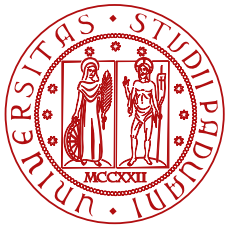

Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Department of Information Engineering
Master Thesis in ICT for Interet and Multimedia

Study and Analysis of Clustering-Based Algorithms for Cooperative Perception in Vehicular Networks

Master Candidate

Alberto Merotto

Student ID 2081117

Supervisor

Prof. Marco Giordani

University of Padova

Co-supervisor

Ch. Prof. Michele Zorzi

University of Padova

Co-supervisor

Dr. Filippo Bragato

University of Padova

Academic Year
2023/2024
Graduation Date 08/10/2024

Alla mia mamma, che mi guarda da lassù.
A mio papà e a mio fratello, che ogni giorno mi sostengono.

Abstract

The increasing capability of 3GPP NR V2X standard for next-generation vehicular systems, will support Vehicle-to-Vehicle (V2V) communication in the Millimeter Waves (mmWaves) spectrum to address the communication requirements of future intelligent automotive networks. This new connectivity will enable the evolution towards Cooperative and Intelligent Transportation Systems (C-ITSs) with the aim of delivering improved traffic safety and efficiency. This concept is crucial because vehicles, even with advanced sensor systems, may not perceive every detail of their surroundings. By establishing connectivity, vehicles can enable a collaborative approach for object perception, where they can collectively enhance their awareness of the environment.
In V2V, data sharing puts a strain on traditional vehicular communication technologies due to high demands in data rate, reliability, and latency. Researchers are exploring new radio systems, like at mmWaves, to address these challenges.
However, the propagation at these frequencies raises many concerns. So, simply increasing channel capacity may not meet the demanding Quality of Service (QoS) requirements of future automotive applications, especially in scenarios with challenging automation levels. Therefore, it is crucial to limit the amount of data broadcast over bandwidth-limited channels.
In this thesis, clustering-based algorithms are studied and analyzed to demonstrate how the burden on the network can be reduced exploiting NR V2V connectivity: first the vehicles are grouped into clusters and exchange information with the master of their cluster using a sidelink connectivity, then the master will communicate with the Base Station (BS), thus reducing the number of vehicles that are exchanging information and enabling the masters to select the information to be shared in a smart way.
Through simulations, we demonstrate how the processing delay required by the approaches that select information to transmit in a smart way may be too high to meet the strict requirements of critical environment such as vehicular networks; hence, a random approach where what to transmit is selected in a random way by the masters of the clusters, performs better. Furthermore, we evaluate the potential of the proposed cluster-based dissemination algorithms as a function of several parameters, including the channel condition and the number of vehicles.

Sommario

La crescente capacità di 3GPP NR V2X supporterà le operazioni veicolo-veicolo (V2V) nello spettro delle onde millimetriche (mmWave) per affrontare le esigenze di comunicazione delle future reti automobilistiche intelligenti. Questa nuova connettività consentirà la creazione di Sistemi di Trasporto Intelligenti Connessi (C-ITS) con l'obiettivo di migliorare la sicurezza e l'efficienza del traffico. Questo concetto è cruciale perché i veicoli, anche con sistemi avanzati di sensori, potrebbero non percepire ogni dettaglio del loro ambiente circostante. Stabilendo la connettività, i veicoli possono adottare un approccio collaborativo per la percezione degli oggetti, dove possono migliorare collettivamente la loro consapevolezza dell'ambiente. Nel V2V, la condivisione dei dati mette a dura prova le tecnologie tradizionali di comunicazione veicolare a causa delle elevate richieste di velocità di trasmissione, affidabilità e latenza. I ricercatori stanno esplorando nuovi sistemi radio, come le mmWave, per affrontare queste sfide. Tuttavia, i problemi di propagazione alle frequenze superiori a 6 GHz pongono ostacoli. Pertanto, aumentare semplicemente la capacità del canale potrebbe non soddisfare le esigenze di Qualità del Servizio (QoS) delle future applicazioni automobilistiche, soprattutto in scenari con diversi livelli di automazione. È quindi cruciale limitare la quantità di dati trasmessi su canali a larghezza di banda limitata.

In questa tesi algoritmi basati su clustering sono studiati e analizzati per capire come ridurre il carico sulla rete sfruttando la connettività NR V2X: innanzitutto, i veicoli vengono raggruppati in cluster e scambiano informazioni con il capo del loro cluster utilizzando un collegamento laterale, quindi il capo comunicherà con la stazione base, riducendo così il numero di veicoli che scambiano informazioni e consentendo ai capi di selezionare in modo intelligente le informazioni da condividere.

Attraverso simulazioni, dimostriamo come il tempo di elaborazione richiesto dagli approcci che selezionano in modo intelligente le informazioni da trasmettere potrebbe essere troppo elevato per soddisfare i requisiti rigorosi di ambienti critici come le reti veicolari; pertanto, un approccio casuale, in cui le informazioni da trasmettere vengono selezionate in modo casuale dai capi cluster, risulta più efficace. Inoltre, valutiamo il potenziale degli algoritmi di disseminazione basati su cluster proposti in funzione di diversi parametri, inclusi le condizioni del canale e il numero di veicoli.

Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

5G 5th Generation

AHP Analytic Hierarchy Process

AoI Age of Information

AV Autonomous Vehicle

BS Base Station

CARLA CAR Learning to Act

CAV Connected Autonomous Vehicle

CD Chamfer Distance

C-ITSs Cooperative and Intelligent Transportation Systems

CV Connected Vehicle

DL Deep Learning

DSRC Dedicated Short-Range Communication

FoV Field of View

GPS Global Positioning System

LiDAR Light Detection and Ranging

LOS Line Of Sight

ML Machine Learning

mmWaves Millimeter Waves

NLOS Non-Line Of Sight

PQoS Predictive Quality of Service

PL Path Loss

QoS Quality of Service

RL Reinforcement Learning

SELMA SEmantic Large-scale Multimodal Acquisitions

SNR Signal to Noise Ratio

SS Semantic Segmentation

TD Teleoperated Driving

UMa Urban Macro

V2I Vehicle-to-Infrastructure

V2V Vehicle-to-Vehicle

V2X Vehicle-to-Everything

VoI Value of Information

1 Introduction

The increasing use of on-board communication devices allows smart vehicles to transmit their sensor data to cloud platforms and other vehicles [1]. While this opens up opportunities for new applications, the large volume of data traffic in vehicular networks is likely to become a significant long-term communication challenge.
In this context, V2V connectivity offers the ability for vehicles to communicate and share data with each other in real-time. This concept is crucial because vehicles, even with advanced sensor systems, may not perceive every detail of their surroundings [2]. By establishing connectivity, vehicles can enable a collaborative approach for object perception, where they can collectively enhance their awareness of the environment.

1.1 Connected Intelligent Transportation Systems

Advancements in vehicle automation are an ongoing process that offers numerous benefits, including reducing driver fatigue, enhancing road safety, improving fuel efficiency, and enabling smart parking solutions. An Autonomous Vehicle (AV), commonly referred to as self-driving car, is a vehicle that require minimal driver assistance [3]. It is essential to distinguish between terms like AV, Connected Vehicle (CV) and Connected Autonomous Vehicle (CAV). Atkins defines an AV as "a car that is capable of fulfilling the operational functions of a traditional car (e.g., driving, lane-change, parking, etc.) without the aid of a human operator" and a CV is described as "a car that is equipped with a tech-

nology enabling it to connect and communicate with other devices within the car, and also to other surrounding cars and external networks (e.g., internet, navigation, environment data, etc.)"[4]. Therefore, a CAV is a vehicle that can not only perform traditional car functions independently but also communicate with surrounding vehicles and infrastructure for safer driving. As pointed out in [5], introduction of CAVs brings many advantages, such as crash avoidance/-severity reduction [6] ,[7], attention monitoring [8] and congestion assistant or traffic jam assist [9], [10].

Besides, in recent years, the automotive sector has shifted towards C-ITSs, both for economic reasons [11] and to enhance safety during travel [12] and streamline traffic control [13]. These applications, for their generality and complementarity, are good representatives of future vehicular services [14].

The introduction of C-ITSs brings the possibility of creating advanced systems like connected adaptive cruise control [15], [10], hence the creation of vehicle platoonings [16].

Moreover, Teleoperated Driving (TD), has been gaining traction in the industry and is becoming crucial in assisting self-driving cars with navigating challenging situations they can't manage independently [17]. The performance of TD is highly dependent on network conditions, as transmitting high-resolution sensor data from the vehicle to the remote control center can require hundreds of megabits per second, thus researches have explored Predictive Quality of Service (PQoS) [18], [19] as a tool to predict unanticipated degradation of the QoS, and allow the network to react accordingly.

This thesis work will focus on a TD scenario, where the vehicles transmit their perception records to the BS, where the (software) operator sits, maximizing the information regarding critical elements such as walkers, bikers and other vehicles, and, at the same time, meeting the strict requirements of the 5G specifications set by the 5G Automotive Association (5GAA). In particular, for Infrastructure-Based TD, the service-level latency between the vehicle and the remote driver is set at 50 ms for both uplink and downlink. The service-level reliability is 99% for data transmission from sensors to the remote host, while the reverse direction, which involves commands from the remote driver, requires a higher reliability level of 99.999% [20].

1.2 NR V2X and Millimeter-Wave Communication

Several studies have shown that to experience the positive impacts of C-ITSs and smart vehicles in general, an high market penetration is required [21], [22]. Thus, to meet the needs of the increasing number of smart vehicles on the network connectivity, in terms of high demands in data rate, reliability, and latency [23], researchers are exploring new radio systems, like mmWaves, to address these challenges [24],[25].
The introduction of 5th Generation (5G) wireless networks brought new standards for peak data rates, latency, mobility, connection density and reliability, which can support critical applications such as vehicular networks. Moreover, since 5G also enables the possibility to communicate in the mmWaves spectrum, it can offer nominal data rates significanlty higher than other V2V communication protocols such as Dedicated Short-Range Communication (DSRC) [26]. To meet the demanding requirements of future vehicular networks, the 3rd Generation Partnership Project (3GPP) has developed the New Radio Vehicle-to-Everything (NR V2X) standard as part of its Release 16 specifications. NR V2X is tailored for Vehicle-to-Infrastructure (V2I) and other vehicle communication modes, also operating at high frequencies, including the mmWaves spectrum. Recent advancements include improvements to sidelink and network architecture, the introduction of a flexible numerology, and a new resource allocation scheme that allows vehicles to autonomously schedule sidelink resources (known as mode 2). Notably, both standards will support operations at mmWaves frequencies, up to 71 GHz. At these frequencies, the large available bandwidth can theoretically enable data connections with rates in the range of several gigabits per second [1], [25].

Despite its potential, mmWaves communication in vehicular networks presents several challenges due to the unique propagation characteristics of high-frequency signals. These challenges include:

- **Path and Penetration Loss**: Signals in the mmWaves spectrum experience significant path loss and are easily obstructed by physical objects, limiting their range.
- **Beamforming and Directional Communication**: To overcome path loss, mmWaves communication relies on highly directional beams that require

precise alignment between the transmitter and receiver. However, this introduces complexity in maintaining stable connections, especially in dynamic vehicular environments.

- **Interference**: Vehicles, particularly their metallic structures, can act as reflectors, causing strong interference in the mmWaves band. Additionally, the high frequency of mmWaves makes the communication link susceptible to time-varying channel conditions, which can affect the reliability of the connection [27].

1.3 Thesis Purpose and Outline

Even with high data rates offered by the transmission at mmWaves, the demanding requirements of C-ITSs applications may exceed those capabilities [28]. Thus, a possible approach to overcome this limitation, is to reduce the data to transmit by investigating the Value of Information (VoI) [29]. With this approach we select the most important and useful data for our purpose and transmit only those. The concept of VoI, although its abstract nature, has already been studied under an economic perspective to support data management and decision making [30] and applied to underwater systems to decide how much information to transmit through resource constrained networks [31]. Traditional approaches in information management focus on monitoring the Age of Information (AoI) [32] to determine if data is outdated and drop it if necessary. However, VoI considers additional factors like proximity and information quality. Various methods, such as heuristic and analytic approaches like Analytic Hierarchy Process (AHP) [33], have been proposed to characterize VoI [34]. However, utilizing VoI effectively in vehicular networks faces several challenges. First of all, VoI should depend on the dynamic nature of the vehicular environment and there is a need to explore vehicular attributes for better VoI assessment. Moreover, algorithms assessing VoI need to consider the prevalence of non-smart vehicles on the roads, and adapt dissemination strategies accordingly.
With this thesis, we propose the design and evaluation of clustering-based algorithms to enhance cooperative perception in vehicular networks, with focus on a TD scenario.
In particular, we exploit V2V connectivity in a smart way, by grouping vehicles in clusters, thus letting vehicles communicate with each other only when they

are near in space, and adapt the number of clusters according to the state of the network and distances between vehicles.
The research focuses on exploring how to assess VoI of vehicles data within the clusters using various dissimilarity metrics and determine optimal data dissemination methods for vehicular communication, with emphasis on reducing the burden on the communication network while maintaining high-quality perception records.
To achieve this, we utilize K-means clustering algorithms to leverage the spatial correlation of vehicles, and hierarchical clustering to prioritize data that carries more information (in terms of higher number of recognized objects) or stands out as more unique compared to other data.
We evaluate the performance of the proposed algorithms using data from SELMA [35], a synthetic dataset for vehicular networks developed by the University of Padova, where we implemented a channel model to characterize V2V communication in the cluster and the communication between the cluster head and the BS. The dataset represents a portion of a city with all its structures and critical elements (pedestrian, bikers, vehicles, etc.) and we test how the performance changes with different level of processing delays at the application level, different channel conditions and different number of smart vehicles (i.e. vehicles which can capture and transmit perception records).
We compare the results of the simulations also with some benchmark approaches, where there is no concept of VoI and what to transmit is selected randomly or everyone tries to transmit to the BS without any form of V2V communication.

1.3.1 Thesis Outline

In chapter 2 different dissimilarity metrics for comparing vehicles' perception records are discussed, such as based on the geometric distance and CD, each with its merits and limitations in a dynamic environment.
In chapter 3 we describe our proposed cluster-based data dissemination methods, and the relative benchmark schemes. Clustering methods and Reinforcement Learning (RL) are key techniques used in the proposed approaches. In particular we propose a double-layer clustering technique where the first layer is addressed at the BS with a K-means clustering algorithm, where the choice of K is carried

out by a RL agent. Then, for every resulting cluster, the vehicle with the best channel with the BS is elected as master, and receives all the perception records from the other vehicles in the cluster; it then assigns different values of VoI and, based also on his channel capacity, selects what to transmit. In one approach, higher VoI will be assigned to perception records more distinct within each other, while, in another approach, higher VoI will be assigned to perception records with higher number of recognized objects. The benchmark methods include an approach where the perception records to transmit are selected randomly by the master vehicles, which we call a single layer approach; Moreover, we implement an approach where every smart vehicle tries to transmit its data, without considering the V2V communication thus there are no clusters.
In chapter 4 we analyze the simulation environment: The SELMA dataset and channel model used for simulating the algorithms are described. Additionally, the object recognition and data streamline techniques to improve computational efficiency are detailed.
Finally we discuss the results of the simulations in chapter 5 analyzing the performance of the different proposed approaches. Processing delays, signal-to-noise ratio (SNR) degradation, and the number of vehicles are evaluated to determine their impact on system performance.
The obtained results show that the processing delays at the application level, that are needed to process data from vehicles to assign different levels of VoI, and select what to transmit, are too high to be efficient in a vehicular environment. However, all the approaches using V2V communication outperforms those that do not, and, with idealized processing delays, the approach that gives higher value of VoI to perception records with higher number of recognized objects, performs better with respect to the approach that gives more VoI to perception records that stands out as more unique compared to other ones.
In chapter 6 we provide the conclusions and future works for this thesis.

2

Dissimilarity Metrics

In this chapter, we explore various dissimilarity metrics that are crucial for comparing the perception records of vehicles in a dynamic environment. These metrics allow us to quantify the differences between vehicles' observations, which is key to enabling more effective data fusion, improving situational awareness, and enhancing decision-making in vehicular networks. We begin with the simplest geometric distance metric and move on to more complex and computationally intensive measures, such as the CD, that take into account the vehicles' surrounding environment and sensor data. Additionally, we discuss other potential metrics and their applicability to this problem space.

2.1 Geometric Distance

The first metric that we consider is also the simplest one, which takes into account the geometric distance between vehicles. Closer vehicles are, in fact, likely to have a perception record more similar than far away vehicles. We use the L2 norm or Euclidean Distance to measure how close the vehicles are. The L2 norm, also known as the Euclidean norm, is a measure of the length (or magnitude) of a vector in Euclidean space. It is defined as the square root of the sum of the squares of its components. For a vector $\mathbf{x}$ in $\mathbb{R}^n$ with components $x_1, x_2, \ldots, x_n$, the L2 norm is given by:

$$\|\mathbf{x}\|_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$$

In our scenario, n is equal to 3, since we are in a 3D space and the components correspond to the x, y, z coordinate of the vehicles and to obtain the Euclidean Distance between two vehicles, V_1 and V_2 with coordinates $[x_1, y_1, z_1]$ and $[x_2, y_2, z_2]$ we compute the L2 norm of the vector $\mathbf{v} = [x_1 - x_2, y_1 - y_2, z_1 - z_2]$:

$$\|\mathbf{v}\|_2 = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

This metric yields the space correlation between vehicles thus it is used to group them in different areas of the map but does not take into account any obstacles or road structures that could make the perception record of two close vehicles very different.

2.2 Chamfer Distance

To consider the surroundings of the vehicles and take into account more precise spatial information, we implement the CD between the point clouds acquired by the on board Light Detection and Ranging (LiDAR) on board of the vehicles. The CD has also previously been used in the PQoS framework [18] to compute the difference between acquired point clouds at different time instants of the same vehicle.

It is defined as following:

Given two point clouds P and Q, where $P = \{p_1, p_2, \ldots, p_m\}$ and $Q = \{q_1, q_2, \ldots, q_n\}$, the Chamfer Distance between P and Q is defined as:

$$CD(P, Q) = \sum_{p \in P} \min_{q \in Q} \|p - q\|^2 + \sum_{q \in Q} \min_{p \in P} \|q - p\|^2 \tag{2.1}$$

This formula can be broken down into two parts:

1. For each point in P, find the closest point in Q and compute the squared Euclidean distance. Sum these distances for all points in P.

2. For each point in Q, find the closest point in P and compute the squared Euclidean distance. Sum these distances for all points in Q.

The CD is symmetric and is a measure of how similar two point clouds are.

From a computational perspective, it is highly expensive and time-consuming to compute, posing significant challenges, particularly in scenarios with tight time constraints such as vehicular networks. To address this issue, in [36] is

proposed to organize the point clouds in a k-dimensional tree structure so that becomes faster to obtain the nearest neighbour. However this technique is still not efficient enough for our purposes, leading to peaks of exectuion times of $700ms$ so we use the built in method of the Open3D library [37] to compute the distance between two point clouds which we empirically measured to be the faster one, lowering down the computation time to $8ms$.

2.3 Other Metrics

Some alternative methods that could be used to address dissimilarity include voxelization and clustering. However, they were not considered in this study due to their high computational demands, which make them impractical for our specific needs, as pointed out in previous studies [38]. While object recognition is also computationally expensive to compute [39], the main advantage is that it can be parallelized assuming all the vehicles have dedicated hardware on board. We decide then to consider it since it introduces a valuable metric: the number of recognized objects by a vehicle. This allows us to assign greater importance to vehicles that identify more objects, and less importance to those recognizing fewer. This approach opens up interesting applications, which will be explored in sec. 3.2.

3

Data Aggregation for Cooperative Perception

In this chapter we will analyze different techniques that can be used to aggregate the data collected by the vehicles. The aim is to gather as much information as possible so that the BS will have as much visibility as possible of what is happening in the scenario, crucial point to take into account in the context of TD.

3.0.1 Clustering

To address this, we will make an extensive use of clustering, a Machine Learning (ML) technique which falls under the category of unsupervised learning which is supposed to organize data in some meaningful way. Formally speaking, clustering is the task of grouping a set of objects such that similar objects end up in the same group and dissimilar objects are separated into different groups.

Lloyd's K-means Clustering

Lloyd's K-means algorithm is a widely used method for clustering that aims to partition a set of points into k distinct clusters minimizing the average distance of every point to the center of the cluster, called centroid. The algorithm operates in an iterative manner, refining the cluster centroids until convergence. The

primary goal of the K-means algorithm is to minimize the sum of squared distances between the data points and their respective cluster centroids.

The K-means algorithm consists of the following steps:

1. Initialize k cluster centroids randomly.
2. Assign each data point to the nearest centroid.
3. Recompute the centroids as the mean of all points assigned to each centroid.
4. Repeat steps 2 and 3 until convergence (until the position of the centroids is stable over successive iterations [40]).

Pseudocode The pseudocode for Lloyd's K-means clustering algorithm is as follows:

Algorithm 1 Lloyd's K-means Clustering Algorithm

Input: Dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, Number of clusters k
Output: Set of clusters $\mathbf{C} = \{C_1, C_2, \ldots, C_k\}$, Centroids $\mu_1, \mu_2, \ldots, \mu_k$
Initialize k centroids $\mu_1, \mu_2, \ldots, \mu_k$ randomly from $\mathbf{X}$
repeat
  Assignment Step: Assign each data point $\mathbf{x}_i$ to the nearest centroid μ_j
  for each cluster C_j **do**
    $C_j \leftarrow \{\mathbf{x}_i : \|\mathbf{x}_i - \mu_j\|^2 \leq \|\mathbf{x}_i - \mu_l\|^2, \forall l \neq j\}$
  end for
  Update Step: Recalculate the centroid μ_j for each cluster C_j
  for each cluster C_j **do**
    $\mu_j \leftarrow \frac{1}{|C_j|} \sum_{\mathbf{x}_i \in C_j} \mathbf{x}_i$
  end for
until centroids do not change

return the set of clusters $\mathbf{C}$ and centroids $\mu_1, \mu_2, \ldots, \mu_k$

Explanation

1. **Initialization:** Centroids are initialized randomly from the data points.
2. **Assignment Step:** Each data point is assigned to the cluster with the nearest centroid. The Euclidean distance is typically used to determine the nearest centroid.

3. **Update Step:** The centroid of each cluster is recalculated as the mean of all points assigned to that cluster. This updated centroid will then be used in the next iteration.

4. **Convergence:** The algorithm iterates between the assignment and update steps until the centroids stabilize, meaning they no longer change significantly or the assignments no longer alter.

Lloyd's K-means clustering is effective for finding spherical clusters but may struggle with clusters of varying shapes and sizes or when the number of clusters is not specified accurately. Its performance can be sensitive to the initial placement of centroids and may require multiple runs to achieve an optimal clustering solution.
The resulting clusters will have a gaussian distribution, with means the position of centroids.

Complexity The time complexity of Lloyd's K-means clustering algorithm is $O(t \cdot n \cdot k \cdot d)$, where t is the number of iterations, n is the number of data points, k is the number of clusters, and d is the number of dimensions. The space complexity is $O(k \cdot d)$ for storing the centroids and $O(n \cdot d)$ for storing the data points and assignments. To improve the performance we use the Scikit python library [40] dedicated method to perform the K-means clustering which is able to achieve a time complexity of $O(k \cdot n \cdot t)$ [**kmeansperformance**].

Hierarchical Clustering

Another important clustering method analysis that we will use extensively is Hierarchical Clustering, which seeks to build a hierarchy of clusters. Unlike partition-based methods like K-means clustering, hierarchical clustering does not require the number of clusters to be specified in advance. Instead, it produces a tree-like structure called a dendrogram, which illustrates the arrangement of the clusters formed at different levels of hierarchy.

There are two main types of hierarchical clustering: *agglomerative* and *divisive*.

Agglomerative Hierarchical Clustering

Agglomerative hierarchical clustering, also known as *bottom-up* clustering, starts with each data point as its own individual cluster. The algorithm then iteratively merges the closest pairs of clusters until all data points are contained

in a single cluster or until a stopping criterion (like a predefined number of clusters) is reached.

The process involves the following steps:

1. **Initialization**: Each data point is initially considered as a separate cluster.

2. **Similarity Computation**: At each step, the algorithm calculates the distance between every pair of clusters. In our framework, we will consider both the CD and the geometric distance as similarity metrics.

3. **Merging**: The two clusters with the smallest distance are merged together to form a new cluster.

4. **Termination**: This process is repeated until the desired number of clusters is achieved or all points are merged into a single cluster.

The result is a dendrogram, where the root represents the final cluster containing all data points, and the leaves represent the individual data points. The height of the connection at which two clusters are merged represents the distance between them.

Divisive Hierarchical Clustering

Although we will not utilize this technique, Divisive hierarchical clustering, or *top-down* clustering, is still worth noting: it begins with all data points in a single cluster and then the algorithm recursively splits the cluster into smaller clusters until each data point is in its own cluster or a predefined stopping criterion is met.

The key steps in divisive clustering are:

1. **Initialization**: All data points start in one cluster.

2. **Splitting**: At each step, the algorithm identifies the cluster that is the most heterogeneous and splits it into two smaller clusters, typically using a clustering method such as K-means.

3. **Update**: The process is repeated by selecting the next cluster to split.

4. **Termination**: The algorithm stops when a specified number of clusters is formed or when splitting further is not meaningful.

Linkage Criteria

A crucial aspect of hierarchical clustering is how the distance between clusters is computed, which is determined by the *linkage criterion*. Common linkage criteria include:

- **Single Linkage**: The distance between two clusters is defined as the shortest distance between any single data point in the first cluster and any single data point in the second cluster.
- **Complete Linkage**: The distance between two clusters is the greatest distance between any single data point in one cluster and any single data point in the other cluster.
- **Average Linkage**: The distance between two clusters is the average of all pairwise distances between points in the two clusters.
- **Wards Method**: This method minimizes the total within-cluster variance. At each step, the pair of clusters with the smallest increase in variance after merging is chosen.

Advantages and Disadvantages
Hierarchical clustering has several advantages:

- It does not require the number of clusters to be specified in advance.
- It produces a dendrogram, which gives a visual representation of the data and can help in determining the number of clusters.
- It is particularly useful for small datasets and for data that naturally forms a hierarchical structure, just like in our case, where the number of vehicle is not too big and the similarity measure between them is suitable to form a hierarchical structure.

However, there are also some disadvantages:

- Hierarchical clustering is computationally expensive, especially for large datasets, since it has a time complexity of $O(n^3)$.
- The choice of distance metric and linkage criterion can significantly influence the final clusters, which may not always be intuitive.
- Once a merge or split is done, it cannot be undone, leading to potentially suboptimal clustering solutions.

3.0.2 Reinforcement Learning Agent

As we have seen, in K-Means clustering the choice of the parameter K is very important and can deeply affect the performance of the whole framework. Since vehicular networks are highly dynamic environments, it's impossible to determine a single optimal value for K that works in all situations. For instance, if we set K to 10 when there are 30 vehicles, we might achieve good clustering and conclude that this is a suitable value. However, if most of the vehicles leave, leaving only 9 or fewer, the same value for K would no longer be effective. Additionally, changes in the spatial distribution of the vehicles could also lead to poor performance. Therefore, we need to develop a method to choose K dynamically and intelligently. To address this requirement we use a RL agent.

RL is a branch of machine learning that deals with how an agent should take actions in an environment to maximize cumulative rewards. It involves learning through interaction with the environment, making decisions that affect the state of the environment and receiving feedback in the form of rewards.

In its simplest form, RL can be understood through the *multi-armed bandit* problem. This problem is a straightforward example of reinforcement learning, where an agent is faced with *multi* different actions (or arms), each associated with an unknown probability distribution of rewards. The agent's objective is to maximize its total reward over time by selecting the best action as often as possible.

Key Concepts in the Multi-Armed Bandit Problem

- **Agent**: The decision-maker that interacts with the environment by choosing actions (pulling arms).

- **Action (A)**: The set of possible choices or arms the agent can pull. Each action has an associated but unknown reward distribution.

- **Reward (R)**: The immediate feedback received by the agent after selecting an action. Each action has a reward distribution, and the agent aims to maximize the sum of the rewards over time.

- **Exploration vs. Exploitation**: A critical trade-off that is faced in the multi-armed bandit problem where the agent must choose between *exploiting* the

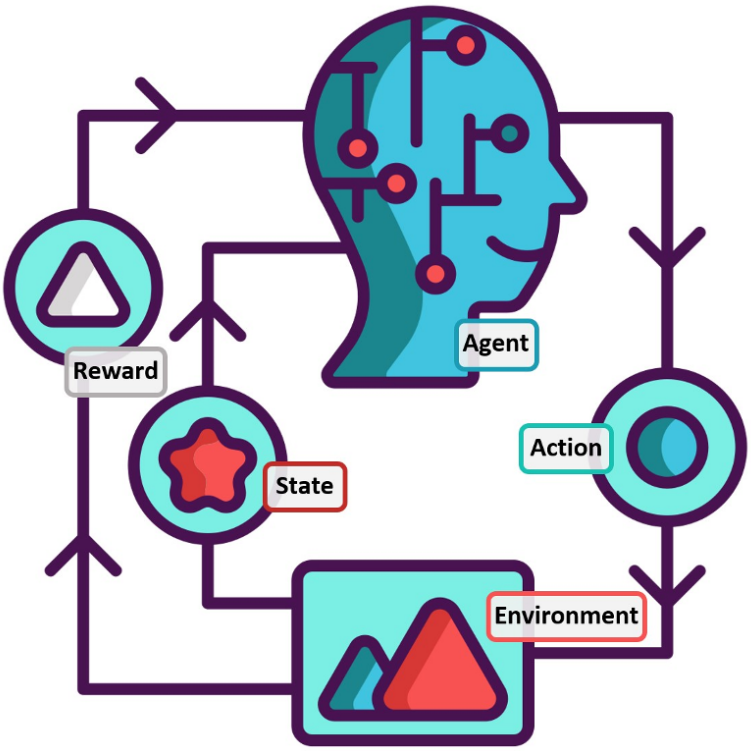

Figure 3.1: Reinforcement Learning Paradigm
Image is taken from [41]

action that has yielded the highest reward so far and *exploring* less-known actions that might lead to better rewards in the future.

The Multi-Armed Bandit Algorithm

The multi-armed bandit problem serves as a fundamental model in RL, illustrating the exploration-exploitation dilemma. Several strategies can be used to solve this problem:

- **Epsilon-Greedy Strategy** is one of the simplest methods where, with probability $1-\epsilon$, the agent exploits the action with the highest observed reward, and with probability ϵ, it explores a random action. This strategy balances exploration and exploitation by adjusting the value of ϵ.
- **Upper Confidence Bound (UCB)** is a more advanced strategy that selects actions based on both the average reward and the uncertainty or variance in the reward estimates. This approach systematically balances exploration and exploitation by considering the potential for higher rewards from less frequently chosen actions.

- **Gradient Bandit Algorithm (GBA)** is an approach that encourages exploration without choosing randomly the action to explore. In GBA, we define a preference for each action and the larger the preference, the more often the action is taken.

In dynamic environments, such as vehicular networks, conditions can change rapidly, and the optimal action may vary over time. The multi-armed bandit algorithm is well-suited to our scenario because the distribution of the rewards is stationary since we do not operate major changes to the Signal to Noise Ratio (SNR) or the capacity within one simulation run, but we do that for different runs, as we show in chapter 5.

For example, in the context of dynamically selecting the parameter K for K-Means clustering in vehicular networks, the multi-armed bandit algorithm can be employed to choose the optimal value of K over time. By treating each possible value of K as an arm, the agent can explore different values and exploit the one that provides the best clustering performance, adapting as the number and arrangement of vehicles change.

This adaptability makes the multi-armed bandit algorithm a powerful tool for managing dynamic systems where the optimal choice can shift due to changes in the environment.

In our framework, since we seek for a faster convergence, we implement the multi-armed bandit algorithm with an Epsilon-Greedy Strategy using optimistic initialization values, meaning that we encourage exploration in the early stages of learning by initializing the estimated values of actions to be higher than their expected values. The pseudocode of the algorithm is taken from [42] and is provided in Algorithm 2.

We define the reward R as follows:

$$R = \frac{n_{\text{recognized objects}}}{n_{\text{recognizable objects}}}$$

and we set $Q_{init} = 2$, so higher than the maximum reward value which will be 1 and $\varepsilon = 0.2$.

Algorithm 2 Optimistic Initialization for K-Armed Bandit with Epsilon-Greedy Strategy

Initialize, for $a = 1$ to k:
$Q(a) \leftarrow Q_{\text{init}}$
$N(a) \leftarrow 0$

Loop forever:
$A \leftarrow \begin{cases} \arg\max_a Q(a) & \text{with probability } 1 - \varepsilon \\ \text{a random action} & \text{with probability } \varepsilon \end{cases}$
$R \leftarrow bandit(A)$
$N(A) \leftarrow N(A) + 1$
$Q(A) \leftarrow Q(A) + \frac{1}{N(A)} [R - Q(A)]$

3.1 Double Layer Clustering

To take advantage of both the local awareness of vehicles and the broader perspective of the BS, we propose a two-layers clustering method. We call it two-layers because it involves two separate clustering steps: first, a K-means clustering, followed by a second clustering process within each resulting group of vehicles.

3.1.1 K-means Clustering at the Base Station

The first clustering layer is handled by BS. Since the BS has continuous access to vehicle data, including their Global Positioning System (GPS) coordinates, it is aware of the exact locations of all vehicles.

Using this location data, the BS performs K-means clustering based on the geometric distance between vehicles. This method groups nearby vehicles into clusters. After this clustering, the BS selects one vehicle from each cluster to act as the master vehicle (or centroid). The selection is based on which vehicle in each group has the highest SNR.

Once the master vehicle is selected, the BS informs all vehicles in the cluster which vehicle has been chosen as their master. Each vehicle then knows both its cluster and the identity of the master vehicle to which it should send its data. This clustering approach takes advantage of V2V communication by allowing nearby vehicles to share information about their local environment in an efficient manner.

Figure 3.2 shows an example of the K-means clustering results applied to our scenario. At this stage, the master vehicles have full access to the data (in the form of point clouds) collected by all the vehicles within their respective clusters.

The decision on how many clusters (K) to use is determined by a RL agent, which optimizes this parameter.

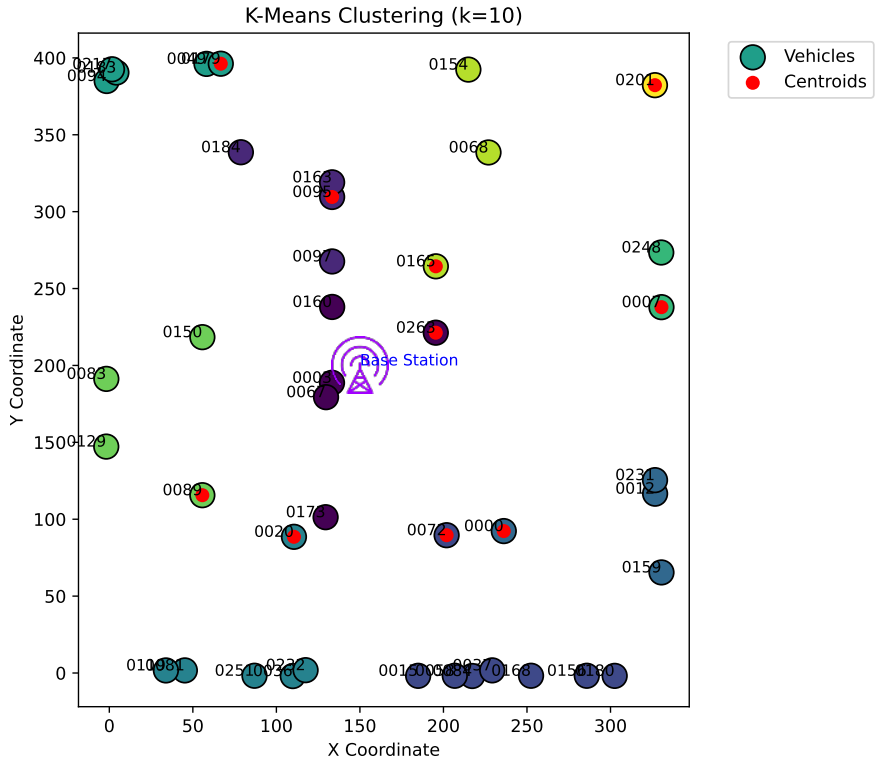

Figure 3.2: K-means clustering at the BS; different colors of the vehicles represent the different clusters they belong to. In red the master vehicle elected for each cluster

3.1.2 Hierarchical Clustering at the Master's Vehicles

All the master vehicles are now able to identify which are the best point cloud that will be transmitted. Based on their channel quality, they compute the number of point clouds they are able to send n_{pcds} with the following formula:

$$n_{\text{pcds}} = \left\lfloor \frac{\left(\frac{C \times 10^6}{8}\right) \times t_a}{s_{\text{pcd}}} \right\rfloor \tag{3.1}$$

where c is the capacity and is computed using the Shannon Capacity Theorem:

Theorem 1 (Shannon Capacity Theorem) *The channel capacity c, which is the maximum rate at which information can be transmitted over a communication channel without error, is given by:*

$$C = B \log_2 \left(1 + \frac{S}{N}\right)$$

Where:

- *C is the channel capacity in bits per second (bps),*
- *B is the bandwidth of the channel in hertz (Hz),*
- *$\frac{S}{N}$ is the signal-to-noise ratio (SNR).*

t_a is the time available that is left for the actual transmission of the data, afterwards the delay required for the processing of the data and is computed as:

$$t_a = t_{\text{target}} - t_{\text{processing}}$$

and we set t_{target} to $50ms$, based on the requirements in the standard [43], s_{pcd} is the average size of the point cloud expressed in KBytes, which we measured to be of $30KB$ after compression (see sec. 4.1) and we set $B = 50MHz$.

The master vehicles know how many point clouds they can afford to send based on their data rate so they have to make an informed choice trying to maximize the amount of information that they send to the BS.
To do that, they perform an agglomerative hierarchical clustering using as distance metric the CD between the point clouds of the vehicles in the clusters and Average Linkage as linkage criteria (Figure 3.3). In this way the master vehicles are able to build the dendrogram and *cut* it based on the number of point clouds they can send. When cutting the dendrogram, in fact, based on the height of the cut, different numbers of clusters are formed: the lower the cut the higher the number of clusters.
The master vehicles then form a number of cluster equals to the number of point clouds that they are able to send and then select one representative at random per cluster; an example is shown in Figure 3.4.
The primary advantage of this approach is that the selected point clouds will be the most distinct from one another, providing a greater amount of information.

In fact, if we only use the spatial information available at the BS we would pick just vehicle far apart in the space. But, if two or more vehicles that are close to each other suffer from an obstructed view or there are obstacles between them, the perception record will be very different, and this knowledge can only be extracted by processing the point clouds.
Nonetheless there is a major drawback: the number of comparisons $n_{\text{comparisons}}$ the master vehicle has to make in order to compute the dendrogram is quadratic in the number of the vehicles in the cluster. It is given by:

$$n_{\text{comparisons}} = \frac{{n_{\text{vehicles}}}^2 - n_{\text{vehicles}}}{2}$$

This results in an high value of $t_{\text{processing}}$. We measured the average time required to compute the CD between two point clouds to be $8ms$.
To reduce the burden of the computation, the CD between vehicle is computed only if their distance is below a certain threshold, which we set to $100m$, which is the LiDAR range of operation, so vehicles that are further than the threshold will not have anything in common in their perception records. This rarely happens since, due to the way the clusters are formed, vehicles are unlikely to be far away from each other if they belong to the same cluster.

3.2 Object Recognition Exploitation

In this approach we first perform a K-means clustering at the BS, and then we elect the master (or centroid) in the same way as before. Then, all the vehicles in the cluster perform the object recognition task for their own point cloud, and send to the master their point cloud with the number of recognized objects.
At this point the master vehicles perform an agglomerative hierarchical clustering using the geometric distance between vehicles as distance metric to build the dendrogram.
They then compute the number of point clouds n_{pcds} that they are able to send based on their SNR and cut the dendrogram to form exactly n_{pcds} clusters. Now, unlike in the previous approach, they do not select one representative per cluster at random but select the vehicle which has the higher number of recognized objects.
This is done to overcome the possibility that two or more vehicles that have an high number of recognized objects but are really close to each other are selected

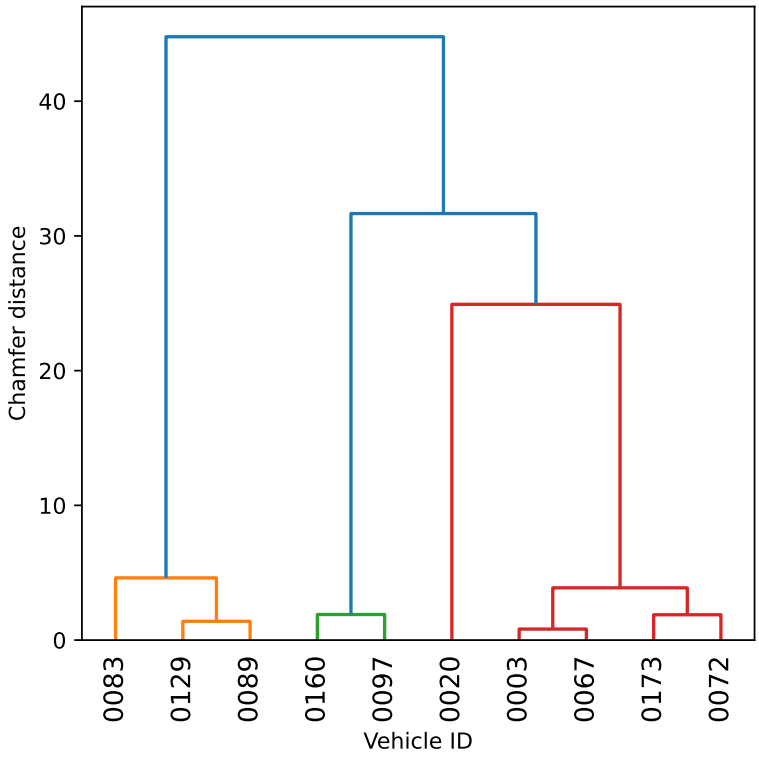

Figure 3.3: Example of Agglomerative Hierarchical Clustering performed by the master vehicle for its cluster

as representatives since they likely see the same objects.
In this approach the main drawback is given by the fact that the object recognition task is pretty expensive to perform, state of the art algorithms takes about $16ms$[39]. On the other hand, if we assume that every vehicle has dedicated on board hardware, the task can be parallelized and performed simultaneously by all the vehicles. In this way the processing delay is fixed and will not grow increasing the number of vehicles in the cluster.

3.3 Single Layer Clustering

In this last approach we firstly perform the K-means clustering at the BS as before, also selecting the master vehicles using the same strategy. All the vehicles will then send their point clouds to the master vehicles without processing them and the master vehicles will select at random the point clouds to send to the BS without any kind of processing. This way, the processing delay will be none and this is the major advantage of this approach, but at the same time the perception

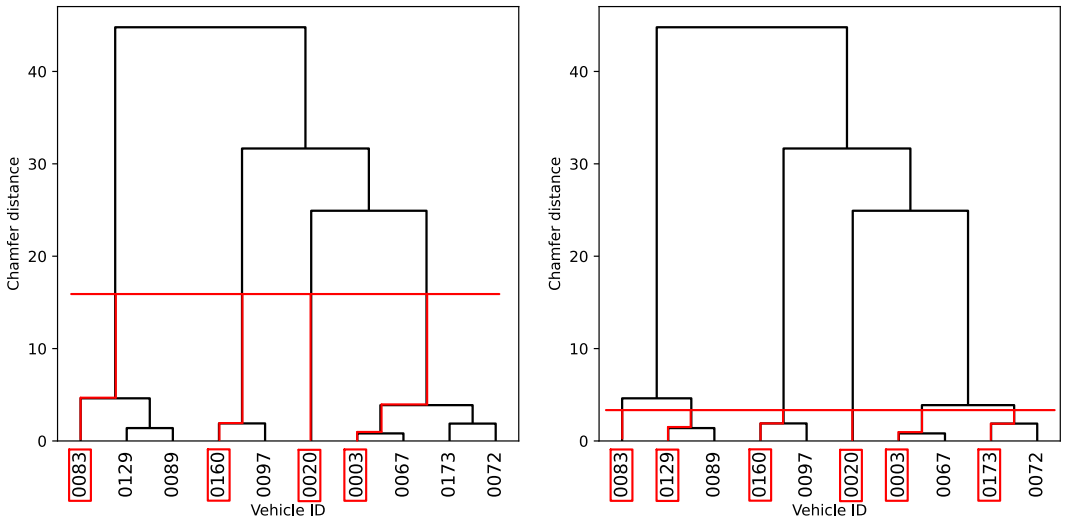

Figure 3.4: Dendrogram cut examples
On the left, the cut performed if we want to send 4 point clouds while on the right if we want to send 6 point clouds. Vehicles which point cloud is selected for transmission are marked in red.

record will likely be less informative, since the choice is not informed at all.

A comparison of the three approaches can be seen in Figure 3.5

3.4 Benchmark Methods

Finally, we implement a couple of methods to have a benchmark comparison with respect to solutions that do not exploit the V2V connectivity, thus vehicles will exchange information only with the BS.

3.4.1 Simple K-means

In this approach the BS will perform a K-means clustering of the vehicles using the geometric distance as distance metric using as K the same value of K used for the approach described in Section 3.1.1. At this point, the BS will only exchange information with the vehicle in the cluster with the higher SNR. The vehicles in the cluster will not exploit the sidelink communication so the BS will receive at most 1 point cluster, the one from the vehicle with the higher SNR in the cluster. If the sending vehicle will not meet the $50ms$ requirement to send the point cloud, it will not be considered valid. The main advantage is given by the fact that there is neither delay associated to sidelink communications nor for

processing the point clouds, but the drawback is given by the fact that in general the BS will always receive less point clouds than the other approaches, since the V2V connectivity is not exploited.

3.4.2 Everyone Transmits

The last approach is the simplest one, since every vehicle tries to transmit its point cloud to the BS. We obviously expect very poor performance from this approach since if everyone is transmitting the bandwidth will be significantly lower for every vehicle, leading to a lower data rate. Moreover, vehicles with a bad channel will affect the performance of vehicles with good channels, since they will try to transmit anyway.

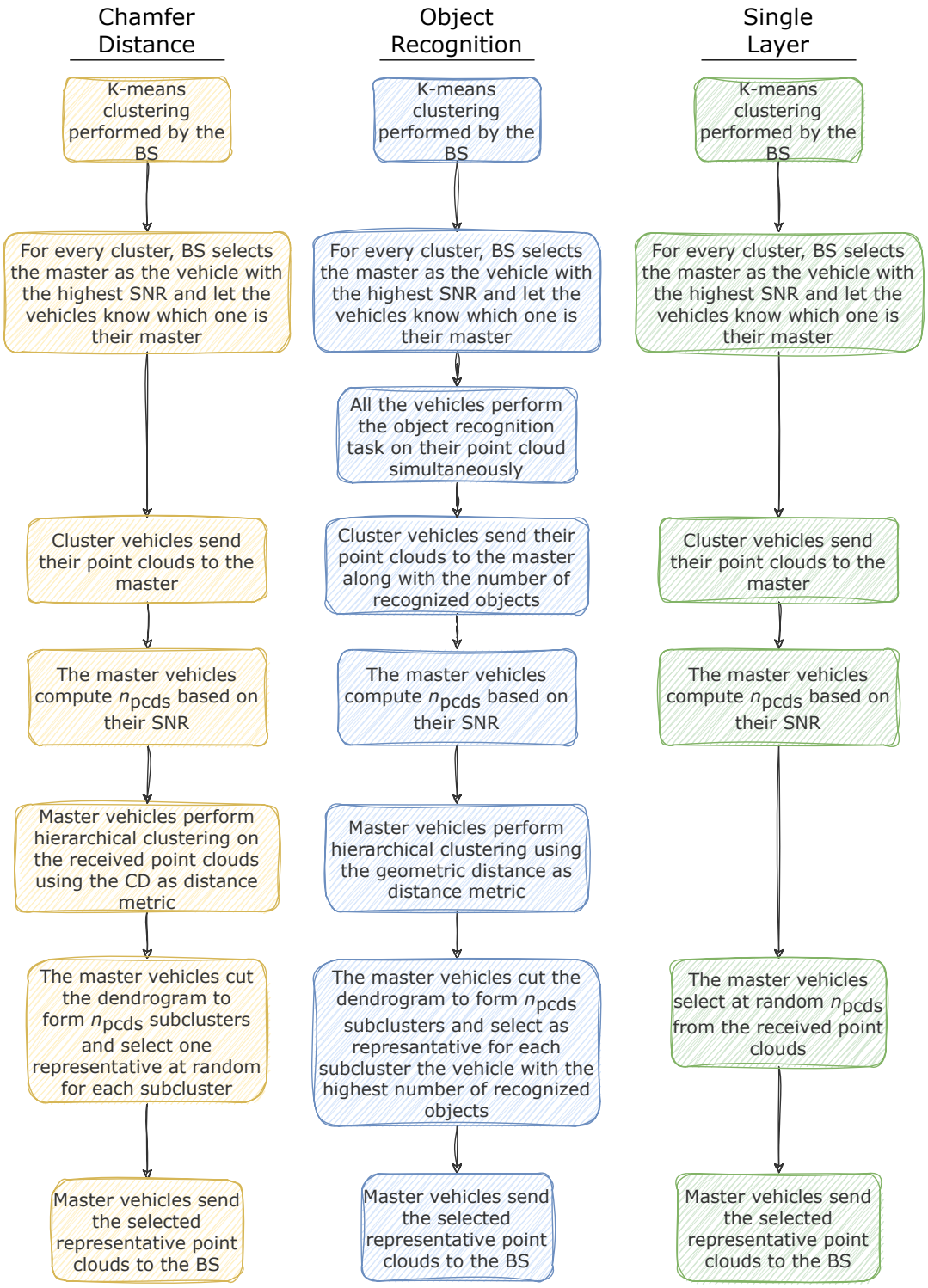

Figure 3.5: Comparison of three different approaches; on the left, the approach based on hierarchical clustering using CD, in the middle the one exploiting object recognition and on the right the single layer cluster approach where the representatives are selected at random.

4

Simulation Environment

In this chapter we provide an overview of the simulation environment used to carry out the analysis of the proposed approaches. We first explore the synthetic dataset used to simulate the city with its structures, vehicles, bikers and walkers and how the data is collected and formatted. We then cover the methodology used for simulating communication channels, essential for transmitting data between vehicles and the BS.

4.1 SELMA Dataset

Autonomous and TD tasks, such as Semantic Segmentation (SS) and Vehicle-to-Everything (V2X) communication, pose several challenges, particularly due to the complex and dynamic environments in which autonomous vehicles operate [44], [45]. In this context, ML and Deep Learning (DL) offer valuable approaches to tackle these challenges and enhance driving decisions [46]. However, these techniques require large amounts of labeled data for effective training, and obtaining and annotating such data is both expensive and time-consuming. Consequently, existing open-source datasets like Waymo [47], Cityscapes [48], and KITTI [49] are often limited in scope and diversity. Many datasets are too small to encompass the complexities of urban environments, lack coverage across various sensor types, and include unlabeled scenes, which compromise the training of ML models [50].

To address these limitations, the research community has turned to synthetic, computer-generated datasets, where simulations manage the entire data

generation process. This approach reduces costs, increases flexibility, and allows for the creation of large quantities of data compared to real-world data [51], [52], [53], [54], [55] . Simulations enable data collection across different conditions, scenarios, and sensor configurations. One open-source simulator that generates synthetic data is CAR Learning to Act (CARLA) [56]. CARLA offers urban layouts, diverse environmental conditions, models for vehicles, buildings, pedestrians, and supports flexible sensor setups. Currently, several synthetic datasets are available for SS in autonomous driving [51], [57], [58], [59], [60]. However, these datasets have limitations. The samples typically cover only a limited number of settings, similar viewpoints, and limited weather, lighting, and time-of-day conditions, often relying on data from a single sensor. Additionally, they often lack detailed control over weather conditions or alignment with common benchmark semantic class sets, such as Cityscapes.

The address these limitations, [35] presents SEmantic Large-scale Multimodal Acquisitions (SELMA). This dataset was designed for SS tasks in autonomous driving systems and provides a rich collection of data across varying environmental conditions. The SELMA dataset includes data from 24 sensors, such as RGB cameras, depth cameras, semantic cameras, and LiDAR, all mounted on vehicles in simulated urban and rural settings. The data was generated using a customized version of the CARLA simulator, which allowed for more diverse weather and lighting conditions, and included sensors specifically optimized for autonomous driving research. In particular, all the vehicles are equipped with a full sensor suite including:

- **7 RGB cameras**: these have a 90-degree horizontal Field of View (FoV) and a native resolution of 5120×2560, which is downsampled to 1280×640 to enable $4\times$ anti-aliasing. Post-processing effects include vignette, grain jitter, bloom, auto exposure, lens flare, and depth of field.
- **7 depth cameras**: these also feature a 90-degree horizontal FoV with a resolution of 1280×640. Anti-aliasing is omitted as it would degrade the quality of depth information.
- **7 semantic cameras**: these share the same specifications as the depth cameras.
- **3 semantic LiDARs**: each capable of producing 64 vertical channels and generating up to 100,000 points per second, with a range of 100 meters.

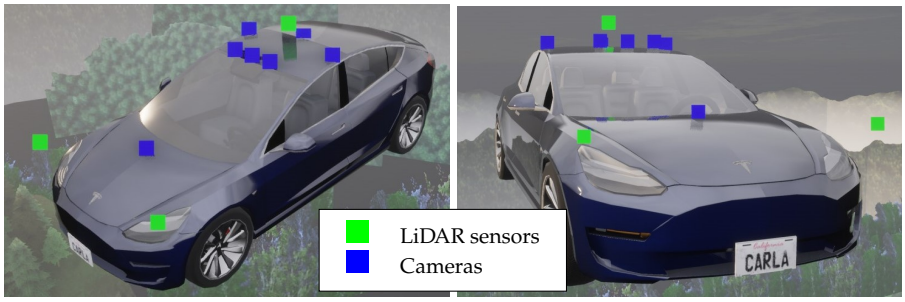

Figure 4.1: Placement of the sensors in a car

This provides a high-resolution, omnidirectional view of the surroundings, which is critical for understanding the vehicle's environment in various conditions, including fog, rain, and different times of day.

The placement of the sensors is illustrated in Figure 4.1.

For our analysis, we utilized the data from the LiDAR sensors, which generates a 3D point cloud representation of the environment around the vehicles, namely, their perception record. In particular, we use the point clouds generated by the LiDAR placed on top of the vehicle.
This choice is driven by many factors:

1. **Semplicity**: taking into account all the sensor on board of the vehicles would simply be too complex in this first explorative phase of our analysis and would introduce even larger processing delays which we definitely want to avoid.

2. **Reliability**: data acquired by the LiDAR on top provides reliable depth information compared to image based detection [61], [62]. Moreover, it offers full view of the surrounding scene of the vehicle, since it acquires data with a 360-degree FoV.

3. **Robustness to different weather and lighting conditions**: LiDAR acquisitions are the more robust to weather conditions, such as fog and rain as well as difficult light conditions such as backlight and night time.

4. **Efficiency**: point clouds can be efficiently compressed via Draco compression [63] while keeping a good level of information [64], making them suitable for fast transmissions.

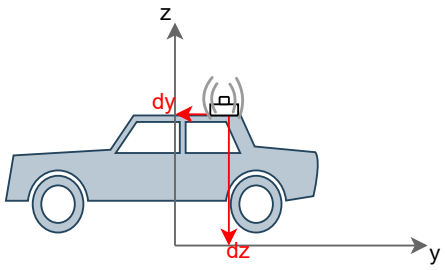

Figure 4.2: Side view of the LiDAR offset

The point clouds generated by the LiDAR sensor mounted on top of the vehicles need to have a common reference frame to allow a correct computation of the CD between them.

The acquired point clouds must be then aligned with respect to the relative location of the LiDAR placement on top of the vehicles and then adjusted with respect to the word reference frame of the simulated city. In particular, the LiDAR placement offset, with respect to the position of the vehicle, can be found in the blueprint of the vehicles provided in the dataset and is expressed by the vector o_{L}:

$$o_{\mathrm{L}} = \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix}$$

whose components represents the offset in the 3 components of 3D space; an example of one side view can be seen in Figure 4.2. It is also worth noting that $dx = 0$ for all the vehicles considered in our simulations. Then we need to take care of the word reference frame; to do that, we have to account for a translation with respect to the vehicle location in the map, which is provided in the vector V_l:

$$V_l = \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix}$$

as well as the orientation of the vehicle with respect to the origin, provided in the vector V_o:

$$V_o = \begin{bmatrix} \theta_x \\ \theta_y \\ \theta_z \end{bmatrix}$$

Using the euler angles convetions [65] we will also refer to these angles as

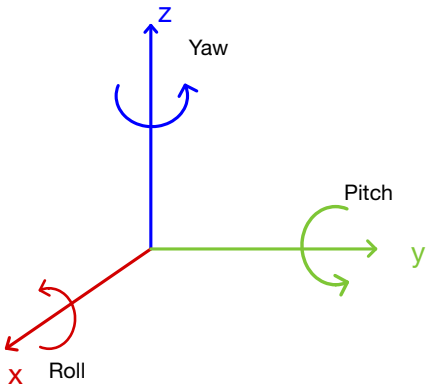

Figure 4.3: Euler's angle conventions

roll for rotations along the x axis, *pitch* for rotations along the y axis and *yaw* for rotations along the z axis, as in CARLA conventions, see Figure 4.3.

The alignment process is an affine transformation composed by a rotation and a translation: a rototranslation. The affine transform in 2D space is formally defined as:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{A} \begin{bmatrix} v \\ w \end{bmatrix} + \vec{b} \tag{4.1}$$

Where x and y are the coordinates of the projected point; $\mathbf{A}$ is a matrix that, based on its content, allows different transformations like scaling, shearing, reflection and, in our case, rotation; v and w are the coordinates of the point to transform while $\vec{b}$ is the translation vector, which will translate the point in the space.
In our case of rototranslation, in the xy coordinate plane, so the top view of the map, eq. 4.1 will become as follows:

$$\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \begin{bmatrix} \cos\theta_z & -\sin\theta_z \\ \sin\theta_z & \cos\theta_z \end{bmatrix} \begin{bmatrix} x_p \\ y_p \end{bmatrix} + \begin{bmatrix} V_x \\ V_y \end{bmatrix} \tag{4.2}$$

with x_p and y_p the coordinate of the point as we read it straight out of the dataset and x_t and y_t the coordinate of the rototranslated point.To ease the computation, we express all the points coordinate using homogenous coordinates (we add an extra dimension equal to 1), as this allows to represent affine transforms in a single matrix multiplication, as well as combining multiple transforms in a single

matrix. A generic point $P = (x_p, y_p)$ of the point cloud will then be expressed as

$$P_h = \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix}$$

and eq, 4.2 will become:

$$\begin{bmatrix} x_t \\ y_t \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_z & \sin\theta_z & V_x \\ -\sin\theta_z & \cos\theta_z & V_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} \tag{4.3}$$

We now want to extend this definitions to the 3D case, so we will combine three matrices to perform the rototranslation along the three axis by following the procedure in [65], with the difference that we also take into account that CARLA uses left-handed triplets, so x and y indices will be swapped. To build the rototranslation matrix $\mathbf{T}_{\text{word}}$ to adjust the point cloud with respect to the position and orientation of the vehicle we define $c_r = \cos(roll)$, $s_r = \sin(roll)$, $c_p = \cos(pitch)$, $s_p = \sin(pitch)$, $c_y = \cos(yaw)$, and $s_y = \sin(yaw)$. The final matrix will be as follows:

$$\mathbf{T}_{\text{world}} = \begin{bmatrix} c_p c_y & c_y s_p s_r - s_y c_r & -c_y s_p c_r - s_y s_r & -V_y \\ s_y c_p & s_y s_p s_r + c_y c_r & -s_y s_p c_r + c_y s_r & -V_x \\ s_p & -c_p s_r & c_p c_r & V_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.4}$$

Then, to adjust the point cloud with respect to the relative position of the LiDAR on the vehicle, we build a transformation matrix $\mathbf{T}_{\text{car}}$. Since the orientation of the LiDAR is the same for every vehicle, to ease the computation, we only take into account the relative location of it, expressed in the vector o_{L}:

$$\mathbf{T}_{\text{car}} = \begin{bmatrix} 1 & 0 & 0 & dy \\ 0 & 1 & 0 & dx \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.5}$$

We now need to take the inverse of this matrix, because the raw data from the dataset have already applied this transformation to the point cloud and we want

to *undo* it. Finally, we are able to apply these matrices together to perform the rototranslation in one step. We compute the dot product as follows:

$$\mathbf{P}_{\text{transformed}} = \mathbf{T}_{\text{world}} \cdot \mathbf{T}_{\text{car}}^{-1} \cdot \mathbf{P} \tag{4.6}$$

To perform this computation, we once again use the Open3D library [37], in particular its method `transform`. Example of rototranslation application can be seen in Fig. 4.4 and Fig. 4.5

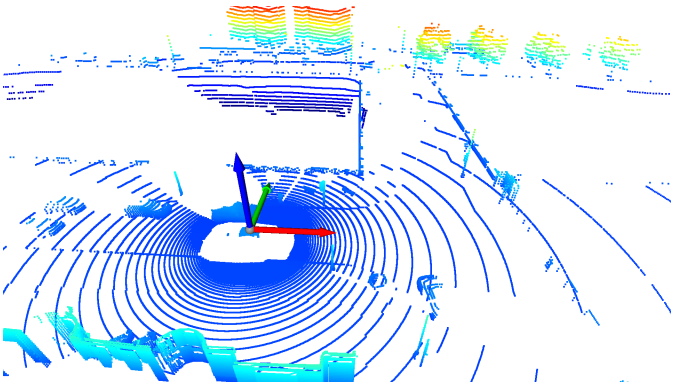

Figure 4.4: Point cloud before transformation; we can see the reference frame origin on top of the car.

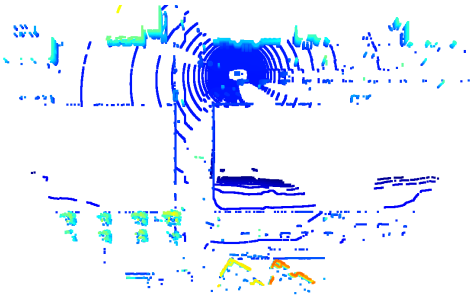

Figure 4.5: Point cloud after transformation; the reference frame origin has moved to the bottom left due to the transformation and defines where are the coordinates $(x, y, z) = (0,0,0)$

4.2 Channel Model

To simulate the transmission of data between the vehicles and the BS, we use the specification provided in the 3GPP 38.901 release 17 [66] focusing on the Urban Macro (UMa) scenario. In particular, we first compute the LOS probability Pr_{LOS} as follows:

$$Pr_{\text{LOS}} = \begin{cases} 1 & , d_{\text{2D-out}} \leq 18\,\text{m} \\ \left[\frac{18}{d_{\text{2D-out}}} + \exp\left(-\frac{d_{\text{2D-out}}}{63}\right)\left(1 - \frac{18}{d_{\text{2D-out}}}\right)\right] \cdot & \\ \cdot \left(1 + C'(h_{\text{UT}})\frac{5}{4}\left(\frac{d_{\text{2D-out}}}{100}\right)^3 \exp\left(-\frac{d_{\text{2D-out}}}{150}\right)\right) & , 18\,\text{m} < d_{\text{2D-out}} \end{cases} \tag{4.7}$$

where

$$C'(h_{\text{UT}}) = \begin{cases} 0 & , h_{\text{UT}} \leq 13\text{m} \\ \left(\frac{h_{\text{UT}}-13}{10}\right)^{1.5} & , 13\text{m} < h_{\text{UT}} \leq 23\text{m} \end{cases}$$

and h_{UT} is the height of the antenna placed on the vehicle, and to simulate it we have used the height of the LiDAR on top of the vehicle; in our case we always have $C'(h_{\text{UT}}) = 0$ since vehicles are never higher than 13m; d_{2D} and d_{3D} are defined in Fig. 4.6. We then compute the Path Loss (PL) using the formulas provided in table 4.1, where we define

$$d'_{BP} = 4h'_{BS}h'_{UT}f_c/c$$

with

$$h'_{\text{BS}} = h_{\text{BS}} - h_{\text{E}},$$
$$h'_{\text{UT}} = h_{\text{UT}} - h_{\text{E}}$$

and we set the environment height $h_{\text{E}} = 1\text{m}$, carrier frequency $f_c = 24.25Ghz$ in Frequency Range 2 (FR2). We then add a term related to the shadow fading as a

Scenario	PL[dB]
LOS	$\begin{cases} PL = 28 + 22\log_{10}(d_{3D}) + 20\log_{10}(f_c), & \text{if } 10 \leq d_{2D} \leq d'_{bp} \\ PL = 28 + 40\log_{10}(d_{3D}) + 20\log_{10}(f_c) - 9\log_{10}(d'^2_{bp} + (h_{bs} - h_{ut})^2), & \text{if } d_{2D} > d'_{bp} \end{cases}$
NLOS	$PL_{NLOS} = \max(PL, PL')$ $PL' = 13.54 + 39.08\log_{10}(d_{3D}) + 20\log_{10}(f_c) - 0.6(h_{ut} - 1.5)$

Table 4.1: Path loss formulas for LOS and NLOS scenarios

Scenario	Shadow Fading std [dB]
LOS	$\sigma_{\text{SF}} = 4$
NLOS	$\sigma_{\text{SF}} = 6$

Table 4.2: Shadow Fading for LOS and NLOS scenarios

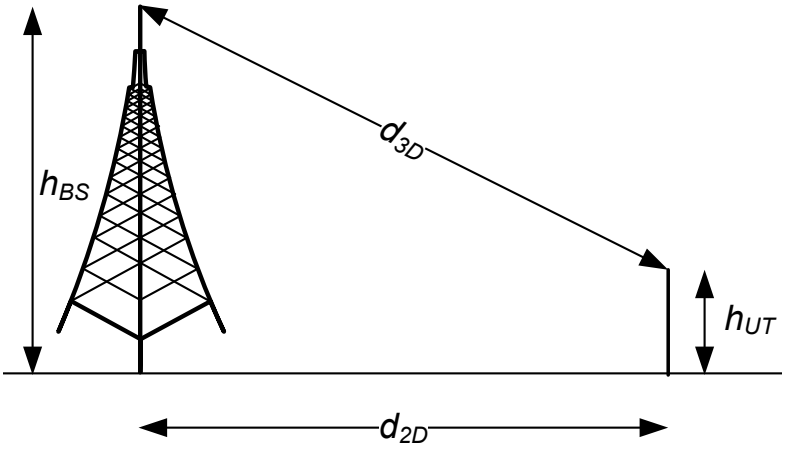

Figure 4.6: Definition of $d_{2\text{D}}$ and $d_{3\text{D}}$

random variable normally distributed, centered in $0dB$ with standard deviation σ_{SF} depending on the LOS and NLOS scenario as described in table 4.2. The final path loss PL_f will be then given by:

$$PL_f = PL + \mathcal{N}(0dB, \sigma_{SF}) \tag{4.8}$$

The final SNR value is then given by:

$$SNR[\text{dB}] = P_{\text{RX}}[\text{dBm}] - P_{\text{N}}[\text{dBm}] \tag{4.9}$$

with the received power $P_{\text{RX}}[\text{dBm}] = P_{\text{TX}}[\text{dBm}] - PL_f[\text{dBm}]$ where we set the transmitting power $P_{\text{TX}} = 30dBm$ and the noise power $P_{\text{N}} = -97dBm$. The SNR value is then converted to linear and can be used to compute the capacity of the channel as explained in sec. 3.1.2.
Moreover, we choose to implement this channel model also to simulate the V2V communication, because, in the initial phase of this project, we assumed ideal the communication between vehicles, meaning that every vehicle has perfect knowledge of the data of other vehicles without the need to exchange information. This assumption led the RL agent to always choose the lowest value of K available, 2 in our case, because having just 2 master vehicles means that

they can take the most informed choice since they can see a lot of vehicles and at the same time they would have the best channels since they can be near the BS.

The downside is that the clusters formed have a huge number of vehicles in them and they are far apart from each other, hence making it an unreal scenario. Thus, by implementing our channel model, we added a delay for the V2V communication which is proportional to the distance between vehicles that led the choice of K to be more sensible, as we show in chapter 5. It is worth noting that a more precise model for V2V communication is provided in [67], but, for our purposes, which are to introduce a realistic delay to make the choice of K more plausible, the goal can be achieved also with the current model.

4.3 Object Recognition

To implement the approach described in sec. 3.2 we need to perform the object recognition task. In our simulation, to define if an object has been recognized or not, we base our object recognition task on the VoxelNet approach [62]. In the paper, they propose a threshold $T = 45$ points to determine if the object recognition task can be performed with a good level of confidence or not.
In our case, if the vehicle, or the BS, can see at least T points of an object, that object is marked as recognized.

4.4 Data Streamline Techniques

To reduce the computation time of the CD between point clouds, we filter points by retaining only those above a given threshold and external to a specified radius. In particular, we retain all the points above $-1.5m$ coordinate of the Z-axis of the LiDAR reference frame to exclude all the road structures and all the points farther than $1m$ to exlcude the points of the vehicle itself, see fig. 4.7. With this approach the CD computation time dropped from roughly $30ms$ to roughly $8ms$.

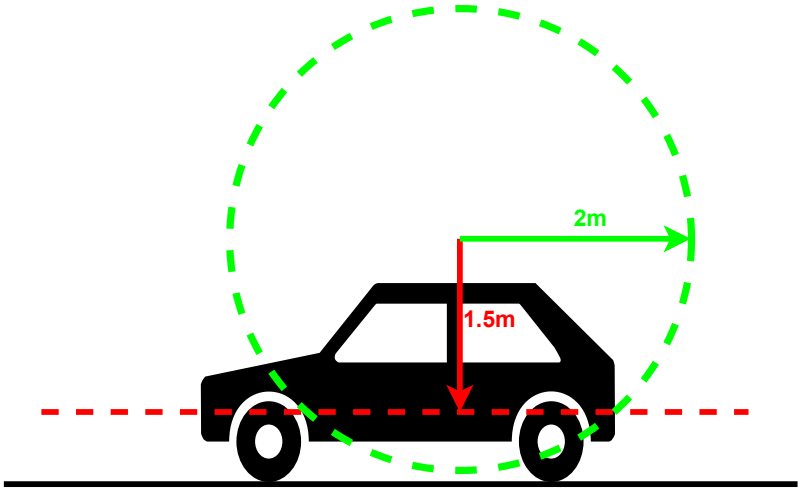

Figure 4.7: How the point clouds are sliced

5

Results Discussion

In this chapter, we analyze the performance of the algorithms presented in chapter 3. We begin by examining the outcomes under ideal conditions and gradually introduce different levels of processing delay at the application. Through simulations involving 100 smart vehicles, we compare our proposed cluster-based solutions that select the most valuable data to transmit against a benchmark scheme that does not implement any form of control. We then proceed by further exploring the impact of varying processing delays and SNR values, as well as the effect of different numbers of smart vehicles on overall system performance.

5.1 Simulation Scenario, Parameters and Metrics

The simulation scenario is the portion of a city with 200 smart vehicles in it. Moreover, it includes a total of 424 critical objects, which are cars, pedestrians and bikers. The number of critical objects is fixed and does not depend on the number of smart vehicles in the simulation. The BS is placed at the center of the city.
To understand how the algorithms perform in different conditions and to explore different trade-offs, we tune the following parameters:

- **SNR**: we apply a scaling factor to the SNR computed with the channel model described in sec. 4.2 for every channel between master vehicles and the BS, progressively decreasing it to simulate a global degradation of the

performance of the network caused by, e.g, a congestion. By doing so, we can observe how the performance changes in different channel conditions.

- **Number of smart vehicles**: we select different percentages of smart vehicles in the simulation, to understand how the performance changes with different levels of market penetration. The number of vehicles in the simulation is fixed to 200, but we consider different percentages of them to be smart. This means that we use different percentages of the available perception records and we divide the network resources accordingly.

- **Processing delay at the application**: we use different values of processing delays for the object recognition task and CD computation. We begin by evaluating the performance in the ideal case, with processing delays equal to $0ms$ for both tasks, up until the real case of $16ms$ for the object recognition task and $8ms$ for the CD computation task. We also test the behavior of the algorithms with some intermediate values, to understand what could be the differences if, in the future, better hardware and software optimization of these task, would lead to lower processing delays.

To evaluate the goodness of the algorithms, we evaluate the following performance metrics:

- **Percentage of recognized objects**: we measure the performance of the algorithms based on the percentage of objects that they are able to recognize. In this way we can observe the trade-off between having a large number of smart vehicles which can collect a lot of data, but at the same time creating a congestion on the channel, making it difficult to transmit them.

- **Ratio of recognized objects by the informed choice approaches and by the single layer approach**: we evaluate this metric to understand how better the informed choice approaches (meaning the ones that select what to transmit by processing the data using object recognition and CD) perform with respect to the single layer approach, with different values of SNR.

5.2 Comparison of Different Approaches

We begin by evaluating the performance of the various approaches discussed in chapter 3 assuming idealized processing delays for each method. The results

of a simulation involving 100 vehicles are presented in Fig. 5.1. As expected, the three algorithms that utilize a cluster-based approach, first three on the left side of the chart, outperform those that do not. This confirms that V2V connectivity and cluster-based algorithms enhance the effectiveness of cooperative perception. In particular, thanks to a communication between vehicles that are grouped together using spatial information, the perception records to transmit can be selected in a such a way to give more importance to those that are more informative for the receiver. The K-means only approach, even though does not exploit any type of V2V connectivity, is still performing better than the everyone transmits approach. This means that the spatial correlation that is exploited by the K-means algorithm and the fact that fewer vehicles try to transmit over the channel, impact positively on the performance.

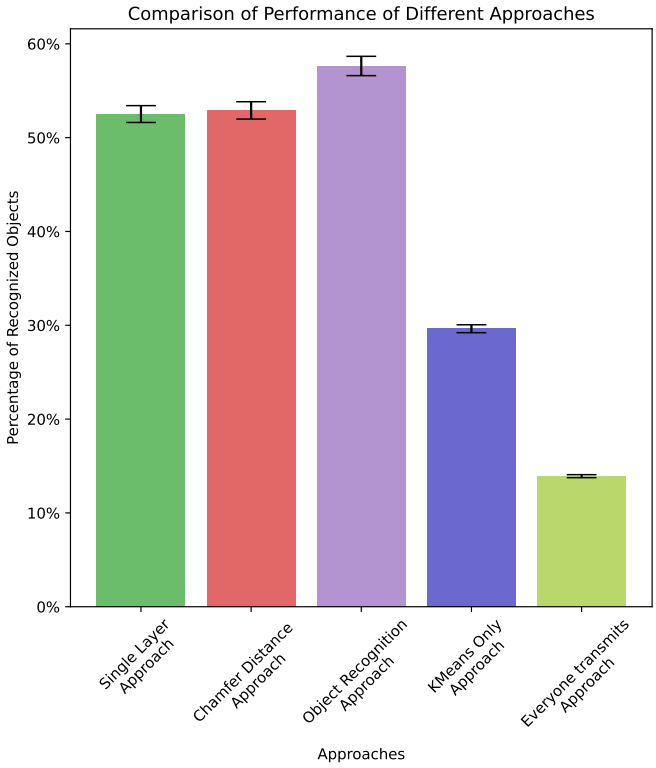

Figure 5.1: Approaches Results Comparison with 100 smart vehicles in the simulation and idealized processing delay at the application; value of K is around 20 for all the cluster based approaches.

We can see that the performance of the single layer approach and the CD approach is nearly identical. This may seem counterintuitive, as we would

expect a random selection to perform worse than an informed one (meaning that the perception records to transmit are processed and then selected in such a way to maximize the overall information), especially when the processing delay for making the informed decision is idealized at $0ms$. There are three main reasons for this:

1. When the channel conditions are good, leading to an high value of SNR, the master vehicles can transmit a large number of point clouds, and in the best-case scenario, they may not even need to make a selection, and send all of them.

2. The CD approach guarantees that we send the point clouds that are the most distinct to each other; this does not imply that they contain the highest number of objects.

3. The random selection is actually more similar to the informed choice than one might expect, as we demonstrate below.

If we have a group of n vehicles, which are the vehicles inside a cluster obtained with K-means, and the master vehicle is able to send m point clouds, it will pick m point clouds from the n available point clouds. Formally speaking the master vehicle is choosing one combination of m vehicles from $\binom{n}{m}$ possible combinations.
Then, the probability P_{ri} of picking the same combination of vehicles chosen by the informed choice with a random guess is given by:

$$P_{ri} = P[\text{random guess} = \text{informed guess}] = \frac{1}{\binom{n}{m}} \tag{5.1}$$

This probability, however, does not offer enough explanation of the problem, because, actually, there can be more than one optimal choice.
As explained in sec. 3.1.2, when we cut the dendrogram we divide the vehicles in m subclusters. At this point, we pick at random one of them per subcluster and transmit its perception record. This means that also the other vehicles in the subcluster could have been picked up, so they also have to be considered when evaluating the possible optimal choices. **We consider as optimal choice every choice that picks one vehicle for each subcluster.**
Formally, we define S_l as the number of vehicles in the *l-th* subcluster formed when cutting the dendrogram, with $1 < l < m$ and $\sum_{l=1}^{m} S_l = n$.

To compute the total number of available optimal choices we have to take the product of all S_l:

$$\#\ optimal\ choices = \prod_{l=1}^{m} S_l \tag{5.2}$$

because we account for how many possibilities we have for each subcluster. Then, the probability P_{ro} of the random guess to be the same as the optimal choice is given by:

$$P_{ro} = P[\text{random guess} = \text{optimal choice}] = \frac{\#\ optimal\ choices}{\#\ possible\ choices} = \frac{\prod_{j=1}^{m} S_j}{\binom{n}{m}} \tag{5.3}$$

We define σ_{v} as the variance of number of vehicles per subcluster and in fig. 5.2 we plot the behaviour of P_{ro} for different values of σ_{v}; in fact, the value of $\prod_{l=1}^{m} S_l$ is maximum when all S_l values are equal (in particular, when they are all equal, $S_l = \frac{n}{m} \forall l$), so each subcluster has the same number of vehicles. When this happens, $\sigma_{\mathrm{v}} = 0$, thus we can observe how the values of P_{ro} are higher in correspondence of that value. In particular we can observe that in the case of $m = 2$, hence bad channel conditions since we can send a few point clouds, the value of P_{ro} for $\sigma_{\mathrm{v}} = 0$ is higher than 55%, and, in the worst case of $\sigma_{\mathrm{v}} > 8$, remains a substantial 25%. Therefore, we can conclude that under poor channel conditions, the random choice is more likely to align with an optimal choice.

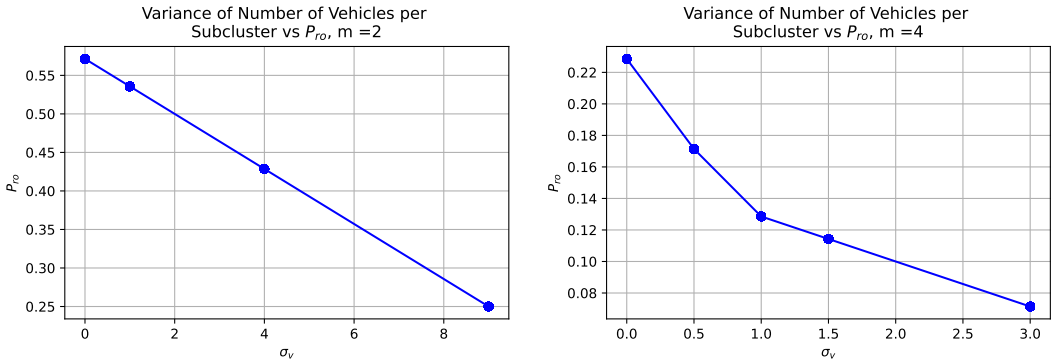

Figure 5.2: Values of P_{ro} vs σ_{v}. In both cases $n = 8$ while $m = 2$ in the plot on the left and $m = 4$ in the plot on the right.

5.3 Comparison of Performance with Different Processing Delays and SNR Values

We then proceed with further analysis on the three main approaches that utilize V2V connectivity. In particular, we test the performance of the approaches varying the processing delays, from the ideal case of $0ms$ to $8ms$ for the CD computation and $16ms$ for the object recognition task, with also some intermediate levels, considering, at first, 100 smart vehicles in the simulation.
Moreover, we vary the value of the SNR for all the vehicles in the scene by applying a scale factor to it.
In fig. 5.3 we can observe how, with a decreasing level of SNR, the performance decreases, as we expect. But, in the ideal case with processing delay set to $0ms$, we notice how the informed choices perform better compared to the random choice, enlightening how, with a degraded channel, the importance of an informed choice increases. However, when processing delays differ from $0ms$, the performance of the CD approach is consistently lower than that of the single-layer approach. In contrast, for the object recognition approach, performance remains comparable at a delay of $5ms$, but for longer delays, it is consistently worse than the single-layer approach.
To highlight even more this behavior, in fig. 5.4 we plot the ratio R_{is} given by:

$$R_{is} = \frac{\textit{Recognized Objects by the Informed Choice Approaches}}{\textit{Recognized Objects by the Single Layer Approach}} \tag{5.4}$$

and the plot actually shows how for lower levels of SNR the informed choices algorithms perform better. Even though we observe a slight increasing trend in the last step of the object recognition approach, it is only of 1.3% and can be traced back to random fluctuations given by the randomness of the simulation in terms of channel capacity, the epsilon greedy policy of the Reinforcement Learning (RL) agent and the randomness component of the K means clustering algorithms.

We then proceed by plotting in fig. 5.5 the values of K chosen by the RL agent, both in the case of idealized processing delays ($0ms$) for CD computation and object recognition task, and in the case of real processing delays ($8ms$ for CD computation and $16ms$ for object recognition). We observe, how, for the idealized processing delays, K is around 20, which appears as the best trade-off

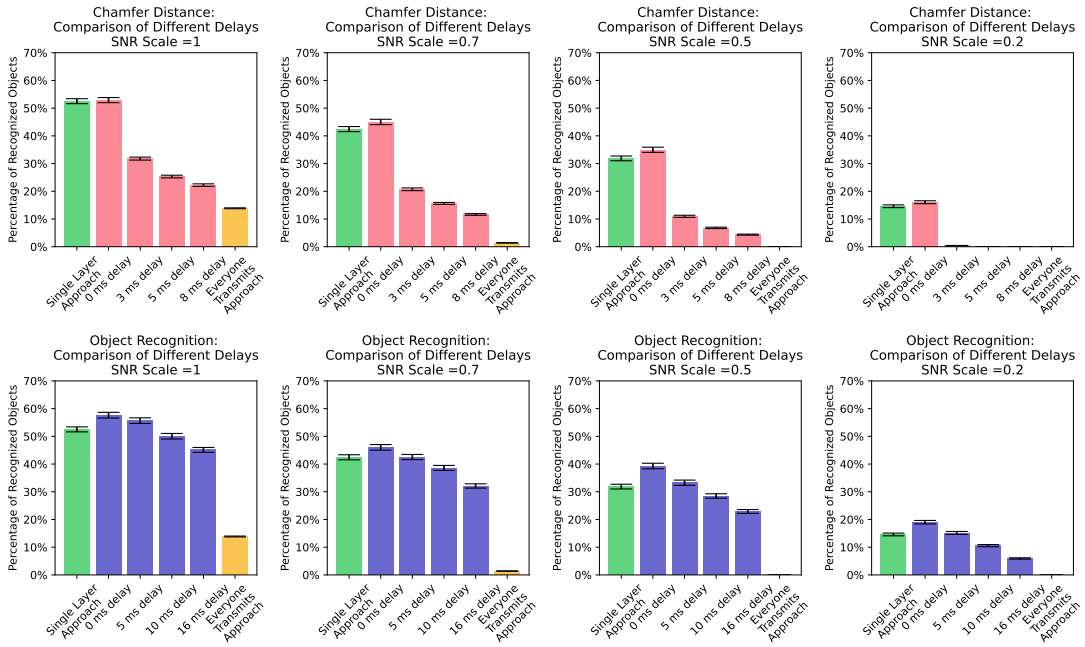

Figure 5.3: Comparison of the performance varying the SNR scale factor. Simulation with 100 smart vehicles in the scenario.

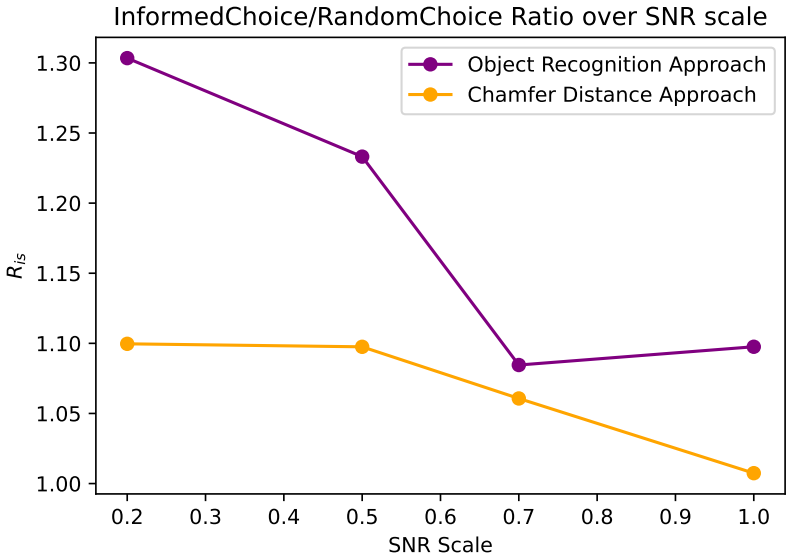

Figure 5.4: Ratio of recognized object varying the SNR scale factor. Simulation with 100 smart vehicles in the scenario. Processing delays are idealized

between a good channel capacity (the more vehicles in the cluster, the lower the capacity for sidelink transmissions), and an high performance of the informed choice algorithms (the more vehicles in the cluster, the better the informed choice

algorithms can select what to transmit). Moreover, by selecting a low value of K, the capacity of the channel between the master vehicles and the BS will be higher, since less vehicles have to transmit to it.
Meanwhile, in the non idealized case, we observe how the choice of K for the single layer and object recognition approaches does not change, since the overall processing delay is still $0ms$ for the single layer approach and, in the case of object recognition, does not vary depending on the number of vehicles in the cluster, as it does in the CD approach, as explained in sec. 3.1.2. This leads the RL agent to choose an higher value of K for the CD approach, in order to reduce the overall processing delay of CD computation between the vehicles in the same clusters.
The sparsity of chosen values that can be observed throughout the whole simulation is given by the epsilon greedy policy that we adopted which keeps selecting other values of K rather than the optimal one to encourage exploration, while during the first steps of the simulation we can observe how all the possible values of K are chosen, as a consequence of the optimistic initialization values, that encourage exploration at the beginning.

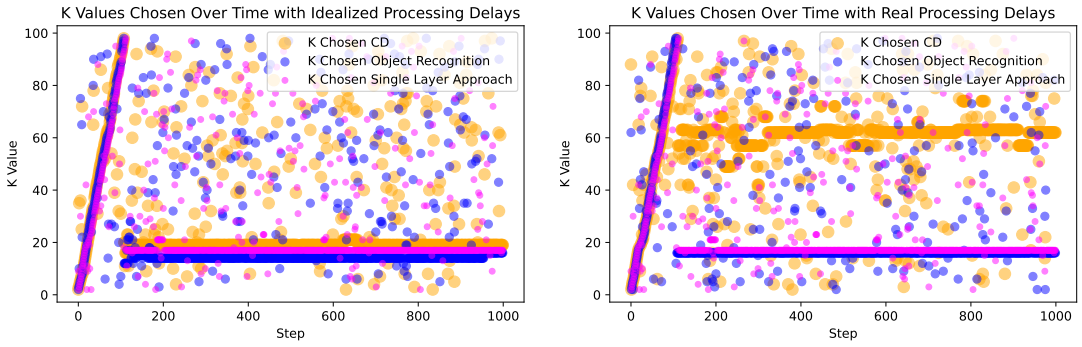

Figure 5.5: Values of K chosen by the RL agent in case of idealized processing delays (on the left) and real processing delays (on the right). 100 smart vehicles in the simulation

5.4 Comparison of Performance With Different Numbers of Smart Vehicles

We then evaluate the performance with different numbers of smart vehicles in the scene, where the smart vehicles are the ones who are actually able to

capture and transmit.
In fig. 5.6 we plot the results of the simulations for 25 smart vehicles in the scene. In fig. 5.7 we can see how the values of R_{is} defined in Eq. 5.4 are lower than the 50 smart vehicles case, thus indicating that the performance advantage with respect to the single layer approach of both the CD approach and the object recognition approach is lower when we have less vehicles in the scene. This is because, with few smart vehicles in the scene, there is a lot of bandwidth available, thus an high capacity that enables the masters of the clusters to send almost all the available point clouds, hence undermining the advantage of an informed choice. Moreover, vehicles with a bad channel and far away from the BS, won't be able to send their data regardless of the selected transmission algorithm, while with more smart vehicles in the scene they could end up in the same cluster of another vehicle that has a good channel with the BS, to which they can send their data.
In fig. 5.6 we can observe how, with respect to the case with 100 smart vehicles in the simulation, the difference between the clustering based approaches and the everyone transmits approach is lower and the impact of increasing processing delays is less evident. Both of these behaviors are given by the fact that with few smart vehicles in the scene, the capacity will likely be high, so in the everyone transmits approach there will be more bandwidth available for all the vehicles and in the cluster based approaches the processing delays will have less impact since the transmission time will be lower.

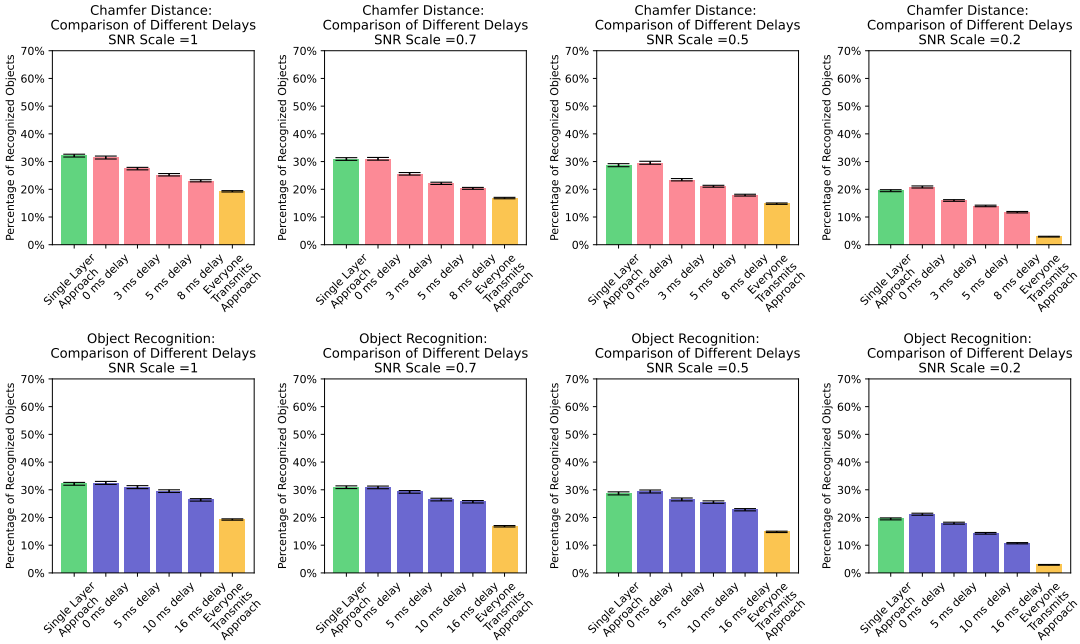

Figure 5.6: Comparison of the performance varying the SNR scale factor. Simulation with 25 smart vehicles in the scenario.

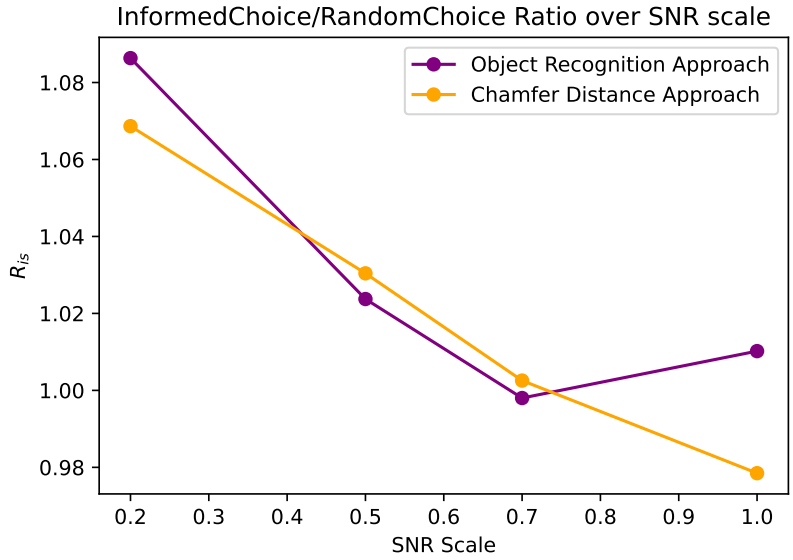

Figure 5.7: Ratio of recognized object varying the SNR scale factor. Simulation with 25 smart vehicles in the scenario. Processing delays are idealized

In fig. 5.8 and 5.9 we can see the results with 50 smart vehicles in the scene. As expected, overall performance is higher with respect to the simulation with 25 vehicles in the scene since there are more smart vehicles that can capture

more perception records and share them; the trend of the values of R_{is} is still descending as far as the object recognition approach is concerned, while for the CD approach the values fluctuate around 1.04, indicating similar performance with the single layer approach. As explained in sec. 5.2, there are three different reasons to explain this behavior.

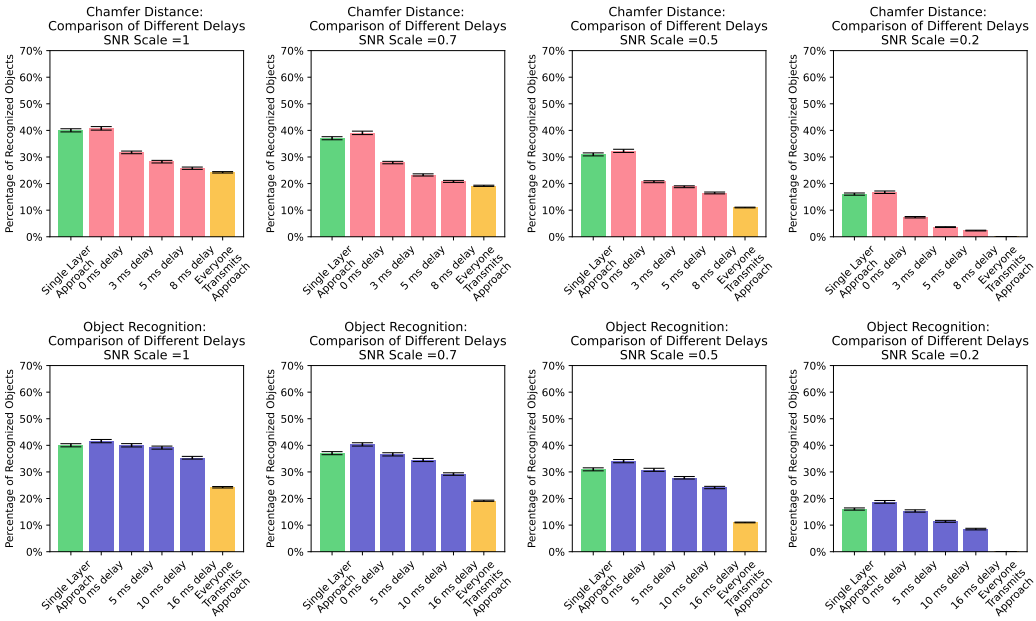

Figure 5.8: Comparison of the performance varying the SNR scale factor. Simulation with 50 smart vehicles in the scenario.

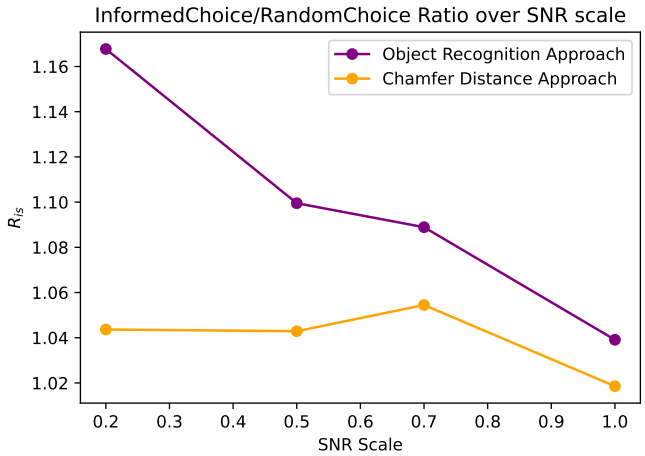

Figure 5.9: Ratio of recognized object varying the SNR scale factor. Simulation with 50 smart vehicles in the scenario. Processing delays are idealized

Finally, in 5.10 and 5.11 we plot the results of simulations with 200 smart vehicles in the scene. We can observe how, for the full channel capacity (SNR scale = 1) the performance is the highest in all the simulations, because in good channel conditions the master vehicles can transmit to the BS a lot of collected perception records. However, at the lowest capacity (SNR scale = 0.2), the performance is the lowest of all the previous scenarios, indicating how, with a degraded channel, the congestion created by the large number of smart vehicles impacts more on the overall performance.

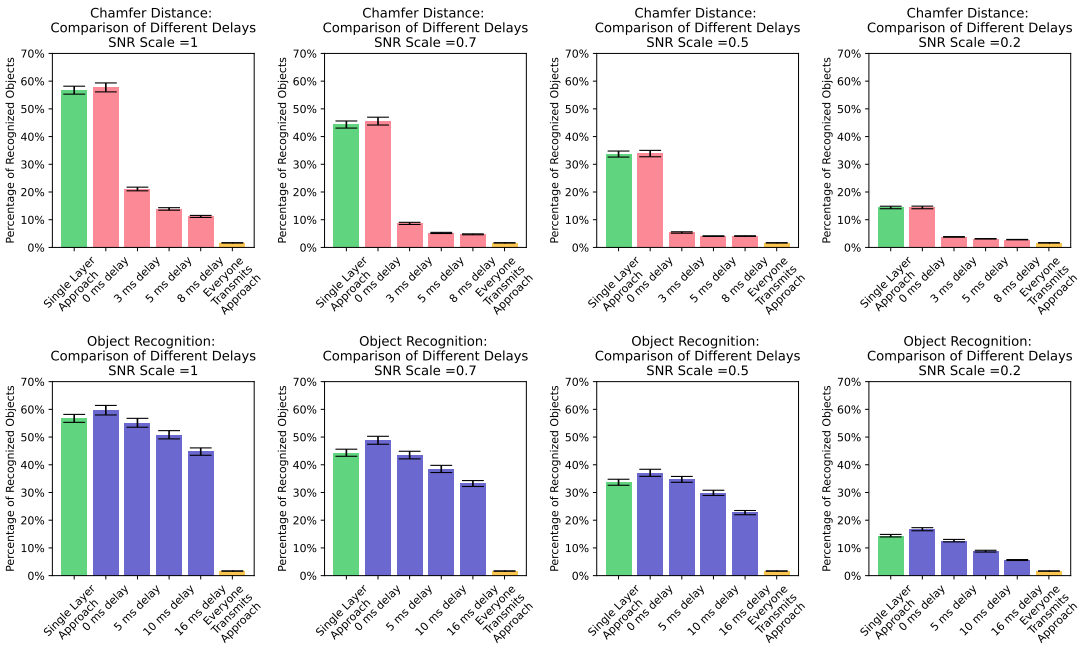

Figure 5.10: Comparison of the performance varying the SNR scale factor. Simulation with 200 smart vehicles in the scenario.

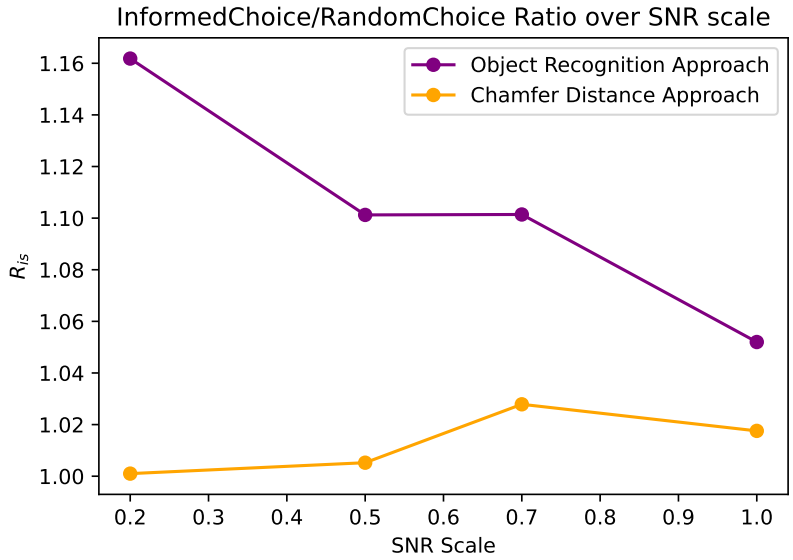

Figure 5.11: Ratio of recognized object varying the SNR scale factor. Simulation with 200 smart vehicles in the scenario. Processing delays are idealized

We enlight this behavior in fig. 5.12 where, using idealized processing delays, we observe how, when the channel conditions are good, the performance of the different approaches increases with increasing number of smart vehicles. This is because, when channel condition allows most of them to be shared, more perception records collected allow to recognize more critical objects.
However, when the channel is degraded (SNR scale = 0.2), the best performance is achieved either with the lowest number of smart vehicles or with a number of smart vehicles in the scene around 75%, because when a lot of smart vehicles try to transmit they create a congestion if the channel conditions are already bad. So with a few smart vehicles there is more bandwidth available, while, to explain the increase of performance when there are 75% smart vehicles, we have to consider that smart vehicles are deployed uniformly in the simulation area so, when we increase the number of smart vehicles in the scene, we also increase the number of smart vehicles that are closer to the BS, and experience a better channel, so can transmit more data. We can also observe the comparison with the performance of the everyone transmits approach, which, when channel conditions are bad, is close to 0% in almost all cases, except when there are few smart vehicles, so the available bandwidth is higher and allows more transmissions. When channel conditions are good, it always perform worse than the cluster-based approaches in the same channel conditions; moreover, with more than 40% smart vehicles,

performs worse than cluster-based approaches in bad channel conditions.

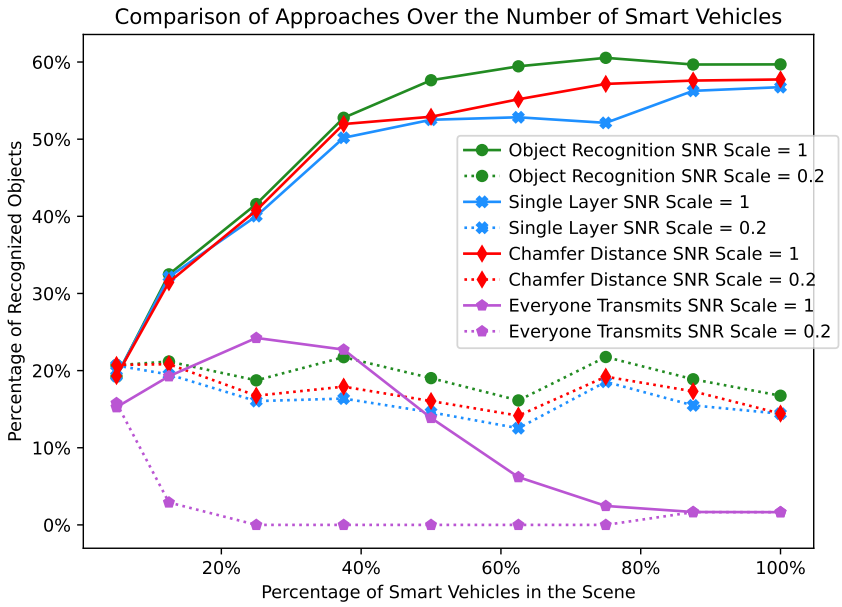

Figure 5.12: Comparison of performance with different scales of SNR values. Processing delays are idealized.

In fig. 5.13 we plot the number of recognized objects by the single layer, object recognition and everyone transmits approaches with a processing delay of $5ms$ for the object recognition task, with different number of smart vehicles, with an SNR scaling factor of 1, hence full channel capacity. With this intermediate level of delay we can observe how the performance of the two cluster-based approaches is similar up until we have 40% of smart vehicles in the scene, then the object recognition approach starts to perform better, showing how an informed choice gets more important when there are more smart vehicles, thus more information to process and select.
When we approach an even higher number of smart vehicles, up to 200 in our case, this behavior changes and the single layer approach performs better, showing how, with a very congested channel, even a small processing delay impacts the performance, thus indicating that the best choice in that case would be to use the single layer approach.
As far as the everyone transmits approach is concerned, we observe how, initially, performance increases with the number of smart vehicles involved, since there are more vehicles, so more perception records, and the channel conditions are good enough to support a reasonable amount of transmissions. Nevertheless, its

performance decreases with a number of smart vehicles in the scene higher than 25% (so approximately 50 smart vehicles), because of the congestion created by them; hence the quality of the perception record with 200 smart vehicles approaches 0%.

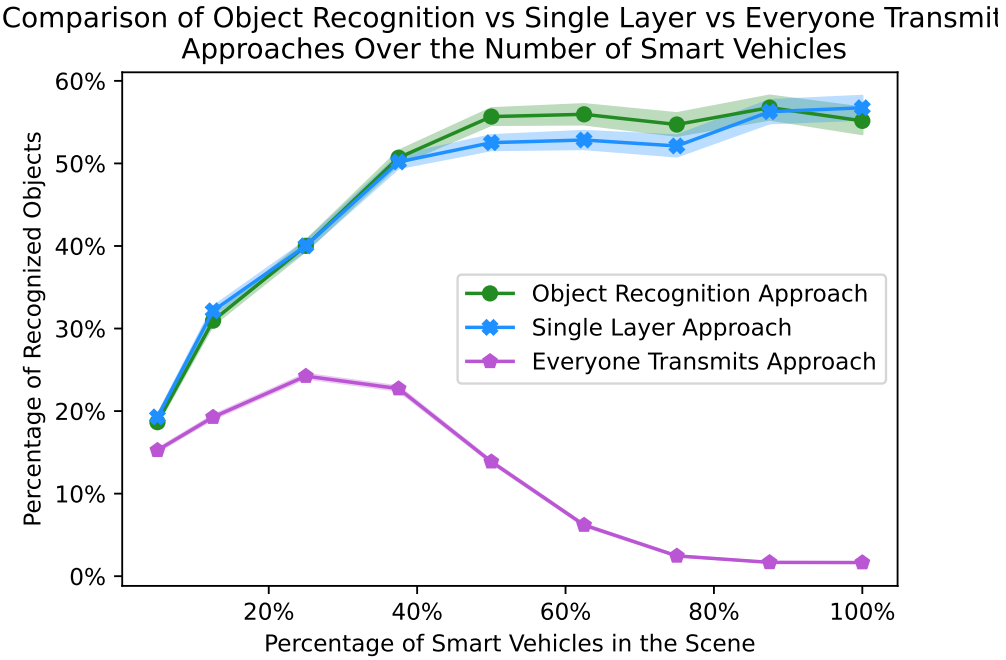

Figure 5.13: Performance of object recognition, single layer and everyone transmit approaches with different numbers of smart vehicles in the simulation. Processing delay for the object recognition task set to $5ms$

6 Conclusions and Future Works

6.1 Conclusions

This thesis explored the effectiveness of different clustering-based approaches for data dissemination in vehicular networks. By leveraging V2V communication and RL techniques, it was demonstrated that clustering-based algorithms significantly enhances the performance of cooperative perception.
We showed how cluster-based approaches using V2V connectivity outperforms those that do not, and how a k-means clustering algorithm can efficiently group vehicles into clusters where they can exchange information.
In the end, comparing the performance using real processing delays for the object recognition ad CD approaches, the best performance is obtained with the single layer approach. This shows how, in critical applications with strict requirements such as vehicular networks, the processing delays are deal-breakers that do not make an informed approach suitable for real time applications.
We demonstrated how a CD based approach does not improve the overall performance with respect to the single layer approach that selects what to transmit in a random way.
Moreover, we showed how, when processing delays are close to zero, the best informed approach is the object recognition based one. Thus, to fully exploit advanced data selection techniques like object recognition, smart vehicles must be equipped with hardware that can handle computationally intensive tasks in real time. Without this capability, the benefits of such methods are outweighed by

the processing delays, which undermines their performance in dynamic vehicular environments. In the end, we provided a solution to exploit V2V connectivity to enhance cooperative perception, which is the single layer approach, and we indicated that, when future hardware and software updates will enable a faster execution of the object recognition task, it will be the best choice to asses the VoI and select what to transmit.

6.2 Future Works

Future works for this project include:

- Implementing the whole simulation framework using a powerful network simulator such as ns3 [68], which will enable a full-stack end-to-end evaluation of the performance of the proposed approaches.
- Testing the algorithms in a different simulated city, expanding the existing SELMA dataset by generating new maps using the CARLA simulator.
- Implementing a stateful RL approach, to understand if the introduction of a state could improve the choices of the RL agent, resulting in better performance, or if the larger processing delays would make it perform worse.

References

[1] Tommaso Zugno et al. *NR V2X Communications at Millimeter Waves: An End-to-End Performance Evaluation*. 2020. arXiv: 2005.10148 [cs.NI]. URL: https://arxiv.org/abs/2005.10148.

[2] Valentina Rossi et al. *On the Role of Sensor Fusion for Object Detection in Future Vehicular Networks*. 2021. arXiv: 2104.11785 [cs.NI]. URL: https://arxiv.org/abs/2104.11785.

[3] Todd Litman. *Autonomous vehicle implementation predictions*. 2017.

[4] ATKINS. *Research on the Impacts of Connected and Autonomous Vehicles (CAVs) on Traffic Flow, Stage 2: Traffic Modelling and Analysis Technical Report, Department for Transport*. 2016.

[5] Hafiz Usman Ahmed et al. "Technology Developments and Impacts of Connected and Autonomous Vehicles: An Overview". In: *Smart Cities* 5.1 (2022), pp. 382–404. ISSN: 2624-6511. DOI: 10.3390/smartcities5010022. URL: https://www.mdpi.com/2624-6511/5/1/22.

[6] Jessica B. Cicchino. "Effectiveness of forward collision warning and autonomous emergency braking systems in reducing front-to-rear crash rates". In: *Accident Analysis and Prevention* 99 (2017), pp. 142–152. ISSN: 0001-4575. DOI: https://doi.org/10.1016/j.aap.2016.11.009. URL: https://www.sciencedirect.com/science/article/pii/S0001457516304006.

[7] I Isaksson-Hellman and M Lindman. "Evaluation of rear-end collision avoidance technologies based on real world crash data". In: *Proceedings of the Future Active Safety Technology Towards zero traffic accidents (FASTzero), Gothenburg, Sweden* (2015), pp. 9–11.

[8] TA Ranney et al. "Driver distraction research: past, present and future". In: *17th International Technical Conference of Enhanced Safety of Vehicles, Amsterdam. Recuperado el*. Vol. 29. 07. 2000, p. 2010.

[9] Michael P. Hunter et al. *Cooperative VehicleHighway Automation (CVHA) Technology: Simulation of Benefits and Operational Issues*. Tech. rep. RP 14-36. Georgia Institute of Technology. School of Civil and Environmental Engineering, Mar. 2017. URL: https://rosap.ntl.bts.gov/view/dot/36001.

[10] Kara M Kockelman et al. *Implications of connected and automated vehicles on the safety and operations of roadway networks: A final report*. 2016.

[11] Lewis M. Clements and Kara M. Kockelman. "Economic Effects of Automated Vehicles". In: *Transportation Research Record* 2606.1 (2017), pp. 106–114. DOI: 10.3141/2606-14. eprint: https://doi.org/10.3141/2606-14. URL: https://doi.org/10.3141/2606-14.

[12] Daniel J. Fagnant and Kara Kockelman. "Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations". In: *Transportation Research Part A: Policy and Practice* 77 (2015), pp. 167–181. ISSN: 0965-8564. DOI: https://doi.org/10.1016/j.tra.2015.04.003. URL: https://www.sciencedirect.com/science/article/pii/S0965856415000804.

[13] G. Silberg et al. *Self-driving cars: The next revolution*. Tech. rep. Center Automot. Res., Ann Arbor, MI, USA, KPMG LLP and Center Automot. Res., 2012.

[14] Marco Giordani et al. "A Framework to Assess Value of Information in Future Vehicular Networks". In: *Proceedings of the 1st ACM MobiHoc Workshop on Technologies, MOdels, and Protocols for Cooperative Connected Cars*. TOP-Cars '19. Catania, Italy: Association for Computing Machinery, 2019, pp. 31–36. ISBN: 9781450368070. DOI: 10.1145/3331054.3331551. URL: https://doi.org/10.1145/3331054.3331551.

[15] Bart van Arem, Cornelie J. G. van Driel, and Ruben Visser. "The Impact of Cooperative Adaptive Cruise Control on Traffic-Flow Characteristics". In: *IEEE Transactions on Intelligent Transportation Systems* 7.4 (2006), pp. 429–436. DOI: 10.1109/TITS.2006.884615.

[16] Bart Van Arem, Chris MJ Tampere, and Kerry M Malone. "Modelling traffic flows with intelligent cars and intelligent roads". In: *IEEE IV2003 Intelligent Vehicles Symposium. Proceedings (Cat. No. 03TH8683)*. IEEE. 2003, pp. 456–461.

[17] Tao Zhang. "Toward Automated Vehicle Teleoperation: Vision, Opportunities, and Challenges". In: *IEEE Internet of Things Journal* 7.12 (2020), pp. 11347–11354. DOI: 10.1109/JIOT.2020.3028766.

[18] Federico Mason et al. "A Reinforcement Learning Framework for PQoS in a Teleoperated Driving Scenario". In: *2022 IEEE Wireless Communications and Networking Conference (WCNC)*. 2022, pp. 114–119. DOI: 10.1109/WCNC51071.2022.9771590.

[19] Filippo Bragato et al. *Towards Decentralized Predictive Quality of Service in Next-Generation Vehicular Networks*. 2023. arXiv: 2302.11268 [cs.NI]. URL: https://arxiv.org/abs/2302.11268.

[20] 5GAA. "C-V2X Use Cases Volume II: Examples and Service Level Requirements". In: *White Paper* (2020).

[21] L. C. Davis. "Effect of adaptive cruise control systems on traffic flow". In: *Phys. Rev. E* 69 (6 June 2004), p. 066110. DOI: 10.1103/PhysRevE.69.066110. URL: https://link.aps.org/doi/10.1103/PhysRevE.69.066110.

[22] S. C. Calvert, W. J. Schakel, and J. W. C. van Lint. "Will Automated Vehicles Negatively Impact Traffic Flow?" In: *Journal of Advanced Transportation* 2017.1 (2017), p. 3082781. DOI: https://doi.org/10.1155/2017/3082781. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2017/3082781. URL: https://onlinelibrary.wiley.com/doi/abs/10.1155/2017/3082781.

[23] 3GPP. *Enhancement of 3GPP support for V2X scenarios (Release 15)*. 2018.

[24] Junil Choi et al. *Millimeter Wave Vehicular Communication to Support Massive Automotive Sensing*. 2016. arXiv: 1602.06456 [cs.IT]. URL: https://arxiv.org/abs/1602.06456.

[25] Marco Giordani, Andrea Zanella, and Michele Zorzi. "Millimeter wave communication in vehicular networks: Challenges and opportunities". In: *2017 6th International Conference on Modern Circuits and Systems Technologies (MOCAST)*. 2017, pp. 1–6. DOI: 10.1109/MOCAST.2017.7937682.

[26] John B. Kenney. "Dedicated Short-Range Communications (DSRC) Standards in the United States". In: *Proceedings of the IEEE* 99.7 (2011), pp. 1162–1182. DOI: 10.1109/JPROC.2011.2132790.

[27] Vitaly Petrov et al. "The Impact of Interference From the Side Lanes on mmWave/THz Band V2V Communication Systems With Directional Antennas". In: *IEEE Transactions on Vehicular Technology* 67.6 (2018), pp. 5028–5041. DOI: 10.1109/TVT.2018.2799564.

[28] Vutha Va et al. "Millimeter Wave Vehicular Communications: A Survey". In: *Found. Trends Netw.* 10.1 (June 2016), pp. 1–118. ISSN: 1554-057X. DOI: 10.1561/1300000054. URL: https://doi.org/10.1561/1300000054.

[29] Marco Giordani et al. *A Framework to Assess Value of Information in Future Vehicular Networks*. 2019. arXiv: 1905.09015 [eess.SP]. URL: https://arxiv.org/abs/1905.09015.

[30] Richard Y. Wang and Diane M. Strong. "Beyond Accuracy: What Data Quality Means to Data Consumers". In: *Journal of Management Information Systems* 12.4 (1996), pp. 5–33. DOI: 10.1080/07421222.1996.11518099. eprint: https://doi.org/10.1080/07421222.1996.11518099. URL: https://doi.org/10.1080/07421222.1996.11518099.

[31] Stefano Basagni et al. "Maximizing the value of sensed information in underwater wireless sensor networks via an autonomous underwater vehicle". In: *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*. 2014, pp. 988–996. DOI: 10.1109/INFOCOM.2014.6848028.

[32] Sanjit Kaul, Roy Yates, and Marco Gruteser. "Real-time status: How often should one update?" In: *2012 Proceedings IEEE INFOCOM*. 2012, pp. 2731–2735. DOI: 10.1109/INFCOM.2012.6195689.

[33] Enrique Mu and Milagros Pereyra-Rojas. "Understanding the Analytic Hierarchy Process". In: *Practical Decision Making: An Introduction to the Analytic Hierarchy Process (AHP) Using Super Decisions V2*. Cham: Springer International Publishing, 2017, pp. 7–22. DOI: 10.1007/978-3-319-33861-3_2. URL: https://doi.org/10.1007/978-3-319-33861-3_2.

[34] Marco Giordani et al. "Investigating Value of Information in Future Vehicular Communications". In: *2019 IEEE 2nd Connected and Automated Vehicles Symposium (CAVS)*. 2019, pp. 1–5. DOI: 10.1109/CAVS.2019.8887791.

[35] Paolo Testolina et al. *SELMA: SEmantic Large-scale Multimodal Acquisitions in Variable Weather, Daytime and Viewpoints*. 2022. arXiv: 2204.09788 [cs.CV]. URL: https://arxiv.org/abs/2204.09788.

[36] Simsangcheol. *Chamfer Distance*. 2023. URL: https://medium.com/@sim30217/chamfer-distance-4207955e8612.

[37] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. "Open3D: A Modern Library for 3D Data Processing". In: *arXiv:1801.09847* (2018).

[38] Filippo Bragato. "Design and Evaluation of Machine Learning algorithms to support Predictive Quality of Service in Vehicular Networks". MA thesis. Università degli Studi di Padova, 2022. URL: https://hdl.handle.net/20.500.12608/50761.

[39] Alex H. Lang et al. *PointPillars: Fast Encoders for Object Detection from Point Clouds*. 2019. arXiv: 1812.05784 [cs.LG]. URL: https://arxiv.org/abs/1812.05784.

[40] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[41] Chris Mahoney. *Reinforcement learning*. June 2021. URL: https://towardsdatascience.com/reinforcement-learning-fda8ff535bb6.

[42] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. 2nd ed. MIT press, 2018.

[43] 3rd Generation Partnership Project (3GPP). *Service requirements for enhanced V2X scenarios (Release 15)*. Tech. rep. TS 22.186. Version 15.0.0. 3rd Generation Partnership Project (3GPP), Sept. 2018. URL: https://www.3gpp.org/DynaReport/22186.htm.

[44] Marco Giordani et al. "Performance study of LTE and mmWave in vehicle-to-network communications". In: *2018 17th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)* (2018), pp. 1–7. URL: https://api.semanticscholar.org/CorpusID:21672391.

[45] Marco Giordani et al. "On the Feasibility of Integrating mmWave and IEEE 802.11p for V2V Communications". In: *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*. 2018, pp. 1–7. DOI: 10.1109/VTCFall.2018.8690697.

[46] Shan Zhang et al. "Vehicular Communication Networks in the Automated Driving Era". In: *IEEE Communications Magazine* 56 (Sept. 2018), pp. 26–32. DOI: 10.1109/MCOM.2018.1701171.

[47] Pei Sun et al. *Scalability in Perception for Autonomous Driving: Waymo Open Dataset*. 2020. arXiv: 1912.04838 [cs.CV]. URL: https://arxiv.org/abs/1912.04838.

[48] Marius Cordts et al. *The Cityscapes Dataset for Semantic Urban Scene Understanding*. 2016. arXiv: 1604.01685 [cs.CV]. URL: https://arxiv.org/abs/1604.01685.

[49] A Geiger et al. "Vision meets robotics: The KITTI dataset". In: *The International Journal of Robotics Research* 32.11 (2013), pp. 1231–1237. DOI: 10.1177/0278364913491297. eprint: https://doi.org/10.1177/0278364913491297. URL: https://doi.org/10.1177/0278364913491297.

[50] Jiageng Mao et al. *One Million Scenes for Autonomous Driving: ONCE Dataset*. 2021. arXiv: 2106.11037 [cs.CV]. URL: https://arxiv.org/abs/2106.11037.

[51] Adrien Gaidon et al. *Virtual Worlds as Proxy for Multi-Object Tracking Analysis*. 2016. arXiv: 1605.06457 [cs.CV]. URL: https://arxiv.org/abs/1605.06457.

[52] Marco Toldo et al. "Unsupervised Domain Adaptation in Semantic Segmentation: A Review". In: *Technologies* 8.2 (2020). ISSN: 2227-7080. DOI: 10.3390/technologies8020035. URL: https://www.mdpi.com/2227-7080/8/2/35.

[53] Francesco Barbato et al. *Latent Space Regularization for Unsupervised Domain Adaptation in Semantic Segmentation*. June 2021. DOI: 10.1109/CVPRW53098.2021.00318.

[54] Francesco Barbato et al. "A Modular System for Enhanced Robustness of Multimedia Understanding Networks via Deep Parametric Estimation". In: *Proceedings of the 15th ACM Multimedia Systems Conference*. MMSys '24. Bari, Italy: Association for Computing Machinery, 2024, pp. 190–201. ISBN: 9798400704123. DOI: 10.1145/3625468.3647623. URL: https://doi.org/10.1145/3625468.3647623.

[55] Umberto Michieli et al. *Adversarial Learning and Self-Teaching Techniques for Domain Adaptation in Semantic Segmentation*. 2020. arXiv: 1909.00781 [cs.CV]. URL: https://arxiv.org/abs/1909.00781.

[56] Alexey Dosovitskiy et al. *CARLA: An Open Urban Driving Simulator*. 2017. arXiv: 1711.03938 [cs.LG]. URL: https://arxiv.org/abs/1711.03938.

[57] Stephan R. Richter et al. *Playing for Data: Ground Truth from Computer Games*. 2016. arXiv: 1608.02192 [cs.CV]. URL: https://arxiv.org/abs/1608.02192.

[58] German Ros et al. *The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes*. June 2016. DOI: 10.1109/CVPR.2016.352.

[59] Yohann Cabon, Naila Murray, and Martin Humenberger. *Virtual KITTI 2*. 2020. arXiv: 2001.10773 [cs.CV]. URL: https://arxiv.org/abs/2001.10773.

[60] Emanuele Alberti et al. "IDDA: A Large-Scale Multi-Domain Dataset for Autonomous Driving". In: *IEEE Robotics and Automation Letters* 5.4 (Oct. 2020), pp. 5526–5533. ISSN: 2377-3774. DOI: 10.1109/lra.2020.3009075. URL: http://dx.doi.org/10.1109/LRA.2020.3009075.

[61] Bo Li. *3D Fully Convolutional Network for Vehicle Detection in Point Cloud*. 2017. arXiv: 1611.08069 [cs.CV]. URL: https://arxiv.org/abs/1611.08069.

[62] Yin Zhou and Oncel Tuzel. *VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection*. 2017. arXiv: 1711.06396 [cs.CV]. URL: https://arxiv.org/abs/1711.06396.

[63] Google. *Draco 3D Data Compression*. 2017. URL: https://github.com/google/draco.

[64] Andrea Varischio et al. "Hybrid Point Cloud Semantic Compression for Automotive Sensors: A Performance Evaluation". In: *ICC 2021 - IEEE International Conference on Communications*. 2021, pp. 1–6. DOI: 10.1109/ICC42927.2021.9500523.

[65] Rose. *Rotations in Three-Dimensions: Euler Angles and Rotation Matrices*. 2015. URL: https://danceswithcode.net/engineeringnotes/rotations_in_3d/rotations_in_3d_part1.html.

[66] 3GPP. *Study on Channel Model for Frequencies from 0.5 to 100 GHz*. 2020.

[67] Marco Giordani et al. "Path Loss Models for V2V mmWave Communication: Performance Evaluation and Open Challenges". In: *2019 IEEE 2nd Connected and Automated Vehicles Symposium (CAVS)*. 2019, pp. 1–5. DOI: 10.1109/CAVS.2019.8887792.

[68] nsnam. *ns-3, a discrete-event network simulator*. URL: https://www.nsnam.org/.

Acknowledgments

Con questa tesi concludo un percorso di studi e una parte della mia vita. Ricorderò questi anni molto intensi con piacere, perchè, alla fine del percorso, sono contento di quello che mi ha insegnato e di come mi ha formato come professionista e come persona.
Ma soprattutto, mi porterò sempre nel cuore le persone che mi hanno accompagnato, aiutato, sostenuto e incoraggiato, con le quali ho riso e scherzato e, soprattutto, imparato molto.
Ci tengo quindi a ringraziare tutti coloro che sono stati parte di questa avventura.
Comincio dalla mia famiglia, i miei sponsor, la mia mamma che mi segue ogni giorno da lassù, mio papà che con affetto mi ha sempre sostenuto e creduto in me e mio fratello, sul quale posso sempre contare per un supporto. Insieme a loro ringrazio anche il nonno Armando che mi ha sempre aiutato con i suoi preziosi consigli (e considerevoli quantità di polvere) e miei nonni Giuliano, Ada e Lisetta che ho sempre sentito molto vicini anche se adesso mi guardano dall'alto.
Ringrazio tutti i miei zii e cugini perchè mi hanno sempre fatto sapere che credevano in me e incoraggiato in questo percorso con affetto.
Ringrazio Marta, per avermi sopportato e supportato durante questa grande avventura e avermi incoraggiato nei momenti più difficili.
Voglio ora ringraziare Marco Giordani e Michele Zorzi, per avermi introdotto al loro gruppo di ricerca e per avermi dato gli strumenti e la libertà per affrontare lo sviluppo di questa tesi, per i loro preziosi consigli e la loro professionalità.
Ringrazio anche Filippo Bragato, perchè i suoi consigli sono stati fondamentali per portare avanti questo progetto, per l'impegno che ha messo nell'aiutarmi, la conoscenza che mi ha trasmesso e la grande professionalità che ha sempre dimostrato. Insieme a lui ringrazio anche tutti i ragazzi del laboratorio SIGNET

che in questi mesi mi hanno accolto e sono sempre stati disponibili per condividere un consiglio, una risata e, soprattutto, un aperitivo.
Ringrazio tutti gli amici conosciuti all'università e tutti quelli che hanno condiviso una parte del loro percorso con me, a partire dagli esami più difficili, che se affrontati insieme sembravano meno impossibili, i progetti di gruppo (ricordiamo SagrONE, presto nelle sagre di tutto il Veneto), i pomeriggi di studio in Se(R.I.P.), Ge, biblioteca Treviso e Silea ma soprattutto, i pranzi alla Piovego. Grazie perchè senza di voi, davvero, non ce l'avrei mai fatta.
Ringrazio anche tutti gli amici di una vita, i quali hanno contribuito a questo traguardo donandomi momenti di svago e confortatomi quando era necessario. Grazie quindi a tutti gli amici delle parrocchie di Frescada e Dosson per esserci sempre stati e per le esperienze che abbiamo condiviso.
Grazie infine a tutti coloro che, se stanno leggendo queste righe, oggi festeggiano questo traguardo con me. Grazie perchè, ognungo a modo proprio, avete contribuito a farmi diventare la persona che sono, e della quale posso dire di essere fiero.