohash presents inside the concerned circles using Algorithm 5. After that, we converted the BDDs to JSON format to make them suitable to save in the databases.

4.2.4 Results

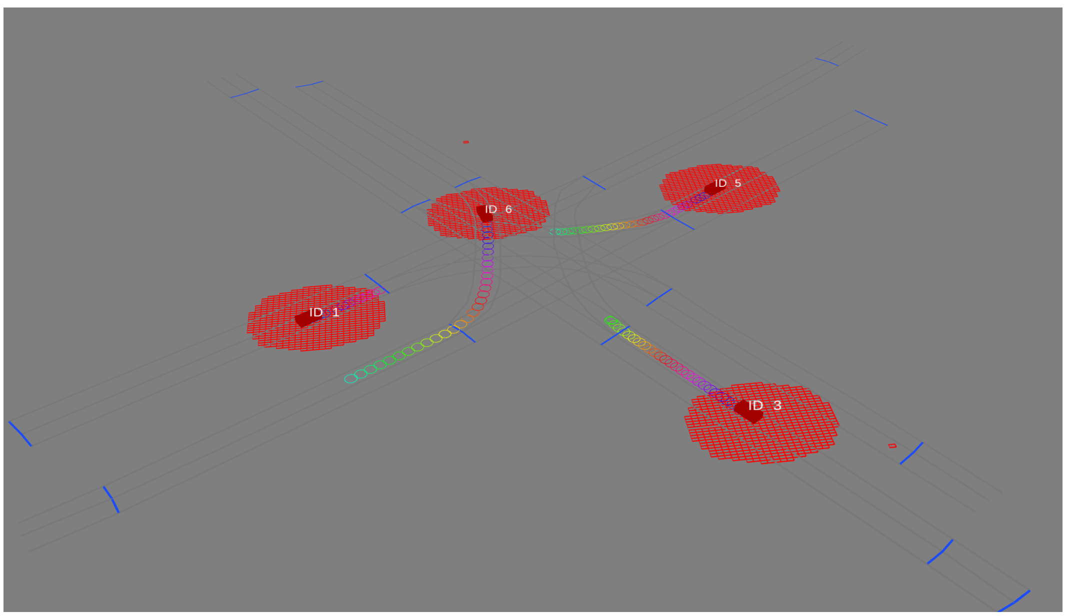


Figure 4-20: Union of geohash corresponding to the Kamm's circle at 0.4, 0.8 and 1.2 seconds.

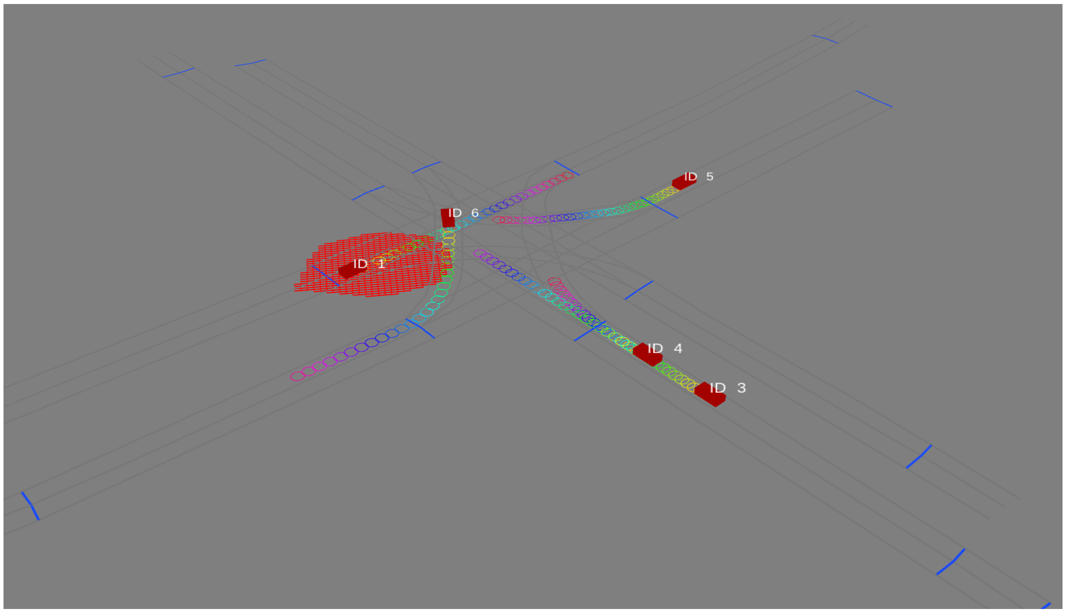


Figure 4-21: Union of geohash corresponding to the Kamm's circle at 0.3, 0.7 and 1.2 seconds.

Figure 4-20 shows the generated Kamm's circle Geohashes for given vehicles moving towards the intersection (scenario '1'). Figure 4-22 shows the insertion time of vehicles data into the given LDM when we add extra information of reachable Kamm's circle Geohashes BDDs in JSON form. It increased the time for insertion operation since we insert more information into the LDM than only inserting point position information data into the Postgres-based LDM. Then also time taken is much lesser than 50 ms for 50 vehicles, which indicates its suitability for real-time operations of VITS (Vehicle Intelligent Transport system) based LDM.

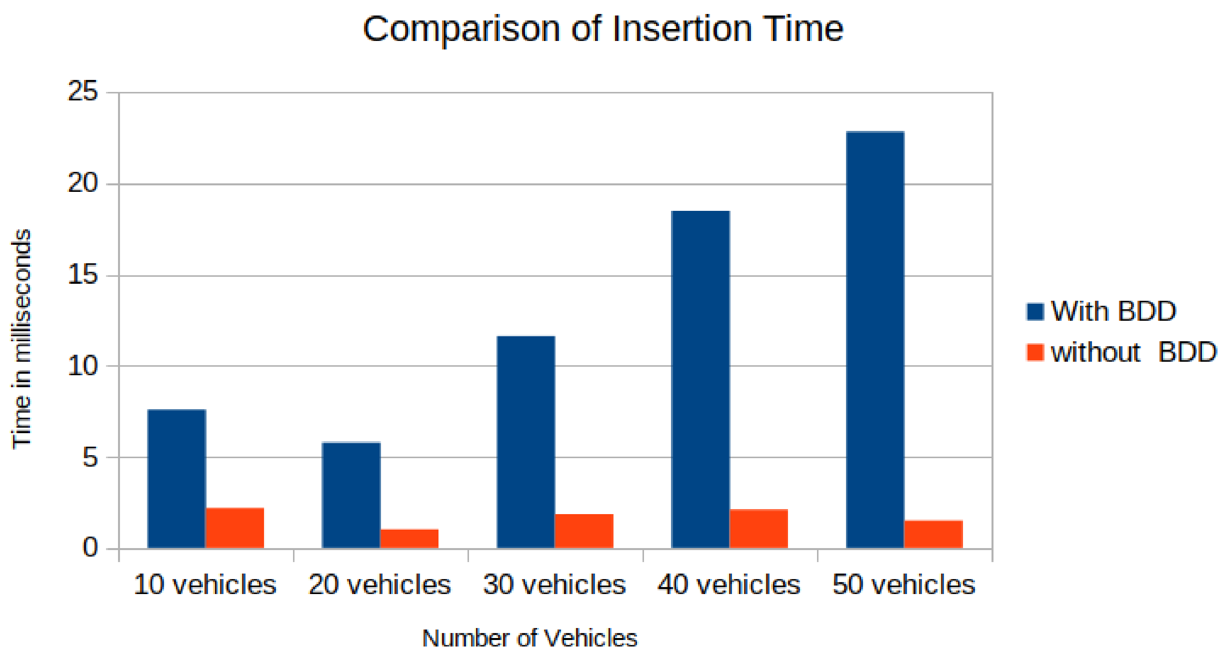


Figure 4-22: Layer 4 data insertion time with BDD vs without BDD

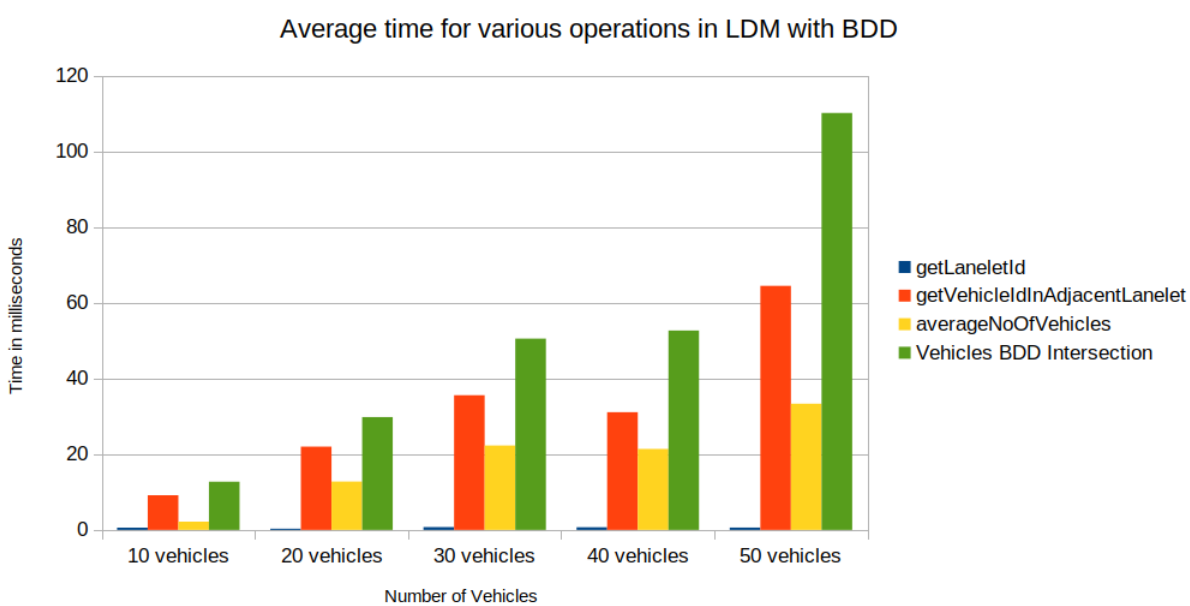


Figure 4-23: Time in milliseconds for operations (get ego vehicle Lanelet id, get vehicles Id's in adjacent Lanelets of ego vehicle, Average number of vehicles in adjacent Lanelets, BDDs Intersection operation with adjacent vehicles for collision avoidance).

Figure 4-23 and 4-24 shows the difference we observed by introducing BDD in the

LDM for the following tasks:-

- *Task1*- getLaneletId (To get the lanelet id and data corresponding to an ego vehicle).
- *Task2*- getVehicleInAdjacentLanelet (To retrieve data of all vehicles (other than ego) present in the ego vehicle current lanelet or its adjacent lanelets).
- *Task3*- averageNoOfVehicles (To retrieve the number of vehicles present around an ego vehicle for a given scenario).
- *Task4*- Collision avoidance, Retrieve BDDs using Task2 and check for collision avoidance following the 'AND' operation on BDDs in figure 4-23 and collision risk warning task following the procedure in [13] for figure 4-24.

We observe an increase in time for various tasks like Task2 and Task4 by introducing BDD (Figure 4-23) in the LDM compared to shimada et al. implementation in Figure 4-24. Till 40 vehicles, the functions performed take less than 100 ms, which is necessary for real-time operation.

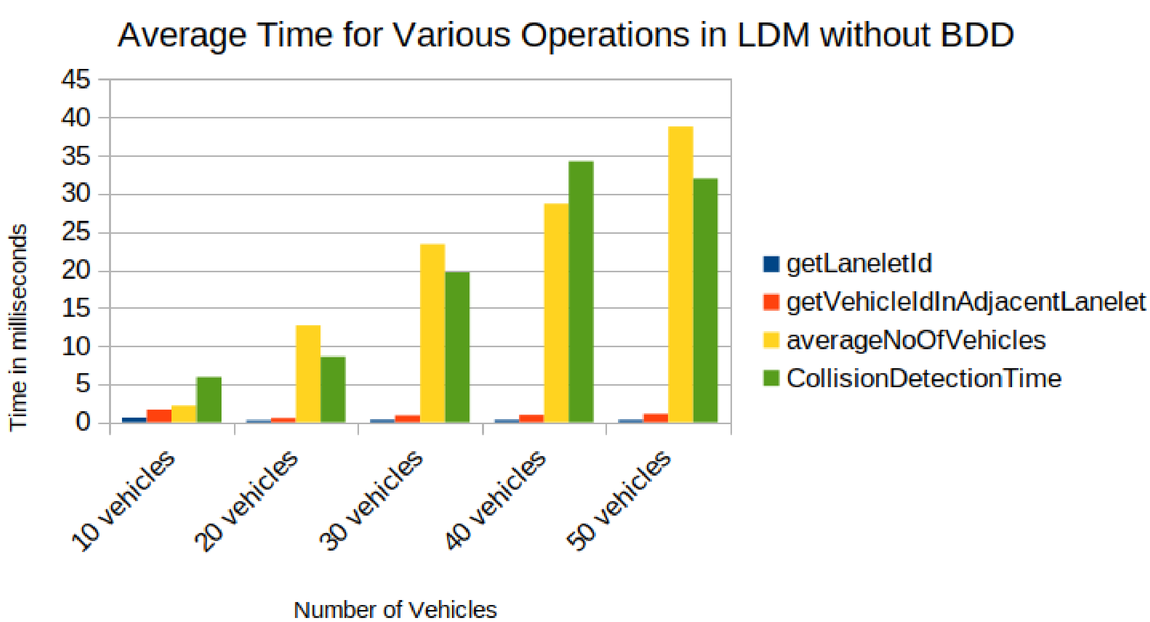


Figure 4-24: Time in milliseconds for operations (get ego vehicle lanelet id, get vehicles id's in adjacent Lanelets of ego vehicle, Average of vehicles in adjacent Lanelets, collision risk warning algorithm from [5].)

The summary above approach is shown in the figure 4-25 below:

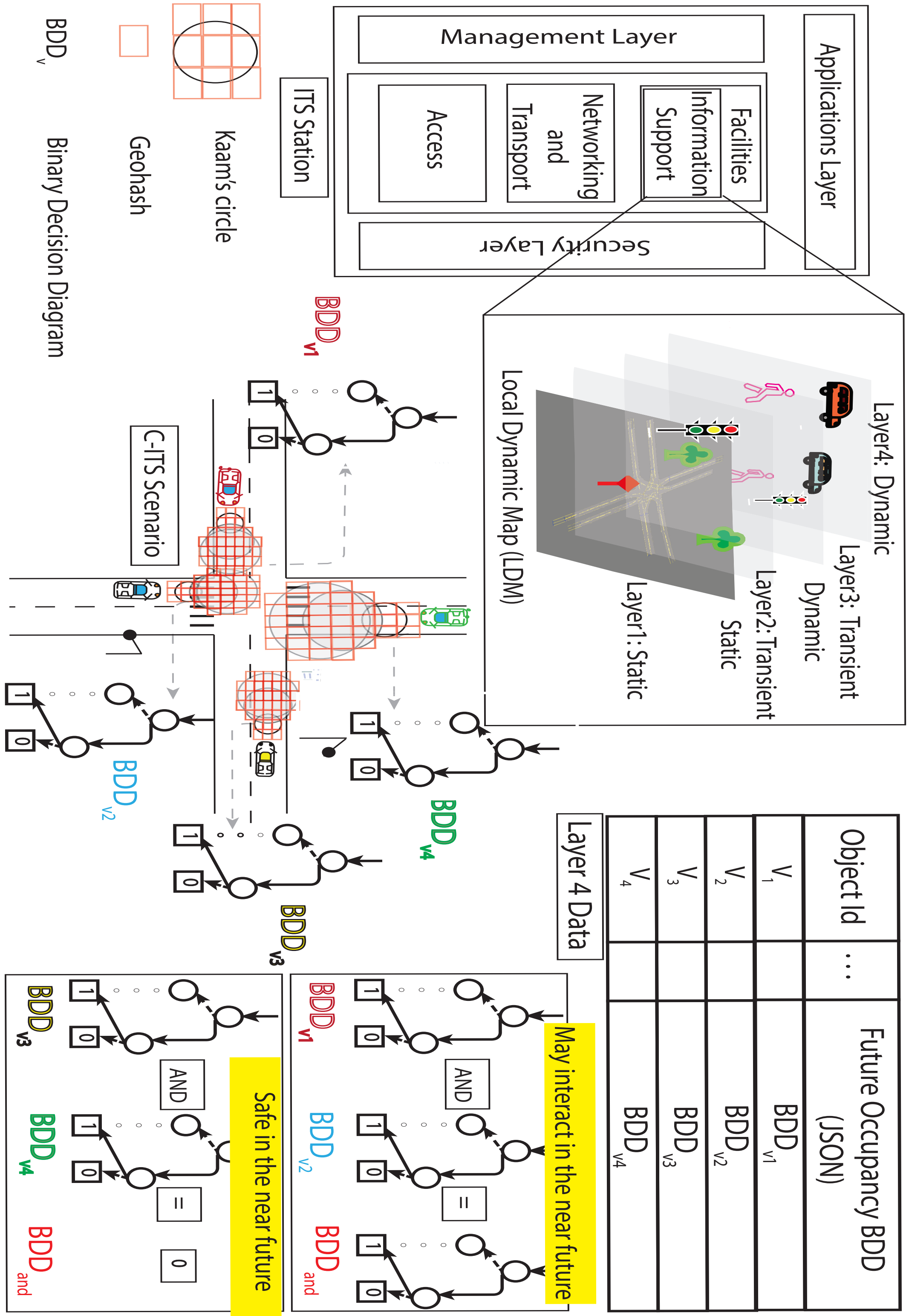


Figure 4-25: Summary of the Potential Risk Estimation Scheme in the Local Dynamic Map using Kamms Circle and Binary Decision Diagrams.

Chapter 5

Discussion

In this Thesis, we successfully presented the use of decision diagrams along with a Geohash-based setup. In addition, we reported a near-miss detection task using SBDD. Also, a novel data representation method was proposed for vehicle future geographical occupancy information using a BDD. Furthermore, it was shown that sharing BDD-based geographical information support algebraic operations between the exchanged BDDs and can confirm the possibility of future interaction, which supports the C-ITS nature of data sharing. Also reported the data insertion and collision avoidance check time of the linked list-based BDD on PostgreSQL database-based LDM.

Our proposal relied on V2V and V2I to support safe navigation in the C-ITS setup. The use of decision diagrams for the quick detection of the nearby traffic participants and the representation of future geographical occupancy was successfully used and associated with the LDM database. On the other hand, verifying the above approach with the actual vehicle was out of range in the current study. Although the above process may have various benefits, like any method, the proposed may suffer multiple drawbacks. Communication delay/ data loss is an inevitable shortage since the heavy dependency of the above procedure on the communication setup. Despite this, the above approach can support safe navigation since it supports sensor-based navigation to overcome the limitations on sensors such as stereo cameras and lidar limitations on the range as well as due to weather conditions. Approaches of sensor or V2X both have their limits and benefits, so a balanced approach between the methods will be necessary for the future to enhance the safety aspect of autonomous vehicles. We used a time of 1.2 seconds to find future occupancy in our approach. Although, minimum swerving time can be used as a more reasonable time parameter in the future. In addition, ITE-based implementation of the BDDs and algebraic properties of the various decision diagrams, such as ZDD/ MTBDDs, may help enhance the performance of the above approach and support even more functionalities using their corresponding algebras. Thereon let us discuss the point by point about the topics of the approaches discussed in previous chapters.

5.1 Challenges

While realizing the above setup in real life, we may face the following challenges.

1. SBDD setup for checking equivalent Boolean is only suitable for fitting in the central server. Since verification of the equivalent function is based on the same memory addresses, this condition limits the equivalence check to the one central memory setup.

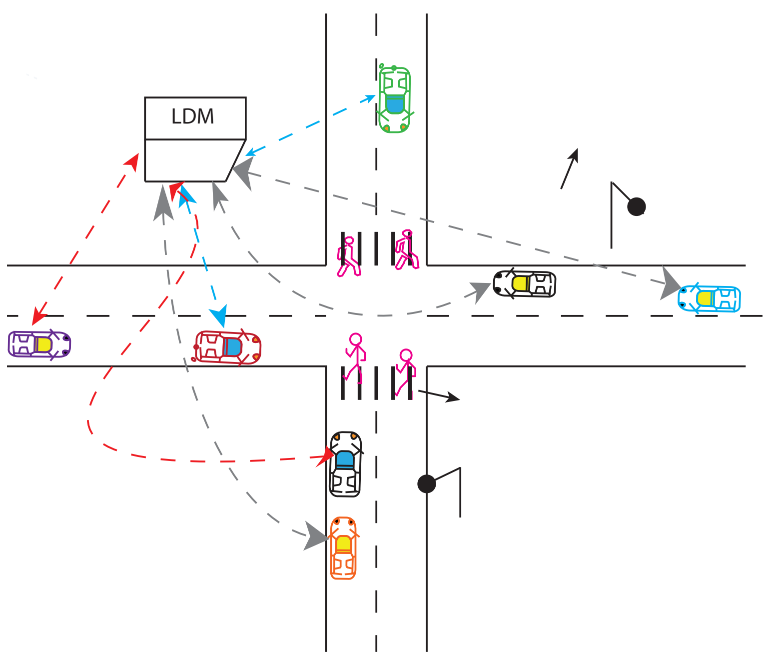


Figure 5-1: Equivalence checks are based on memory locations in the central LDM.

In such a situation, we must maintain the SBDD into one memory-based LDM. Otherwise, we may need to maintain virtual memory addresses if the distributive memory setup is available.(See figure: 5-1)

2. In the case of ROBDD sharing, we can afford distributed LDM checking for the collision avoidance task. Still, We must take care that the node id assigned to the nodes of the BDDs must be unique among the participating vehicles since algorithms for the BDD operations use this unique node id feature to support various functions (e.g. Apply [77]).

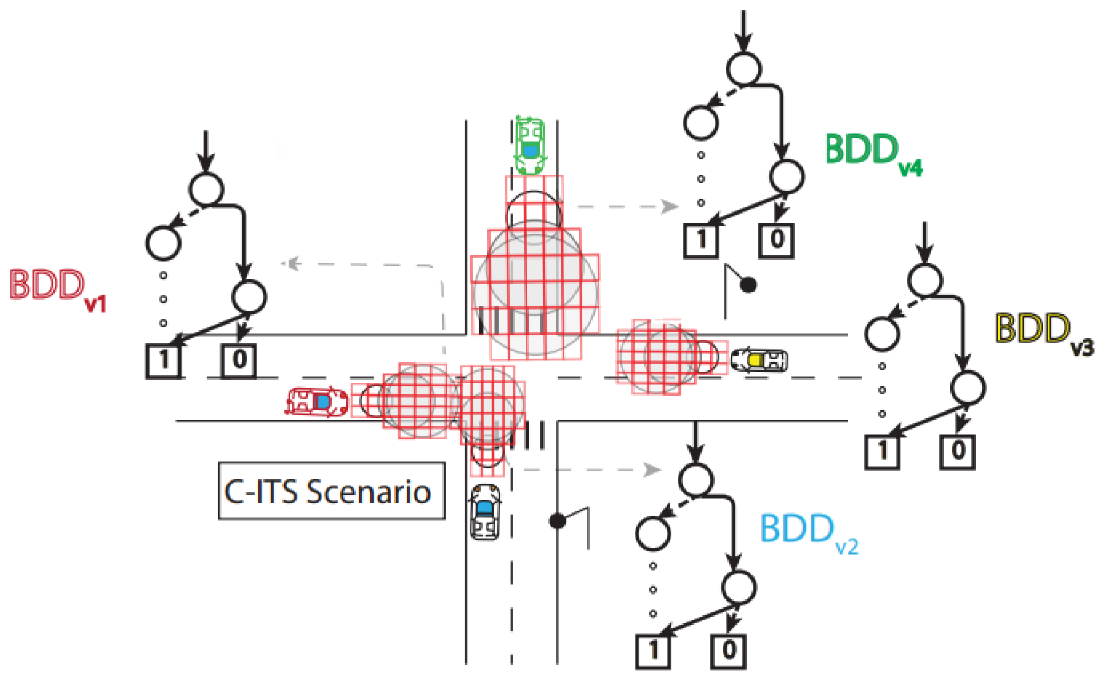


Figure 5-2: Node id of the shared BDD nodes must be unique among the participating vehicles.

3. According to the vehicle's speed, we may need to sample more of the Kamm's circle. Since we may miss space in between the circles 5-3 or we can also consider all the geohash present inside the polygon enveloping the Kamm's circles. Polygon envelop is given as [46]:

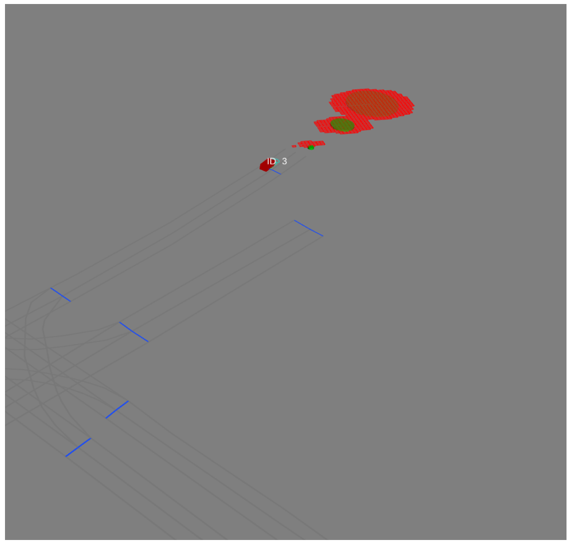


Figure 5-3: Kamm's circle at 0.4, 0.8 and 1.2 seconds for the vehicle at speed of 40 m/s.

$$
\begin{aligned}
q_1 &= [c_x(t_k) - r(t_k), c_y(t_k) + r(t_k)]^T \\
q_2 &= [b_x(t_{k+1}), c_y(t_{k+1}) + r(t_{k+1})]^T \\
q_3 &= [c_x(t_{k+1}) + r(t_{k+1}), c_y(t_{k+1}) + r(t_{k+1})]^T \\
q_4 &= [c_x(t_{k+1}) + r(t_{k+1}), c_y(t_{k+1}) - r(t_{k+1})]^T \\
q_5 &= [b_x(t_{k+1}), c_y(t_{k+1}) - r(t_{k+1})]^T \\
q_6 &= [c_x(t_k) - r(t_k), c_y(t_k) - r(t_k)]^T
\end{aligned}
$$

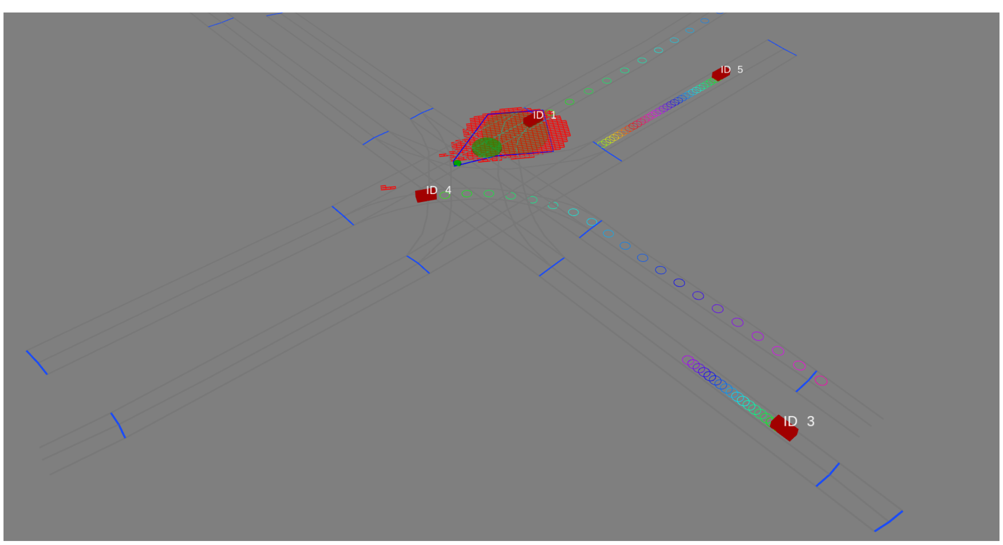


Figure 5-4: Polygon envelop for the kamm's circle.

5.2 Variable Order

BDD are very sensitive to the variable ordering; If we choose the wrong variable order, we may end up representing a BDD with too many nodes. However, in our case, many variable orders are possible. Nevertheless, we checked for 10000 different variable orders. As a result, we found that the variable order used in this thesis (Please refer figure 5-7) took the least number of nodes to represent the ROBDD. (See figure: 5-5). The following figure represents the change in the number of nodes needed to represent a set of given Geohash list (refer, Appendix A) while randomly varying the variable orders.

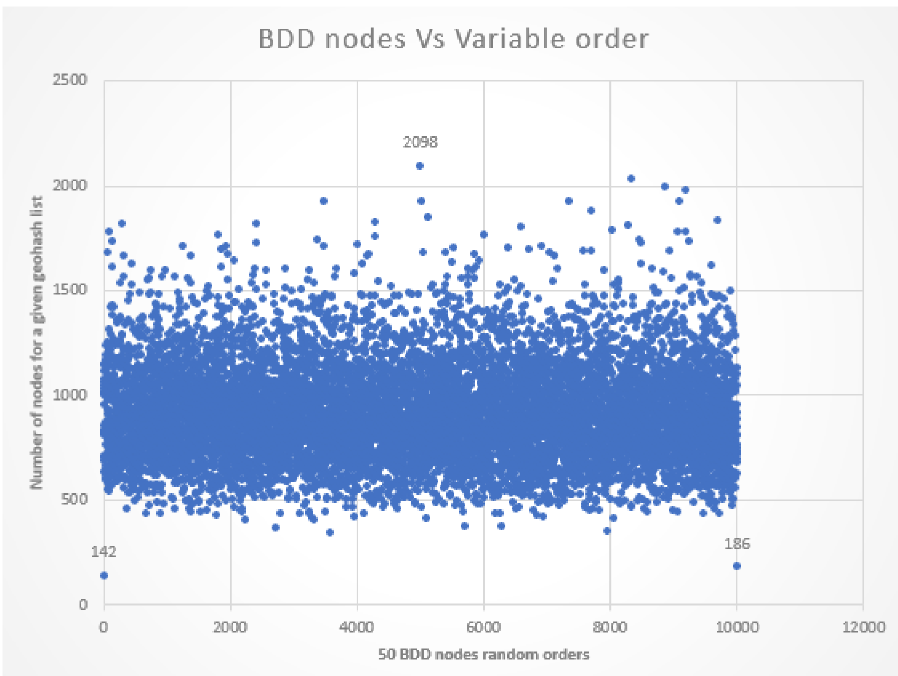


Figure 5-5

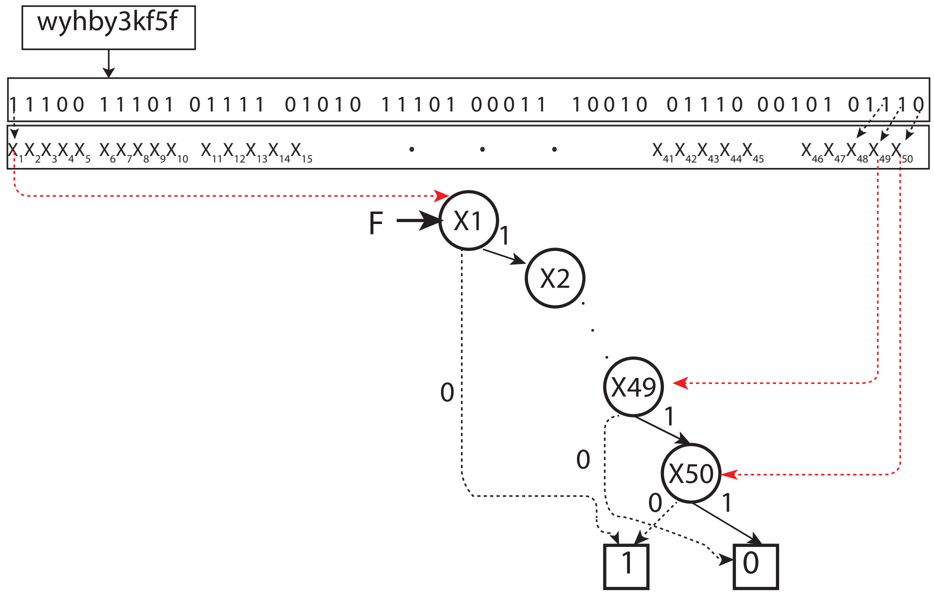


Figure 5-6: Variable order the has minimum number of nodes.

5.3 Traffic rules

Using the Kaam/friction circle to predict vehicle position in the future is over-approximated. The underlying assumption is that the vehicle has the freedom to drive in whatever direction. However, traffic rules and road geometries must always be respected in reality. For instance, a car cannot cross the solid line to enter an adjacent lane. Restriction regarding lane restriction can be achieved in the LDM setup. LDM has four layers: Layer 1 stores static information, and Layer 4 stores dynamic information. Vehicles future occupancy using Kamm's circle. If we also store the Geohash of the road in the static Layer 1, then; the above restrictions can be quickly achieved. Rule to be within the lane and not crossing adjacent lanes can be computed using the AND operation between layer 4 BDD_{l4} and BDD_{l1} corresponding to layer 1 of the LDM.

Similarly, for the traffic rules, lane/ curb restrictions can be achieved by taking controlled AND (BDD AND operation are equivalent to set intersection operation) between the Layer 4 BDDs and Layer 1 BDDs. The output regarding the concerned point can also be understood as per the following figure. However, since the domain for the traffic rules is significantly large, all traffic rules are out of the scope of the current study.

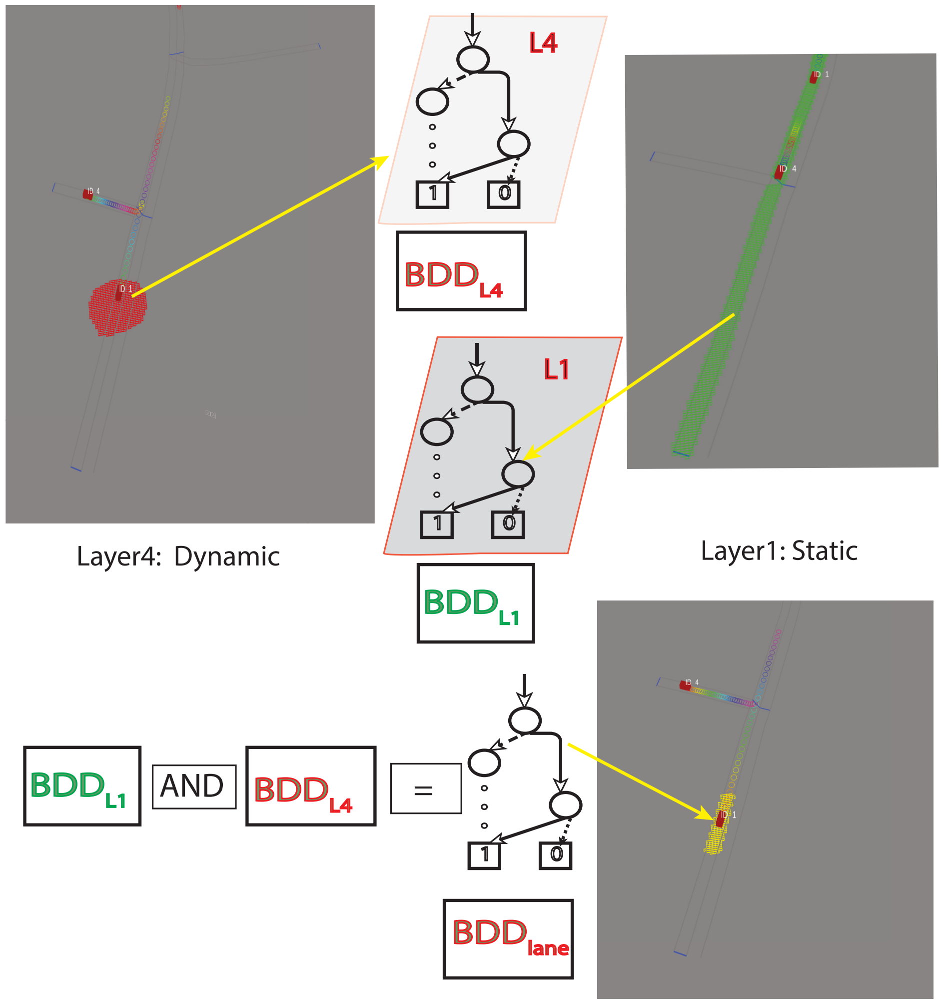


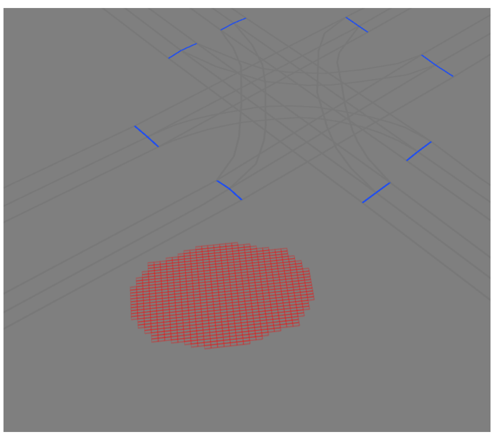
Figure 5-7: Lane restriction traffic rule using BDDs and LDM.

5.4 Communication Delay/ Loss

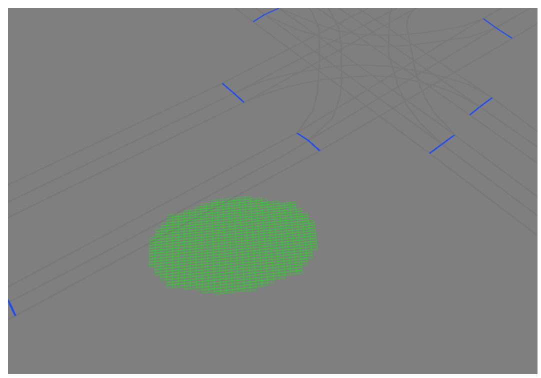
The main problems which could occur are the delay or loss of the packet. For a plan to handle this problem, one approach can be using 'Abstract data type'. 'Abstract data type' is the type that could be made for data to grow itself [79]. So, in case of packet loss/ delay, we could add a bigger Kamm circle to the existing representation of the BDD in the LDM to cover the uncertainty involved in the occupancy of the vehicle, which could be refreshed on the arrival of the new packet. Since, this could involve number of new experiments that could be studied as future work.

5.5 Spatial Operations using BDD

We encoded Geohash using BDD to support spatial operations using decision diagrams. Furthermore, as mentioned in the previous chapter, we used ROBDD AND operation for collision avoidance. The outcome of XOR and OR operation over ROBDD is shown below:

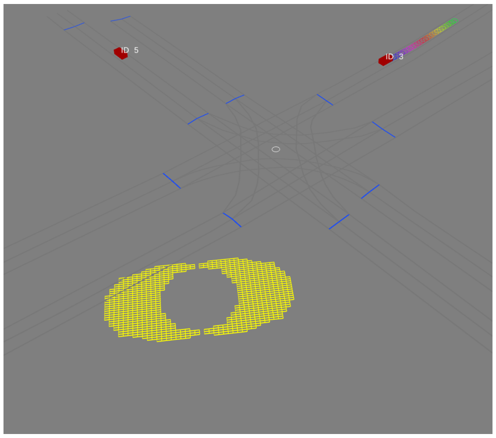


(a) Geohash set1.

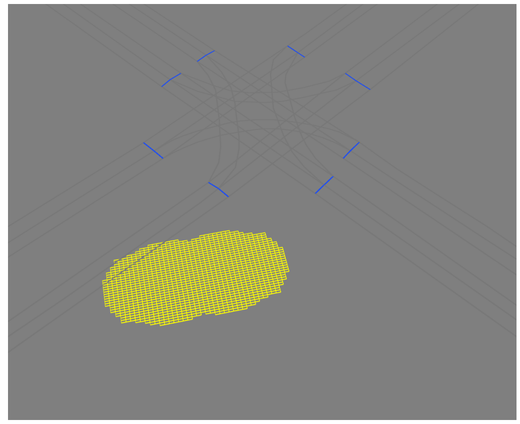


(b) Geohash set2.

Figure 5-8: $ROBDD_{red}$ and $ROBDD_{green}$ represents ROBDD for the red and green Geohash sets.



(a) $ROBDD_{red}\, xor\, ROBDD_{green}$.



(b) $ROBDD_{red}\, OR\, ROBDD_{green}$

Figure 5-9: Spatial effect of algebraic operations over ROBDDs.

Chapter 6

Summary

We hypothesized that integrating Geohash encoding of geographical positions of vehicles and the SBDD minimizes the computation time in verifying multiple vehicle positions. We successfully established the testable framework based on ROS and the Lanelet system. Results of computer experiments demonstrated the effectiveness of reducing the necessary computational time compared with other conventional conditions. This fact implies that Geohash or discrete representations in encoding vehicle conditions are effective if the SBDD can be applied. We applied the proposed method to the position and occupancy information in the present study's analysis. At the same time, it may have a new capability to include other vehicle information in further analysis.

We introduced and included a BDD representation for a set of Geohash representing reachable Kamm's circle for a given vehicle in the LDM. It supports algebraic operations between BDDs. Here, we used AND operation between two BDDs for collision avoidance. Finally, we converted BDD to JSON format to make it usable for most databases. Furthermore, Kamm's circle has been used previously in complex traffic scenarios for safe navigation. Therefore, introducing it in LDM may greatly benefit the safety aspect and including them as a decision diagram benefited us using algebraic properties available among the stored data. The current approach may benefit from storing the information that has supported algebra.

In the current approach, we concentrated on storing the data in the LDM. We hope the current work gives a new direction for the development of the LDM from a data point of view.

Appendices

Appendix A

Geohash list used for the calculation of the BDD variable orders in figure 5-5: GeohashList = ['wyhby3hxxu', 'wyhby3kc6f', 'wyhby3hxrt', 'wyhby3kc67', 'wyhby3hwhq', 'wyhby3kcew', 'wyhby3kf5b', 'wyhby3hxxr', 'wyhby3kceg', 'wyhby3hwv9', 'wyhby3hxqx', 'wyhby3hwt5', 'wyhby3hwkm', 'wyhby3hwjp', 'wyhby3hwhu', 'wyhby3hxpj', 'wyhby3kcg0', 'wyhby3kc5p', 'wyhby3hxq3', 'wyhby3hwh1', 'wyhby3hwkk', 'wyhby3kce9', 'wyhby3kcef', 'wyhby3kc6b', 'wyhby3hxx6', 'wyhby3kc74', 'wyhby3htut', 'wyhby3kb01', 'wyhby3hxnf', 'wyhby3hwtu', 'wyhby3hxz8', 'wyhby3hxr6', 'wyhby3hwky', 'wyhby3hwh2', 'wyhby3hw5g', 'wyhby3kc6g', 'wyhby3kcgt', 'wyhby3hwhk', 'wyhby3kb8c', 'wyhby3hwvz', 'wyhby3hxz9', 'wyhby3kb0q', 'wyhby3kcg6', 'wyhby3hwhh', 'wyhby3hxx0', 'wyhby3hxqe', 'wyhby3hxzb', 'wyhby3hxq1', 'wyhby3kceh', 'wyhby3hxnk', 'wyhby3hxrp', 'wyhby3hz8p', 'wyhby3kc4n', 'wyhby3kbfk', 'wyhby3kc70', 'wyhby3hxnu', 'wyhby3hxq7', 'wyhby3hxjg', 'wyhby3hxqw', 'wyhby3kcsh', 'wyhby3hxz0', 'wyhby3kc6u', 'wyhby3kfh1', 'wyhby3hzbp', 'wyhby3kc7s', 'wyhby3hwt2', 'wyhby3kb2t', 'wyhby3kc19', 'wyhby3hwtw', 'wyhby3hxrx', 'wyhby3hwjn', 'wyhby3kcu4', 'wyhby3hwvv', 'wyhby3kc66', 'wyhby3kc61', 'wyhby3kb0k', 'wyhby3hxnw', 'wyhby3kbcz', 'wyhby3htun', 'wyhby3kc76', 'wyhby3hz8n', 'wyhby3hwkg', 'wyhby3hwkc', 'wyhby3kb9x', 'wyhby3hxw8', 'wyhby3kb8u', 'wyhby3hwth', 'wyhby3hwv1', 'wyhby3hxqc', 'wyhby3hxqu', 'wyhby3kbcw', 'wyhby3kb1n', 'wyhby3kbcs', 'wyhby3hwht', 'wyhby3kf5c', 'wyhby3kcur', 'wyhby3kc5h', 'wyhby3hzb0', 'wyhby3kb35', 'wyhby3kb2s', 'wyhby3hwhv', 'wyhby3hxzx', 'wyhby3kb30', 'wyhby3hxn0', 'wyhby3kb0s', 'wyhby3kb8g', 'wyhby3hzb1', 'wyhby3hzbx', 'wyhby3hwvt', 'wyhby3hxz6', 'wyhby3hwv2', 'wyhby3kc6d', 'wyhby3kbc2', 'wyhby3hwtz', 'wyhby3hw5b', 'wyhby3kb0x', 'wyhby3kcdg', 'wyhby3hwkx', 'wyhby3kbf3', 'wyhby3hxrw', 'wyhby3hwvg', 'wyhby3kc4q', 'wyhby3kc1f', 'wyhby3hwkt', 'wyhby3hwtk', 'wyhby3hwyh', 'wyhby3hwt9', 'wyhby3kc4m', 'wyhby3hwy6', 'wyhby3kcsn', 'wyhby3kc4w', 'wyhby3hxxx', 'wyhby3kcd9', 'wyhby3hxnz', 'wyhby3htgv', 'wyhby3kc18', 'wyhby3hzbw', 'wyhby3kb0h', 'wyhby3kf58', 'wyhby3hwh5', 'wyhby3kbc9', 'wyhby3hxrr', 'wyhby3htgx', 'wyhby3hwk2', 'wyhby3hwhy', 'wyhby3kb2m', 'wyhby3hwy1', 'wyhby3kce4', 'wyhby3hxnq', 'wyhby3hwhx', 'wyhby3htur', 'wyhby3hxny', 'wyhby3hwhc', 'wyhby3htuh', 'wyhby3kcev', 'wyhby3hxq8', 'wyhby3hxnc', 'wyhby3kbc3', 'wyhby3hxwc', 'wyhby3kbfn', 'wyhby3kc6t', 'wyhby3htum', 'wyhby3hwhd', 'wyhby3hwvw', 'wyhby3hzb6', 'wyhby3kcu1', 'wyhby3kb0e', 'wyhby3kb99', 'wyhby3kce0', 'wyhby3hxze', 'wyhby3htuk', 'wyhby3kbfq', 'wyhby3kce7', 'wyhby3kb08', 'wyhby3hwhz', 'wyhby3hwvy', 'wyhby3kcu2', 'wyhby3kb88', 'wyhby3kb90', 'wyhby3hwsc', 'wyhby3hxqb', 'wyhby3hwkw', 'wyhby3kc4j', 'wyhby3kc41', 'wyhby3kb9k', 'wyhby3hzb4', 'wyhby3kb9d', 'wyhby3kb04', 'wyhby3kb9f', 'wyhby3kb0d', 'wyhby3kc47', 'wyhby3hwm0', 'wyhby3kb8v', 'wyhby3kcet', 'wyhby3kb0c', 'wyhby3kceb', 'wyhby3kcg8', 'wyhby3kb29', 'wyhby3kcg2', 'wyhby3hxxq', 'wyhby3hxzd', 'wyhby3kcez', 'wyhby3hw5f', 'wyhby3kc63', 'wyhby3kc4f', 'wyhby3kcg1', 'wyhby3kcu3', 'wyhby3hwtc', 'wyhby3kcgw', 'wyhby3hxxy', 'wyhby3kc6m', 'wyhby3kcej', 'wyhby3kbf7', 'wyhby3hwtf', 'wyhby3kb36', 'wyhby3hxxt', 'wyhby3hxxg', 'wyhby3hxnx', 'wyhby3hxqz', 'wyhby3hwm7', 'wyhby3kcgz', 'wyhby3htux', 'wyhby3hwmq', 'wyhby3kbck', 'wyhby3kb2y', 'wyhby3kb2v', 'wyhby3kc71', 'wyhby3hxxk', 'wyhby3kb92', 'wyhby3hws8', 'wyhby3hwmk', 'wyhby3kb95', 'wyhby3kb2f', 'wyhby3kb9h', 'wyhby3kc6k', 'wyhby3kb0u', 'wyhby3hxzs', 'wyhby3kb0t', 'wyhby3kbf6', 'wyhby3kbfm', 'wyhby3hwhp', 'wyhby3hxnm', 'wyhby3kbcg', 'wyhby3hwyn', 'wyhby3hwvb', 'wyhby3kbc8', 'wyhby3kcgv', 'wyhby3hwy5', 'wyhby3kce6', 'wyhby3hwvs', 'wyhby3kces', 'wyhby3kb2u', 'wyhby3kcum', 'wyhby3kb03', 'wyhby3hxzv', 'wyhby3hwmn', 'wyhby3kb33', 'wyhby3hxn6', 'wyhby3hxrk', 'wyhby3kcun', 'wyhby3hwhg', 'wyhby3hxz3', 'wyhby3hxqs', 'wyhby3hwm6', 'wyhby3hwyk', 'wyhby3kb00', 'wyhby3kc46', 'wyhby3kcge', 'wyhby3hxpp', 'wyhby3hwt4', 'wyhby3kc6z', 'wyhby3kc4e', 'wyhby3kc4x', 'wyhby3kc73', 'wyhby3hw5c', 'wyhby3hzbj', 'wyhby3kb8f', 'wyhby3hxn4', 'wyhby3kb94', 'wyhby3hxrh', 'wyhby3kbfj', 'wyhby3hxj8', 'wyhby3kc43', 'wyhby3hwmx', 'wyhby3hwwn', 'wyhby3hxzw', 'wyhby3kb89', 'wyhby3hw58', 'wyhby3kb9v', 'wyhby3hwv0', 'wyhby3k8pv', 'wyhby3kbfh', 'wyhby3kb2c', 'wyhby3hxqy', 'wyhby3kcgc', 'wyhby3kbdp', 'wyhby3hwhm', 'wyhby3hxr5', 'wyhby3kb9r', 'wyhby3kb3x', 'wyhby3kb2w', 'wyhby3hxzg', 'wyhby3kcgu', 'wyhby3hwy0', 'wyhby3kc6w', 'wyhby3hwtp', 'wyhby3kb2g', 'wyhby3kb0f', 'wyhby3kc4h', 'wyhby3kb0n', 'wyhby3kcgg', 'wyhby3kc4r', 'wyhby3k8pc', 'wyhby3kb9w', 'wyhby3kcuk', 'wyhby3kceu', 'wyhby3kbf2', 'wyhby3kb02', 'wyhby3kcem', 'wyhby3hzbm', 'wyhby3hxr1', 'wyhby3kcu7', 'wyhby3hxnv', 'wyhby3hwk9', 'wyhby3kc6s', 'wyhby3htuj', 'wyhby3kc1c', 'wyhby3hxwg', 'wyhby3kcg9', 'wyhby3hxqv', 'wyhby3hxzc', 'wyhby3hxj9', 'wyhby3kcg7', 'wyhby3kcuh', 'wyhby3hxw9', 'wyhby3hwhj', 'wyhby3hwwp', 'wyhby3hxxm', 'wyhby3hxq9', 'wyhby3hwvm', 'wyhby3hxne', 'wyhby3hwsv', 'wyhby3htgy', 'wyhby3kbf0', 'wyhby3hwtg', 'wyhby3hxrn', 'wyhby3kbfw', 'wyhby3kbf1', 'wyhby3kb28', 'wyhby3hwmr', 'wyhby3kf5f', 'wyhby3kc69', 'wyhby3kcuq', 'wyhby3kc6y', 'wyhby3hxn8', 'wyhby3hwtd', 'wyhby3kb0j', 'wyhby3hwv3', 'wyhby3hxjb', 'wyhby3kc42', 'wyhby3hxqf', 'wyhby3kb98', 'wyhby3hxn7', 'wyhby3hzbn', 'wyhby3hwy4', 'wyhby3hxnh', 'wyhby3kce2', 'wyhby3kc77', 'wyhby3hwyw', 'wyhby3hwvu', 'wyhby3hwv8', 'wyhby3kb1j', 'wyhby3kb2d', 'wyhby3kb37', 'wyhby3hxq6', 'wyhby3hxn2', 'wyhby3kc4b', 'wyhby3kc4c', 'wyhby3hw5u', 'wyhby3kb1p', 'wyhby3kcu5', 'wyhby3hwvc', 'wyhby3kcup', 'wyhby3kc4z', 'wyhby3hxns', 'wyhby3kb3n', 'wyhby3kc49', 'wyhby3hzbq', 'wyhby3hwh7', 'wyhby3kbcf', 'wyhby3kc1u', 'wyhby3hwym', 'wyhby3hwtj', 'wyhby3hwm1', 'wyhby3hwjj', 'wyhby3kcex', 'wyhby3hwtq', 'wyhby3kc7q', 'wyhby3hwsb', 'wyhby3hwk0', 'wyhby3hwku', 'wyhby3hwkb', 'wyhby3hxx3', 'wyhby3hxx2', 'wyhby3kc6v', 'wyhby3hwsu', 'wyhby3hzb3', 'wyhby3kbc6', 'wyhby3kcen', 'wyhby3hwwj', 'wyhby3kb96', 'wyhby3kcep', 'wyhby3hxnn', 'wyhby3hz8j', 'wyhby3kc75', 'wyhby3hzb5', 'wyhby3hwy3', 'wyhby3kc4u', 'wyhby3hwh6', 'wyhby3kcer', 'wyhby3kcus', 'wyhby3kb2e', 'wyhby3kb0z', 'wyhby3kcu6', 'wyhby3hwvx', 'wyhby3hxng', 'wyhby3kb9p', 'wyhby3kb2x', 'wyhby3hxz7', 'wyhby3hwv7', 'wyhby3hws9', 'wyhby3hxxc', 'wyhby3kbfr', 'wyhby3hzbr', 'wyhby3kc4p', 'wyhby3hwhs', 'wyhby3k8p8', 'wyhby3hxx5', 'wyhby3hwyx', 'wyhby3hzbh', 'wyhby3hxzy', 'wyhby3kc7n', 'wyhby3hwhn', 'wyhby3kc7k', 'wyhby3k8p9', 'wyhby3hxr4', 'wyhby3kb97', 'wyhby3kc7h', 'wyhby3kb3t', 'wyhby3hxnr', 'wyhby3hwtv', 'wyhby3kb34', 'wyhby3kbft', 'wyhby3hwvd', 'wyhby3kb06', 'wyhby3kce1', 'wyhby3kcgf', 'wyhby3hxxs', 'wyhby3hxzu', 'wyhby3kb91', 'wyhby3kc62', 'wyhby3hxn1', 'wyhby3kbcv', 'wyhby3hwt0', 'wyhby3kc7x', 'wyhby3kbct', 'wyhby3kcsj', 'wyhby3kcd8', 'wyhby3kc44', 'wyhby3kb9c', 'wyhby3hxzm', 'wyhby3hzbk', 'wyhby3kb2k', 'wyhby3hxzf', 'wyhby3kc68', 'wyhby3hxz2', 'wyhby3kced', 'wyhby3kb9q', 'wyhby3hxxj', 'wyhby3hwm5', 'wyhby3kcuj', 'wyhby3hwmt', 'wyhby3kcgy', 'wyhby3kcdf', 'wyhby3kc45', 'wyhby3hxzt', 'wyhby3hxn3', 'wyhby3hwk8', 'wyhby3kc4k', 'wyhby3hxn5', 'wyhby3hwmw', 'wyhby3hwke', 'wyhby3htgz', 'wyhby3kcgk', 'wyhby3hwmp', 'wyhby3hxjc', 'wyhby3kb9m', 'wyhby3hwtn', 'wyhby3hwmj', 'wyhby3hwt7', 'wyhby3hxxf', 'wyhby3kce5', 'wyhby3kc72', 'wyhby3kcdb', 'wyhby3kb0g', 'wyhby3hwyq', 'wyhby3kbdj', 'wyhby3kbfs', 'wyhby3hxq0', 'wyhby3kb21', 'wyhby3kb09', 'wyhby3kbcy', 'wyhby3htup', 'wyhby3kc6x', 'wyhby3hwtt', 'wyhby3kfh0', 'wyhby3kb31', 'wyhby3kc4d', 'wyhby3kb0w', 'wyhby3hxx8', 'wyhby3kc1g', 'wyhby3kbf5', 'wyhby3hzbt', 'wyhby3hxz1', 'wyhby3kb9j', 'wyhby3kb0r', 'wyhby3hwsg', 'wyhby3kb27', 'wyhby3kb3w', 'wyhby3hwv6', 'wyhby3hxrj', 'wyhby3kcec', 'wyhby3kb22', 'wyhby3kb3r', 'wyhby3hxxp', 'wyhby3hxrq', 'wyhby3hxwv', 'wyhby3kb0m', 'wyhby3kb07', 'wyhby3kb23', 'wyhby3kbdh', 'wyhby3kb05', 'wyhby3kb3h', 'wyhby3kcdv', 'wyhby3hxxd', 'wyhby3hwm4', 'wyhby3hwkf', 'wyhby3kb0y', 'wyhby3kcgb', 'wyhby3kbfp', 'wyhby3hxnp', 'wyhby3hxzz', 'wyhby3kb9y', 'wyhby3hwte', 'wyhby3hxr3', 'wyhby3hxjv', 'wyhby3kcsp', 'wyhby3hxqg', 'wyhby3hxr0', 'wyhby3kbc7', 'wyhby3hxqk', 'wyhby3kbfx', 'wyhby3hxrm', 'wyhby3kb9t', 'wyhby3hwm3', 'wyhby3kcgs', 'wyhby3hxjf', 'wyhby3hwyj', 'wyhby3kb0p', 'wyhby3kc1b', 'wyhby3hwve', 'wyhby3kcgx', 'wyhby3kc4s', 'wyhby3kbcb', 'wyhby3htuq', 'wyhby3kb2z', 'wyhby3hwk7', 'wyhby3kbdn', 'wyhby3kc5j', 'wyhby3kc7m', 'wyhby3hxn9', 'wyhby3hwsf', 'wyhby3hwks', 'wyhby3kc7t', 'wyhby3kb2b', 'wyhby3kcgm', 'wyhby3hxx9', 'wyhby3kc4v', 'wyhby3kb3p', 'wyhby3kb9g', 'wyhby3kbc0', 'wyhby3k8pg', 'wyhby3kc7w', 'wyhby3hxnd', 'wyhby3hwhe', 'wyhby3hwhw', 'wyhby3kc7r', 'wyhby3kb9u', 'wyhby3kcdu', 'wyhby3kbce', 'wyhby3k8pu', 'wyhby3kbcu', 'wyhby3hwk6', 'wyhby3kcee', 'wyhby3kcg3', 'wyhby3hwyp', 'wyhby3hwtm', 'wyhby3hxnj', 'wyhby3hxpn', 'wyhby3hwh9', 'wyhby3kb9e', 'wyhby3hwt8', 'wyhby3hxx1', 'wyhby3hxxw', 'wyhby3hxju', 'wyhby3hwvf', 'wyhby3kf59', 'wyhby3kc4y', 'wyhby3kb3k', 'wyhby3hzb7', 'wyhby3kcgd', 'wyhby3hwmm', 'wyhby3hwh3', 'wyhby3kbcd', 'wyhby3hwt1', 'wyhby3kbcx', 'wyhby3hwhf', 'wyhby3hxnt', 'wyhby3hwty', 'wyhby3kb0v', 'wyhby3hxq2', 'wyhby3kc6e', 'wyhby3htuw', 'wyhby3hwvk', 'wyhby3hwhr', 'wyhby3hwk3', 'wyhby3kc5n', 'wyhby3kb9n', 'wyhby3hwmh', 'wyhby3kb8b', 'wyhby3kcek', 'wyhby3kb3m', 'wyhby3kc4t', 'wyhby3kcey', 'wyhby3hwts', 'wyhby3kce8', 'wyhby3hwh0', 'wyhby3htu7', 'wyhby3hxqm', 'wyhby3hxr7', 'wyhby3hwh8', 'wyhby3hxxn', 'wyhby3hxqd', 'wyhby3hwyt', 'wyhby3kcu0', 'wyhby3hxwf', 'wyhby3hxxv', 'wyhby3kc4g', 'wyhby3hxxe', 'wyhby3hwyr', 'wyhby3kbcm', 'wyhby3kce3', 'wyhby3hwkv', 'wyhby3kb26', 'wyhby3kc7j', 'wyhby3k8pb', 'wyhby3k8pf', 'wyhby3hwkz', 'wyhby3hwt6', 'wyhby3kc60', 'wyhby3hwt3', 'wyhby3hxxh', 'wyhby3kb20', 'wyhby3hxx7', 'wyhby3hwh4', 'wyhby3kc6c', 'wyhby3hwy7', 'wyhby3kbc1', 'wyhby3hxxz', 'wyhby3kc1v', 'wyhby3hwtx', 'wyhby3kbf4', 'wyhby3htgw', 'wyhby3kb9z', 'wyhby3kc7p', 'wyhby3hxqt', 'wyhby3kc40', 'wyhby3hxwu', 'wyhby3kb93', 'wyhby3hwkd', 'wyhby3kc48', 'wyhby3hwtr', 'wyhby3kcdc', 'wyhby3kceq', 'wyhby3hxzk', 'wyhby3kbcc', 'wyhby3kb3q', 'wyhby3kb9s', 'wyhby3hxx4', 'wyhby3hxwb', 'wyhby3kb3j']

Bibliography

[1] Shin-ichi Minato, Nagisa Ishiura, and Shuzo Yajima. Shared binary decision diagram with attributed edges for efficient boolean function manipulation. In *27th ACM/IEEE Design Automation Conference*, pages 52–57. IEEE, 1990.

[2] ETSI TR 102 863 (V1.1.1): Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM), 2011.

[3] Cabinet Office. Society 5.0.

[4] Xun Xu, Yuqian Lu, Birgit Vogel-Heuser, and Lihui Wang. Industry 4.0 and industry 5.0-inception, conception and perception. *Journal of Manufacturing Systems*, 61:530–535, 10 2021.

[5] Jorge Vargas, Suleiman Alsweiss, Onur Toker, Rahul Razdan, and Joshua Santos. An overview of autonomous vehicles sensors and their vulnerability to weather conditions. *Sensors*, 21(16):5397, 2021.

[6] Lidar vs. camera: driving in the rain. `https://ouster.com/blog/lidar-vs-camera-comparison-in-the-rain/`. Accessed: 2022-08-24.

[7] SAE International, Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles, J3016_202104. `https://www.sae.org/standards/content/j3016_202104`. Accessed: 2022-08-25.

[8] Administrator of the CAR 2 CAR Communication Consortium,CAR 2 CAR Communication Consortium, Administrator of the CAR 2 CAR Communi-

cation Consortium,CAR 2 CAR Communication Consortium. https://www.car-2-car.org/about-c-its/. Accessed: 2021-11-20.

[9] Hiroshi Makano. C-its and connected automated driving in japan. *Routes/Roads*, (373), 2017.

[10] Cooperative intelligent transport systems (C-ITS) Guidelines on the usage of standards. https://www.itsstandards.eu/app/uploads/sites/14/2020/10/C-ITS-Brochure-2020-FINAL.pdf, June 2020. Accessed: 2022-04-19.

[11] Etsi En. 302 665 v1. 1.1: Intelligent transport systems (its), communication architecture. *ETSI EN 302 665 V1*, 1, 2010.

[12] BD Netten, LJHM Kester, Harry Wedemeijer, Igor Passchier, and Bart Driessen. *Dynamap: A dynamic map for road side its stations*. Washington, DC: Intelligent Transportation Society of America, 2013.

[13] Hideki Shimada, Akihiro Yamaguchi, Hiroaki Takada, Kenya Sato, et al. Implementation and evaluation of local dynamic map in safety driving systems. *Journal of Transportation Technologies*, 5(02):102, 2015.

[14] Julian Eggert, D Aguirre Salazar, Tim Puphal, and Benedict Flade. Driving situation analysis with relational local dynamic maps (r-ldm). In *Proc. Symp. Future Active Safety Technology*, 2017.

[15] Thomas Eiter, Herbert Füreder, Fritz Kasslatter, Josiane Xavier Parreira, and Patrik Schneider. Towards a semantically enriched local dynamic map. *International Journal of Intelligent Transportation Systems Research*, 17(1):32–48, 2019.

[16] Giulio Vivo. The safespot integrated project: an overview. In *2007 IEEE Intelligent Vehicles Symposium*, pages 14–14. IEEE, 2007.

[17] L Andreone, R Brignolo, S Damiani, F Sommariva, G Vivo, and S Marco. Safespot final report. *Technical Report D8. 1.1*, 2010.

[18] Christian Zott, Sheung Ying Yuen, Christopher L Brown, Christine Bartels, Zoltan Papp, and BD Netten. Safespot local dynamic maps: context-dependent view generation of a platform's state & environment. 2008.

[19] Javier Ibanez-Guzman, Stephanie Lefevre, Abdelkader Mokkadem, and Sylvain Rodhaim. Vehicle to vehicle communications applied to road intersection safety, field results. In *13th International IEEE Conference on Intelligent Transportation Systems*, pages 192–197. IEEE, 2010.

[20] Tobias Schendzielorz, Jaap Vreeswijk, and Paul Mathias. Intelligent cooperative intersection safety implementation, test and evaluation. In *16th ITS World Congress and Exhibition on Intelligent Transport Systems and ServicesITS AmericaERTICOITS Japan*, 2009.

[21] Bart Netten and Harry Wedemeijer. Testing cooperative systems with the mars simulator. In *13th International IEEE Conference on Intelligent Transportation Systems*, pages 186–191. IEEE, 2010.

[22] Mikel García, Itziar Urbieta, Marcos Nieto, Javier González de Mendibil, and Oihana Otaegui. ildm: An interoperable graph-based local dynamic map. *Vehicles*, 4(1):42–59, 2022.

[23] ETSI EN 302 895 (V1.1.0): Intelligent transport systems - Extension of map database specifications for Local Dynamic Map for applications of Cooperative ITS, 2014.

[24] ISO/TS 17931:2013: Intelligent transport systems - Extension of map database specifications for Local Dynamic Map for applications of Cooperative ITS, 2013.

[25] ISO/TS 18750:2015: Intelligent transport systems - Cooperative systems - Definition of a global concept for Local Dynamic Maps, 2015.

[26] Eric Koenders, Dick Oort, and Klaas Rozema. An open local dynamic map. In *Procedings of ITS European Congress*, 2014.

[27] Nicole El Zoghby, Véronique Cherfaoui, and Thierry Denoeux. Evidential distributed dynamic map for cooperative perception in vanets. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 1421–1426. IEEE, 2014.

[28] Julius Ziegler, Philipp Bender, Markus Schreiber, Henning Lategahn, Tobias Strauss, Christoph Stiller, Thao Dang, Uwe Franke, Nils Appenrodt, Christoph G Keller, et al. Making bertha drivean autonomous journey on a historic route. *IEEE Intelligent transportation systems magazine*, 6(2):8–20, 2014.

[29] Ömer Şahin Taş, Niels Ole Salscheider, Fabian Poggenhans, Sascha Wirges, Claudio Bandera, Marc René Zofka, Tobias Strauss, J Marius Zöllner, and Christoph Stiller. Making bertha cooperate–team annieways entry to the 2016 grand cooperative driving challenge. *IEEE Transactions on Intelligent Transportation Systems*, 19(4):1262–1276, 2017.

[30] Michael Aeberhard, Sebastian Rauch, Mohammad Bahram, Georg Tanzmeister, Julian Thomas, Yves Pilat, Florian Homm, Werner Huber, and Nico Kaempchen. Experience, results and lessons learned from automated driving on germany's highways. *IEEE Intelligent transportation systems magazine*, 7(1):42–57, 2015.

[31] OpenDrive.. `https://www.opendrive.com/`. Accessed: 2022-05-08.

[32] Philipp Bender, Julius Ziegler, and Christoph Stiller. Lanelets: Efficient map representation for autonomous driving. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 420–425. IEEE, 2014.

[33] Fabian Poggenhans, Jan-Hendrik Pauls, Johannes Janosovits, Stefan Orf, Maximilian Naumann, Florian Kuhnt, and Matthias Mayr. Lanelet2: A high-definition map framework for the future of automated driving. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1672–1679. IEEE, 2018.

[34] GM Morton. A computer oriented geodetic data base and a new technique in file sequencing tech. rep. *IBM Ltd., Ottawa, Ontario, Canada*, 1966.

[35] SH Song, CQ Cheng, GL Pu, FG An, and X Luo. Global remote sensing data subdivision organization based on geosot. *Acta Geod Cartographica Sin*, 43(8):869–876, 2014.

[36] Z Sun and C Cheng. true 3d data expression based on geosot-3d ellipsoid subdivision. *Geomatics world*, 23(3):40–46, 2016.

[37] Alexander Barth and Uwe Franke. Where will the oncoming vehicle be the next second? In *2008 IEEE Intelligent Vehicles Symposium*, pages 1068–1073. IEEE, 2008.

[38] Andreas Eidehall. Multi-target threat assessment for automotive applications. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 433–438. IEEE, 2011.

[39] Simon Danielsson, Lars Petersson, and Andreas Eidehall. Monte carlo based threat assessment: Analysis and improvements. In *2007 IEEE Intelligent Vehicles Symposium*, pages 233–238. IEEE, 2007.

[40] Adrian Broadhurst, Simon Baker, and Takeo Kanade. Monte carlo road safety reasoning. In *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.*, pages 319–324. IEEE, 2005.

[41] Taewung Kim and Hyun-Yong Jeong. A novel algorithm for crash detection under general road scenes using crash probabilities and an interactive multiple model particle filter. *IEEE Transactions on Intelligent Transportation Systems*, 15(6):2480–2490, 2014.

[42] Alain Lambert, Dominique Gruyer, Guillaume Saint Pierre, and Alexandre Ndjeng Ndjeng. Collision probability assessment for speed control. In *2008 11th International IEEE Conference on Intelligent Transportation Systems*, pages 1043–1048. IEEE, 2008.

[43] Matthias Althoff, Olaf Stursberg, and Martin Buss. Model-based probabilistic collision detection in autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 10(2):299–310, 2009.

[44] Matthias Althoff and Alexander Mergel. Comparison of markov chain abstraction and monte carlo simulation for the safety assessment of autonomous cars. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1237–1247, 2011.

[45] Matthias Althoff and John M Dolan. Online verification of automated road vehicles using reachability analysis. *IEEE Transactions on Robotics*, 30(4):903–918, 2014.

[46] Matthias Althoff and Silvia Magdici. Set-based prediction of traffic participants on arbitrary road networks. *IEEE Transactions on Intelligent Vehicles*, 1(2):187–202, 2016.

[47] SB Ackers. Binary decision diagrams. *IEEE Transactions on Computers*, 100(6):509–516, 1978.

[48] Shin-ichi Minato. Zero-suppressed bdds for set manipulation in combinatorial problems. In *Proceedings of the 30th International Design Automation Conference*, pages 272–277, 1993.

[49] Masahiro Fujita, Patrick C. McGeer, and JC-Y Yang. Multi-terminal binary decision diagrams: An efficient data structure for matrix representation. *Formal methods in system design*, 10(2):149–169, 1997.

[50] Mordechai Haklay and Patrick Weber. Openstreetmap: User-generated street maps. *IEEE Pervasive computing*, 7(4):12–18, 2008.

[51] Mercedes-Benz deploys HERE HD Live Map for DRIVE PILOT system.. `https://www.globenewswire.com/news-release/2021/09/06/2291764/0/en/Mercedes-Benz-deploys-HERE-HD-Live-Map-for-DRIVE-PILOT-system.html`. Accessed: 2022-05-08.

[52] HERE powers BMW Maps with Predictive Routing.. https://www.globenewswire.com/en/news-release/2022/04/25/2427778/0/en/HERE-powers-BMW-Maps-with-Predictive-Routing.html. Accessed: 2022-05-08.

[53] BMW Motorrad and TomTom.. https://www.tomtom.com/customers/bmw-motorrad/. Accessed: 2022-05-08.

[54] Microsoft Connected Vehicle Platform and TomTom.. https://www.tomtom.com/customers/microsoft-connected-vehicle-platform/. Accessed: 2022-05-08.

[55] JOSM.. https://josm.openstreetmap.de/. Accessed: 2022-05-08.

[56] Elwan Héry, Stefano Masi, Philippe Xu, and Philippe Bonnifait. Map-based curvilinear coordinates for autonomous vehicles. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–7. IEEE, 2017.

[57] DEVELOPMENT OF DYNAMIC MAP FOR AUTOMATED DRIVING AND ITS POTENTIAL TO BE NEXT-GEN INDUSTRIAL INFRASTRUCTURE . https://www.mitsui.com/mgssi/en/report/detail/__icsFiles/afieldfile/2018/09/05/1807t_wariishikinjo_e.pdf. Accessed: 2022-05-09.

[58] SCENARIO DESCRIPTION AND KNOWLEDGE-BASED SCENARIO GENERATION . https://www.pegasusprojekt.de/files/tmpl/Pegasus-Abschlussveranstaltung/05_Scenario_Description_and_Knowledge-Based_Scenario_Generation.pdf. Accessed: 2022-05-09.

[59] PEGASUS RESEARCH PROJECT . https://www.pegasusprojekt.de/en/. Accessed: 2022-05-09.

[60] Mesh: GIS data beyond raster and vector. https://av.tib.eu/media/43361. Accessed: 2022-05-10.

[61] 3D and mesh forms of spatial data. https://mdsumner.github.io/geo-comp-graphics-oghub/d-and-mesh-forms-of-spatial-data.html. Accessed: 2022-05-10.

[62] Ning Guo, Wei Xiong, Ye Wu, Luo Chen, and Ning Jing. A geographic meshing and coding method based on adaptive hilbert-geohash. *IEEE Access*, 7:39815–39825, 2019.

[63] Christopher C Tanner, Christopher J Migdal, and Michael T Jones. The clipmap: a virtual mipmap. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 151–158, 1998.

[64] JieQing Yu, LiXin Wu, GuoJie Zi, and ZengZhang Guo. Sdog-based multi-scale 3d modeling and visualization on global lithosphere. *Science China Earth Sciences*, 55(6):1012–1020, 2012.

[65] Tibor Vukovic. Hilbert-geohash-hashing geographical point data using the hilbert space-filling curve. Master's thesis, NTNU, 2016.

[66] Matthias Althoff. *Reachability analysis and its application to the safety assessment of autonomous cars*. PhD thesis, Technische Universität München, 2010.

[67] Reachability. https://ocw.mit.edu/courses/16-412j-cognitive-robotics-spring-2016/c71b2cf371b216faa50aca186b305a9f_MIT16_412JS16_L18.pdf. Accessed: 2022-05-13.

[68] CommonRoad: Vehicle Models. https://gitlab.lrz.de/tum-cps/commonroad-vehicle-models/-/blob/master/vehicleModels_commonRoad.pdf. Accessed: 2022-07-17.

[69] Piotr F Orzechowski, Annika Meyer, and Martin Lauer. Tackling occlusions & limited sensor range with set-based safety verification. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1729–1736. IEEE, 2018.

[70] Klaus Havelund and Doron Peled. Bdds for representing data in runtime verification. In *International Conference on Runtime Verification*, pages 107–128. Springer, 2020.

[71] Matthias Althoff, Markus Koschi, and Stefanie Manzinger. Commonroad: Composable benchmarks for motion planning on roads. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 719–726. IEEE, 2017.

[72] Munehiro Matsuura, Tsutomu Sasao, Jon T Butler, and Yukihiro Iguchi. Bi-partition of shared binary decision diagrams. *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, 85(12):2693–2700, 2002.

[73] Klaus Havelund, Doron Peled, and Dogan Ulus. First-order temporal logic monitoring with bdds. *Formal Methods in System Design*, 56(1):1–21, 2020.

[74] Matthias Althoff, Stefan Urban, and Markus Koschi. Automatic conversion of road networks from opendrive to lanelets. In *2018 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, pages 157–162. IEEE, 2018.

[75] Maximilian Naumann, Fabian Poggenhans, Martin Lauer, and Christoph Stiller. Coincar-sim: An open-source simulation framework for cooperatively interacting automobiles. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–6. IEEE, 2018.

[76] Donald Ervin Knuth. The art of computer programming: Bitwise tricks & techniques. *Binary Decision Diagrams*, 4, 2009.

[77] Randal E Bryant. Graph-based algorithms for boolean function manipulation. *Computers, IEEE Transactions on*, 100(8):677–691, 1986.

[78] Watis Leelapatra et al. *Geometric transformations of binary decision diagram encoded images*. PhD thesis, Asian Institute of Technology, 2007.

[79] Esteban Zimányi, Mahmoud Sakr, and Arthur Lesuisse. Mobilitydb: A mobility database based on postgresql and postgis. *ACM Transactions on Database Systems (TODS)*, 45(4):1–42, 2020.

Research achievements

6.1 Conferences

- A. Kumar and H. Wagatsuma. An Implementation of Linear Temporal Logic for Driving Safety Suitable for the Concept of Local Dynamic Map. 2021 IEEE/SICE International Symposium on System Integration (SII) (2021): 708-709.
- A. Kumar and H. Wagatsuma. A Graph-Based Safe Region Prediction Method in the Lanelet Framework Toward Automated Driving System Assessments. SAES 2020, Kyutech, JAPAN

6.2 Journals

- A. Kumar and H. Wagatsuma. A binary decision diagram based approach for refining road safety scenarios in the local dynamic map, ICIC-ELB, Vol.13, No.6, 597-605, June 2022.
- A. Kumar and H. Wagatsuma. A Kamms Circle Based Potential Risk Estimation Scheme in the Local Dynamic Map Computation Enhanced by Binary Decision Diagrams. Sensors. 2022.