

Perception enhancement of an autonomous vehicle through the use of UAV

*Amélioration de la capacité de perception d'un véhicule
autonome grâce à l'utilisation de drones*

Thèse de doctorat de l'université Paris-Saclay

École doctorale n°580 Sciences et Technologies de l'Information et de la
Communication (STIC)

Spécialité de doctorat : Informatique

Graduate School : Informatique et sciences du numérique, Référent :
Université de Versailles -Saint-Quentin-en-Yvelines

Thèse préparée dans l'unité de recherche LISV (Université Paris-Saclay, UVSQ),
sous la direction de **Amar RAMDANE-CHERIF**, professeur des universités, et
le co-encadrement de **Hongyu GUAN**, ingénieur de recherche et de **Manolo
HINA**, maître de conférence.

Thèse soutenue à Paris, le 06 juillet 2022, par

Abderraouf KHEZAZ

Composition du jury

Samir OTMANE Professeur des universités, Université d'Evry	Président du jury
Hakima CHAOUCHI Professeure des universités, Télécom Sud Paris	Rapporteuse & Examinatrice
Nicole LEVY Professeure des universités, CNAM	Rapporteuse & Examinatrice
Jaouhar FATTAHI Professeur, Université Laval	Examineur
Hongyu GUAN Ingénieur de recherche, LISV	Examineur
Manolo Dulva HINA Maître de conférence, ECE Paris	Examineur
Ravi TOMAR Professeur, University of Petroleum and Energy Studies	Examineur
Amar RAMDANE-CHERIF Professeur des universités, UVSQ	Directeur de thèse

Titre : Amélioration de la capacité de perception d'un véhicule autonome grâce à l'utilisation de drones

Mots clés : Véhicules autonomes, Drones, VLC, Base de connaissances

Résumé : Du fait de l'évolution des méthodes électronique de calcul et de communication, le concept de ville intelligente prend rapidement forme. Les nombreux capteurs et actionneurs positionnés dans les espaces habités permettent un meilleur contrôle des données qui transitent entre les usagers, et une optimisation du confort de vie. L'utilisation de véhicules reste l'un des moyens de déplacement les plus utilisés dans le monde. Là-aussi, les avancées technologiques ont permis des évolutions permettant d'avoir des voitures qui se déplacent par elle-même, avec pas ou peu d'assistance humaine. Il est cependant important de

considérer la sécurité des usagers de la route dans cet environnement de plus en plus peuplé.

Cette thèse propose d'améliorer la capacité de perception d'un véhicule autonome grâce à l'utilisation de drones. Du fait de leur positionnement avantageux et de leur taille réduite, ces derniers peuvent récolter des données de perception et les transmettre au véhicule grâce à une méthode sécurisée, par exemple une communication VLC. Les données des différentes sources sont ensuite fédérées et traitées grâce à une base de connaissance et un jeu de règles logiques.

Title : Perception Enhancement of an Autonomous Vehicle through the use of UAV

Keywords : ITS, UAV, VLC, Knowledge-base

Abstract : Due to the evolution of electronic calculation and communication methods, the concept of the intelligent city is rapidly taking shape. The numerous sensors and actuators positioned in inhabited spaces allow a better monitoring of the data that passes between users, and an optimization of living comfort.

The use of vehicles remains one of the most widely used means of travel in the world. Here too, technological advances have allowed evolution enabling cars to drive themselves, with little to no human

assistance. However, it is important to consider the safety of road users in this increasingly populated environment. This thesis proposes to improve the perception capability of an autonomous vehicle through the use of drones. Due to their advantageous positioning and small size, they can collect perception data and transmit them to the vehicle using a secure method, such as VLC communication. The data from the different sources are then federated and processed thanks to a knowledge base and a set of logical rules.

To my family

Acknowledgements

Cette thèse est l'aboutissement de trois années de travaux durant lesquelles j'ai eu la chance de rencontrer et de collaborer avec de nombreuses personnes. Voici pour moi l'occasion de leur exprimer ma gratitude.

Je tiens tout d'abord à remercier mes encadrants Manolo Hina, Hongyu Guan et Amar Ramdane-Cherif pour leur suivi constant et toutes leurs remarques constructives. Mes remerciements également à Assia Soukane pour m'avoir offert l'incroyable opportunité de faire ma thèse à l'ECE.

Tous mes remerciements vont également à Mme. Nicole Lévy et Mme. Hakima Chaouchi pour avoir accepté d'être rapporteuses de ma thèse, à M. Samir Otmane pour avoir présidé le jury, ainsi qu'à M. Jaouhar Fattahi et M. Ravi Tomar qui ont accepté d'être examinateurs de mon jury.

J'ai eu la chance de réaliser ma thèse au sein de l'ECE Paris, et je tiens à exprimer ma gratitude à tous les membres du staff que j'ai eu l'occasion de côtoyer. Mes pensées vont en particulier à mes chers collègues du cinquième étage, passé et présent, pour l'environnement de travail jovial et énergétique qui règne dans nos bureaux : Ben, Rafik, Sebti, Serena, Naila, Abderrahmane, Rola, Haïfa, Aakash, ainsi que les doctorants Aghiles, Nicolas, Maxime, Cécile et Alexandre. J'ai une gratitude toute particulière envers Thomas et Quentin pour m'avoir forcé à continuer lorsque moi-même je n'y croyais plus. De même, j'étends mes remerciements à Maxime pour toutes nos discussions passionnantes et nécessaires sur l'électronique, ainsi qu'à Eugénie pour nos bien trop nombreuses et bien trop longues pauses-déjeuners.

Ces trois années de thèse étaient chargées de tracas et d'incertitudes, et je ne serais probablement pas arrivé au bout sans le soutien de mes amis . Merci à Maxime et Mel pour tous ses films vus ensemble et sur lesquelles nous n'étions jamais d'accord. Merci à Charles pour les sessions d'escalade, les kebabs, les blagues nulles, et bien plus encore. Une pensée toute particulière à Marie, qui m'a donné l'énergie et les calories nécessaires sur la dernière ligne droite. And a heartfelt thanks to Lexi who helped me keep my sanity from the other side of the world.

Enfin, et surtout, ma gratitude la plus profonde et la plus sincère à ma famille qui a toujours été présente pour moi et m'ont toujours soutenu dans mes choix. A ma mère, mon père et ma sœur.

Contents

1	General Introduction	13
1.1	Introduction	14
1.1.1	Context and motivation	14
1.1.2	Problem statement	16
1.2	Methodology and Contribution	17
1.3	Report Outline	20
2	State of the art	23
2.1	Introduction	24
2.2	A review of the perception process	25
2.2.1	Sensors limitations	28
2.2.2	Amelioration axes	29
2.3	Use of an external sensory organ	30
2.4	Federation of data from multiple sources	34
2.5	Communication protocol	35
2.6	Proposed perception enhancement model	40
2.7	Conclusion	44
3	A knowledge-based driving context detection approach	47
3.1	Introduction	48
3.2	Definitions	48
3.2.1	Knowledge	48

3.2.2	Context	49
3.2.3	Knowledge-base and ontology	51
3.2.4	Reasoning	53
3.3	Proposed model	54
3.3.1	Dealing with missing data	54
3.3.2	Knowledge-base	68
3.3.3	Decision-making process	77
3.3.4	Summary of the process	82
3.4	Conclusion	85
4	Secured communication process	87
4.1	Introduction	88
4.2	Secured communications	88
4.2.1	Cryptography	90
4.2.2	Physical layer security	91
4.3	Communication protocols	92
4.3.1	RF communication	92
4.3.2	VLC	94
4.4	Hybrid vehicular application	99
4.4.1	Advantages of a hybrid approach	99
4.5	Typical applications	106
4.5.1	Train station use case	106
4.5.2	UAV/Car application	108
4.6	Conclusion	110
5	Use case study	111
5.1	Introduction	112
5.2	Simulator	113
5.2.1	Presentation of the simulator	113

5.2.2	Experimentation description	116
5.3	Scenarios	122
5.3.1	Intersection with a moving vehicle	122
5.3.2	Foggy area	129
5.3.3	Dark environment	134
5.3.4	Rainy weather with inadequate sensors	139
5.3.5	Complex scenario	145
5.3.6	Discussion	153
5.4	Conclusion	154
6	General conclusion and future works	157
6.1	General conclusion	158
6.2	Future works	159
A	Presentation of the knowledge-base	163
B	Résumé de thèse	175

List of Figures

1.1	Overview of the autonomous navigation process	15
1.2	Illustration of a road	15
1.3	A vehicle in a limited-perception area	16
1.4	The autonomous navigation process with improved perception	17
1.5	Methodology	18
2.1	Past and future evolution toward automated and cooperative driving. An interesting comparison can be made with the vehicular challenges of the same eras	25
2.2	Local perception radius of a vehicle	27
2.3	The 5 milestones of ADAS. The consensus is that current commercial vehicles are on Level 3	28
2.4	Schematic description of UAV usecase	29
2.5	Using UAV as communication relays	32
2.6	Illustration of UAV/Vehicle interactions [22]	33
2.7	Comparison between Camera and Photodiode	37
2.8	Illustration of the mirror-less vehicle from Jin et al.	37
2.9	Model of the scooter used in the work of Yu et al.	38
2.10	Illustration of the vehicle used in Yu et al.	39
2.11	Environment light as a noise source	40
2.12	Proposed methodology	41

2.13	Illustration of the general model	42
3.1	Data-reinforcement step in the general model	54
3.2	Defuzzification rules.	64
3.3	Defuzzification rules for sensors.	64
3.4	Fuzzy representation of weather	65
3.5	Fuzzy representation of brightness	66
3.6	Fuzzy representation of sensors	67
3.7	Knowledge-base step in the general model	68
3.8	Different types of properties	69
3.9	Hierarchy of the Sensors class	71
3.10	Relationships and properties of the Sensors class	72
3.11	Relationships and properties of the Vehicle class	73
3.12	Relationships and properties of the Communication class	74
3.13	Hierarchy of the Object class	75
3.14	Relationships and properties of the Action class	75
3.15	Decision-making step in the general model	78
3.16	Detection of weather using sensors.	79
3.17	Classification of the environment depending of the weather	79
3.18	Detection of brightness using sensors.	80
3.19	Obstructed view	80
3.20	Activation of a UAV	80
3.21	Activation of the sensors of the UAV	80
3.22	Communication protocol	81
3.23	Focus on the decision-making process	83
4.1	The communication part happens for data transfer between entities in the same environment. The protocols are also managed by the set of rules used by the reasoner.	89

4.2	Strength of the wireless signal depending of the orientation of the antenna	94
4.3	Illustration of the fire experiment	94
4.4	Steps of a VLC communication process	95
4.5	Headlight of a vehicles used for data transmission	96
4.6	Distance between the emitter and the receiver	97
4.7	Photodiode as a receiver	98
4.8	Setup of the VLC testing	100
4.9	Iperf results	100
4.10	Illustration of the proposed communication protocol	102
4.11	Illustration of the hybrid testing setup	104
4.12	Illustration of the hybrid testing setup	105
4.13	Illustration of the usecase in Ahamed work[113]	107
4.14	Illustration of the proposed model for hybrid train application.	107
4.15	Camera activation	108
4.16	Communication protocol activation	108
4.17	Example of a situation where the VLC protocol is not activated. Due to the light diffraction caused by fog, it is harder for the message to be transmitted.	109
5.1	Simulation steps in the model. The experiments actually cover all the steps, but only the main components are represented in this illustration.	112
5.2	View of the development aspect of the simulator.	113
5.3	Computed inferring time for the Java tool	114
5.4	Computed inferring time for the Python tool	115
5.5	Illustration of the technical implementation.	115
5.6	Experimentation setup	116
5.7	Interactions between different agents. The human agent controls the car, the generated driving data are stored in the knowledge base, and the even- tual important information are displayed to the user in a widget.	118
5.8	Global architecture of the simulator.	119

5.9	SWRL logical rules editor as it appears in the Protégé software.	121
5.10	Informing the driver about the distance to an obstacle	121
5.11	Environment test with an obstructed view	125
5.12	Environment test with an obstructed view	126
5.13	Avoiding obstacle in fog	128
5.14	Communication in a simple obstructed view scenario	129
5.15	Environment test with fog	130
5.16	Illustration of the foggy simulated environment	132
5.17	Avoiding obstacle in a foggy situation	133
5.18	Communication in a foggy situation	133
5.19	Environment test in a dark area	134
5.20	Environment test in a dark area	136
5.21	Presence of the obstacle in a dark environment	137
5.22	Obstructed view	138
5.23	Communication in dark environment	138
5.24	Environment test in a rainy area	139
5.25	Set of rules for the management of a rainy environment.	142
5.26	UAV agent as seen in the ontology	143
5.27	Communication protocol	144
5.29	Illustration of the complex scenario.	146
5.30	Some of the gathered data with random values dropped.	146
5.31	Data set after the completion process.	147
5.32	Set of rules for the obstructed section of the scenario	149
5.33	Set of rules for the foggy section of the scenario	150
5.34	Set of rules for the fire hazard section of the scenario	151
A.1	Example of a logical rule.	171
A.2	Individual reclassification as inferred by the reasoner. The highlighted part represent the output of the reasoner.	172

A.3	Example of a logical rule targeted to a specific individual.	172
A.4	Example of a logical rule targeted to a specific individual.	172
A.5	New property inferred by the reasoner. The highlighted part represent the output of the reasoner.	173

List of Tables

- 3.1 Environment parameters 58
- 3.2 Car parameters 58
- 3.3 Event parameters 58
- 3.4 Comparison between the completion algorithms. 61

- 4.1 Comparison of various properties of VLC and RF 99
- 4.2 Transmission speed of some VLC studies 103

- 5.1 Main classes of the ontology and their associated variables from the simulator. The virtual data generated by the experiments are sent to the ontology which classify and process them according to the declared properties and logical rules. 117
- 5.2 Elements of the first scenario as they are identified in the simulator and their matching variable in the logical rule 124
- 5.3 Results of the obstructed view driving experiment. 125
- 5.4 Elements of the second scenario as they are identified in the simulator and their matching variable in the logical rule 131
- 5.5 Results of the foggy driving experiment. 131
- 5.6 Elements of the third scenario as they are identified in the simulator and their matching variable in the logical rule 135
- 5.7 Results of the darkness environment driving experiment. 137

5.8	Elements of the fourth scenario as they are identified in the simulator and their matching variable in the logical rule	141
5.9	Results of the rainy experiment.	141
5.10	Results of the rainy experiment.	145
5.11	Elements of the scenario as they are identified in the simulator and their matching variable in the logical rules	148
5.12	Summary of the different situations with a faulty assistance.	148
5.13	Results of the complex course experiment without assistance	152
5.14	Summary of the different situations with adequate assistance.	152
5.15	Results of the complex course experiment with full assistance.	152
5.16	Feedbacks of the test subjects on the system	154
A.1	A representative example of an ontology components	163
A.2	The Vehicle Class as it appears in the ontology	165
A.3	The Sensors class as it appears in the ontology, and the different sub-classes and sub-sub-classes	166
A.4	Properties of the Sensors class	166
A.5	The Vehicle Class as it appears in the ontology	168
A.6	The Action Class as it appears in the ontology	169
A.7	The Environment Class as it appears in the ontology	170

Acronyms

ADAS Advanced Driver Assistance Systems.

BER Bit Error Rate.

DSRC Dedicated Short Range Communications.

FOV Field Of View.

FPGA Field-programmable Gate Array.

IR Infra-Red.

KNN K-Nearest Neighbour.

LED Light-Emitting Diode.

LiFi Light-Fidelity.

LOS Line Of Sight.

OWC Optical Wireless Communication.

RF Radio-Frequency.

SOC System On Chip.

SVD Singular Value Decomposition.

SWRL Semantic Web Rule Language.

UAV Unmanned Aerial Vehicle (Drone).

V2X Vehicle-To-Everything communication.

VANET Vehicular Ad-hoc Network.

VLC Visible Light Communication.

Wi-Fi Wireless Fidelity.

XML eXtensible Markup Language.

Chapter 1

General Introduction

Contents

1.1	Introduction	14
1.1.1	Context and motivation	14
1.1.2	Problem statement	16
1.2	Methodology and Contribution	17
1.3	Report Outline	20

1.1 Introduction

This chapter introduces the thesis context with four parts: context and motivation, problematic, thesis contributions, and finally, the thesis outline. The context and motivation explains the reason of this subject. The problem statement is the expression of the problem specified in the motivation. The contribution outlines how this work can be useful for the scientific community. Finally, the thesis outline details all chapters in the manuscript.

1.1.1 Context and motivation

Like many other fields, transportation technologies have seen important developments during the last century. Thanks to the popularization of computers and electronics, we have reached a level of advancement where a car can drive itself alone, without the need of a human agent.

This progress has been happening at par with the road environment, which greatly expanded too. There is an estimated 1.4 billion cars in the world[1], without considering bicycles, motorcycles, scooters, or any other types of vehicle. It is then clear that when on the road, the driver must constantly be aware of their surroundings and exercises caution. This is all the more true for an autonomous vehicle, which is most of the time evolving in an environment made up of both connected and non-connected agents. It would be then dangerous to rely on the communication grid alone (which might be compromised), and there is a necessity to have a dependable perception system.

Indeed, and as shown in figure 1.1, autonomous navigation process can be split in 4 steps [2]: Perception is the step where the vehicle uses its sensors in order to gather information about its surroundings. Localization and Mapping is the step where those data are processed in order to generate a precise knowledge of the environment it is evolving in, Decision-making is the step where an intelligence layer generates an optimal action depending of said environment, and Action is the step where the vehicle physically executes the selected decision. It is interesting to note that Perception is the very first node of the

process. This is the proof that without a good grasp of its surroundings, it is difficult, even dangerous for a vehicle to make a safe decision, and by improving it, we can improve the whole process.



Figure 1.1 – Overview of the autonomous navigation process

As an example, let us consider Figure 1.2 as a road with multiple smart vehicles and a smart building. In a connected environment, those actors can exchange information and allow for quick decision-making.

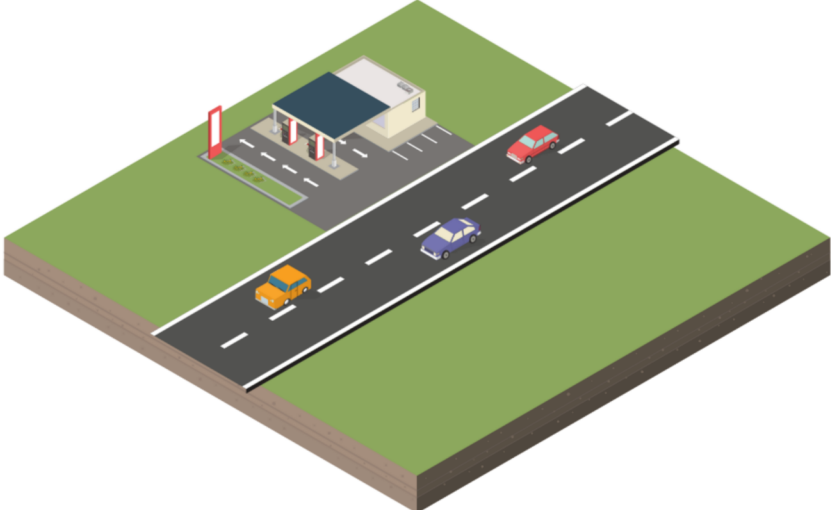


Figure 1.2 – Illustration of a road

However, in Figure 1.3, a single vehicle is in a situation where the perception is limited (in this case because of an obstacle), and cannot take a safe decision because it is lacking important data about the environment.

Hence, it is then clear that perception plays a key-role in ensuring the safety of road users: It is only when provided with enough data that we can make sure an autonomous vehicle can take the optimal choice with minimal risk of injuring passengers and other drivers. Hence the need to make sure that the perception process is reliable.

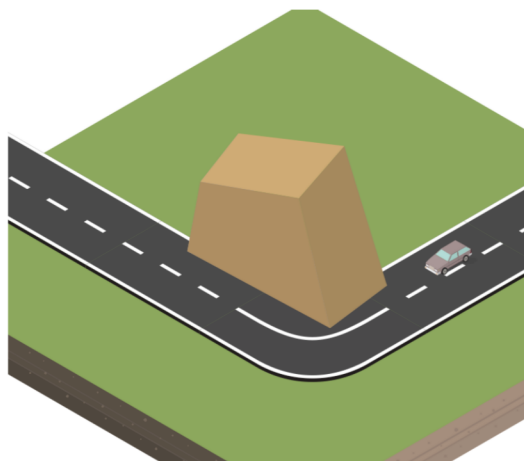


Figure 1.3 – A vehicle in a limited-perception area

1.1.2 Problem statement

Nowadays, vehicles need to evolve in a more complex environment and need to react faster. Improving their ability to sense and recognize their surroundings is a necessity to ensure the safety of road users, hence the main problematic of this thesis :

“How can we improve a vehicle ability to perceive its surroundings?”

As stated above, vehicles are evolving in a more connected environment, and it is interesting to ask if the surrounding flow of data cannot be used for the vehicle’s interest. Or even if we cannot redirect the surrounding sensors to serve the vehicle’s purpose :

“Can we use a mobile external actor to expand the vehicle’s perception range?”

However, using an element that is outside of the vehicle raises the question of data exchange. The transmitted information are of a sensitive nature, and we must also ensure that there is no security compromise during the transmission:

“How can we ensure a safe and reliable way to exchange perception data with the surroundings? ”

And finally, if the vehicle is going to use information from an external source, we must find a way to fuse them with the data gathered by the vehicle while making sure the process remains secured:

“How can we safely manage the information gathered by both local and distant sensors while maintaining their integrity and reliability? ”

Ultimately, the process shown in Figure 1.1 will be improved to look like the one in Figure 1.4: the UAV sensors will provide an external perception happening in parallel with the local perception of the vehicle, and those data will be transmitted through a secured channel. All the information are then stored in a knowledge-base from where they will be managed and processed. The additional steps shown in the figure correspond to the different contributions of this work.

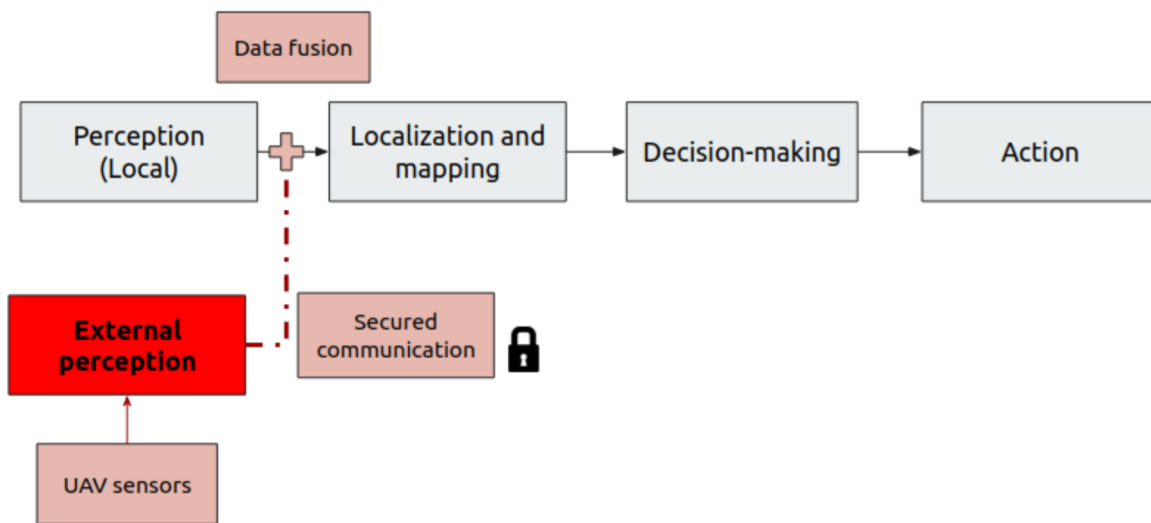


Figure 1.4 – The autonomous navigation process with improved perception

1.2 Methodology and Contribution

The goal of this work is to design a new methodology to efficiently expand the vehicle’s ability to perceive its surroundings. As stated above, we also need to consider the question of communication safety. In order to complete the perception chain, it is important to think about how the data are treated once they are safely received. The different parts of this work can be represented as a methodology in Figure 1.5.

This methodology presents the different phases of this work and their assigned chapters which are more detailed in the next section. The first step is a standard approach of

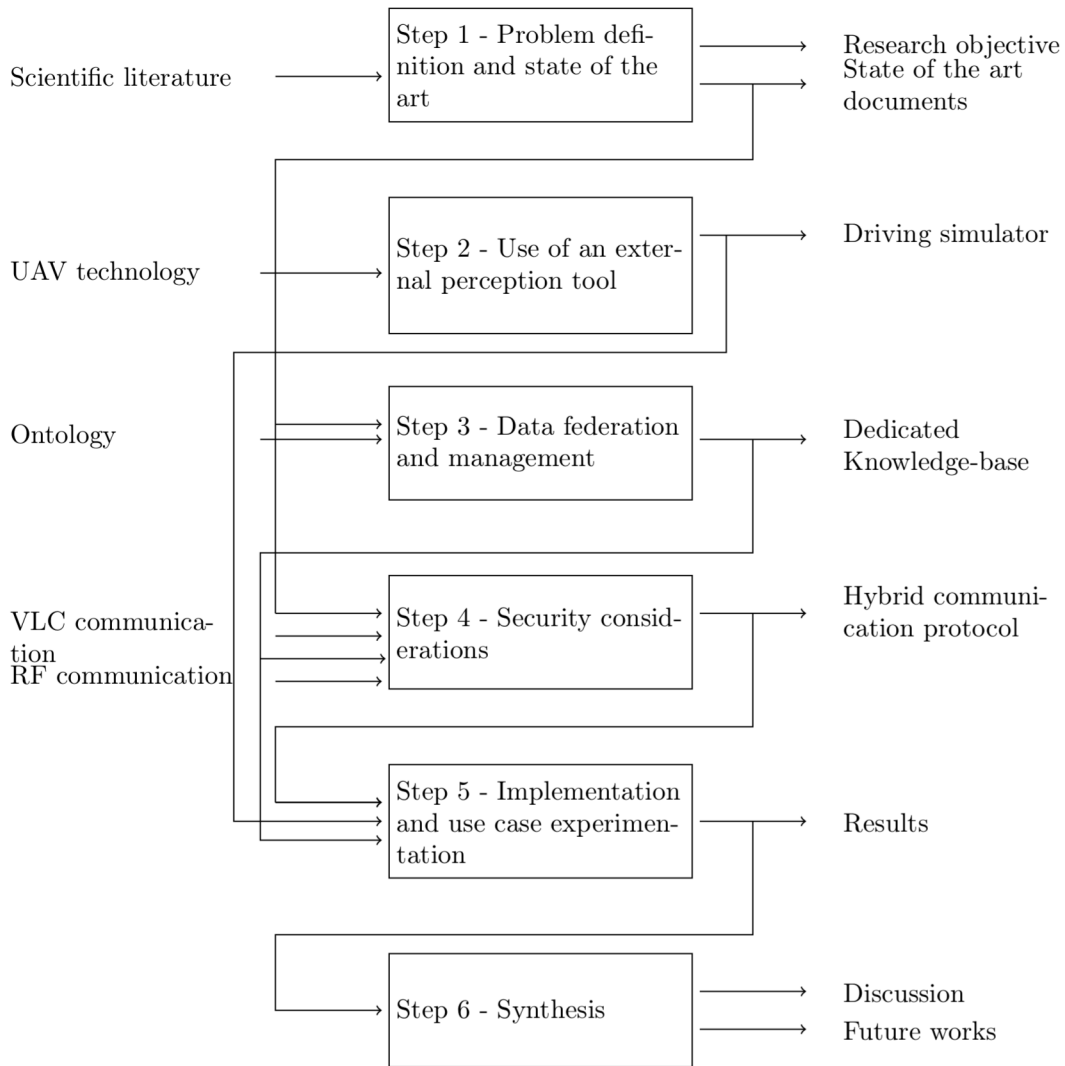


Figure 1.5 – Methodology

scientific literature exploration in specific domains in order to obtain common knowledge to find a way to resolve a problem.

The second step is about adding an external sensory tool (a drone) and using it to transmit gathered information to the vehicle.

The third step is about the setup of a knowledge-base that will act as a way to store the perceived element in both local and distant range of a vehicle, and process them through a set of logical rules for a decision-making purpose. This set of rules will also include the management of different entities and communication protocols available.

The fourth step is about ensuring the integrity of the exchanged data by building a robust communication protocol based on a hybrid VLC/RF protocol.

The fifth step is about experimenting the automated methodology on use cases to get results. This is done in a simulated environment and by having multiple candidates navigating different situations.

The sixth and last step is about a synthesis in which results are discussed and future works are presented.

This thesis proposes the perception enhancement of an autonomous vehicle through 4 different steps:

1. Identification of the correct contextual environment by gathering perception data and reinforcing them with completion algorithms.
2. The identification of the sensors working correctly in the environment inferred in step **1**.
3. The identification of the available entities (Drones) carrying the sensors chosen in step **2**.
4. The identification of the communication protocols working correctly in the environment inferred in step **1**, and that can be used with the entities chosen in step **3**.

The first step relies on neural network and fuzzy logic algorithms for the completion of the data, and logical rules for the correct identification of the environment. For the other steps, an inference engine relies on a set of logical rules in order to identify the correct elements.

The main contributions proposed from this work are the following :

- A knowledge-base dedicated to the road environment, able to store objects detected from different sources and classify them.
- A set of logical rules relying on the knowledge-base elements for the identification of the correct environment as well as the management of the different actors in a

low-perception area.

- A comparison on different completion algorithms for the road environment data, as well as a fuzzy logic approach for sensors' reliability
- A study on the different types of UAV-vehicle interactions, and a proposition of use for perception purposes
- A VLC/RF hybrid communication protocol for the protection of data integrity through the transmission redundancy on two different channels
- A driving simulator with realistic physics engine allowing the quick development of scenario for testing, and a virtual data gathering and interfacing with the knowledge-base and the reasoner.

1.3 Report Outline

In this report, Chapter 2 discusses the state of the art. It introduces a history of vehicular perception, before identifying the current limitations and axes for improvement. This work focuses on the use of an external tool in order to reinforce the perception of a vehicle, hence it includes a review on the use of UAV (drones) for vehicular applications, as well as review on the VLC protocol and knowledge-bases, respectively for safe communication and data storing and federation. The chapter concludes with a summary of the state of the art as well as a presentation of the overall proposed system.

Chapter 3 starts with the definition of the notions around the thematic of knowledge. It then introduces the works made for the management of perception data: a comparison of different ways of dealing with missing data, and the presentation of the knowledge-base developed for this work. The latter introduces the classes, objects and properties making up the ontology, as well as a set of rules that the reasoner tool uses for the inferring process. Indeed, in addition to storing the detected elements, the system also needs to

manage the environment detection, management and activation of the different actors and sensors, and the adequate communication protocols in order to ensure an optimal output: we end up with an ontology dedicated to the driving environment which can record the different road actors (vehicles, UAV, etc.) and their components (sensors and communication protocols) in order to have an accurate representation of the environment. We built the ontology progressively, by implementing each new elements added to the study, as well as its properties. We also set up a collection of logical rules which allows inference of some more information about the context, for example the general context the vehicle is evolving in (weather, brightness, etc.), as well as manage the perception elements depending on the identified environment and based on the outputs of the rules (Is the weather bad? Is there a UAV nearby? Does it hold the adequate sensors? Should we require its assistance?). The work in this chapter leads to conference papers[3] [4] and was featured in a journal [5].

Chapter 4 covers the topic of communication. It introduces different common ways of securing communications. Most vehicular communications relies on radiofrequency (RF), whose limitations are also discussed here. This work proposes the use of VLC (Visible Light Communication) for the purpose of reinforcing the process and establish a more secured communication channel, and its advantages are also presented in this chapter. In order to ensure an optimal security, this thesis proposes a hybrid RF/VLC approach with data and their hash transmitted on two different channels for data redundancy and protection goals. The heavy information are sent through VLC, and the digest of the payloads (which is smaller in size) is sent using an RF channel. The hash is re-computed on the receiver side and if a corruption is detected, it requests another transmission. This process offers data security thanks to redundancy. The communication protocols, similarly to the sensors and entities presented in the previous chapter, are also managed by the knowledge-base. They are identified in the ontology and this chapter also returns to the logical rules specifically used for communication. Some potential usecases are shown

in order to better illustrate the process (Train station and Vehicle/UAV interaction). The work in this chapter was featured in a journal paper[5] and leads to a submitted under review conference article.

The general model was tested in a simulated environment, detailed in Chapter 5. It introduces the driving simulator, developed using the Unity engine and allowing for the quick building and testing of scenarios. The driving data (speed, environment, sensor values...) are gathered in the simulation and logged in the knowledge-base. By using the Unity tool, we built multiple different scenarios varying in complexity, and we interfaced it with the previously designed ontology in order to log and process the elements that inhabit it. Multiple use cases were tested where human agents were asked to drive through defined situations, wherein a driving assistance was provided thanks to the inferred deductions of the reasoner based on the gathered driving data. The different behaviours are covered and discussed in this chapter. The work in this chapter was also featured in a journal paper[5].

And finally, the last chapter concludes this report and presents some future research perspectives brought upon by this thesis.

Chapter 2

State of the art

Contents

2.1	Introduction	24
2.2	A review of the perception process	25
2.2.1	Sensors limitations	28
2.2.2	Amelioration axes	29
2.3	Use of an external sensory organ	30
2.4	Federation of data from multiple sources	34
2.5	Communication protocol	35
2.6	Proposed perception enhancement model	40
2.7	Conclusion	44

2.1 Introduction

The gathering and management of data is a critical step for every intelligent system. Autonomous vehicles model their environment using a variety of sensors. This perception process has greatly changed in the last decades, and the related research topics have been shifting and adapting to the modern technological advances happening in parallel.

Indeed, the whole environment has been evolving, and smart cities are slowly becoming a real concept. The whole surrounding is becoming an intelligent system thanks to the presence of sensors, actuators and processing units. Another critical step of an intelligent entity is the ability to communicate: different intelligent bodies must be able to exchange information between them to guarantee comfort and security of concerned actors.

As presented in the previous chapter, this work's goal is to find a way to improve the perception abilities of a vehicle. This can be done by using an external perception tool which will transfer the gathered information to the vehicle. In order to optimize the gathering process, the additional perception source should present some advantageous features, for example an advantageous positioning. There have been various studies on the use of UAV (Drones) in a smart city environment, and they offer a strong potential for our situation. Even so, there would still be a need to communicate safely and federate the data from different sources. By its nature, the problem topic is the intersection of multiple topics, hence the need to make a scientific review of subjects seemingly independent from each other, but which will allow us to build an optimal solution.

This chapter presents the state of the art in four main areas: 1) a review of the perception process in autonomous vehicles and the different ways to optimize it 2) UAV for vehicular applications 3) The data fusion of multiple sources thanks to ontologies 4) VLC as a communication protocol for vehicles

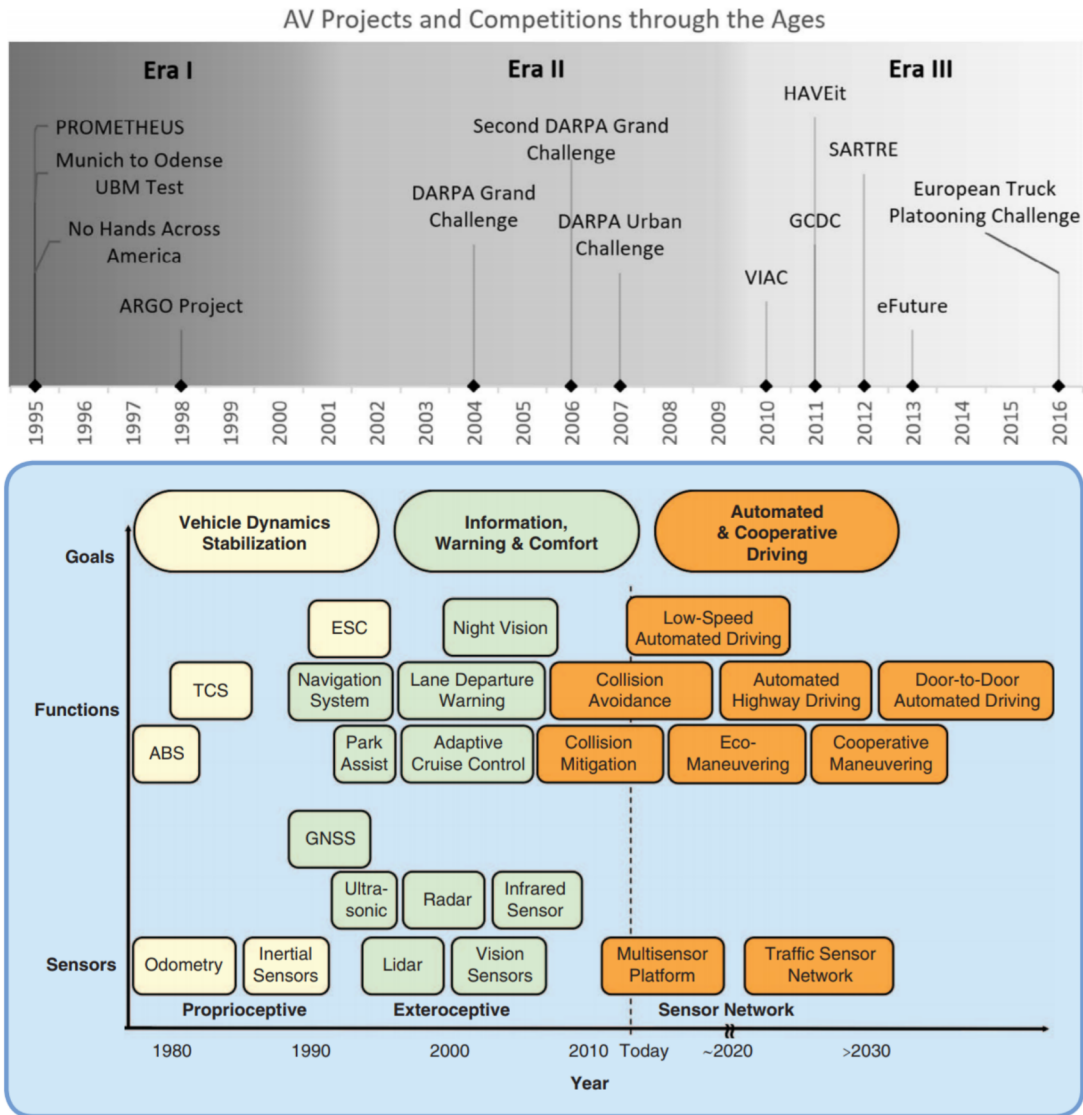


Figure 2.1 – Past and future evolution toward automated and cooperative driving. An interesting comparison can be made with the vehicular challenges of the same eras [8] [9]

2.2 A review of the perception process

Visibility plays a key role in traffic accident prevention [6]. In fact, the World Health Organization (WHO) states that "Seeing and being seen are fundamental prerequisites for the safety of all road users" and "Inadequate visibility is an important factor that influences the risk of a road crash among all types of road user" [7]. Those criteria are initially aimed at a human driver visibility but can easily be extended to an autonomous vehicle's ability to perceive its surroundings.

The ability to perceive and understand the environment is a vital part of intelligent sys-

tems, including smart vehicles. In their 2018 work, Van Brummelen et al. [9] made an extensive review of the current state of vehicular perception. They define autonomous vehicle navigation with four main components : **Perception, Localization and Mapping, Decision Making, and Vehicle Control**:

- **Perception:** using sensors in order to gather physical data about the surroundings.
- **Localization and mapping:** processing the raw data from the sensors in order to assess the status of the environment.
- **Decision-making:** using the processed data in order to infer the optimal action to take.
- **Vehicle Control:** physically executing the chosen operation thanks to the vehicle's actuators.

Perception is described as using "sensors to continuously scan and monitor the environment, similar to human vision and other senses". In order to achieve that, a considerable amounts of different sensors can be considered [9][10][11]:

- Radars have been used for decades for vehicular applications[12][13]. This technology has proved itself to be great in mid-to-long range measurement and have a great accuracy, in addition to doing well in a poor weather situation [14]. It is still heavily present in vehicles but has a small FOV (Field Of View) and show poor results in near-distance measurement and static object detection. There is also the disadvantage of receiving interference from other sources or vehicles.
- Cameras have shown an interesting potential, in both single and stereo vision. When considering the perception quality, they are the least expensive sensor that can be used [11]. They allow a quick classification of an obstacle and a potential 3D mapping of the area. Stereoscopy in particular shows very good results in detecting forms, depth, colors and velocity, although they require a substantial computational power[15]. The most advanced models can also be used for long-range precise

detection, but they have a more important cost [16]. However, its performance highly depends on the weather and brightness [14], and the required computation power can sometimes be heavy.

- LIDAR technology relies on measuring laser light reflection to infer the distance to a target. It has been studied since the 1980's[17] but it is only in the early 2000 that it has found its way in vehicular application [18][19]. It is a useful tool for 3D mapping and localization, and can be used on a large FOV [14], but it relies heavily on good weather conditions and is not efficient outside a defined range.



Figure 2.2 – Local perception radius of a vehicle

Other types of sensor can be found on vehicles, such as ultrasounds. The combination of different type of sensors allows for an efficient perception of the environment.

It is now clear that vehicular perception has known a consequent expansion in the previous decades. The first "autonomous vehicle" can be dated back to 1926, where a radio-controlled car strolled in Milwaukee, but it was followed by a human controller who still needed to watch and monitor the surroundings[20]. The study to develop fully autonomous vehicles has since been formalized and regulated, particularly thanks to the definition of the 5 ADAS (Advanced Driver-Assistance Systems) levels milestones, as shown in Figure 2.3 [21]. Those milestones cover situations where the driver is decreasingly focusing on

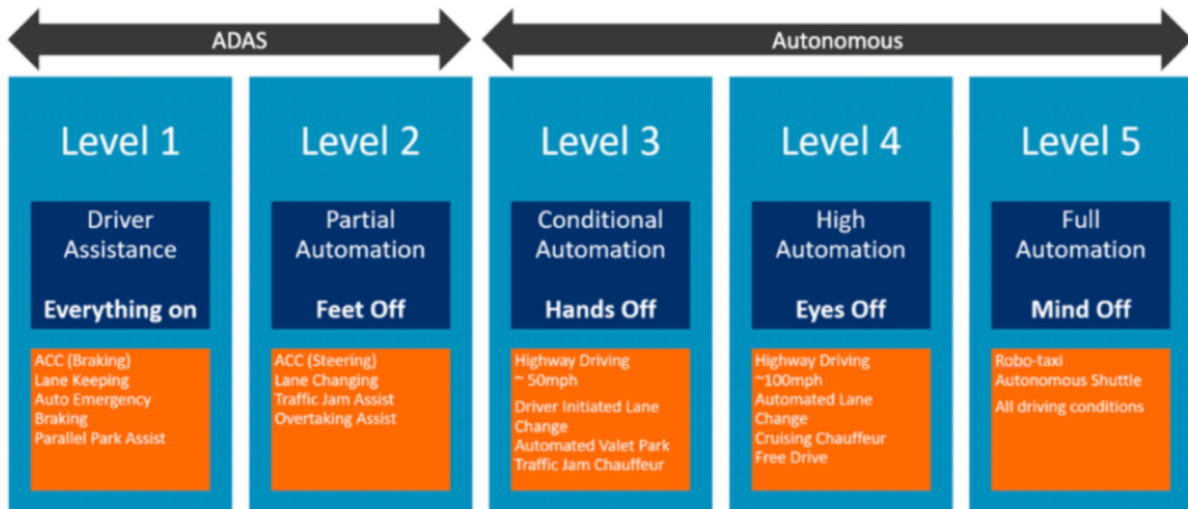


Figure 2.3 – The 5 milestones of ADAS. The consensus is that current commercial vehicles are on Level 3

the driving activity: the first level of automation corresponds to traditional driving, with no form of assistance. The second level offers assistance for the speed management of the vehicle. The third level deals with basic maneuvers like parking or self-driving in optimal situations. The fourth level takes the process even further with self-driving at higher speed and complex actions like overtaking. And finally the last level is a situation where the vehicle can fully drive itself in any situation with no assistance from a human driver.

2.2.1 Sensors limitations

No matter how reliable the sensors can be, they will still be limited by physical constraints like line of sight requirement (LOS), range limitation or other forms of interference. Furthermore, and as pointed by [9], due to limitations and high costs of available sensors, most commercial vehicles only include Level 1 to Level 2 autonomy, which require constant driver attention and control. The autonomous features in these vehicles generally consist of emergency braking, blind spot detection, and/or lane keeping.

2.2.2 Amelioration axes

Considering the current state of vehicular perception, at least five axes of amelioration can be considered [9]:

- Vehicular perception in poor weather and lighting conditions
- Vehicular perception in complex urban environments
- Autonomous driving without heavy reliance on a priori perception data
- Development of safety measures in case of faulty sensors/perception
- Utilization of connected vehicle technology to improve accuracy, certainty and reliability of perception

With the advent of the IOT (Internet of Things) and Smart Cities, more components of daily life are getting connected and interacting with each other. The last point of improvement might present an interesting potential: By requesting information from another close intelligent entity, the vehicle could significantly improve its understanding of the environment.

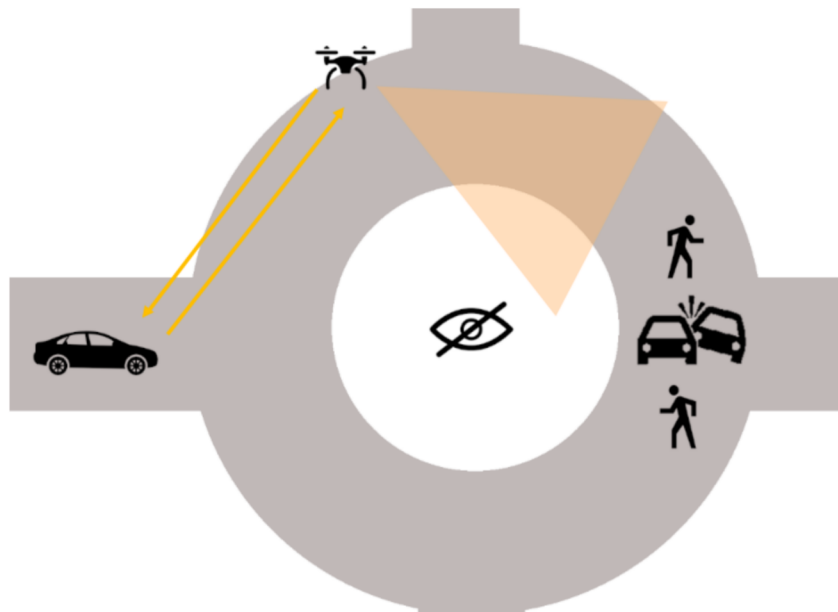


Figure 2.4 – Schematic description of UAV usecase

2.3 Use of an external sensory organ

This enhancement of the perception ability can be made through the use of UAV (Unmanned Aerial Vehicle), also referred to as Drones.

In their 2017 paper, Menouar et al.[22] submitted the idea of using UAV (Unmanned Aerial Vehicles i.e. Drones) as supporting items of ITS (Intelligent Transport System), proposing their multiple possible use. Indeed, UAV's abilities to move in a 3D space at high speed, as well as their advantageous small size that allows to transport packages while remaining smaller than cars, give them important benefit in a world where transportation is mainly 2D oriented.

For example, drones can be used for the delivery of goods. While terrestrial delivery-drone are already starting to get exploited, such as Starship Technologies that has already made more than 100.000 deliveries in 2019[23], there is still no aerial drone-delivery for the mass market. Two GAFAM (Google Amazon Facebook Amazon Microsoft) companies, Google and Amazon, are already actively working on UAV for deliveries, with their respective projects Wing and Prime Air[24][25]. Another use would be for police assistance, patrolling over roads looking out for possible offences, like speeding or improper lane usage. They might then register the car's license plate and send the information to the authorities. However, as pointed out by Sakiyama in her PhD. Thesis, there might currently be strong legal and social opposition for this specific use of UAV [26].

The strength of UAV remains its ability to access spaces that vehicles or people cannot access easily. They show a strong potential in case of dangerous events, such as a leak of dangerous liquid or gas, industrial incidents or terrorist attacks. Back in 2011, *Daniel et al.* already proposed the idea of UAS (Unmanned Aerial Systems) for homeland security, deploying a fleet of drones to distribute mobile sensors in incident areas. They define the main requirements for rescue-oriented UAV as follow:

- **Low-cost:** Police departments and fire brigades' financial resources are usually

very limited

- **High Throughput:** Ideally, the drone should be able to livestream both audio and video feeds, both requiring a high transmission speed to guarantee real-time reaction
- **Operating Time:** A UAV should have enough autonomy to ensure reaching the place of incident and staying on it until human operators arrive[27]
- **Frequency Ownership:** Spectrum Management is a real problem nowadays with a lot of the saturated bands. When controlling a drone through RF, we must ensure that there is no interference in case of incident.
- **Payload weight:** The UAV is not expected to carry heavy charges. It could eventually be used to transport small things (i.e. first aid kits) but we favor speed and autonomy over pulling strength.

Given their aerial position, UAVs have a good view of events happening on a road, in addition to having less impact on the observed traffic. When equipped with a powerful enough camera, it can easily detect and identify moving cars. In 2017, Xu et al. [28] trained a Convolutional Neural Network (CNN) to find and follow cars from a height of 100m to 150m. Their neural networks showed a car detection success of 98.43%, with a robustness to illumination changing and moving vehicles - allowing to use this algorithm with both a static and moving UAV. The training time of the algorithm was about 21 hours, a decent time for this kind of applications, but the training was only made on car detection, neglecting other road users (buses, pedestrians,etc.)

So far, UAV have shown great potential in a vehicular environment. If we want to build a viable model, we need to set up a communication link between the drones and the other actors of the vehicular environment. The integration of UAV in V2X protocol has already been investigated by many researchers before : In 2018, Hadiwardoyo et al. [29] managed

to establish a communication link between a stationary drone and a moving vehicle. They experimented on many parameters, such as the height of the drone and the orientation of the transmitting antenna (horizontal or vertical) and made real tests on an empty road. They managed to reach a transmitting ratio between 60% and 98% transmitting over a maximum distance of 2.5km. They obtained their best results by pointing the antennas to the vertical and positioning the drone at a 100m, the highest distance allowed by European laws being 120m.

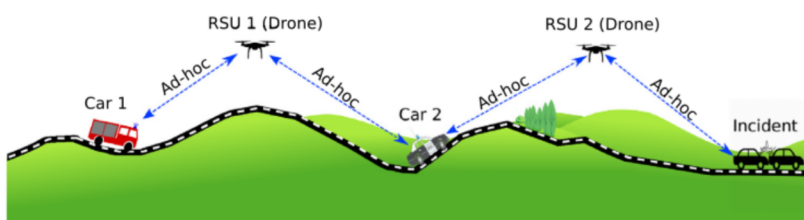


Figure 2.5 – Using UAV as communication relays [29]

V2X communication is sometimes referred to as VANET (Vehicular Ad-hoc NETWORK), the vehicular application of MANET (Mobile Ad-hoc NETWORK) principles. MANET are described as being "a continuously self-configuring, infrastructure-less network of mobile devices connected wirelessly", meaning that there is no router in the network and the nodes are used for routing instead, despite being in motion and connecting to different other nodes over time. In the case of VANETs, the nodes of the network are mostly vehicles, but it can also be other infrastructures, such as RSU (RoadSide Unit) or even pedestrians. Studies have already been made focusing on the use of UAV as network's nodes. In their review of applications and challenges of UAV in Smart Cities, Menour et al. [30] already introduced the idea of UAV being part of a larger network. They proposed a deployment method based on the CDS (Connected Dominating Set) graph theory algorithm, that ensures that all nodes of the same area are always connected by constraining the movements of the UAV (a UAV that goes too far might break the connection with its closest nodes). They then could be used as hot-spots that can at the same time be used for monitoring the traffic and as a device for routing informations in the network. Shi et

al. [31] made a similar study focusing on the throughput and reached a speed of 1 to 2 MB/s in their simulations with UAV, comparing a regular 802.11p car-only communications with a 2.4GHz DSRC (Dedicated Short Range Communications) with a swarm of drones. They also made an interesting remark on the quality of the service by pointing out that a higher vehicle density generates a higher delay between messages.

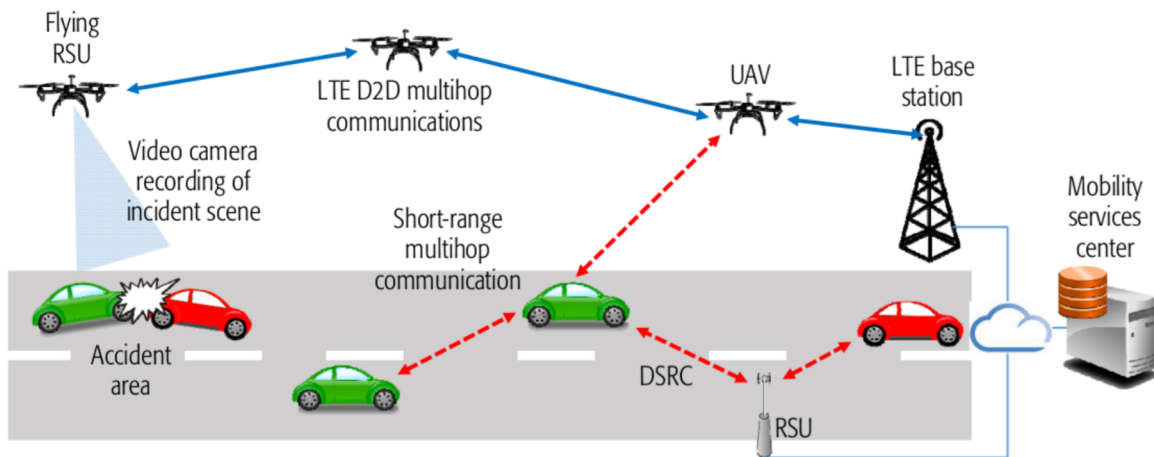


Figure 2.6 – Illustration of UAV/Vehicle interactions [22]

When studying the deployment of a swarm of UAV, one of the major constraints that stood out is power supply. Currently, most drones have a battery lifetime of 20 to 30 minutes, and the longer they stay on, the less performant they become. Galkin et DaSilva [27] addressed this issue in their 2018 study and proposed 3 battery management solutions.

- **UAV swapping:** Having many drones to use and cycling between them. When the active one is getting low in battery, it returns to the station to recharge and another one takes its place. This solution requires having an important swarm of drones
- **Battery hot-swapping:** A single drone is used and when its battery is getting low, it returns to the station where an automatic infrastructure quickly changes its battery with a fully-charged one. However, this means that the network would be lacking a node for the duration of the battery swap, and this can have critical repercussions

- **Wireless power transfer:** In 2018, Ouyang et al. submitted a model that would technically allow to power an UAV with a high-power laser. Unfortunately, this solution has not been tested in real conditions yet, and would require an almost permanent LOS (Line Of Sight)

The proposed solutions are interesting, but they each have their own constraints. This could be overcome by combining some of them together, like having a dedicated UAV that would just serve as an energy refill and make short-time travel to the other UAV that serve as nodes to recharge them quickly with a laser, since hot-swapping and a direct physical connections would be dangerous or difficult to implement.

All those considerations highlight the potential of UAV as a dynamic data gathering tool that can provide a useful service to intelligent vehicles. There will then be multiple feeds of data that need to be stored and processed to allow perception enhancement

2.4 Federation of data from multiple sources

If data are gathered from multiple sensors and multiple sources, it is also important to consider a way of merging and managing them.

There are a multitude of reasons that could lead to performance issues, for example the accumulation of error over time. Through the combination and association of sensing methods, it is possible to overcome the weaknesses of individual components.

In a broader sense, sensor fusion is considered as the *"process of managing and handling data and information coming from several types of sources in order to improve some specific criteria and data aspects for decision tasks."* [2]. Thanks to the redundancy and complementarity of information, the obtained perception is optimized in order to guarantee the best decision-making. It is a method generally applied to sensors embedded on a single body, but in a smart city environment, this could also concern sensors from

different entities.

Some studies have taken an ontological approach to this solution, as shown by the review work of Bendadouche et al. [32]. For example, Calder et al.[33] used a reasoning approach in order to validate the behaviour of multiple sensors in a coastal ecosystem. Through the use of logical rules, they tried to infer if a sensor is functioning properly : Did the sensor log a measurement? Was it done at the correct time? Is the registered value in an acceptable range?

Compton et al. also worked on a sensors-dedicated ontology [34]. They aimed to make a model abstracted enough to allow it. This consequent work would then allow an easy way of both adding new sensors, and reading the gathered value. This work, as well as that of Calder et al.[33] was later on merged in the SNN (Semantic Sensor Network) ontology [35], which is described as an *"ontology for describing sensors and their observations, the involved procedures, the studied features of interest, the samples used to do so, and the observed properties, as well as actuators"*.

Those cited works highlight the necessity of federating the data from different sources in a common base, and classify and treat them in the correct way. Knowledge-bases, especially ontologies, seem to be an adequate candidate for this kind of task.

2.5 Communication protocol

Previous sections introduced the idea of gathering extra data from an UAV sensors and merging them with the ones from the vehicles. This process would imply that some form of communication exists between both entities, and we must ensure the safety of the communication process.

VLC is an Optical Wireless Communication (OWC) technology where the signal is produced by a light source and the data are transmitted over the visible light spectrum. The transmitter can either be a LED or a Laser Diode (LD). It has been around for a while

now, the first practical use being that of Alexander Graham Bell in 1880, who invented the photophone : By vibrating a mirror at a specific frequency, he managed to modulate a voice signal and received it over 200m away with a parabolic receiver.

The invention of Laser (Light amplification by stimulated emission of radiation) enabled a breakthrough in VLC communication : US scientists in the 1960's managed to transmit a voice signal modulated by laser over 190km. Nowadays, Japanese road networks use IR (Infrared) devices to send traffic information directly to the vehicles: There are more than 30000 IR beacons on the Japanese roads in a system called Vehicle Information and Communication Systems (VICS)[36]. While VICS is not technically VLC, it proved that we could use near-IR signals for V2X communications[37].

In 2001, Akanegawa et al. [38] suggested to extend the VICS system to LED traffic light. Whilst they focused on downlink and one-way communication, their work is often qualified as the first real VLC V2X study because they focused on the real qualities of VLC and proposed a theoretically functional model.

In his PhD thesis, Bechadergue [39] focused on VLC in V2X communications and made an in-depth comparative study between the use of Photodiode (PD) and cameras as VLC receivers. As shown in Figure 2.7, the high-speed camera is better on every aspects, save the data rate. However, in the general context of the work, it was deemed more interesting to select a PD instead of a camera. Considering the linear and short-range transmission in a platooning system, it makes sense to use a PD because of its narrow FOV and high sensitivity.

In 2016, Goto et al.[40] created a new type of Optical Communication Image sensor (OCI). This innovative high-speed camera model allows for the acquisition and treatment of data with a much higher speed than what was previously used (up to 55mb/s). It has also been successfully used in other experiments, such as the one conducted by Yamazato et al. in 2017 [41]: Yamazato and his team built a 32x32 LED array that modulates specific data, and an OCI-equipped vehicle is driving towards it. The team managed to clearly receive

Characteristic	Photodiode	Camera	
		Low-speed	High-speed
Data rate	> 100 kbps	< 1 kbps	~10 kbps
Range	< 100 m	> 100 m	> 100 m
Coverage	Medium	Wide	Wide
Mobility	Limited	Good	Good
Spatial distinction	No	Yes	Yes
Sensitivity	High	Low	Low
Cost	Low	Low	High

Figure 2.7 – Comparison between Camera and Photodiode



Figure 2.8 – Illustration of the mirror-less vehicle from Jin et al.

and read the transmitted data regardless of the vibrations due to the car displacement, and could also define the relative position of the vehicle from the LED array with an error of about 0.5m for a distance of 30 to 60m.

In Jin et al.[42], a South Korean team from the Hoseo University tried to use the already-present cameras in mirrorless cars as VLC receivers. Mirror-less cars are a new type of concept car that replace rear-view and side-view mirrors with camera, hence granting the driver a better FOV (Field of view) than with classical mirrors. This experiment was conducted on a classical cars by positioning cameras near side-view mirrors, and using daylight running lamps (DRL) of the following vehicle to transmit information, as shown in Figure 2.8. The following car can transmit 2 kinds of information : the type of vehicle and the kind of action it will execute.

In 2013, Yu et al.[43]. were amongst the first to make VLC experiments in a dynamic context (with a moving vehicle). Their studies was aimed at comparing VLC and DSRC

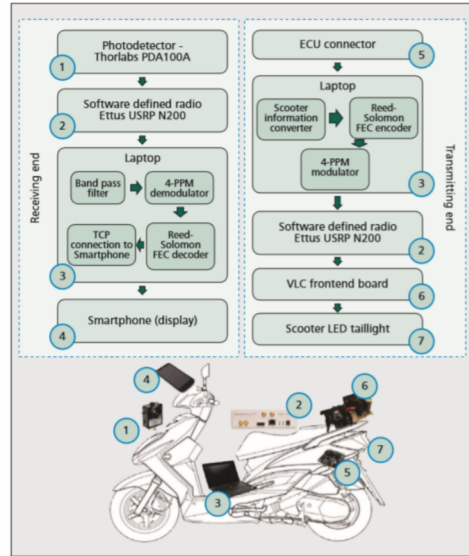


Figure 2.9 – Model of the scooter used in the work of Yu et al.

(dedicated short-range communications) technologies in a V2X context. DSRC uses radio frequency for short-to-medium range communications. They built a VLC receiver on a scooter and made tests at a varying speed from 10km/h to 40km/h. DSRC seems to be better performing when having a vehicle communicating with its neighbors, but the mutual interference is important, and the complexity grows with the distance between the vehicles. Figure 2.9 describes the different modules mounted on the vehicle. Also according to the same study, weather can have a strong impact on VLC communications. VLC still showed good results. The FOV might be too narrow but in an overtaking scenario the information were transmitted at a decent speed. The size and speed of a scooter, and the ease of setting-up VLC tools on it, make it a good environment to run dynamic tests.

In 2012 Cui et al. [44] made a study on how weather conditions and environment could affect VLC communication. They isolated multiple source of noises, such as the sun and neon lights on the streets (Figure 2.11) They then used electrical Passband Bandwith Filters as well as plastic to make sure that the photodiode only reads from the stoplight's LEDs.

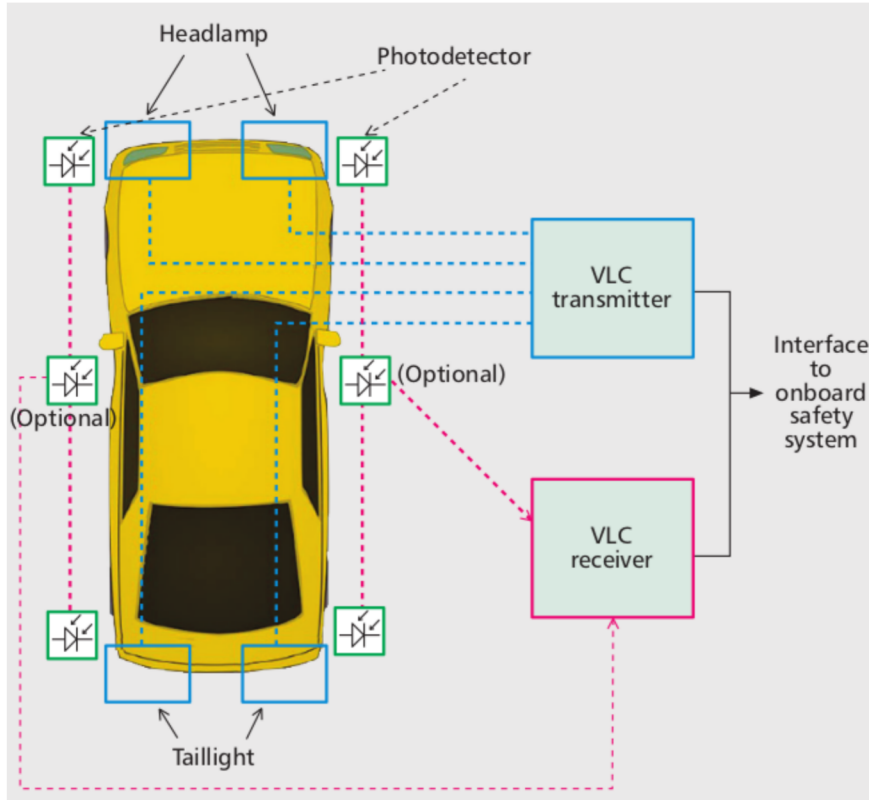


Figure 2.10 – Illustration of the vehicle used in Yu et al. [43]

They also showed that weather can have a strong impact on VLC communications [43][44]. An in-depth study has been led by Kim et al. in 2015 [45], mostly focusing on fog. In a controlled environment, they tried to establish a VLC communication using LEDs of different colors and fog of varying thickness. The results showed that red light passes through fog more easily than green or blue lights. They paired it with a Fresnel lens to focus the incoming light and saw an improvement in the overall data transmission.

VLC has the advantage of being immune to RF interference, but there are still many sources of noise that can disturb the communications: working within the visible light bandwidth, anything that can be visually perceived by the human eye is a potential source of noise. This can be a real problem in V2X communications, considering that both the emitter and receiver are moving and being outdoor, which means being exposed to multiple sources of lightning.

Multiple solutions have been proposed over the years. Most of them are for indoor applications but some can be exported to an outdoor environment. One of the easiest and most

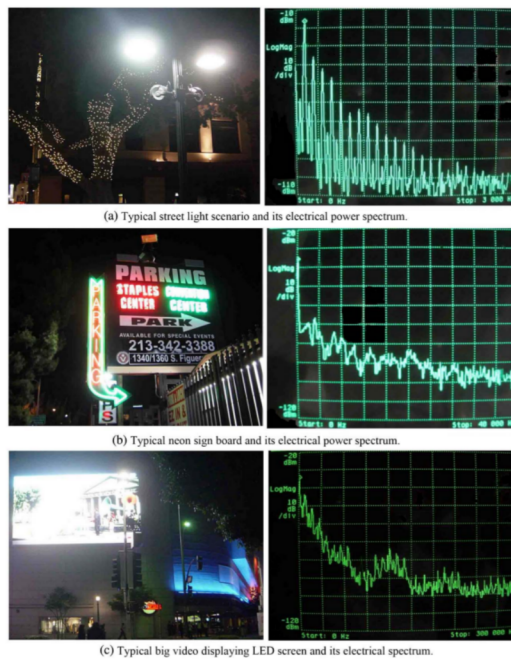


Figure 2.11 – Environment light as a noise source

common way is to use hardware analog filters [46] [47]. Cheap and small analog filters have shown good performances in indoor environments and have potential for outdoor environment too.

Software filtering has also been investigated. Since 2013, Paul Antony Haigh and his teams have been working on the use of Neural Networks and Artificial intelligence for the equalization of signals transmitted through VLC[48][49][50][51][52].

Considering all the works made on VLC and specifically in vehicular application, it is then clear that there is a real interest for us for its use.

2.6 Proposed perception enhancement model

As shown in Chapter 1, the core problem of this thesis is "**How can we improve a vehicle ability to perceive its surroundings ?**" The previous sections presented a literature review of vehicular perception as well as the current state of the art on UAV/Vehicle interactions, VLC communications and knowledge-base for vehicular applications.

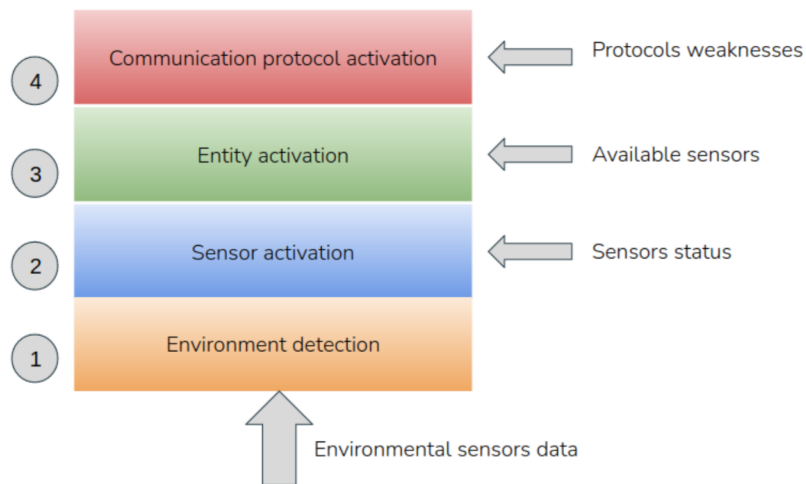


Figure 2.12 – Proposed methodology

We believe that those elements allow for the formulation of a solution to this question. Indeed, thanks to their embedded sensors and their advantageous positioning, UAVs can gather information on events happening on the road and transmit them to the vehicle through a secured means of communication, which can be comprised of VLC. The data from the vehicle's and UAV's sensors are then gathered in the ontology and processed thanks to the numerous logical rules implemented in the knowledge-base. An illustration of the process can be found in Figure 2.12.

Even in a harsh environment where the UAV's perception is also limited, its position and small size would still allow it to gather data without endangering road users, making it an interesting candidate for this process.

The general model proposed in this work is described in Figure 2.13:

1. The actions made by the vehicle (either through automatic driving or manual control from the driver) will have a direct impact on the environment where it is evolving and by moving around, the perceived surroundings will also change (1).
2. The Environment (2) is the context where the vehicle evolves (e.g rainy road, dark area, etc.) . The sensors will allow a representation of this context and will generate perception data (3 and 4) reflecting the different actors and elements of the road.
3. The perception data can be divided in two categories: surroundings (3) and envi-

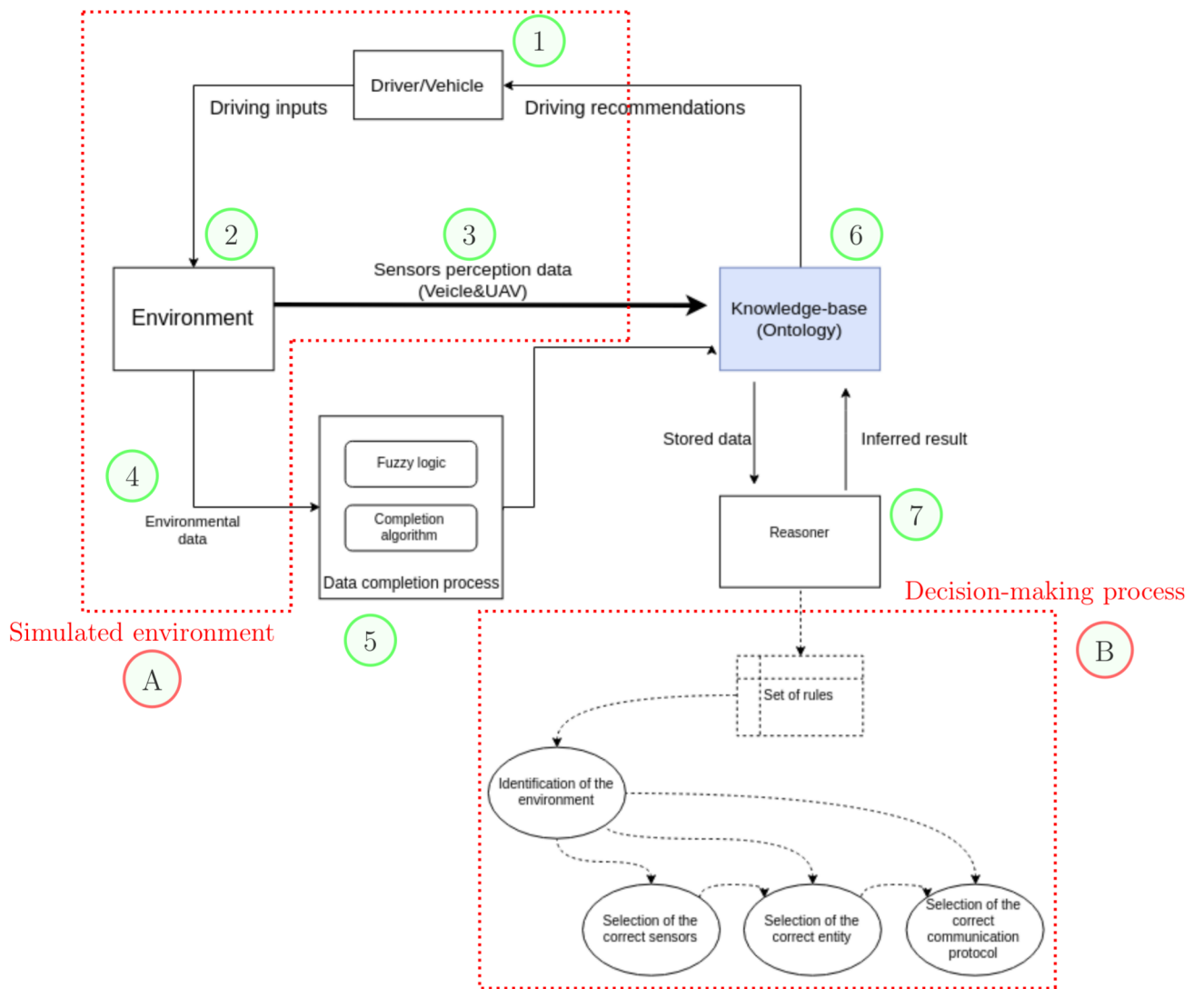


Figure 2.13 – Illustration of the general model

ronment (4) information. The latter serves for the identification of the actors and elements (obstacles, other vehicles...), while the former is used for the correct identification of the environment. The determination of the correct environment is a critical step of the process, which is why those data first go through a completion process (5) in order to make sure that the environment is correctly inferred.

4. All the gathered data are then stored in a knowledge-base (6) where they will be classified and assigned with corresponding properties.
5. The knowledge-base will process the information it was provided through a reasoner (7), which will then start the decision-making process (B)
6. The reasoner will cross-reference the objects in the knowledge-base with a pre-defined set of rules in order to identify the rules returning a positive result: After inferring the correct environment, the system will also identify the adequate sensors and communication protocols working correctly in said environment, then locate the entities (drones) carrying them and request their activation if necessary. This is the decision-making process (B), which is also represented in Figure 2.12.

Various data of different natures are generated during the driving process, some can be elements related to the ego vehicle, which is the vehicle on which the model is centred, (e.g. speed and position) or gathered from the vehicle's exteroceptive sensors (e.g. brightness or distance to obstacle). These elements can come from either the vehicle or another intelligent agent, such as a drone, whose help could be requested through a secured communication channel (for example VLC or a hybrid VLC/RF protocol). These data are stored in an ontology, where they are classified in order to formally represent the environment the vehicle is evolving in.

A critical step in the process is the correct identification of the environment. In order to strengthen this operation, the environmental data go through a data cleaning stage in order to increase their reliability.

Once the storing process is completed, a reasoner is called upon by the knowledge-base

in order to process the data stored in it. It is a tool that relies on logical rules in order to infer an adequate output considering the situation. Once provided with the correct set of rules, the system can identify the current environment and infer on what sensors and communication protocols would perform best under given conditions, as well as to determine the agents carrying those sensors and protocols and request for their assistance.

This approach offers multiple benefits:

- **Extension of the perception range:** thanks to their size and positioning, UAV can cover areas where it would normally be dangerous for the vehicle to venture to
- **Transmission integrity:** by using two different physical communication protocol for redundancy, the system makes sure the transmission is safe and robust
- **Proper context identification:** knowledge-bases offer a formal way of describing an environment. When correctly populated, it can be used for an accurate recognition of the settings.
- **Rigorous management of the enhancement process:** logical rules make it possible for the reasoner to manage the different actors of a scene. When feed with the correct rules, the system can quick decide which sensors and communication protocols should be activated to undertake proper response.

2.7 Conclusion

This chapter introduced a summary of today's state of autonomous vehicle's perception, and raised the idea of using an external entity to gather information. Indeed, UAVs have a strong potential as shown by some promising works that have already been conducted. However, as pointed out in Chapter 1, having another data-gathering system raises the problem of federating the locally perceived information with the ones collected by another entity. Data need to be classified, merged and processed in order to assess an optimal

inference of the environment and the adequate decision to take. A knowledge-base system shows great potential for this type of situation. There is also the question of safely transmitting data from one entity to another, which can be done with the use of VLC as a stand-alone protocol or as part of a more hybrid approach

Chapter 3

A knowledge-based driving context detection approach

Contents

3.1	Introduction	48
3.2	Definitions	48
3.2.1	Knowledge	48
3.2.2	Context	49
3.2.3	Knowledge-base and ontology	51
3.2.4	Reasoning	53
3.3	Proposed model	54
3.3.1	Dealing with missing data	54
3.3.2	Knowledge-base	68
3.3.3	Decision-making process	77
3.3.4	Summary of the process	82
3.4	Conclusion	85

3.1 Introduction

In the previous chapter, we presented the state of the art and explained the advantages of using a knowledge-base for the federation and management of data from various sensors.

A knowledge-base allows for a formal representation of a context by offering a rigorous classification of elements, their properties and the relationship existing between them. With the use of an additional tool called a reasoner, it is possible to add a layer of intelligence to the objects of the knowledge-base thanks to logical rules. Different situations can be addressed thanks to set of rules, for example the identification of the environment the vehicle is currently evolving in, or the selection of the adequate communication protocol. This approach allows for the system to better manage the perception enhancement steps while ensuring data integrity.

In this chapter, we will present our work towards our knowledge-based approach for driving context detection. We will introduce the different notions of knowledge-representation, including the different practical ways of doing so, as well as the logical rules to manage them. The previous chapter discussed about a methodology for the perception process, and a breakdown of it will be essayed in this chapter. Finally, the ontology developed in the context of this thesis will be explained and detailed in this chapter, with both the data elements and the reasoning process discussed in details.

3.2 Definitions

3.2.1 Knowledge

Knowledge is an abstract notion which exact definition changes depending of the topic it is applied to. Zagzebski[53] define the cognitive meaning of knowledge as a "...*relation between a conscious subject and the portion of the reality to which the knower is directly or*

indirectly related. Nickols [54] gives a more formal definition, identifying 3 subcategories of knowledge:

- Knowing as a **state**, which is about being aware or familiar with facts, methods, principles or techniques.
- Knowledge as an **action**, making the effort to create a relationship between an individual and an abstract or concrete element.
- And finally Knowledge as a **codified set of facts** that allows the precise and distinct definition of an element.

In the scope of this thesis, knowledge will take a definition related to the last one. It will represent the distinctive characteristics of an abstract or physical object, as well as the different types of relationship linking it to the other elements of its surroundings.

3.2.2 Context

The literal definition of context, according to the Collins dictionary is "*the set of facts or circumstances that surround a situation or event*". There is however no real consensus of the word's meaning in the topic of context representation. In the scope of this thesis, the definition chosen was the one given in Endsley's work in 1995 [55][56] and characterizes it as "*...the perception of elements in the environment within a span of space and time, the comprehension of their meaning and the projection of their status in the near future*".

Given the heterogeneity of data and the complexity of the notion, categorizing contexts allows for a better management of information. Several propositions have been made in this regards.

Schilit et al.[57] categorize context into two categories : *primary* and *secondary*. The former contains all the information that can be directly gathered, and the latter is made up of what can be deduced from them.

Chen et Kotz[58] also propose two different categories : *active* and *passive* contexts, splitting the information into what has a direct influence on the action, and the secondary, non-critical data.

Hofer et al[59] have a dual-class system too : *physical context* and *logical context*, with data that can be measured with sensors and those that contains information about the interactions between objects.

Fuchs et al.[60] focused their work on a specific case, the Road context. They identified multiple levels on which the the context could be subdivided.

- **Spatial context** corresponds to the whole physical environment the main vehicle is currently evolving in
- **Local context** is a specific situation the vehicle will encounter in a defined space, such as an intersection
- **Traffic context** the elements affecting traffic, like signs, pedestrians, markings, etc.
- **Participants context** comprises the elements related to the driver, such as his tiredness level or driving experience
- **Road conditions** are the properties of the road the vehicle is currently evolving in, for example the maximum speed allowed or the physical state of the road

An interaction context corresponds to a *scenario*. A scenario is a practical situation from where knowledge is gathered and processed, and the inferred actions are applied. A scenario needs a minimum amount of information related to the environment and the objects to be modelled. Ontology will keep the meaning of any situation, current or planned interaction and running scenario in a representation language close to natural language.

3.2.3 Knowledge-base and ontology

The "Principles of Modeling" [61] defines a model as a simplification of a real-world problem. The modeling of a problem presents various advantages, such as a formalization and logical description of a problem, a better understanding of the concerned data, and a simplification of the testing procedure.

A knowledge-base is an object-model way of representing data. As an expert system [62], it contains not only all the different actors and entities of a given situation, but also the abstract concepts, properties and relationships among stored elements.

In addition to the structured data storage, the other important feature of a knowledge-base is the intelligence layer that can be obtained through the use of inference rules. By setting up an appropriate set of logical instructions, the stored data can be analyzed, processed, compared and rearranged in order to produce an output that can be reused.

There are different ways to implement a knowledge-based model, such as logic programming [63], a knowledge-graph [64] or an expert system [62]. This work focuses on the use of **ontologies**.

An ontology usually serves as a hierarchical data structure containing all the entities of a specific context and the different rules, axioms and properties regulating them. In addition to the technical interest, the ontological approach shows some functional features that make it an interesting choice :

- **Scalability:** Once the classes, properties and rules are defined, the instantiation is managed by a Java API and it is easy to populate the ontology with new elements
- **Exportability:** The knowledge-base and its actors are ultimately independent from the application and can be used for another operation set in a vehicular environment

The Stanford 101 Guide defines Ontology [65] as "a formal explicit description of concepts in a domain of discourse, properties of each concept describing various features and

attributes of the concept, and restrictions on slots". An ontology basically defines the main actors within a domain of discourses and the different interactions and relationships between them[66]. Ontology may be represented by:

- **Classes** : Describe the concepts in the domain, whether they are abstract ideas or concrete entities. Classes can be hierarchized by levels, for example having a Vehicle as a top-class containing Car, Bus and Bike as sub-classes.
- **Properties** : The specific attributes related to a. They can be intrinsic to an object, or extrinsic, representing the interconnections between different concepts
- **Individuals** : Real instances of classes, representing the elements of the ontology.

An ontology which is complete and filled with a full set of individuals, rules and properties is referred to as a knowledge base. In technical terms, the knowledge base is composed of the **Tbox** and **Abox**, respectively Terminological Box and Assertion Box. The former represents the ontology where the information are stored, and the latter encompasses the rules and properties.

3.2.3.1 Data fusion

Multimodal data fusion is defined by [67] as "the analysis of several data sets such that different data sets can interact and inform each other", meaning that the information from different sources can be compared and cross-referenced to offer a better understanding of the situation where an intelligent agent is evolving. The implementation of this process requires an architecture capable of efficiently classifying data, and a necessary processing power [68]. In a real driving situation, an important quantity of data needs to be considered, and most of the time they are of different types. Being able to manage and join seemingly unrelated information could be critical in this context.

As seen in the previous chapters, our solution relies on using a UAV to gather traffic data and send them back to the vehicle. These data can then be merged with the ones from

the vehicle sensors by storing them in an ontology and processing them with logical rules. This concept can be extended to every other source of information.

3.2.4 Reasoning

An ontology allows the storage and management of different objects from a common context. In addition to being able to represent all the elements of a situation, it is also possible to add a layer of intelligence and reflection through the use of reasoners. A reasoner is a tool that can infer logical conclusion from a set of given facts, making the classification of an ontology easier. For example, if we declare an instance V as a Car, and the class Car is a sub-class of vehicle, then the reasoner infers that V is a vehicle[69].

For a more complex situation, some reasoners can be supplied by logical rules. A rule is a presentation of an Aristotelean syllogism, defined as "a discourse in which certain things having been supposed, something different from the things supposed results of necessity because these things are so" [70], meaning that if we assume a set of propositions to be true, we can conclude another independent proposition to also be true. Rules are made up of two distinctives parts : an antecedent set of conditions, and a consequent set of actions. They are given in the form : **IF** <conditions> **THEN** <actions>.

There are multiple ways to implement rules into a knowledge-base. This work used the SWRL approach (Semantic Web Rule Language)[71].As previously stated, it is a language of logic description that enables the combination of different rules to build a more complex axiom. The official documentation gives the following basic example to define the syntax : $hasParent(?x1,?x2) \wedge hasBrother(?x2,?x3) \rightarrow hasUncle(?x1,?x3)$. By joining the two axioms *hasParent* and *hasBrother*, it is possible to apply the *hasUncle* relation to the individuals, hence making the individual X1 the child of X2 and the nephew of X3.

3.3 Proposed model

3.3.1 Dealing with missing data

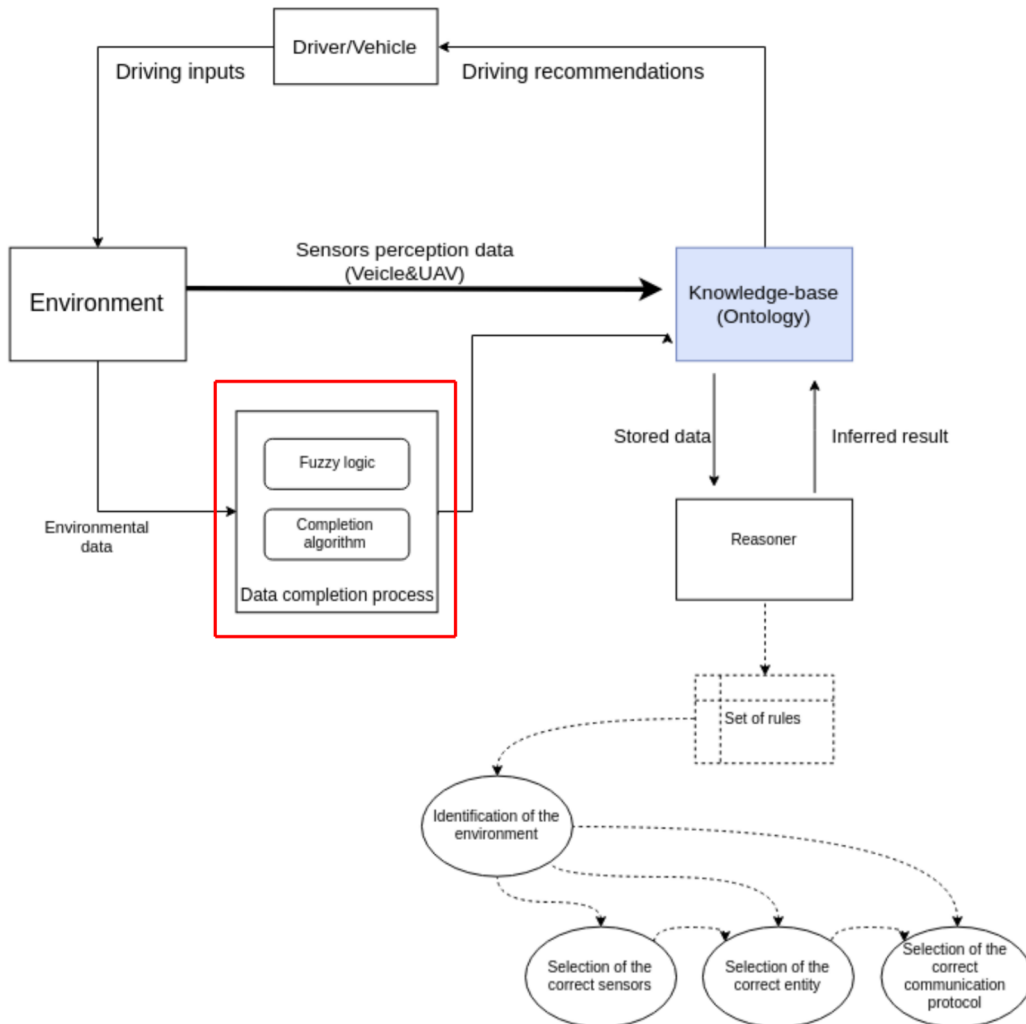


Figure 3.1 – Data-reinforcement step in the general model

In order to guarantee an optimal logging and inferring process, we must ensure that we have sufficient amount of data. However, due to various factors such as sensor malfunctioning or an error in the logging process, we might end up with a lack of or faulty set of data. It is interesting to consider this aspect to optimize the output of the system. There are multiple way of preventing [72] and dealing with missing data[73]. In this thesis, we consider this aspect and envision two approaches : A computational approach through

comparison of different algorithms, and a fuzzy-logic approach to generate a "perception score" of the existing data.

3.3.1.1 Data Cleaning

3.3.1.1.1 Presentation of the dataset

A set of data can take multiple forms, depending on the type of data that are being considered. This section will introduce the dataset that served as a base for the data completion process. It is important to note that this will not be the final set of data on which this thesis work was focused on, but was instead used for the comparison of different algorithms. On the road context, it is important to consider multiple factors in order to infer the correct state of traffic. There are many types of data that can be used for traffic prediction.

Traffic prediction has long been regarded as a statistical problem. In one of the earliest studies in 1991, Davis and Nihan [74] compared the simple univariate linear prediction on a regression model in an empirical measurement of traffic congestion. They choose a Nearest-Neighbor approach and showed that this lazy-learning method was just slightly better than a classical parametric regression method. However, and as noted by the authors, the optimisation was not significant enough to be relevant, and the predictions were still not accurate enough, sometimes being up to 30% incorrect.

In 2003, with access to higher-quality traffic data, Clark[75] proposed a Non-parametric Regression model that would include other variables to consider in addition to speed, like the day of the week. Non-parametric regression is a form of regression that is based on available data, rather than a predetermined prediction function, hence being relevant in the traffic topic since there is not a single "fixed" behaviour. Their model showed a great potential, but they did not have a database big enough to accurately train it, and were also lacking the computational power to properly calibrate it.

As pointed out by Vlahgioanni et al. [76], traffic forecasting has been studied for almost 3 decades now. In their literature review, they came out with 10 possible axis of improvement:

- Developing responsive algorithms and prediction schemes
- Freeway, arterial and network traffic predictions
- Short-term predictions: from volume to travel time
- Data resolution, aggregation and quality
- Using new technologies for collecting and fusing data
- Temporal characteristics and spatial dependencies
- Model selection and testing
- Compare models or combine forecasts
- Explanatory power, associations and causality
- Realizing the full potential of artificial intelligence

The last point has become the most interesting consideration over the years. This review was made in 2014, a few months before the real surge of Deep Learning[77] and Artificial Intelligence. Many studies have since been focused on the improvement of the proposition, i.e. building neural networks and using AI for traffic forecasting.

Artificial Intelligence has already been used in recent works for traffic forecasting. In 2015, Ma et al. [78] implemented a Long-short-term memory neural network to predict traffic situation. They used microwave detector to collect real-life data for a month and trained their model with them. They showed good results with 97% accuracy on their predictions. Even though the model made very accurate predictions, the only input it had was the speed of cars that were going through the testing road, with no consideration of environmental data.

In 2011, Min et Wynter [79] developed a scalable method for traffic prediction up of to 15 min in a dynamic environment. The mathematical model they proposed was built upon two variables : the distance, and average speed of vehicles. They showed excellent results but using only two parameters made the computation process light enough to be fast. One of the possible improvements they mentioned was adding external parameters, such as "weather, incident data and roadwork, current or planned".

In order to compare algorithms, it was necessary to have an adequate dataset filled with correct elements. Considering the review of previous works, the following variables have been considered for the dataset :

- **Weather** : Sunny, Cloudy or Rainy
- **Location** : City, Highway, Isolated Road
- **Day** : Weekday, Week-end
- **Time** : Rush hours, calm hours
- **Speed** : Up to 120km/h
- **Roadwork** : Whether there are works on the road or not
- **Traffic Incident** : Whether there is a traffic incident or not

These specific parameters were chosen after the literature review made in Chapter 2, which showed that traffic forecasting should at least include them. Data are either collected by the vehicle's sensors or the Smart City Broadcast, which is assumed to be by RF transmission of notable event. These data are also classified into 3 categories : 1) Those related to the surrounding of the vehicles (Weather, Day and Time) 2) the ones directly related to the car's behaviour (Speed and Location), and 3) the events that can happen independently of the vehicle (Roadwork and traffic Incident). These data categories are respectively detailed in Tables 3.1, 3.2 and 3.3. Based on the gathered information, we

can predict the state of the traffic and categorize it into 4 different types : Light, Medium, Heavy and Extremely Heavy. The main parameters to define the output are the Roadwork and Traffic Incident variables, both being occasional and spontaneous events. They play an important role in traffic congestion, and coupled with the other parameters, such as the weather or the speed limitation, the traffic flow can be temporarily fully stopped.

Parameters name	Weather	Day	Time
Parameters values	Sunny, Cloudy or Rainy	Weekday, Week-end	Rush hours, calm hours
Gathering channel	Car Sensors	Car System	Car Sensors

Table 3.1 – Environment parameters

Parameters name	Position	Speed
Parameters values	City, Highway or Isolated Road	Up to 120km/h
Gathering channel	Car Sensors	Car Sensors

Table 3.2 – Car parameters

Parameters name	Roadwork	Traffic Incident
Parameters values	Yes or No	Yes or No
Gathering channel	City Broadcast	City Broadcast

Table 3.3 – Event parameters

3.3.1.1.2 Algorithm comparison

For this part, three completion algorithms from different methods are compared : 1) the algebraic SVD (Singular Value Decomposition), 2) the statistical Mean Imputation and 3) the learning-based classifier KNN (K-Nearest Neighbor).

- The dataset technically being an integer matrix, it is possible to use algebraic algorithms on it. One of the appropriate method is the SVD (singular value decomposition), which has been proven to be useful for matrix completion [80].

SVD is an algebraic method, taking a mathematical approach to the problem. It is used in the case of rectangular a rectangular that is split into three different matrixes of smaller sizes for a better computation of the missing data.

- The mean imputation is one of the easiest and most straightforward completion methods and consists of calculating the mean value of each column and using it as a replacement for the missing values. This method requires all the values to be numerical.[81]
- The KNN method takes an incomplete row of data and compares it to the most similar ones in order to predict the correct output. In this case it takes the road and traffic conditions as inputs and tries to find similar cases in the knowledge base and determine the traffic situation.[82]

The efficiency of these algorithms is evaluated based on the time they take to reconstruct the dataset and the accuracy of their outputs. The one with the best results is chosen as the optimal solution for the model. Different data sets were built, then some of their elements were randomly dropped in order to simulate a malfunction in the gathering process. The newly generated incomplete data set was then processed by the completion algorithms, and the different outputs were compared to the original data. The difference ratio and the overall execution time were used to evaluate the performance of the algorithm.

The first step in construction a model of traffic forecasting is the building of the dataset. A Python program was developed that would generate 1000 set of events with their respective outputs. As described in Algorithm 3.1, the parameters are randomly assigned a numerical value that corresponds to the state of the variable. For example, Weather = 0 means that the weather is sunny and presents no problem, whereas a value of 2 means it is a rainy day that may have a strong influence on the traffic. The Traffic Situation output is computed by fusing the values of Weather, Location, Day, Time, Roadwork and Traffic Incident. To simulate a more realistic situation where we could have a data loss, a complementary script was added that would go through the dataset and randomly delete an information, with a chance of 5%.

Once the incomplete dataset is ready, we started the augmentation phase. Data augmentation is the process wherein the initial dataset is reinforced and completed. We searched

for the optimal algorithm to clean it. Three methods have been considered and tested : Mean Imputation, SVD and KNN classification methods. Table 3.4 shows a comparison of the three algorithms performance. The KNN requires calibration in finding the value of K , and the best results seem to be reached for $K = 1$, meaning that the algorithm replaces a missing value with the closest set resembling it. This specific case of KNN, known as the 1-nearest-neighbour, has already been shown to have excellent results for low-dimension problems [83].

Algorithm 3.1: Complete Set
<p>Input : None Output: 1000*8 Training Set</p> <p>Create Training Set; <i>Write the headers first;</i></p> <p>for $i \leftarrow 0$ to 1000 do</p> <p style="padding-left: 2em;">weather = randomValueOfWeather(); location = randomValueOfLocation(); speed = randomValueBetween[0:120] day = randomValueOfDay(); time = randomValueOfTime(); trafficIncident = randomBoolean(); roadWork = randomBoolean(); trafficSituation = trafficComputation(weather,location,speed,day,time, trafficIncident,roadWork); set = [weather,location,speed,day,time, trafficIncident,roadWork, trafficSituation]</p> <p style="padding-left: 2em;">Write the vector <i>set</i> in row i of the Training Set</p> <p>end</p>

The SVD shows overall poor performances, which is not surprising considering that it must make complex operations on the dataset. The algorithm needs to split the matrix of data into smaller ones and process them in a specific order, hence the long execution time. The performance value means that roughly a third of the predicted data were correct and similar to the initial ones. This poor result could be explained because of the complex size of the data set. The performance greatly increases in the two other algorithms. Mean Imputation is a straightforward mathematical method and show great results as long as all the data are numerical, while KNN is a ML algorithm relying on similar patterns existing in the same data set. They both show overall similar results with an error rate

Table 3.4 – Comparison between the completion algorithms.

Algorithm	Execution time (in ms)	Performance
SVD	173	32%
Mean imputation	107	99,3%
KNN	132	99,5%

of less than 1% and the KNN having a very small advantage, but the Mean Imputation is slightly faster. The dataset being local, of 7 dimensions and of relatively small size, these conditions happens to be the most advantageous for the algorithm.

Algorithm 3.2: Incomplete Set
<p>Input : 1000*8 Training Set</p> <p>Output: 1000*8 Incomplete training Set</p> <p>Import the Training Set;</p> <p>for $i \leftarrow 0$ to 1000 do</p> <p style="padding-left: 2em;"> $j = \text{randomValueBetween}[0:100];$</p> <p style="padding-left: 2em;"> if $j > 95$ then</p> <p style="padding-left: 4em;"> Random Cell from the row $i = NA;$</p> <p style="padding-left: 2em;"> end</p> <p>end</p> <p>Save the new set as Incomplete Set;</p>

3.3.1.2 Fuzzy Logic

3.3.1.2.1 Definition

Fuzzy logic is a method employed to handle the concept of partial truth, as opposed to Boolean logic. It is a mathematical means of representing vagueness and imprecise information. A fuzzy logic model is built on different parts:

- **Meaning representation:** The first step is to translate the given propositions into a set of rigorous and usable constraints. The example given by Zadeh [84] is the following : A fuzzy approach tries to copy a human’s ability to infer based on

previous experiences and social norms, thus when we state that "usually Swedes are blond", the term "usually" can be given an accuracy value between 50% and 80%

- **Knowledge-base system:** By definition, a knowledge-base is a way to store complex structured and unstructured information, making it an "advanced" database. The storing function is then completed by an inference engine, a tool that allows the logical processing and reasoning on the stored elements. This process is enabled using IF-THEN type of rules, which can particularly be useful when dealing with fuzzy logic.
- **Defuzzification:** the last part of the process is the inverse transformation that is opposite to the first one. It associates the results of the previous rules to a crisp logic equivalent value. The receiving algorithm requires a real value in order to function, and relies on the defuzzification output in order to do so.

By its nature, fuzzy logic allows for a better management of uncertainty in engineering, and has found a fundamental role in many AI fields due to its potential in dealing with a lack data.

3.3.1.2.2 Application

In Section 3.3.1.1.2, we tried to fill in empty information by looking for similar patterns from the rest of the dataset. The fuzzy logic approach takes a different angle without trying to generate new data.

Fuzzy logic relies on the notion of **Membership function:** an uncertain statement is represented by a mathematical function covering a value ranging from 0 to 1. When we refer to a sensor as functioning "Perfectly" or "Completely not functioning", those are the extreme possible value. But taking uncertainty into consideration means the meanings of "Good" and "Poor" should be considered and represented. Membership functions allow the representation on how reliable an information is: when the Y-axis

value of a fuzzy logic function is at 1, that is the equivalent of the literal expression "Always" or "Definitely", while when it is 0, that would mean "Never" or "Definitely Not". This is a proven way of representing just how certain an information can be.

We tried to apply this process for the management of sensor's reliability. Indeed, due to diverse reasons (shown in Chapter 2), sensors precision can sometimes drop because of internal or environmental issues. A fuzzy logic approach would allow us to combine inputs from multiple potentially faulty sources and try to infer a correct state of the environment where the vehicle is evolving.

As stated in Chapter 2, the most common sensors in an autonomous vehicles are : **Mono-scope camera, stereoscopic camera, infrared camera, radar, sonar and lidar**. They would however still need to reach a threshold of precision. In addition to internal issues, the main uncertainty source on sensors come from brightness and weather states [9]. When fusing their data, it is important to consider a margin of uncertainty.

In order to have a better grasp of the environment, all those variables are considered and treated in order to produce a **Perception score**. This parameter will allow to categorize the environment in which the vehicle is evolving, and influence the sensors activation.

The previous subsection mentioned that the Fuzzy logic method relied on three steps. The membership functions allow for the transposition of fuzzy expressions into numerical, exploitable values. Once quantified, the inference and defuzzification processes can happen. In the context of this work, we are looking for the evaluation of the perception ability, meaning Perception will be quantified into different values: Poor, Average and Good.

The logical rules combine the outputs of different variables in order to infer a general perception value. They appear under the logical IF-THEN form, making it easy to combine different variables. Examples can be found in Figure 3.2: In the first case, the brightness is poor but the weather is great, making the general perception Average. In the second case, both brightness and weather are good, making perception good too.

This approach can be applied to sensors too. Examples are shown in Figure 3.3. The first

1. IF (brightness IS dark) AND (weather IS good) THEN (perception IS average)
2. IF (brightness IS good) AND (weather IS good) THEN (perception IS high)

Figure 3.2 – Defuzzification rules.

1. IF (cameraMono IS poor) THEN (perception IS poor)
2. IF (cameraMono IS good) AND (NOT(brightness IS good)) THEN (perception IS average)
3. IF (cameraMono IS good) AND (brightness IS good) AND (weather IS good) THEN (perception IS good)

Figure 3.3 – Defuzzification rules for sensors.

rule is in the case the camera sensor is declared as poor, meaning it is unreliable due to some malfunction or environmental condition. This situation would automatically impair the perception score. In the second example, the sensor works well but the environment is hindering, making the perception average. In the third example, the sensor functions properly and the conditions are good, making the overall perception good.

The perception scores from the different functions are averaged to a single value which will be the general output of the system. If the majority of the functions have a "Good" perception value, it would be returned value to the Perception Class.

3.3.1.2.3 Weather has a value that ranges from "Sunny" to "Rainy" with numerical associations in the range of [0;10]. It has a fuzziness equivalent of "Good", "Poor", depending of what is considered a "good weather" or a "bad weather". The representation can be found in Figure 3.4.

The values were chosen in an arbitrary way of what a "good" or a "poor" value means. The chart has two different functions: one for "poor" and one for "good". As stated above, when the weather value is close to 0, the environment is sunnier, which is universally considered to be the best weather, as well as the optimal situation for visual perception. The "Good" function is at the max value for 0, which is a perfect sunny weather, and

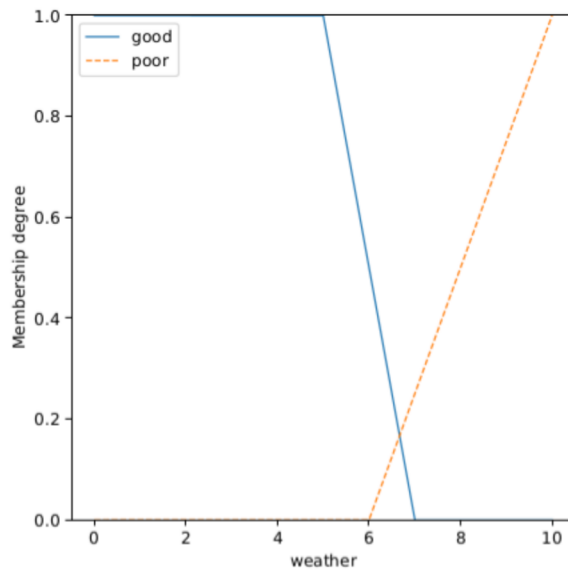


Figure 3.4 – Fuzzy representation of weather

starts to slowly decrease proportionally to the weather, reaching a null value for the bad weather. On the other hand, the "Poor" function only starts to rise at the value of 6, which would consist on a cloudy or light-rain situation (0 equals sunny, while 10 equals rainy). At the max value, the membership function is at 1, meaning a rainy situation is indeed considered a bad weather.

3.3.1.2.4 Brightness is a value that ranges from "Dark" to "Overbrightness" with numerical associations in the range of [0;10]. It has a fuzziness equivalent of ["Dark", "Good", "Bright"], with an optimal brightness achieved at 5. The lowest values correspond to Darkness and the highest to Overbrightness, both having a negative effect on perception. The representation can be found in Figure 3.5.

The membership values for Brightness are split between three functions: "Dark", "Good" and "Bright". The lower the brightness, the higher the "Dark" function membership value is, while a too bright environment is considered as too "Bright", meaning overbright.

3.3.1.2.5 Sensors have each their own membership functions, but they all are subdivided into "Good" and "Poor" reliability. This can be altered due to weariness, environment, quality of the sensor or under-voltage. For example, a Stereoscopic camera,

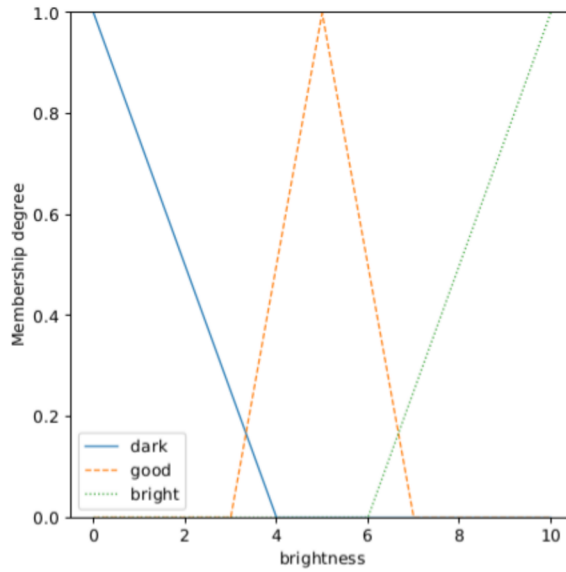


Figure 3.5 – Fuzzy representation of brightness

which is more expensive and of high quality, will be required to have a stricter definition of "good functioning" than a Sonar, a cheaper and more common component. The details of each membership functions are represented in Figure 3.6.

The X-axis defines the quality of the sensor, a variable depending on elements such as the sensor current state or maintenance routine. The higher this value is, the more confident we can be on the reliability of the gathered data. This is represented by a higher membership value. The differences in the functions is related to the type of sensor: some sensors can generate more data, or of better quality, hence the need to be more strict in the the way they are trusted.

3.3.1.2.6 Knowledge-base implementation All those data are treated in order to generate an output called **PerceptionAccuracy** which value varies in $[0;3]$ with a higher score representing a better Perception. This is later logged in the knowledge base.

In the knowledge-base, the perception score is classified as Poor, Average or Good. Its value has an impact on how trusted the environmental detection is.

As stated at the beginning of this chapter, an ontology serves as the storage of different elements making up a specific context as well as their properties and interlinked relation-

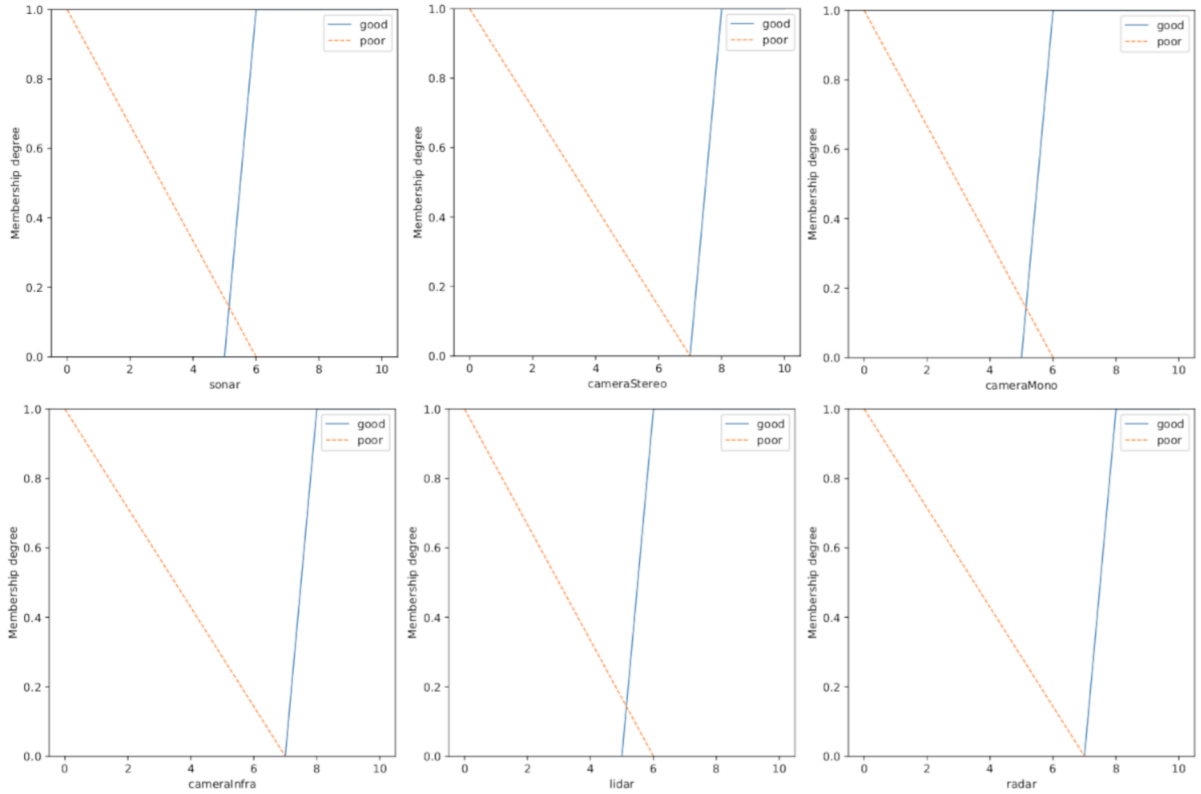


Figure 3.6 – Fuzzy representation of sensors

ships. An important amount of elements classified will naturally result in a more precise ontology.

This section's work was made in order to minimize the amount of data that could potentially be lost due to hardware or low-level software issues. The dataset can either be supplemented through completion algorithms, or fuzzy logic methods can be applied in order to alleviate the uncertainty of the perception. This approach was mainly applied to the environment detection.

Indeed, the correct identification of the environment is one of the most critical steps of the process, from which most of the logical rules and decision-making process rely on (cf. Figure 2.13), hence the need to reinforce it. This will allow for a better management of the process for the rest of the process.

3.3.2 Knowledge-base

3.3.2.1 Presentation

The work of this thesis relies on the use of a UAV in order to gather perception data and sends them to a vehicle in order to merge and process them. As shown previously, an ontology can do both of these operations.

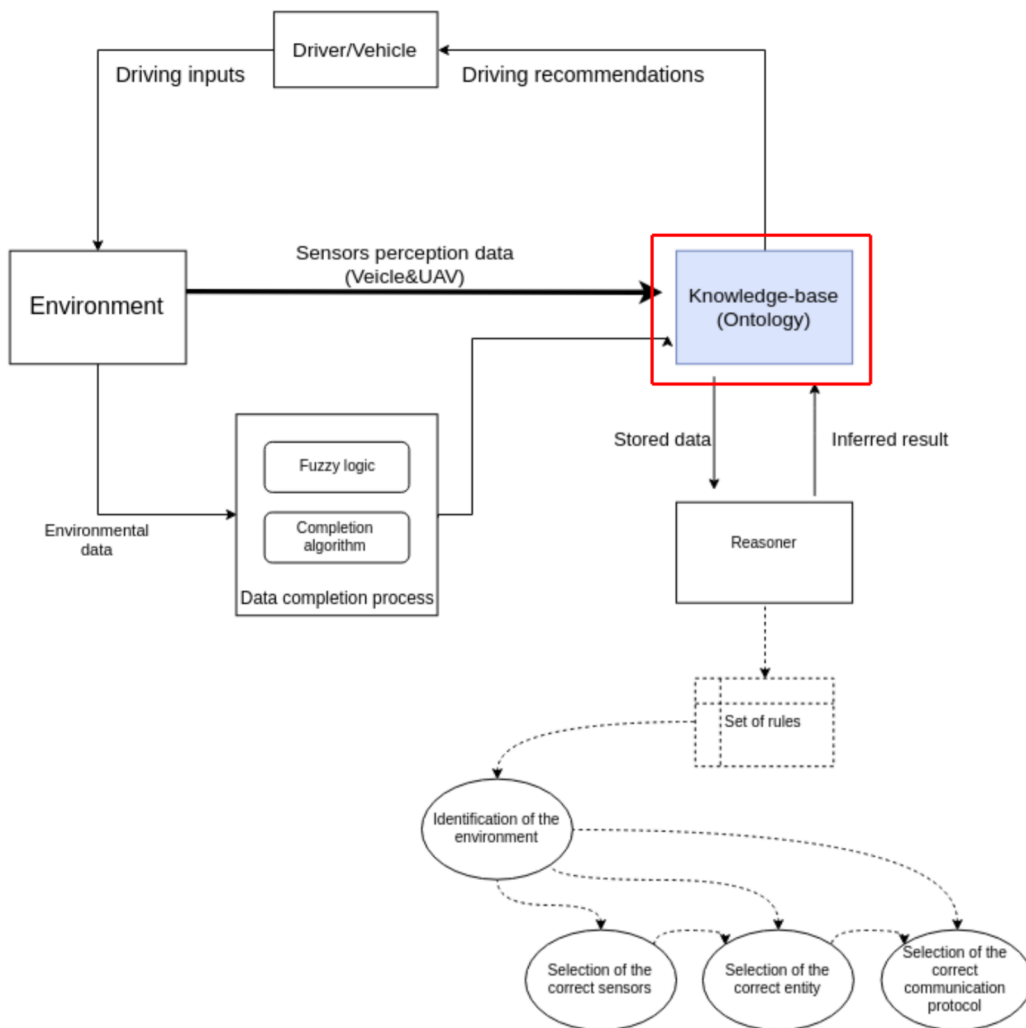
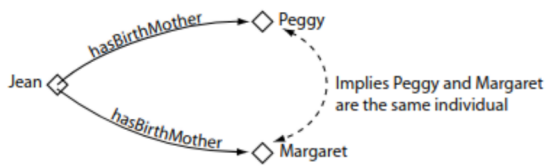
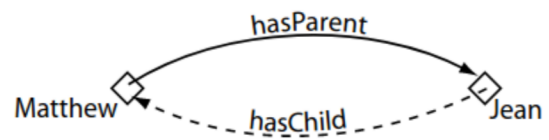


Figure 3.7 – Knowledge-base step in the general model

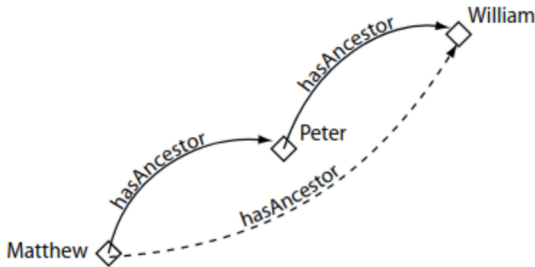
An object in the knowledge-base can have properties. There are two type of properties: 1) Data Properties link an individual to a numerical, literal, or boolean value and 2) Object properties link individuals of the knowledge-base between them, regardless of their initial class. There are multiple types of object properties, as illustrated in Figure 3.8, courtesy



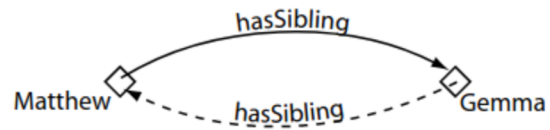
(a) Functional property



(b) Inverse property



(c) Transitive property



(d) Symmetric property

Figure 3.8 – Different types of properties

of the work of Horridge[85]:

- **Functional property:** is a property that links an individual to **at most** one other individual.
- **Inverse functional property:** is a property where the inverse relationship is functional. This means that the original individual can have multiple inverse functional properties, but the targeted individuals only have that property toward the original individual.
- **Transitive property:** is a property that get passed to other individuals. If individual A has a property related to individual B, and individual B has the same property related to individual C, then individual A shares the property with C too.
- **Symmetric property:** is a property that applies similarly to all individuals that share it.
- **Asymetric property:** is a property that can only happen in one way between two individuals.

3.3.2.2 Concepts of the knowledge-base

The ontology used for this study focuses on both the vehicle and its surroundings. The general presentation can be found in Appendix A. It contains a variety of interlinked class:

3.3.2.2.1 Sensors

The *Sensors* class includes all the sensors that can be used in a road environment, as shown in Chapter 2. They are subdivided in different sub-classes, as shown in Figure 3.9:

- *Active Sensors*: Sensors that rely on their own source of emission in order to gather data. Lidars use a laser ray in order to map their surroundings. Radars generate a high-frequency electromagnetic impulse and uses the Doppler effect[86] in order to calculate the distance. Ultrasounds sensors rely on the same principle, but with ultrasonic waves.
- *Passive Sensors*: Sensors that gather data without the need for generating any form of emission. They mostly refer to cameras. There are 3 different types of cameras : 1) Monoscopic cameras are the classical and most common ones, used for classification and image processing. 2) Stereoscopic cameras rely on the same principle except with more cameras, allowing for a depth perception. 3) Thermal cameras are cameras using infrared radiations, allowing their use in bad weather and poorly illuminated environments, but make the detection and classification of elements harder.
- *Environmental Sensors*: Sensors used for the detection of environmental variables, including Rain, Fog and Brightness.

More details about the Sensors can be found in the representation in Figure 3.10, especially concerning their different properties:

- *isWeakToRain* property means that the sensor performs poorly in the case of rain.

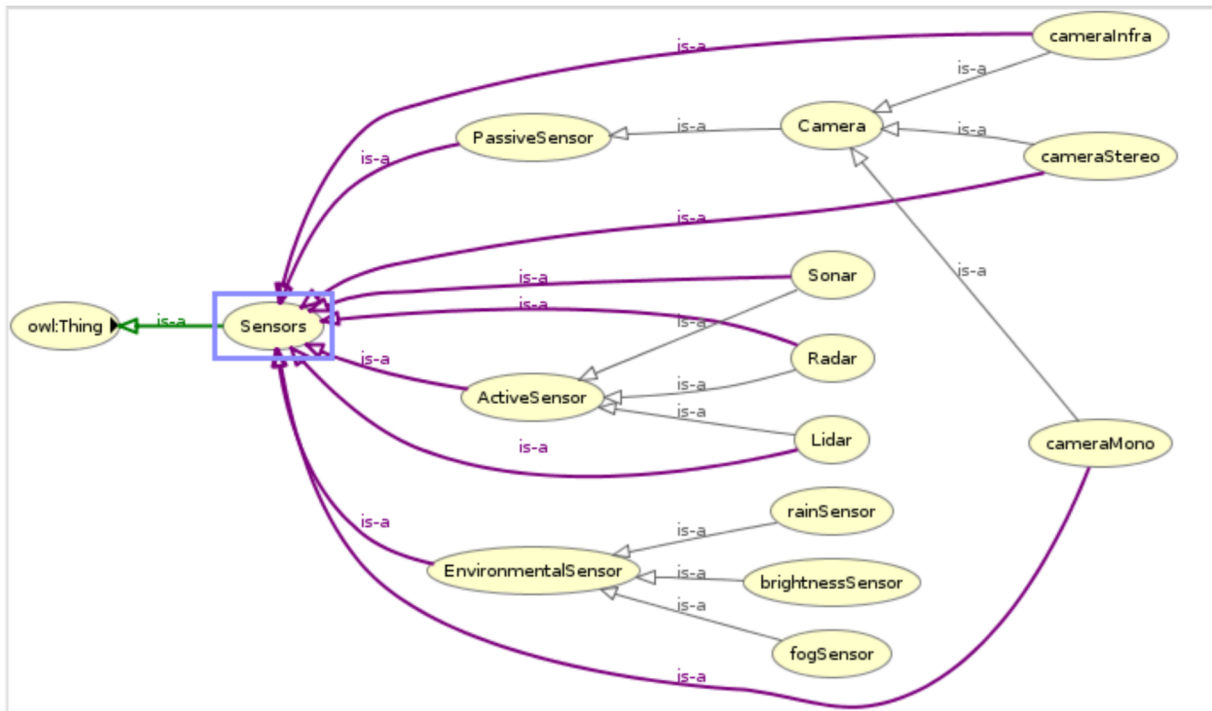


Figure 3.9 – Hierarchy of the Sensors class

- *isWeakToFog* property means that the sensor performs poorly in the case of fog.
- *isWeakToBrightness* property means that the sensor performs poorly in the case of over brightness.
- *isWeakToDark* property means that the sensor performs poorly in the case of lack of illumination.
- *isUnreliable* property means that the sensor performs poorly due to some reasons.
- *hasMinRange* property represents the minimal functioning range of a sensor.
- *hasMaxRange* property represents the maximal functioning range of a sensor.
- *isActiveSensor* property defines if a sensor should be activated or not depending on the environmental situation.

3.3.2.2.2 Vehicle

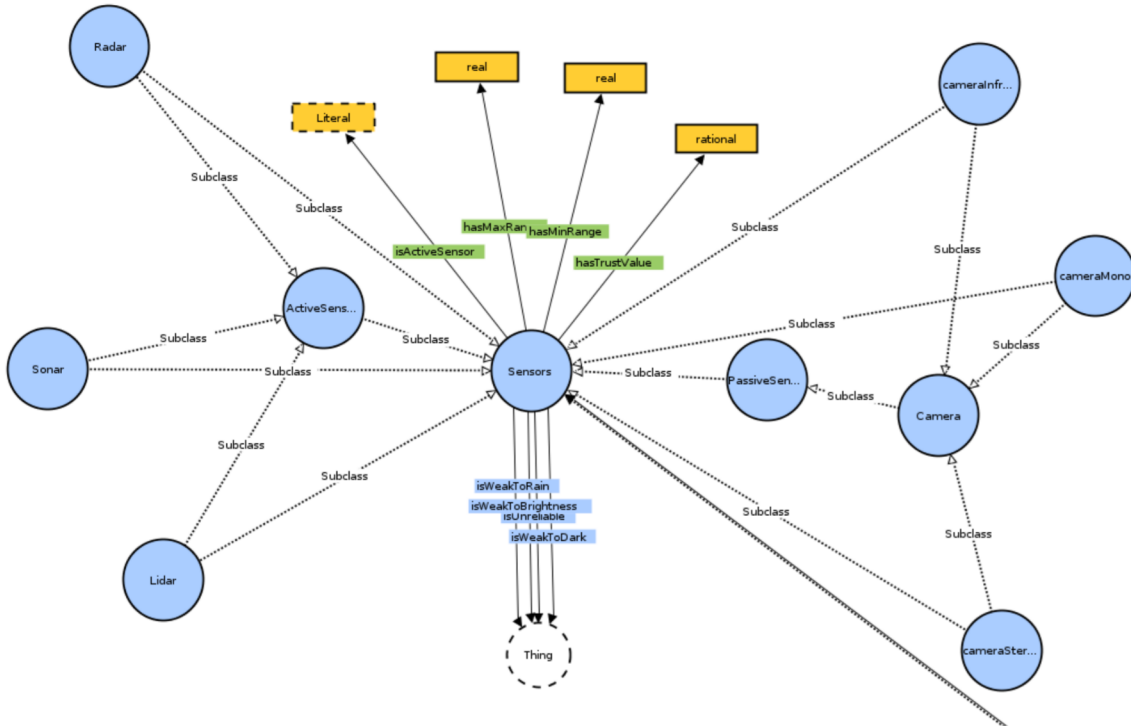


Figure 3.10 – Relationships and properties of the Sensors class

The *Vehicle* class includes all the different vehicles that can be encountered in a road environment, including Cars, Trucks, Bikes, but also UAV. Here are the details of some properties :

- *hasDriver* is a functional property between a vehicle and a driver. it allows for the identification of the Ego Vehicle which will contain the *MainDriver* individual.
- *hasSensor* is a property that associates a vehicle with all the sensors embedded on it
- *hasDistanceFromVehicle*: property allows the classification of the distance between a vehicle and a physical obstacle on the road. It can have a value in the set {Far,Medium,Close}. It can be associated to the boolean data property *hasObstructedView*.
- *isCloseToUAV*: applies when the ego vehicle is in communication reach of a UAV. If the conditions are met, it can be associated to the boolean data property *isAc-*

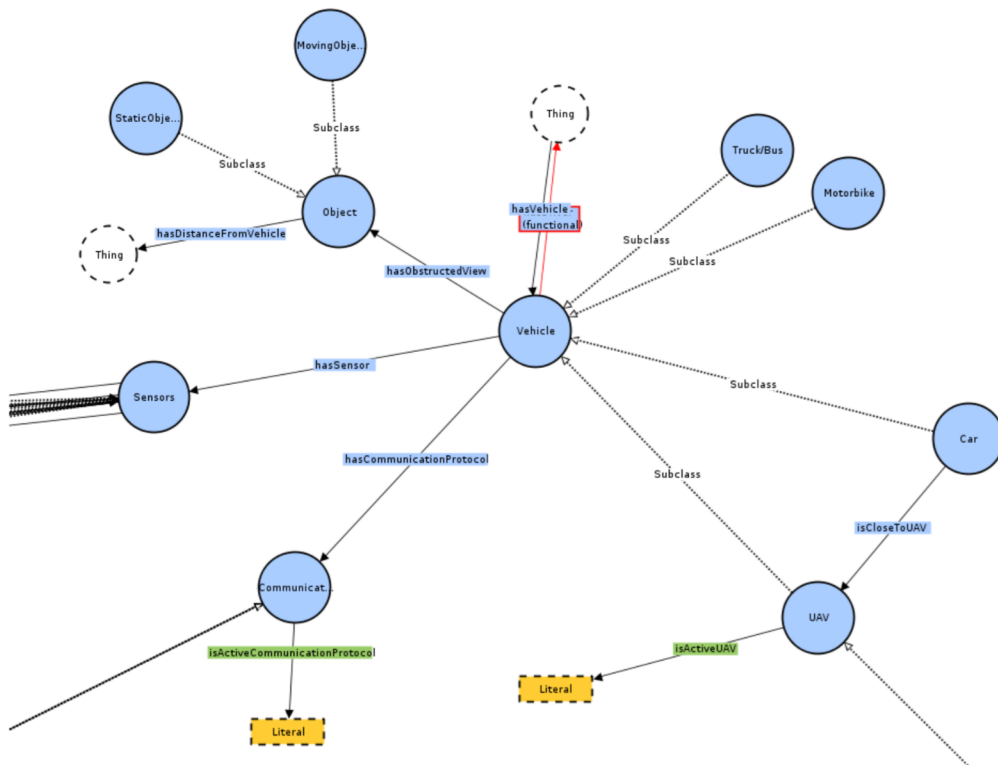


Figure 3.11 – Relationships and properties of the Vehicle class

tiveUAV for the data gathering request.

3.3.2.2.3 Communication Protocol

The *CommunicationProtocol* class includes the three possible disjoint communication protocol classes: RF, VLC and Hybrid. The only notable property of this class is *isActiveCommunicationProtocol* which is a boolean indicating which communication protocol should be used depending on the situation. Protocols function differently depending on the environment, and the hybrid approach, when available, allows for a reinforced data transmission.

3.3.2.2.4 Objects The *Object* class includes all the different objects that can be encountered in a road environment, such as traffic light, pedestrians or road signs. It is subdivided in two generic sub-classes *MovingObject* and *StaticObject*. The individuals here

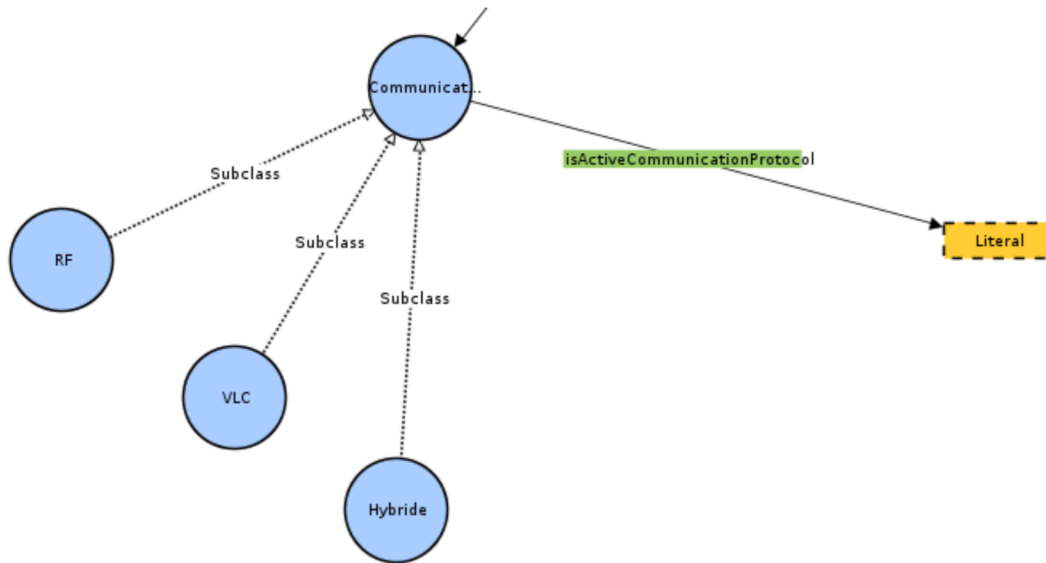


Figure 3.12 – Relationships and properties of the Communication class

mainly have a role of populating the environment and being considered as "obstacles" for the vehicle, but some of them can have special role, like a Building as a FireHazard situation.

3.3.2.2.5 Action The *Action* class covers the main results of the decision-making process, including the driving recommendations and the notable detected situations.

3.3.2.3 Logical rules

Once populated, the elements in the knowledge-base are processed through a reasoner and a set of rules. Rules follow the logical IF-THEN pattern.

Reasoners work in a concurrent way, meaning that the rules are executed simultaneously and not sequentially. However, for readability and clarity purposes, they have been re-grouped in categories based on their common goal.

3.3.2.3.1 Environmental detection

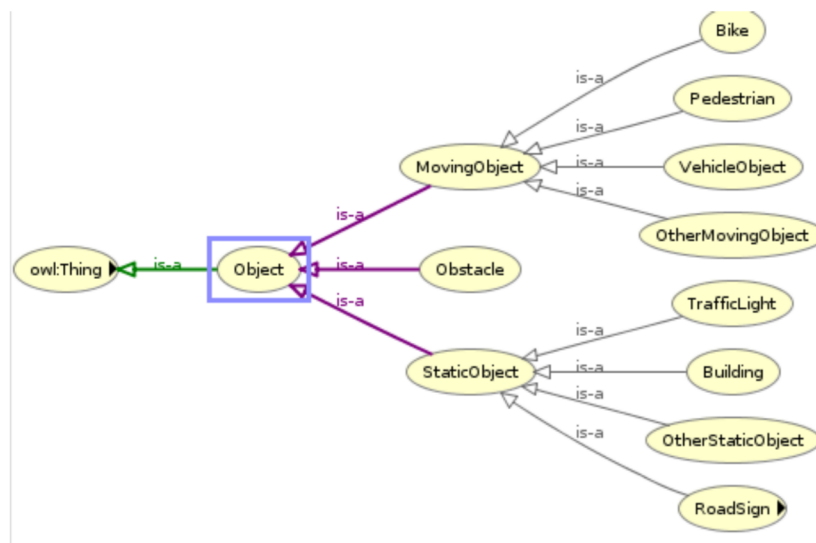


Figure 3.13 – Hierarchy of the Object class

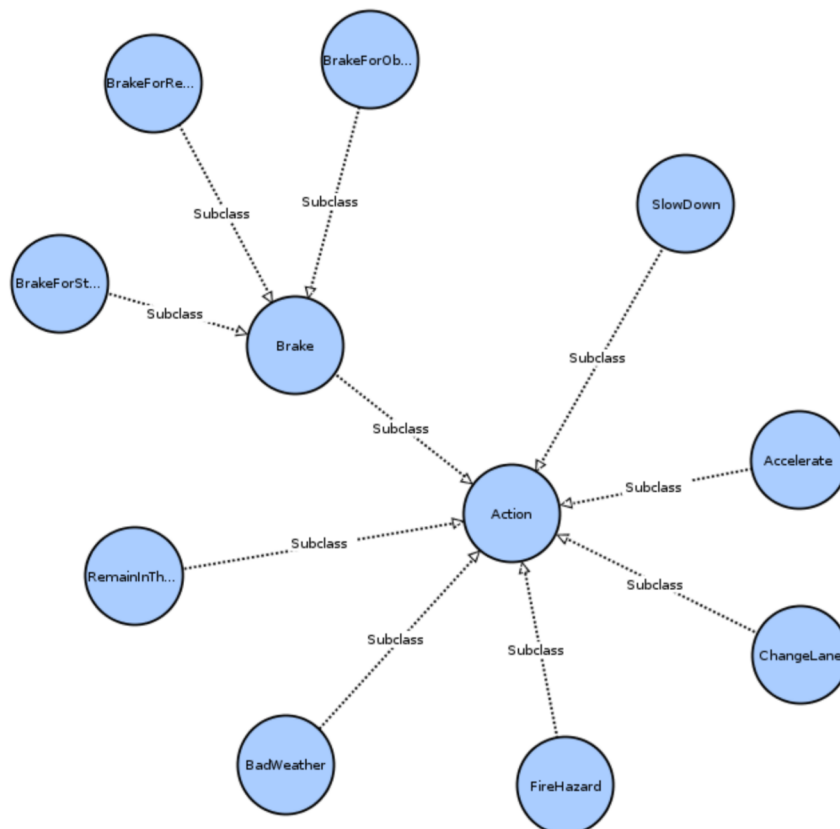


Figure 3.14 – Relationships and properties of the Action class

Identifying the correct environment is a critical part of the process. After the data reinforcement process, the object fed to the knowledge-base can be used in order to correctly infer the environment. This is done thanks to logical rules.

- The environmental sensors gather the value of an environment variable and compare it to a threshold **IF (envSensor exists for environment X) AND (envSensor returns envValue) AND (envValue bigger than threshold value) THEN (environment is of type X)**.
- In case different contexts can be linked to a single environment (for example, a rainy or foggy context both correspond to a Bad Weather situation), this can also be simplified by using logical rules **IF (environment is Rain) THEN (environment is BadWeather) IF (environment is Fog) THEN (environment is BadWeather)**
- Some other situations can be covered, for example limited visibility **IF (Object on the road) AND (vehicle on the road) AND (object is very close to the vehicle) THEN (the view is obstructed)** or a fire hazard event **IF (vehicle on the road) AND (building nearby) AND (building is on fire) THEN (there is a hazard of type FireHazard) AND (environment is of UnusualEnvironment type)**

3.3.2.3.2 Entity, sensors and communication protocols management

The model is aware of what entity is in the vicinity of the vehicle, and based on the environment, can also infer what sensors and communication protocols function properly in a specified context. Those elements can be managed thanks to logical rules too.

- When a UAV is identified, there is a need to check if it contains the correct sensors **IF (there is a vehicle) AND (there is a UAV) AND (there is an environment of type X) AND (there are sensors working on that specific environment)**

AND (the UAV is carrying the sensors) AND (the UAV is close enough to the vehicle) THEN (the UAV is activated). *Activating a UAV in the knowledge-base means requesting perception data from it.*

- Once activated, there is also a need to activate the sensors **IF (there is a UAV) AND (there are sensors working on that specific environment) AND (the UAV is carrying the sensors) AND (the UAV is active) THEN (activate the sensors on the UAV)**
- The same methodology is applied to the communication protocol **IF (there is an environment of type X) AND (there are communication protocols working on environment X) AND (there is a UAV) AND (the UAV can communicate using the defined protocols) THEN (activate the communication protocols).**

Due to the nature of a reasoning engine, these different tasks are actually implemented in parallel. However, for better readability manipulation, the rules are regrouped in a modular way.

3.3.3 Decision-making process

The previous sections introduced the different elements of the model as well as an introduction to the logical rules used for this work. This section will dive deeper into the SWRL rules covering the different situations.

3.3.3.1 Rules description

3.3.3.1.1 Environmental detection

In order to deal with the varying status of weather, different rules exist in order to identify the situation that the vehicle is currently evolving in.

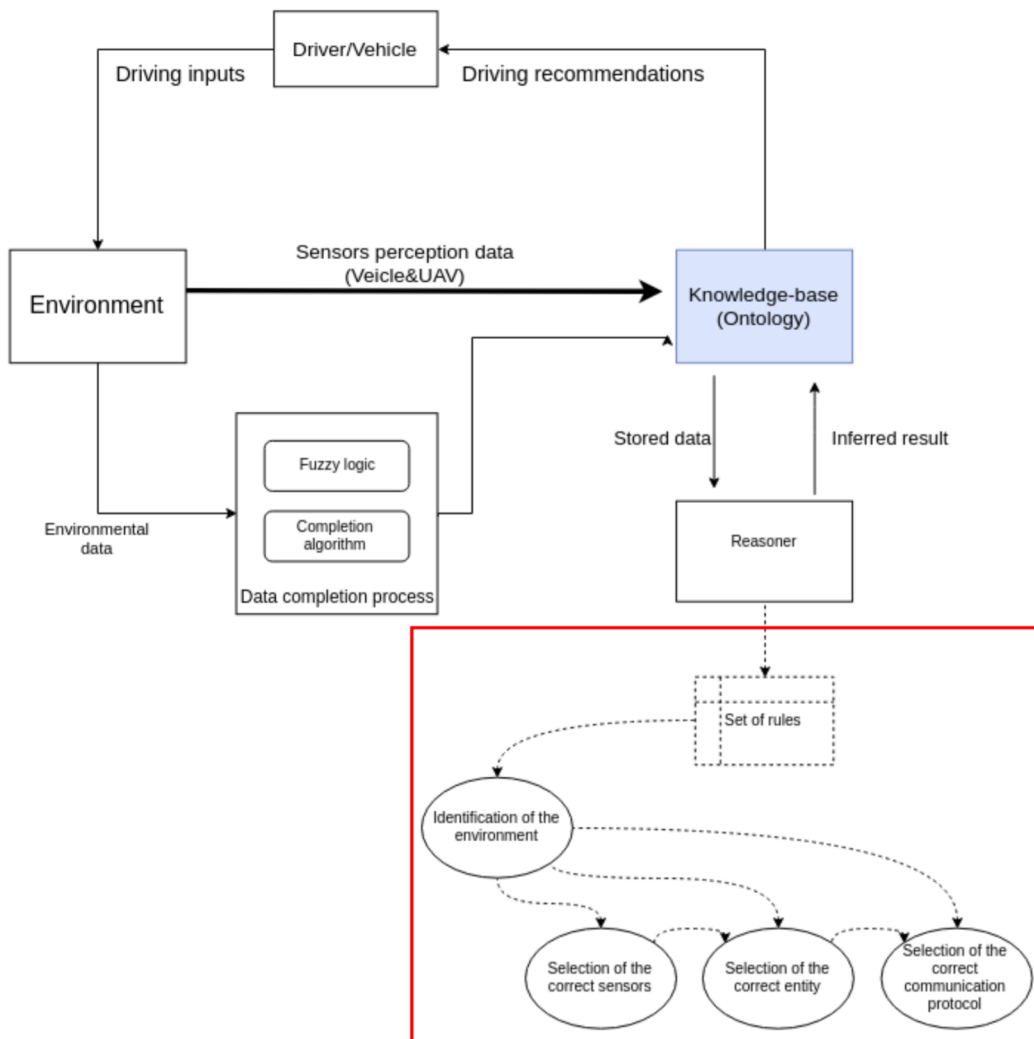


Figure 3.15 – Decision-making step in the general model

Due to the importance of a correct environmental detection, there is an additional Perception Accuracy variable (cf. Section 3.3.1) which makes the data only trusted if the general state of perception is considered good enough.

```

fogSensor(?fogS) ^ hasFogValue(?fogS, ?fogV) ^
PerceptionAccuracy(?P)^ Good(?P)^
swrlb:greaterThan(?fogV, 50) ^ Weather(?W) -> Fog(?W)

rainSensor(?rainS) ^ hasRainValue(?rainS, ?rainV) ^
PerceptionAccuracy(?P)^ Good(?P)^
swrlb:greaterThan(?rainV, 50) ^ Weather(?W) -> Rain(?W)

snowSensor(?snowS) ^ hasSnowValue(?snowS, ?snowV) ^
PerceptionAccuracy(?P)^ Good(?P)^
swrlb:greaterThan(?snowV, 50) ^ Weather(?W) -> Snow(?W)

```

Figure 3.16 – Detection of weather using sensors.

```

Sun(?S) ^ Environment(?Env) -> NormalEnvironment(?Env)

Fog(?F) ^ Environment(?Env) -> BadWeather(?Env)
Snow(?S) ^ Environment(?Env) -> BadWeather(?Env)
Rain(?R) ^ Environment(?Env) -> BadWeather(?Env)

```

Figure 3.17 – Classification of the environment depending of the weather .

The values from the environmental sensors are gathered and processed using the rules shown in Figure 3.16. Using the environmental sensors, if the detected value is above a certain threshold, the weather status is changed to the detected one (Fog, Rain or Snow). The set of rules in Figure 3.17 will then rely on the Weather variable in order to infer the Environment status. The default "Sun" value results in a Normal environment, while the other status will lead to a BadWeather environment.

A similar approach is taken for the identification of brightness level of an environment. The associated rule found in Figure 3.18 allows for the detection of a dark environment, which can influence the contrast of the road.

```
brightnessSensor(?brightS) ^ hasBrightnessValue(?brightS, ?brightV) ^
swrlb:lessThan(?brightV, 50) ^ Environment(?Env) -> Dark(?Env)
```

Figure 3.18 – Detection of brightness using sensors.

3.3.3.1.2 Entity and sensors activations

```
hasSensor(?U, ?cm) ^ UAV(?U) ^
cameraStereo(?cm) ^ Vehicle(?V) ^ Object(?O) ^
hasObstructedView(?V, ?O) ^ NormalEnvironment(?Env)
^isCloseToUAV(?V, ?U)
-> isActiveUAV(?U, true) ^
isActiveSensor(?cm, true)
```

```
Vehicle(?V) ^ hasDistanceFromVehicle(?O, NearDistance)
^ Environment(?Env) ^ Object(?O) ->
hasObstructedView(?V, ?O)
```

Figure 3.19 – Obstructed view

```
Fog(?f) ^ UAV(?u) ^ Radar(?r) ^ cameraInfra(?c) ^
hasSensor(?u, ?c) ^ hasSensor(?u, ?r) ->
isActiveUAV(?u, true)
```

Figure 3.20 – Activation of a UAV

```
UAV(?u) ^ Radar(?r) ^ cameraInfra(?c) ^
hasSensor(?u, ?c) ^ hasSensor(?u, ?r) ^ isActiveUAV(?u, true) ->
isActiveSensor(?r, true) ^ isActiveSensor(?c, true)
```

Figure 3.21 – Activation of the sensors of the UAV

Once an environment has been correctly assessed, the model needs to ask for perception enhancement. Figure 3.2.3 illustrates a simple activation scenario : suppose that a vehicle is in a situation wherein the view is obstructed. There is a UAV stationed there with adequate sensor (in this case, a simple camera). If the vehicle is close to the UAV (the **isCloseToUAV** condition is marked as **True**), then the system asks for the activation of

the UAV and its sensors, effectively meaning the start of data gathering and transmission by the drone.

3.3.3.1.3 Communication protocol

```

A. RF(?r) ^ hasCommunicationProtocol(?U, ?v) ^
   hasCommunicationProtocol(?U, ?h) ^ Hybride(?h) ^ VLC(?v) ^
   BadWeather(?b) ^ UAV(?U) ^ hasCommunicationProtocol(?U, ?r)
   -> isActiveCommunicationProtocol(?r, true) ^
   isActiveCommunicationProtocol(?v, false) ^
   isActiveCommunicationProtocol(?h, false)

B. RF(?r) ^ hasCommunicationProtocol(?U, ?v) ^ Dark(?d)
   ^ hasCommunicationProtocol(?U, ?h) ^ Hybride(?h) ^
   VLC(?v) ^ UAV(?U) ^ hasCommunicationProtocol(?U, ?r)
   -> isActiveCommunicationProtocol(?h, true) ^
   isActiveCommunicationProtocol(?v, true) ^
   isActiveCommunicationProtocol(?r, true)

C. RF(?r) ^ hasCommunicationProtocol(?U, ?v) ^
   FireHazard(?f) ^ hasBrightnessValue(?br, ?bv) ^
   hasCommunicationProtocol(?U, ?h) ^
   brightnessSensor(?br) ^ Hybride(?h) ^ VLC(?v) ^
   UAV(?U) ^ hasCommunicationProtocol(?U, ?r) ^
   UnusualEnvironment(?ue) ^ swrlb:greaterThan(?bv, 70)
   -> isActiveCommunicationProtocol(?r, true) ^
   isActiveCommunicationProtocol(?v, false) ^
   isActiveCommunicationProtocol(?h, false)

D. RF(?r) ^ hasCommunicationProtocol(?U, ?v) ^
   FireHazard(?f) ^ hasBrightnessValue(?br, ?bv) ^
   hasCommunicationProtocol(?U, ?h) ^
   brightnessSensor(?br) ^ swrlb:lessThan(?bv, 70) ^
   Hybride(?h) ^ VLC(?v) ^ UAV(?U) ^
   hasCommunicationProtocol(?U, ?r) ^
   UnusualEnvironment(?ue) ->
   isActiveCommunicationProtocol(?h, true) ^
   isActiveCommunicationProtocol(?v, true) ^
   isActiveCommunicationProtocol(?r, true)

```

Figure 3.22 – Communication protocol

As stated earlier, there are multiple possible communication protocols : a classical RF protocol, a VLC-only protocol in case of interferences, and an optimal hybrid protocol with a better quality of service thanks to the redundancy of data. The choice is made according to the environmental situation the vehicle is evolving in.

As shown in Chapter 2, VLC relies on light and performs poorly in a bad weather environment, because of the diffraction of the raindrops, fog and snowflakes. Rule A covers this situation but only activating the RF protocol in a bad weather environment.

However, those characteristics also allow VLC to bring light in spaces with poor illumination, enhancing the general visibility and perception in a dark area. Thanks to that, the hybrid approach presents a real interest in a Dark environment. This situation is illustrated by Rule B.

In some specific situations, such as in a Fire Hazard, the illumination of an environment can have an impact on the VLC sensors. Rules C and D cover two different scenarios where, depending of the brightness of a fire hazard, the VLC protocol is activated or not.

3.3.4 Summary of the process

This work revolves around the perception enhancement of an autonomous vehicle, a process done in four steps, as shown in Figure 3.23:

1. Environment detection : The system infers what kind of environment the vehicle is advancing in
2. Sensor activation : The system decides what sensors works best in the current environment
3. Entity activation : The system looks up nearby entities (UAV) carrying at least the adequate sensors

4. Communication protocol activation : The system selects the adequate communication protocol according to the environment.

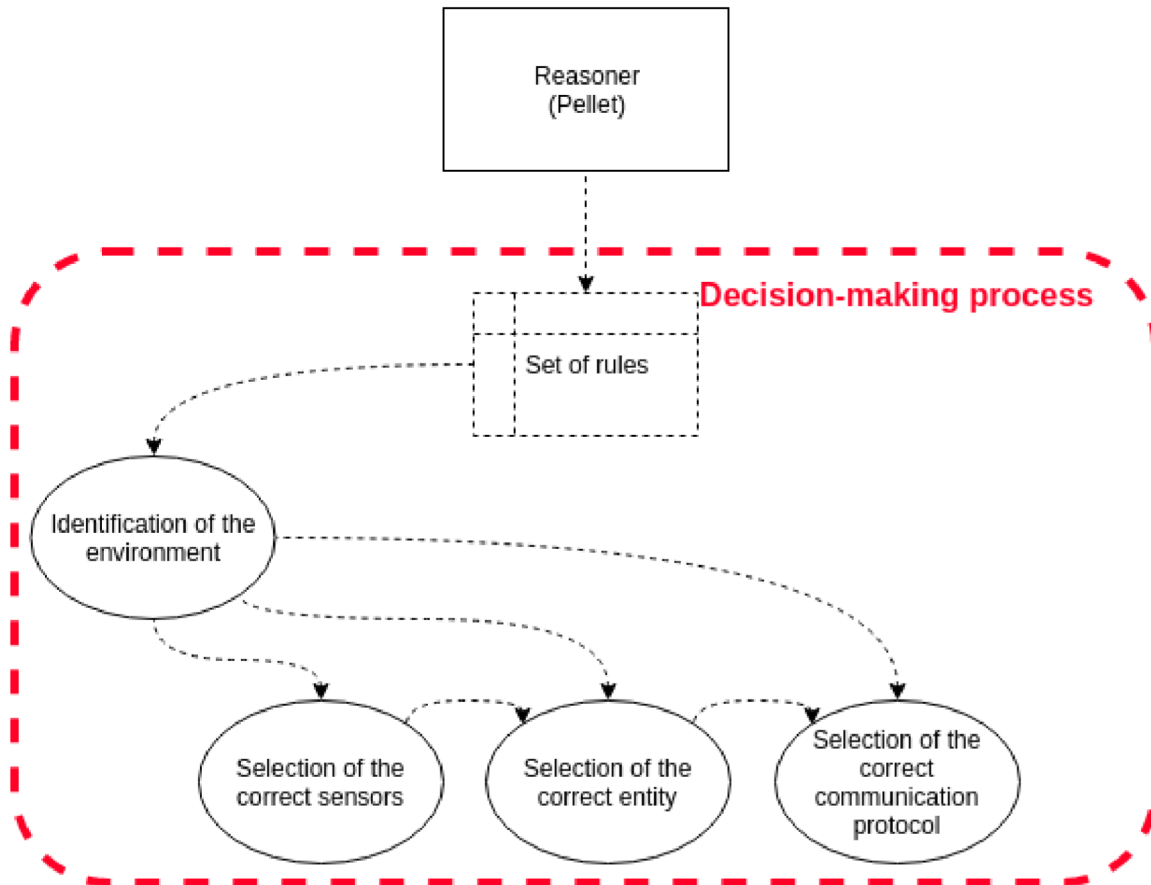


Figure 3.23 – Focus on the decision-making process

This modular approach ensures that all stacks of the process are addressed according to their own parameters. The environment situation plays the most important role into the proper functioning of the elements, but some other variables can influence each steps, such as if the sensors are functional or not, or if the correct ones are present on the UAV or not.

The reasoner will consult the set of rules and apply them on the elements classified in the knowledge-base. It will first use the available elements in order to identify the environment, as shown in Figure 3.17. This will allow for the correct identification of the context where the vehicle is evolving, and will have a direct impact on the resulting reasonings:

- Thanks to the logical rules and data properties, the reasoner can correctly identify which sensors can function well and which ones are weak in a particular situation.
- The knowledge-base also contains a list of different entities (UAV) available on site and their embedded sensors, allowing for the system to select which ones should be requested.
- Is also included the different communication protocols available on each entity, as well as their respective weaknesses.

The logical rules were implemented by taking into consideration all these elements, and allow for the reasoner to output different inferred elements: **Current environment, request assistance on a specific entity, activation on specific sensors, activation of specific communication protocols and, if possible, driving recommendations.**

It is important to remember that an inference engine will run all the rules simultaneously, meaning that the results are processed at the same time. However, for a specific situation, if one of the steps is not validated, it will cancel the whole process. For example, if the vehicle is in a rainy area and has no rain sensors, the environment will not be correctly assessed and the perception enhancement process will not happen. Or if a UAV does not have the correct sensors, then it will not be asked to transmit perception data.

A simple scenario: suppose that a vehicle is approaching an intersection where an obstacle (a rock) is limiting its visibility, and the environment is a normal one, with correct brightness and a sunny weather. There is a UAV stationed at the intersection. The sensors of the vehicle will log their outputs in the knowledge-base and the reasoner will infer that there is a need for perception enhancement. It will check that the UAV possesses the correct sensors, in this case a simple camera, and will request its help. It will also decide on the communication protocol depending on the environment. The hybrid protocol works fine in those conditions, and offers a better integrity of the data thanks to the

transmission redundancy. The UAV informs the vehicle of an incoming vehicle, as well as its position, allowing the ego vehicle to take act accordingly to the situation.

3.4 Conclusion

This chapter presented the notion of a knowledge-base as well as the ontology developed for this thesis. The different classes making up the system were introduced, as well as the different properties linking them. The individuals are managed through a system of rules.

The knowledge-base gathers various driving and surrounding information from the vehicle's sensors and store them unto the appropriate classes, as well as their different properties and relations. The inference engine is then called to process all those elements with the use of logical rules, and outputs its conclusions : the status of the environment as well as the actions that should be taken. If the visibility is limited, and if an adequate UAV is available nearby, then it will be requested to collect additional data thanks to its sensors and transmit these data to the vehicle.

That last part is also critical because perception data may be seen as sensitive and their broadcasting needs to be done in a safe way. Indeed, the correct selection of the communication protocol is also managed by the knowledge-base, based on the available ones and their compatibility with the current environment.

The details on the secured communication protocols is covered in the next chapter.

Chapter 4

Secured communication process

Contents

4.1	Introduction	88
4.2	Secured communications	88
4.2.1	Cryptography	90
4.2.2	Physical layer security	91
4.3	Communication protocols	92
4.3.1	RF communication	92
4.3.2	VLC	94
4.4	Hybrid vehicular application	99
4.4.1	Advantages of a hybrid approach	99
4.5	Typical applications	106
4.5.1	Train station use case	106
4.5.2	UAV/Car application	108
4.6	Conclusion	110

4.1 Introduction

In the context of the emergence of IoT and its implementation in smart cities, new issues have appeared, notably in terms of services, infrastructures and safety. The onset of autonomous vehicle and UAV are one of the most important reasons of these issues, including the passengers' safety and traffic issues. Considering the consequent intelligent elements evolving in the road environment, it is vital to ensure that they have some mean to communicate between them. VLC (Visible Light Communication) and Li-Fi (Light Fidelity) are communication protocols that use visible light as information vector. The emitters of these systems are luminous sources (LEDs or LASER) blinking at a high enough frequency that it becomes invisible to the human eye. The blinking process allows for the transmission of a signal. Equipped with emitters and sensors, the different parts of the network can communicate through this system.

This chapter will introduce communication security, before focusing into the strengths and weaknesses of RF and VLC for vehicular communications. It will then present a hybrid communication protocol merging both technologies and illustrates a few scenarios where it would be interesting to use it.

4.2 Secured communications

Security threats are a key issue in today's connected environment. They may compromise the integrity of private data and could even endanger the lives of people. As a result, real concerns about security have led to the advent of multiple research tracks in order to ensure the protection of information.

The primary focus of this thesis is not on communication security, but it is still important to know where the proposed solution fits in the existing methodology.

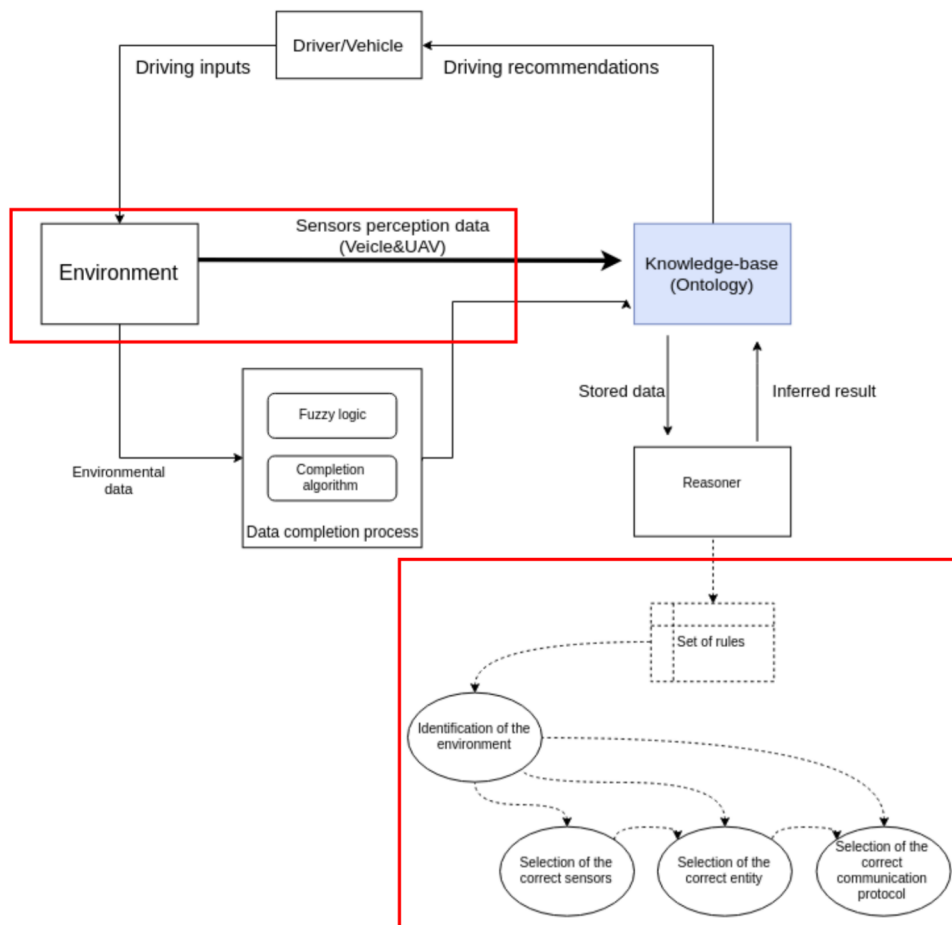


Figure 4.1 – The communication part happens for data transfer between entities in the same environment. The protocols are also managed by the set of rules used by the reasoner.

4.2.1 Cryptography

Encryption is a concept where a computational algorithm is used in order to alter the content of a message and generate a new unintelligible message instead. The receiver will then need a security key in order to backtrack the process and find the clear message. A security key is a string of characters or numerals that is used with the associated decoding algorithm in order to understand the encoded message.

Encryption is one of the oldest and most common way of protecting data. Julius Caesar proposed a cipher system based on letter shifting[87], and the most advanced protocols rely on quantum mechanic in order to guarantee unbreakable protection[88].

There are two different types of encryption algorithms used for data security :

- **Symmetric:** The key used for the encryption is the same one used for the decryption. This means the distribution of the keys needs to be done before the transmission of information. Some algorithms using this approach : Triple DES, Blowfish, AES.
- **Asymmetric:** Two different keys are used for encryption, a public and a private key. The public key is available for everyone and can be used for the encryption of any message. The private key is kept secret and can only be used by the authorized agents for the decryption of the message. Some algorithms using this approach : RSA, DSS.

The actual applications can take a variety of forms, sometimes even mixing multiple approaches. For example, the Off-the-Record approach [89] relies on the AES encryption and the SHA-1 hashing in order to ensure an optimal security. It also guarantees that previous messages cannot be read in case the key is lost.

Identity-based encryption is another way of securing communication where the encryption

key is defined by the receiver's identity. The sender must first ensure that the receiver is the correct one, and only if so a secured link is established between them. This ensures that no malicious entity can access the message [90].

Encryption is then considered as a major way of protecting data privacy, but it is a method relying mainly on software and mathematical approaches. It is possible to bring data protection to an even lower abstract level.

4.2.2 Physical layer security

Most of the modern communications rely on electromagnetic RF technologies. There are many advantages to this approach, for example the possibility to broadcast to multiple agents or to send a message through opaque obstacles (walls). However, these properties can also cause security issues because the signals can be intercepted or jammed by ill-intentioned individuals.

These concerns stem from physical vulnerabilities, and can be addressed the same way. As shown in Section 4.2.1, encryption algorithms rely on public and private keys, usually digitally generated and stored. Saving the private key on a physical media would drastically reduce the number of potential attackers[91]. Security between two different intelligent agents can also be reinforced by ensuring that a physically-wired connection is required for them to communicate [92]. New communication technologies could allow the use of an alternative physical link.

Taking everything into consideration, there are multiple ways to ensure that a communication between two agents is secured. The application presented in this thesis revolves around Vehicle/UAV interaction, meaning that this aspect should not be left out. The PHY layer (Physical layer) security increase would work well in our situation, especially by using new technologies such as VLC.

4.3 Communication protocols

4.3.1 RF communication

Wireless communication is the main method of communication in modern technological agents. There are multiple wireless communication protocols [93], such as WiFi, Bluetooth, or even DSRC (Dedicated Short Range Communications) for vehicular communication. These protocols however have their own set of drawbacks.

Indeed, communication issues can stem for a variety of reasons, such as the presence of blocking material (concrete) or electromagnetic interference. These interferences can come from malignant jamming or more natural causes. There are different events on a road environment that can negatively impact RF communications, for example a Fire Hazard.

Heat can be transferred through three different ways [94]:

- Conduction : The energy is transferred from one molecule to the other through direct contact.
- Convection : The energy is transported by a moving fluid, such as water or air.
- Radiation : The energy is transferred by electromagnetic waves.

Wireless protocols rely on electromagnetic waves in order to transmit data. The presence of other mitigating signals could alter the proper communication between two agents. A high enough temperature is already known to be able to generate an electrical voltage, as proven by the Seebeck Effect[95], but there are two ways by which fire could have an impact on electromagnetic waves:

- Environment change : This situation is mostly known for underwater communications[96], where the attenuation factor of water restricts the communication range of RF protocols. In a broader way, the phase difference between the air/water interface

creates a discontinuous environment that can alter the propagation of electromagnetic waves [97]. A similar concept can be applied to a cold air/hot air situation if the heat difference is consequent enough.

- Electromagnetic generation : At a high enough temperature, fire can produce plasma through air, which is one of the fourth state of matter. The exact conditions to generate plasma are not exactly known with the current physics knowledge, but plasma is known to have electromagnetic components by ionizing the air around it[98].

Hence, in addition to the obvious damages a fire hazard could cause to communicating infrastructures, there is also the risk of data being directly corrupted.

4.3.1.1 High-temperature behaviour

In order to verify these theories, some basic tests were made: A wireless communication was established between a computer and a SOC (System On Chip), and a controlled fire was ignited between them. Data were then exchanged and monitored.

In order to increase the chances of having the electromagnetic waves passing through the ignited area, the devices were put in Faraday Cages. A Faraday cages is an enclosure used to block electromagnetic fields : by wrapping a space in multiple layers of conductive material, the incoming electromagnetic waves will be absorbed by the charges of the covering material. As shown in Figures 4.2b-4.2a, there is a clear attenuation of the signal depending of the orientation of the receiver.

An illustration of the process can be found in Figure 4.3.

The data were sent with the **ping** command. Packages were sized at 1024 bytes and sent every 10ms. Around 5000 packages were sent for each experiment. The fire was located near the emitter.

Results showed that the fire only had a small impact on the travelling time of packages, with an average speed of **1.93 ms** for the unlit fire situation and **2.1ms** for the lit fire

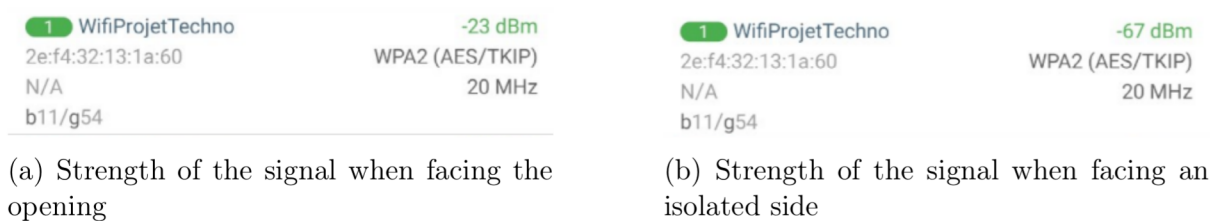


Figure 4.2 – Strength of the wireless signal depending of the orientation of the antenna

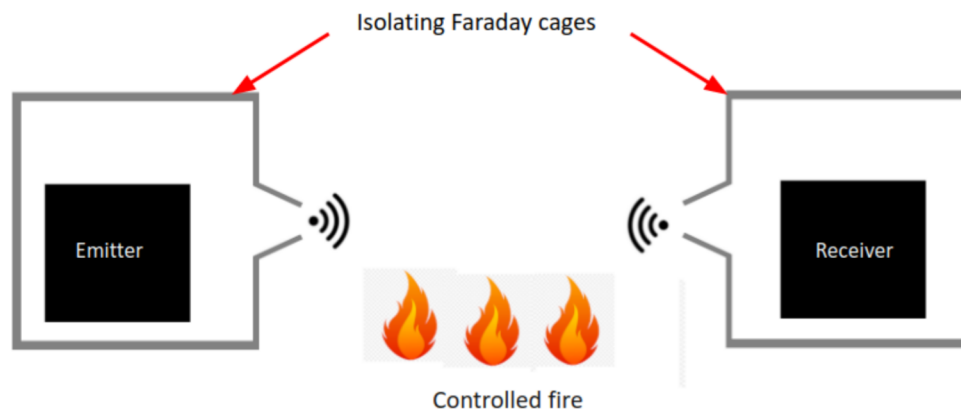


Figure 4.3 – Illustration of the fire experiment

situation. There was however a difference in the packet loss rate : **36** packets were lost for the unlit fire situation, and **129** for the lit fire situation. In the various repeated experiments, the highest BER (Bit Error Rate) reached was **0.0314**.

Tests were made by making adjustments to the temperature of the fire, varying between 350°C and 1750°C. A higher temperature seems to have more impact on the communication.

Although elemental, the results of this experiments reinforced the idea that a fire hazard would have an effect on both communication structures and wireless protocol.

4.3.2 VLC

VLC (Visual Light Communication) is an optical wireless communication (OWC) technology wherein the signal is produced by a light source and the data are transmitted over the visible light spectrum. The transmitter can either be a LED or a Laser Diode (LD). It has been around for many decades now, the first practical use being by Alexander

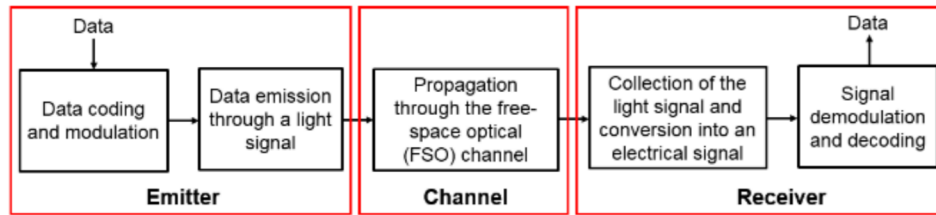


Figure 4.4 – Steps of a VLC communication process

Graham Bell in 1880, who invented the photo-phone : By vibrating a mirror at a specific frequency, he managed to modulate a voice signal and receive it over 200m away with a parabolic receiver.

The invention of light amplification by stimulated emission of radiation (laser) enabled a breakthrough in VLC communication : US scientists in the 1960's managed to transmit a voice signal modulated by laser over 190km. Nowadays Japanese road networks use IR (Infra-Red) devices to send traffic information directly to the vehicles: There are more than 30000 IR beacons over the Japanese roads in a system called Vehicle Information and Communication Systems (VICS)[36]. While VICS is not technically VLC, it proved that we could use near-IR signals for V2X communications[37].

In general, the concept of VLC is a simple one: by modulating light above a certain frequency, information can be transmitted while ensuring there is no noticeable difference for the human eye. The different steps of the technical process can be found in Figure 4.4[39].

As shown, the process is split in different steps :

Data emission In electronic communications, data are constituted of a succession of bits. In order to improve the size, readability and security of the transmission, the data can be encoded and modulated. **Coding** is the process where the bits are regrouped according to different patterns and algorithms in order to compress the overall size of the payload, and reduce the amount of data that needs to be transmitted. **Modulation** depends of the mean of transportation and aims to adjust the carrier signal into the



Figure 4.5 – Headlight of a vehicles used for data transmission

adequate form depending of the physical transportation method.

VLC being different from classical radiowaves-based protocol, there are multiple ways of modulating a signal, for example Intensity Modulation of the light for generating data according to the brightness of the LED. However, the most popular approach would also be the easiest one : the On-Off-Keying (OOK) consists of turning the LED on and off depending of the bit to be sent (On for 1, Off for 0), and over a certain frequency is completely transparent for the naked eye.

The message emission can be done with the use of an LED. There is a significant advantage in doing so considering LEDs are present in a lot of existing infrastructures (vehicles, traffic light, etc.) and only need signal modulation in order to transfer data. In Figure 4.5, a car's headlights are used for transmission

Channel Transmission A VLC signal is emitted by using a light source, generally a LED bc of the advantageous reliability, robustness and low-power consumption. While other traditional means of communication rely on either physical connections (wires, optical fiber) or high-frequency radio spectrum, VLC messages travel through natural environment, referred to as the Free-Space Optical (FSO) environment. This can be air, outer space or vacuum.

"Light" has a frequency ranging from 10nm to 1mm, including Ultraviolet and Infrared.

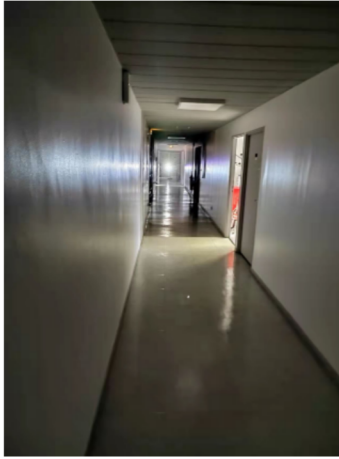


Figure 4.6 – Distance between the emitter and the receiver

Visible light is comprised between 400nm and 800nm. Compared to the wavelengths of other frequencies, light can not go through opaque objects. This is the reason why VLC requires a clear line-of-sight between the emitter and the receiver.

Depending of the environment, some parameters need to be considered too, such as light diffraction or light noise. Those can be treated by using additional tools, for example lenses or filtering algorithms .In Figure 4.6, the emitter and the receiver are separated by 15m in an indoor environment.

4.3.2.0.1 Data reception This is the opposite and complementary step to the Emission process. A light receiver is used in order to gather the transmitted signal, which is then demodulated and decoded in order to find the original data payload.

There are different ways to read the incoming VLC data. The two most popular method are through the use of a camera or a Photodiode (PD). Cameras with a good enough resolution and processing power can catch on the smallest variation of brightness and reconstruct the message from it. A PD is a semiconductor device that can convert a stream of photons into a current, and can generate an electronic message by receiving a modulated light. In Figure 4.7, the receiver is a PD which is logging the data sent by the emitter in Figure 4.5.

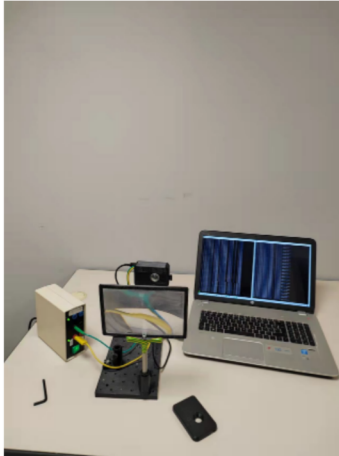


Figure 4.7 – Photodiode as a receiver

4.3.2.1 Benefits of VLC

This thesis focuses on the communication between a vehicle and an UAV. In that regard, VLC offers various benefits as a communication protocol[39] :

- The visible light spectrum corresponds to a huge bandwidth of around 300 THz that is not regulated. In comparison, the RF communications cover a range of 400GHz which are extremely regulated and overcrowded.
- The corresponding frequencies do not interfere with radio waves and cannot pass through obstacles, which means VLC system can be used along with RF systems while providing an increased privacy
- The LED, on the other hand, can be modulated at very high speed which means data rates of hundreds of Mbps can be achieved with adapted modulations and off-the-shelf, low-cost components
- The information is carried through visible light. This would mean that an ill-intentioned individual would need to be physically close to the source in order to intercept the data.

A summary of the main differences between VLC and RF communications can be found in Table 4.1.

Property	VLC	RF
Bandwidth	400-800nm	300 GHz
Line Of Sight	Yes	No
Range	Short to medium	Short to long
Service	Illumination and communication	Communication
Noise sources	Sunlight and light sources	Electromagnetic interferences
Power consumption	Low	Medium
Mobility	Limited	Good
Coverage	Narrow	Wide

Table 4.1 – Comparison of various properties of VLC and RF

4.3.2.2 Setup validation

The VLC setup used for making tests was based on the OpenVLC project[99]. It is an open-source, low-cost VLC platform allowing for the fast prototyping of VLC communication scenarios. The VLC setup can be seen in Figure 4.8: There is an emitter system plugged on a car’s taillight and interfaced with the network commands, while the receiver was positioned at a 2m distance with a lens helping focus the signal. The network payloads are transmitted through the taillight LED matrix and logged by the receiver. For the basic communication experiments, the system was evaluated on the *iperf* network measurement tool[100], sending packages at the speed of 100kb/s, a frequency at which the blinking was undetectable to the human eye. As shown in Figure 4.9, the communication works perfectly with only **0.34%** of data loss, due to some slight displacement of the focusing lens.

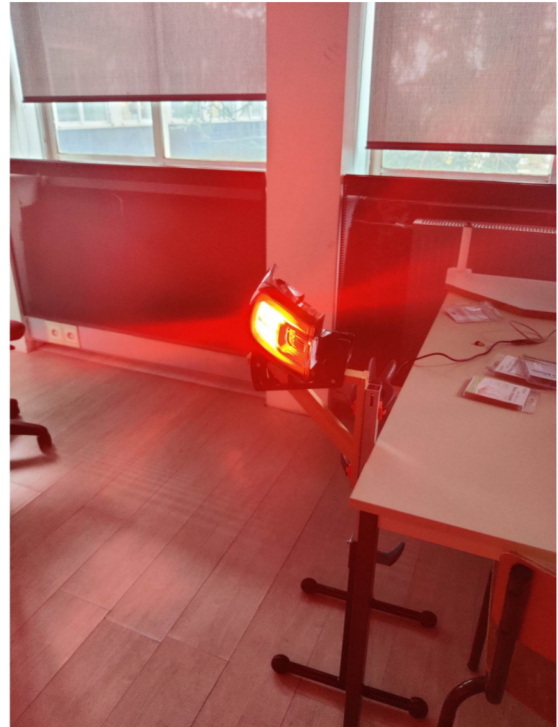
4.4 Hybrid vehicular application

4.4.1 Advantages of a hybrid approach

No matter all its benefits, no system is perfect on its own. VLC is also considered as a complementary means in hybrid communications.



(a) Transmitter VLC



(b) Receiver VLC

Figure 4.8 – Setup of the VLC testing

```

debian@beaglebone:~/r$ cat test.txt
-----
Server listening on UDP port 10001
Receiving 100 byte datagrams
UDP buffer size: 160 KByte (default)
-----
[ 3] local 192.168.0.2 port 10001 connected with 192.168.0.4 port 36595
[ ID] Interval      Transfer    Bandwidth   Jitter    Lost/Total Datagrams
[ 3] 0.0- 3.0 sec  36.5 KBytes 99.7 Kbits/sec  0.664 ms  0/ 374 (0%)
[ 3] 3.0- 6.0 sec  36.6 KBytes 100 Kbits/sec  0.742 ms  0/ 375 (0%)
[ 3] 6.0- 9.0 sec  36.7 KBytes 100 Kbits/sec  0.638 ms  0/ 376 (0%)
[ 3] 9.0-12.0 sec  36.5 KBytes 99.7 Kbits/sec  0.705 ms  0/ 374 (0%)
[ 3] 12.0-15.0 sec 36.7 KBytes 100 Kbits/sec  0.525 ms  0/ 376 (0%)
[ 3] 15.0-18.0 sec 36.5 KBytes 99.7 Kbits/sec  0.740 ms  0/ 374 (0%)
[ 3] 18.0-21.0 sec 36.6 KBytes 100 Kbits/sec  0.335 ms  0/ 375 (0%)
[ 3] 21.0-24.0 sec 36.7 KBytes 100 Kbits/sec  0.440 ms  0/ 376 (0%)
[ 3] 24.0-27.0 sec 36.5 KBytes 99.7 Kbits/sec  0.586 ms  0/ 374 (0%)
[ 3] 27.0-30.0 sec 36.7 KBytes 100 Kbits/sec  0.844 ms  0/ 376 (0%)
[ 3] 30.0-33.0 sec 36.6 KBytes 100 Kbits/sec  0.803 ms  0/ 375 (0%)
[ 3] 33.0-36.0 sec 36.6 KBytes 100 Kbits/sec  0.467 ms  0/ 375 (0%)
[ 3] 36.0-39.0 sec 36.5 KBytes 99.7 Kbits/sec  0.162 ms  0/ 374 (0%)
[ 3] 39.0-42.0 sec 36.7 KBytes 100 Kbits/sec  0.361 ms  0/ 376 (0%)
[ 3] 42.0-45.0 sec 36.6 KBytes 100 Kbits/sec  0.423 ms  0/ 375 (0%)
[ 3] 45.0-48.0 sec 36.6 KBytes 100 Kbits/sec  0.354 ms  0/ 375 (0%)
[ 3] 48.0-51.0 sec 36.6 KBytes 100 Kbits/sec  0.512 ms  0/ 375 (0%)
[ 3] 51.0-54.0 sec 36.6 KBytes 100 Kbits/sec  0.295 ms  0/ 375 (0%)
[ 3] 54.0-57.0 sec 36.6 KBytes 100 Kbits/sec  0.228 ms  0/ 375 (0%)
[ 3] 57.0-60.0 sec 36.6 KBytes 100 Kbits/sec  0.257 ms  0/ 375 (0%)
[ 3] 60.0-63.0 sec 36.6 KBytes 100 Kbits/sec  0.279 ms  0/ 375 (0%)
[ 3] 63.0-66.0 sec 36.6 KBytes 100 Kbits/sec  0.180 ms  0/ 375 (0%)
[ 3] 66.0-69.0 sec 36.6 KBytes 100 Kbits/sec  0.281 ms  0/ 375 (0%)
[ 3] 69.0-72.0 sec 36.6 KBytes 100 Kbits/sec  0.093 ms  0/ 375 (0%)
[ 3] 72.0-75.0 sec 33.0 KBytes 90.1 Kbits/sec  0.180 ms  37/ 375 (9.9%)
[ 3] 75.0-78.0 sec 36.1 KBytes 98.7 Kbits/sec  0.163 ms  5/ 375 (1.3%)
[ 3] 78.0-81.0 sec 36.6 KBytes 100 Kbits/sec  0.130 ms  0/ 375 (0%)
[ 3] 81.0-84.0 sec 36.6 KBytes 100 Kbits/sec  0.107 ms  0/ 375 (0%)
[ 3] 84.0-87.0 sec 36.6 KBytes 100 Kbits/sec  0.117 ms  0/ 375 (0%)
[ 3] 87.0-90.0 sec 36.6 KBytes 100 Kbits/sec  0.070 ms  0/ 375 (0%)
[ 3] 90.0-93.0 sec 36.6 KBytes 100 Kbits/sec  0.069 ms  0/ 375 (0%)
[ 3] 93.0-96.0 sec 36.6 KBytes 100 Kbits/sec  0.341 ms  0/ 375 (0%)
[ 3] 96.0-99.0 sec 36.6 KBytes 100 Kbits/sec  0.040 ms  0/ 375 (0%)
[ 3] 0.0-100.0 sec 1.19 MBytes 99.7 Kbits/sec  0.195 ms  42/12501 (0.34%)

```

Figure 4.9 – Iperf results

In [101], it is stated that RF communication is sensitive to jamming attacks and interference, and even if the use of Cognitive Radio (intelligent detection of unused transmission channels) can minimize the risks, they still propose the addition of VLC communication to strengthen security. Another experiment was conducted by [102], where a joint 5G/VLC prototype was set up. The smart city sensors data were gathered and transmitted to the road infrastructures (traffic light) through 5G, and then to the cars via VLC. This hybrid solution allows for the data to quickly reach vehicles while making sure the wireless network is not saturated. A similar study is led by Rahaim et al. [103], where VLC would act as a complementary protocol that would take over when WiFi reaches maximum capacity. In 2016, Rakia et al. [104] introduced a dual-hop data transmission system. The first hop transmits data on VLC to a relay node where an RF protocol will take over. In order to optimize the energy consumption, the DC component of the received optical signal is harvested and then used to power the RF communication. The proposed system showed great throughput results, even if the DC bias and power-harvesting component could still be improved, according to the authors. The work of Pan et al. [105] was also based on a VLC energy-harvesting feature in a hybrid RF/VLC settings, this time focusing more on data privacy. The hybrid VLC/RF approach allows to ensure only the designated receiver acquires the message, preventing eavesdropping.

Section 3 introduced the global concept of this work as well the different communication protocols considered in this study (RF, VLC and Hybrid) A VLC/RF hybrid approach is suitable in this situation because :

- Intelligent Transport Systems can, by design, communicate with their surroundings [106]
- VLC is a technology revolving around light, making for a brighter environment.
- The redundancy of information allows for a more secured communication and robust system.

The method proposed in this work is to have VLC and RF as a hybrid communication

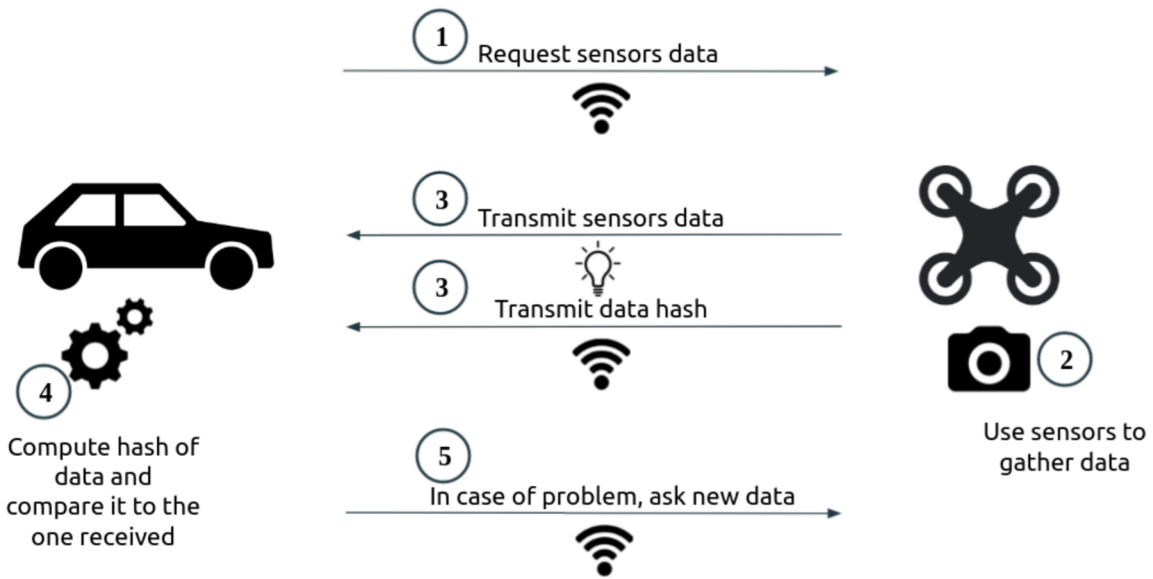


Figure 4.10 – Illustration of the proposed communication protocol

protocol in order to ensure that no information is lost during data transmission between 2 agents : The "heavy" data is sent through VLC, because of its high-speed and reliability, and the hash of the data (much smaller and used to verify the integrity of the transmitted information) are sent through an RF channel. An illustration of this process can be found in Figure 4.10.

4.4.1.1 Hashing algorithms

"Hashing" refers to a process where data are passed through a function that produces a fixed-sized string of characters. There are some benefits to hashing, and it is a useful tool in data security and data integrity: the same set of data will always produce the same string of characters as output, meaning that if the initial information is compromised, even by a few bits, the returned hash will be completely different from what is expected.

There are indeed many types of hash functions, with different level of safety guarantees. The concept proposed in this thesis focuses on the protection against data loss during transmission and speed transmission, and to this end a basic hashing algorithm such as MD5 is acceptable [107]. Pamula et Ziebinski [108] proposed the real-time hashing of

Table 4.2 – Transmission speed of some VLC studies

Study	Speed	Transmission time (for 200Gb)
Haigh et al. 2013 [110]	3Mb/s	18hrs
Haigh et al. 2016 [111]	170Mb/s	19mins
Shi et al. 2019[112]	5Gb/s	40s

a buffered live video stream by using FPGA (Field Programmable Gate Arrays). That study managed to reach a hashing throughput of more than 400Mb/s for blocks of 512 bits thanks to hardware acceleration, generating a signature for each frame.

An FPGA validation system setup at the receiver would then be able to generate the hash of 1Gb of data in about 65ms, and quickly ask for a resend if an error is detected.

4.4.1.2 VLC transmission speed

As stated in the previous sections, VLC also presents an advantageous transmission speed. This would mean that, for a similar Round-Trip Delay (RTD), the size of the data frames would be bigger than when using RF communication. This would allow for the acknowledgement process and the hashing data check to happen more regularly too, enabling a faster transmission of the data.

Depending on the type of LED and receivers used, the transmission rate can greatly vary: Table4.2 offers an illustration of some of the throughputs reached by other works, as well as the average unloading time of 200Gb of data. The first study used of OLED in order to reach a speed of 3Mb/s, and the same team made an Artificial Neural Network as an equalizer and high speed receivers in order to improve the speed to 170Mb/s. Another study made use of FPGA and 64 QAM (Quadrature Amplitude Modulation[109]) to reach a VLC throughput of 5Gb/s.

With the use of VLC the data can then be transmitted at an extremely high-speed, ensuring a good transmission of the gathered data. Most digests generated by hashing

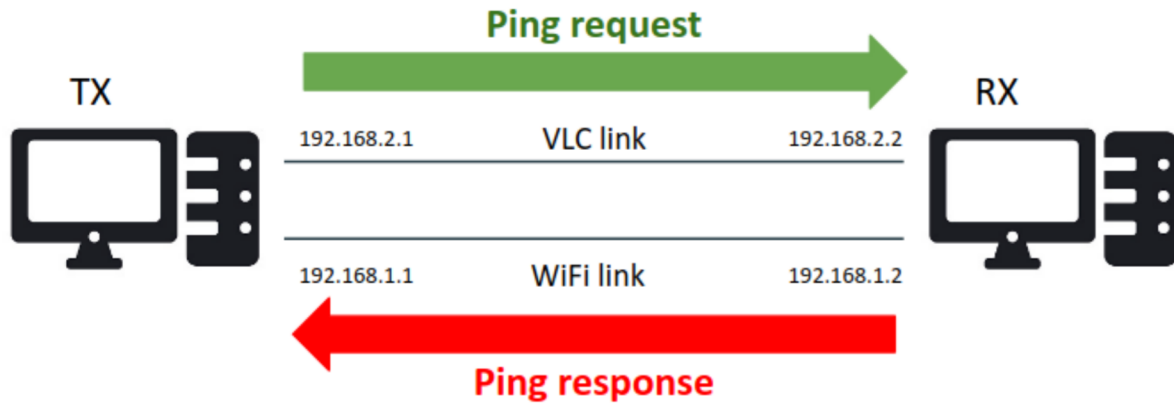


Figure 4.11 – Illustration of the hybrid testing setup

functions are only a few hundred bits long, making their transmission via RF light. In addition, and as stated above, the computing of a new hash can be reduced to a few nanoseconds with the use of FPGA technology, guaranteeing a fast verification of the received data.

4.4.1.3 Hybrid setup

Once the setup was validated in Sec 4.3.2.2, we tested a hybrid communication approach : both sides were equipped with WiFi dongles and a wireless sub-network was established between them, in addition to the VLC one already existing. The **ping** function was then used for validation : the emitter would send a payload by VLC, and the receiver would acknowledge it using WiFi.

An illustration of the process can be found in Figure 4.11. The transmitter TX and the receiver RX are connected to two different sub-networks : a WiFi wireless one, and a VLC one. The devices are equipped of the physical accessories to connect on both networks (WiFi dongles and VLC physical layers), and are assigned an IPv4 address for each interface. Each technology using a different sub-network, they have different network address: they can communicate via WiFi on the 192.168.1.X sub-network, and with VLC using the 192.168.2.X addresses

This setup would allow us to experiment on hybrid communication. The concept proposed

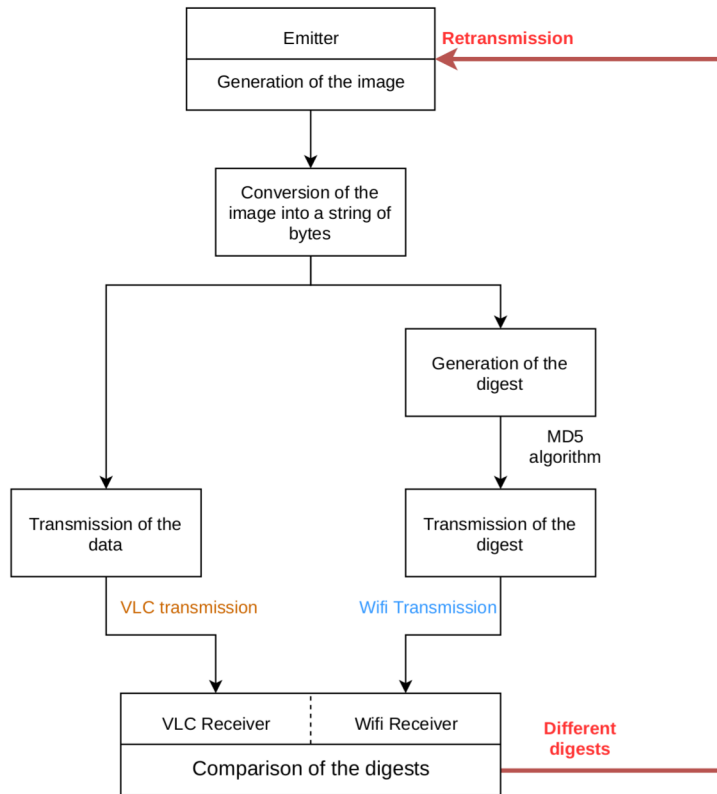


Figure 4.12 – Illustration of the hybrid testing setup

in this thesis is illustrated in Figure 4.12.

- First, the image data is turned into a string of bytes.
- The MD5 digest is generated using the string of bytes. The output is a much smaller string of data attesting of the integrity of the data.
- On the emitter side, two communication sockets are opened, managed by two different threads. The first socket is on the VLC sub-network, while the second in on the WiFi one.
- On the first communication channel, the raw string of bytes is transmitted.
- On the second communication channel, only the generated digest is transmitted.
- The receiver gathers data from both channels. It will use the MD5 algorithm on the raw data in order to generate the digest and compare it to the received one.

With this approach, the integrity of the data would be ensured thanks to the redundancy. Using two parallel channels allows for a fast detection of error, and enables a quick answer in case of detection of data corruption.

4.5 Typical applications

The hybrid approach shows an interesting potential for the transmission and verification of heavy data. There are some scenarios where this approach would be interesting.

4.5.1 Train station use case

This is an ideal scenario which supports the hybrid communication concept presented beforehand. The use case chosen here is *a train transmitting an important amount of data to a train station*. In 2016, Ahamed [113] underlined that the UK Network Rail, similarly to the APTA (American Public Transportation Association) , requires all footage inside trains to be downloaded on a server when entering a depot/siting [114], and proposed the idea of using VLC in order to do so. The concept proposed here is based on the hybrid approach. The train uses its headlights in order to communicate with photosensors located in front of it. Once the communication is established, the video data are transmitted via VLC into the station, after being parsed in smaller packets. In parallel, the hash of each packet is transmitted via RF to the receiving server, which will compare it to the one it computes from the transferred data. If an error occurs, the receiver can ask for a re-send of the previous packet

Indeed, The VLC/RF hybrid approach is suitable in this situation because :

- The train is immobile and usually sheltered from light noise sources (the sun)
- Trains already possess light bulbs, and can modulate the output in order to communicate

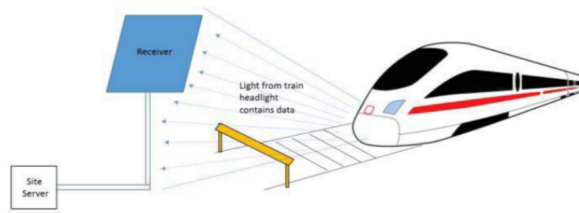


Figure 4.13 – Illustration of the usecase in Ahamed work[113]

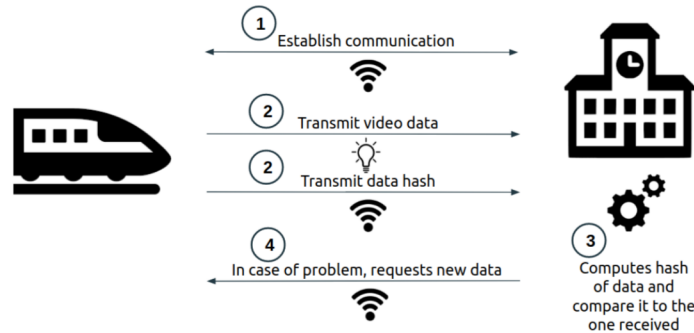


Figure 4.14 – Illustration of the proposed model for hybrid train application.

- Train stations infrastructures are usually made of metal and concrete [115], making it hard for RF waves to pass through[116].

As stated by the last point, the architectural infrastructure of a train station can make RF communication difficult, so it would be interesting to pair this protocol with a complementary channel of communication.

The concept proposed in this section is illustrated in Figure 4.14. The train uses its headlights in order to communicate with stationary photo-sensors located at the end of the line ahead of it, as illustrated by Ahamed in Figure 4.13. Once the communication is established, the video data are transmitted via VLC into the station, after being parsed in smaller packets. In parallel, the hash of each packet is transmitted via RF to the receiving server, which will compare it against the one it computes from the transferred data. If an error occurs, the receiver can ask for a re-send of the previous packet.

This scenario could be one potential application for the hybrid system.

```

UAV(?U) ^ hasSensor(?U, ?cm) ^
cameraStereo(?cm) ^ Vehicle(?V) ^ hasObstructedView(?V, ?O) ^
NormalEnvironment(?Env) ^ isCloseToUAV(?V, ?U)
-> isActiveUAV(?U, true) ^ isActiveSensor(?cm, true)

```

Figure 4.15 – Camera activation

```

UAV(?U) ^ RF(?r) ^ Hybride(?h) ^ VLC(?v) ^
hasCommunicationProtocol(?U, ?v) ^
hasCommunicationProtocol(?U, ?h) ^
hasCommunicationProtocol(?U, ?r) ^
NormalEnvironment(?e)
-> isActiveCommunicationProtocol(?r, true) ^
isActiveCommunicationProtocol(?v, true) ^
isActiveCommunicationProtocol(?h, true)

```

Figure 4.16 – Communication protocol activation

4.5.2 UAV/Car application

This thesis focuses on the use of a UAV for perception assistance of a vehicle. The UAV uses its sensors to gather information and send them to the vehicle.

There are multiple possible sensors that can be embedded, for example a camera. In Chapter 3, we presented the logical rules that could result in the data gathering request and the activation of sensors. A simple camera activation rule can be found in Figure 4.15. These same rules can be used for the management of communication protocols. In Figure 4.16 there is a rule covering a simple case: if a UAV is equipped with both RF and VLC technologies (making it equipped for the hybrid protocol), and if the environment has no particular hazard, then the system can require the activation of the hybrid protocol in order to reinforce communication. This would allow for a safer data communication.

In other cases, the system might evaluate that the VLC protocol cannot function properly in some environment because of constraints. In Figure 4.17 is a situation wherein the VLC protocol is not activated (and incidentally the hybrid one) because of the weather. Bad

```

fogSensor(?fogS) ^ hasFogValue(?fogS, ?fogV) ^
PerceptionAccuracy(?P)^ Good(?P)^
swrlb:greaterThan(?fogV, 50) ^ Weather(?W) -> Fog(?W)

Fog(?F) ^ Environment(?Env) -> BadWeather(?Env)

UAV(?U) ^ RF(?r) ^ Hybride(?h) ^ VLC(?v) ^
hasCommunicationProtocol(?U, ?v) ^
hasCommunicationProtocol(?U, ?h) ^
hasCommunicationProtocol(?U, ?r) ^
BadWeather(?b)
-> isActiveCommunicationProtocol(?r, true) ^
isActiveCommunicationProtocol(?v, false) ^
isActiveCommunicationProtocol(?h, false)

```

Figure 4.17 – Example of a situation where the VLC protocol is not activated. Due to the light diffraction caused by fog, it is harder for the message to be transmitted.

weather can have a strong impact on light diffraction and makes it harder for VLC to function properly. In this case only the RF protocol is activated since it is not affected by weather conditions.

Depending on the computing power of the two agents, the UAV could either parse the data beforehand, or directly sends the video stream to the vehicle. In this case, it would then be sending a consequent amount of data. According to the APTA (American Public Transportation Association) guidelines [117], a CCTV camera covering a passenger area must be able to record at a frequency of 5 to 15 fps at a minimum of 800*600 resolution. With a H.264 coding [118], this would correspond to around 1.1 Mb/s of data generated by one camera.

Considering, the sensitive nature of the environmental information, we must ensure that the data are safely transmitted and minimize the loss and corruption rate during the transfer.

As stated in Section 4.2, data protection can take place in the physical layer, for example thanks to the VLC protocol. It also offers a throughput strong enough to carry heavy

data and can hardly be intercepted. VLC would be suited for the data transfer of some heavy data.

Also, in order to guarantee additional security and integrity during the transmission, the hybrid approach would permit the reinforcement of the process thanks to the redundancy of the data hash. The receiver would then be sure that no corruption happened during the transmission.

4.6 Conclusion

The previous chapter introduced the knowledge-base model making the complete assistance system. Multiple aspects are managed by it, including the communication protocol (cf. Section 3.3.3.1.3), but it does so from a higher abstraction layer, with little involvement in the technical methods. The communication protocols were detailed in this chapter.

This chapter introduced the VLC technology and its potential for vehicular application. Optical communications offer advantages that cannot be found with other electromagnetic waves technologies, and an interest for their indoor and outdoor applications is starting to grow. This chapter also presented a hybrid RF/VLC which would allow for a reinforced data privacy thanks to information redundancy. Data are sent through a physical channel, and their digest are transferred through another one. If the generated hash does not match the received one, the receiver then asks for a resend of the corrupted data. In an environment where communications can be disturbed by external elements, this method would allow for a sturdier exchange of data.

Chapter 5

Use case study

Contents

5.1	Introduction	112
5.2	Simulator	113
5.2.1	Presentation of the simulator	113
5.2.2	Experimentation description	116
5.3	Scenarios	122
5.3.1	Intersection with a moving vehicle	122
5.3.2	Foggy area	129
5.3.3	Dark environment	134
5.3.4	Rainy weather with inadequate sensors	139
5.3.5	Complex scenario	145
5.3.6	Discussion	153
5.4	Conclusion	154

5.1 Introduction

Previous chapters introduced the different concepts of this work. In order to validate the model as a whole, there should be a way to make different tests in various use cases and evaluate the performances of the decision-making system.

This chapter presents the simulator developed in order to make those tests. It will introduce the Unity-based simulator as well as the different tools allowing for the gathering and processing of driving data. Multiple users took part in different scenarios with the assistance of the driving knowledge-base, and their outputs were logged and compared with tests subjects who went through the same situation without assistance.

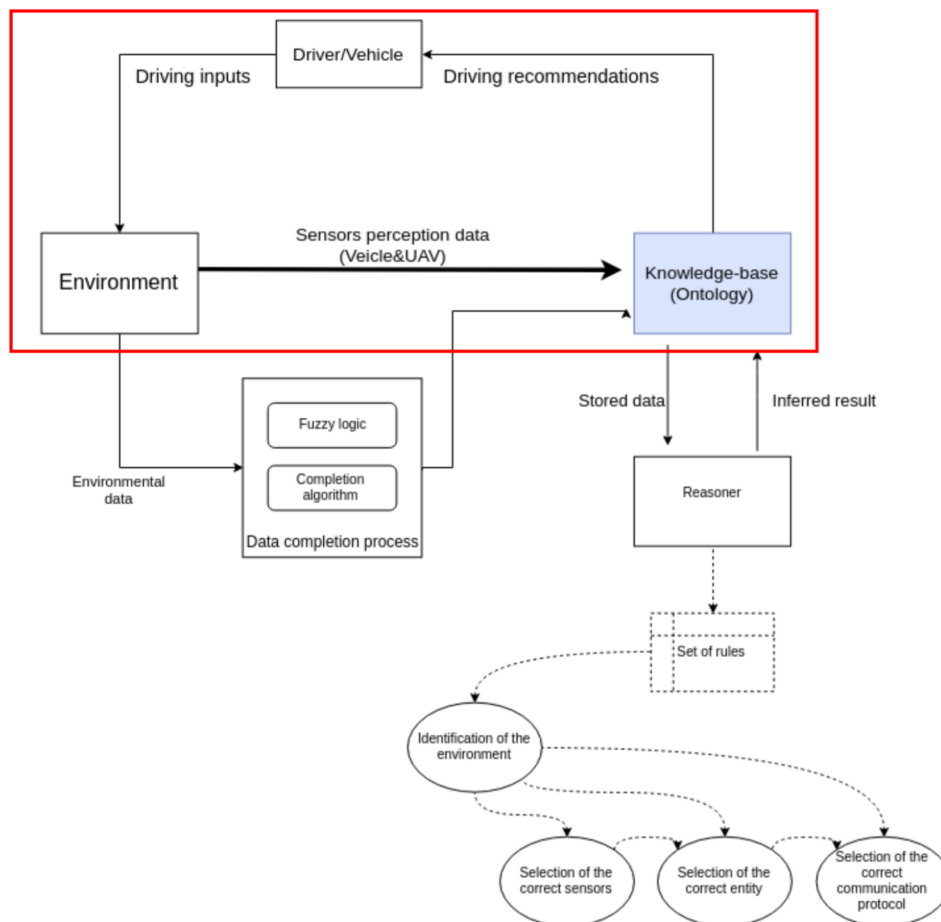


Figure 5.1 – Simulation steps in the model. The experiments actually cover all the steps, but only the main components are represented in this illustration.

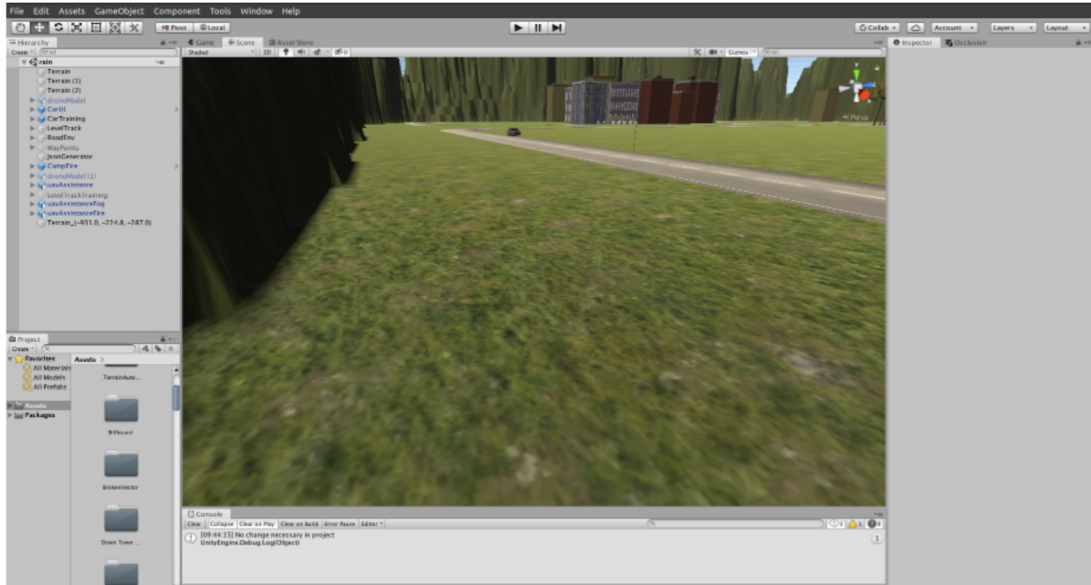


Figure 5.2 – View of the development aspect of the simulator.

5.2 Simulator

5.2.1 Presentation of the simulator

The validation of the model was done in a virtual simulator. The interface was based on the Udacity[119] project, a car simulator built with the Unity engine[120]. It allows for the building of driving surroundings (Roads, obstacles), driving conditions (Rain, Fog, Physics constraints, etc.), and the manual control of the vehicle. The development environment can be seen in Figure 5.2

On a technical level, the driving data are logged in a JSON format and sent via an engine to the knowledge-base. The reasoner will then be called and infer the environmental status.

The communication between the simulator and the knowledge-base is done through a socket connection. We considered 2 different software tools to do so : the Java OWL API [66] and the Owlready Python library. We compared both of those approaches in order to choose the optimal one for our study.

In Figures 5.3 and 5.4, we compared both engines execution times. We executed both of them on a similar environment where the vehicle encounters some heavy-processing events.

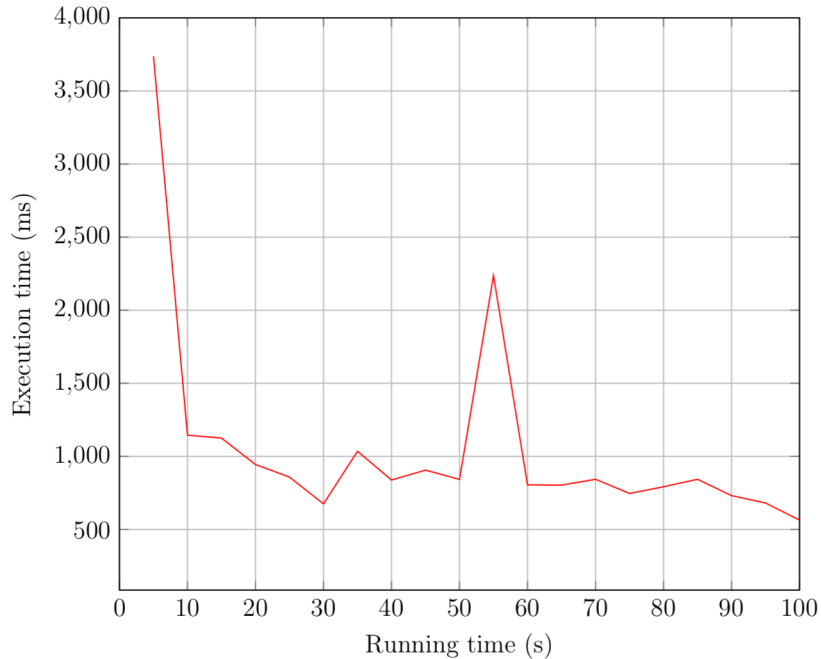


Figure 5.3 – Computed inferring time for the Java tool

As we can see, and depending on the situation, the Python tool offers steady performances with a processing time of around 1 second. The Java occasionally outperforms it, but it has more trouble in harder contexts : we can observe a high peak when the car encounters a fire hazard. The execution speed being an important factor, the choice was made to go with the Owlready tool for this study. An illustration of the process can be found in Figure 5.5, in which the XML object represents the ontology.

Each sensor has its own way of gathering and processing data. For example, cameras relying on deep learning algorithms in order to classify objects [121], LIDAR technology supporting this process with depth-computation [122] or radar sensors for the detection of close elements [123]. The multiple types of methods and algorithms are not considered in this work. The methodology proposed in this paper focuses on a higher lever of processing aimed at the decision-making operation. This is made possible thanks to the simulated environment that allows to virtually generate the necessary environmental data while retaining the constraints of the studied sensors, which have been defined according to the state of the art in Section 2.

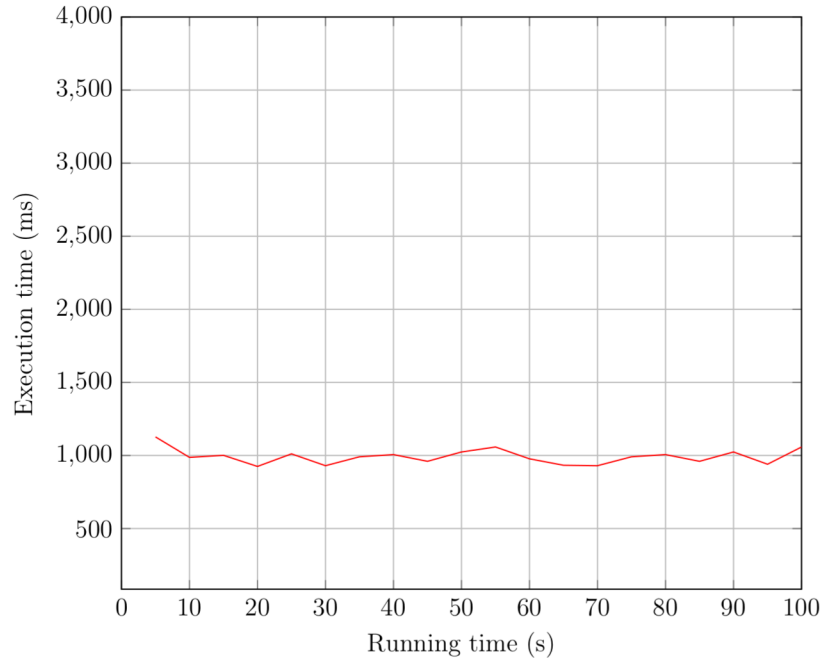


Figure 5.4 – Computed inferring time for the Python tool

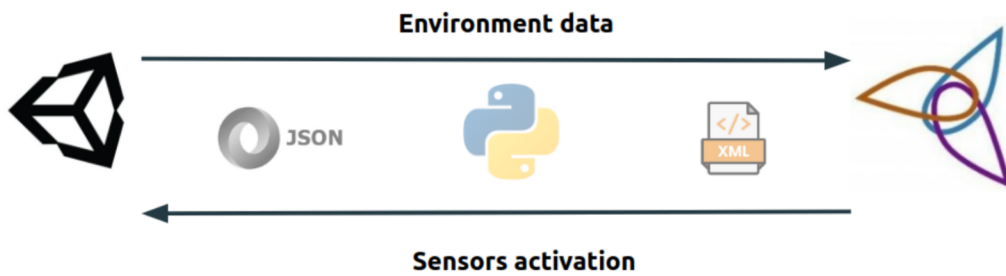


Figure 5.5 – Illustration of the technical implementation.

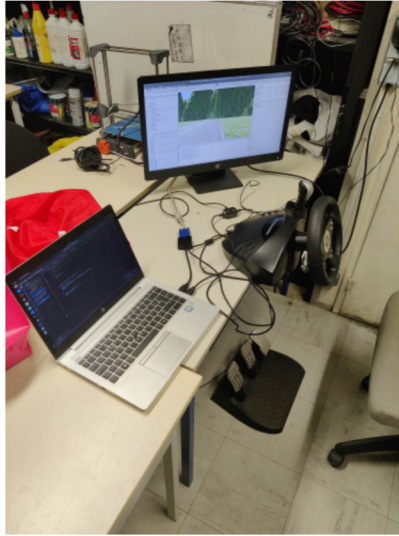


Figure 5.6 – Experimentation setup

5.2.2 Experimentation description

As specified in the previous section, the Udacity tool allows for the quick development of multiple scenarios. The main goal of this study is to show that the use of a UAV allows for the perception enhancement of a vehicle.

In the experiments, this concept was represented as an ADAS system. The experimentation using the case description was made by having different human agents drive in predetermined situations. The driver control the vehicle with an external racing wheel and brakes(cf. Figure 5.6) and the simulator acts as as proxy for the vehicle’s sensors, logging the environmental driving data into the knowledge base. These data are of various natures and correspond to what a real vehicle would collect, such as the localisation, speed of the vehicle or the presence of an obstacle in front of the vehicle. The simulator allows the virtual generation of these data.

For instance, when the vehicle reaches an area where there is fog, the fog sensor of the vehicle will receive a numerical value of 70, which is higher than the threshold value fixed at 50 for fog-detection and the *Weather* individual will be classified as *Fog*. An illustration of this process is shown in Figure 5.7. The raw data from the simulator are sent to the ontology through the Python pipeline. Details of the transmitted data can be found in Table 5.1.

Variables	Ontology Class Values and/or Linking Property	Associated Simulator Values	Comment
Vehicle Speed	hasSpeed [NoSpeed,ExtraLowSpeed,LowSpeed,NormalSpeed,HighSpeed,Overspeed]	(float) VehicleSpeed	
Position of the vehicle	isOnRoad [Roads]	(string) Name of the Road where the vehicle is	
Distance to Obstacle	hasDistanceFromVehicle [FarDistance,MediumDistance,NearDistance]	(float) DistanceToVehicle	
Weather status	[Fog,Sun]	(int) FogSensorValue	Default value "Sun"
Brightness status	[Dark,Normal,Overbright]	(int) brightnessValue	Inferred from other elements
Environmental Status	[Normal,Dark,BadWeather,Hazardous]	-	Not declared if there is no Hazard
Hazard	[FireHazard]	(int,int) X & Y Position of the hazard	
Sensors available	hasSensor [cameraMono,cameraStereo,cameraInfra,fogSensor,brightSensor,lidar,radar,sonar]	(string) Names of the sensors on the vehicle	For both the car and the UAV
Communication protocols	hasCommunicationProtocol[RF,VLC,Hybrid]	(string) Name of the communication protocol	
UAV data	isActiveUAV [true,false]	-	Inferred from other elements

Table 5.1 – Main classes of the ontology and their associated variables from the simulator. The virtual data generated by the experiments are sent to the ontology which classify and process them according to the declared properties and logical rules.

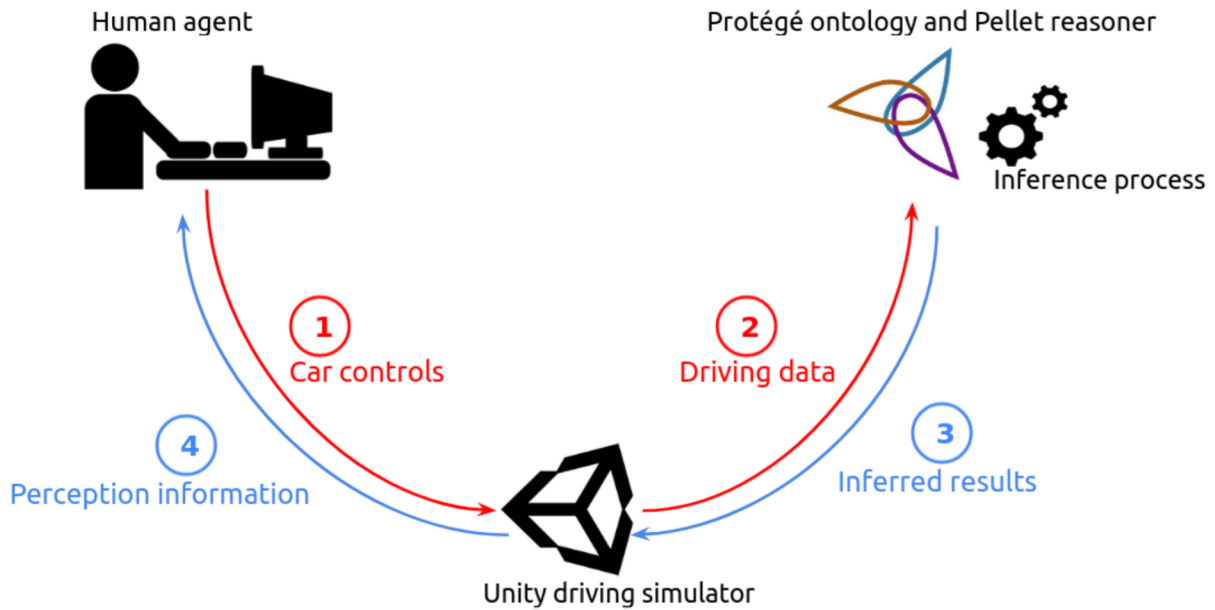


Figure 5.7 – Interactions between different agents. The human agent controls the car, the generated driving data are stored in the knowledge base, and the eventual important information are displayed to the user in a widget.

Once logged in the knowledge base, they are processed by the set of logical rules and the tools described in Section 5.2. If the system inference requires a perception enhancement, and if the conditions are verified (i.e., available UAV within reach with all the correct sensors), the additional information will be displayed on screen, as shown in Figure 5.10: a message informs the user that an obstacle is at a certain distance of the vehicle. This distance is computed thanks to the localization of the ego vehicle and the localization of the obstacle detected by the UAV.

The limited-visibility areas are built so that the controlled vehicle would crash into stationary or moving obstacles if the driver is not careful enough. The obstacles are purposely positioned to maximize the chances of a hit in case of bad driving, for example by over-speeding. In each location, a stationary UAV is positioned in order to cover a specific site. The UAV communicates with the vehicle to provide information on the covered area. Upon request, it will transmit the gathered data to the vehicle, giving important information such as the distance to an obstacle, as shown in Figure 5.10. In this way, the ego vehicle gets knowledge of obstacles in a specific area in advance.

An illustration of the general architecture can be found in Figure 5.8.

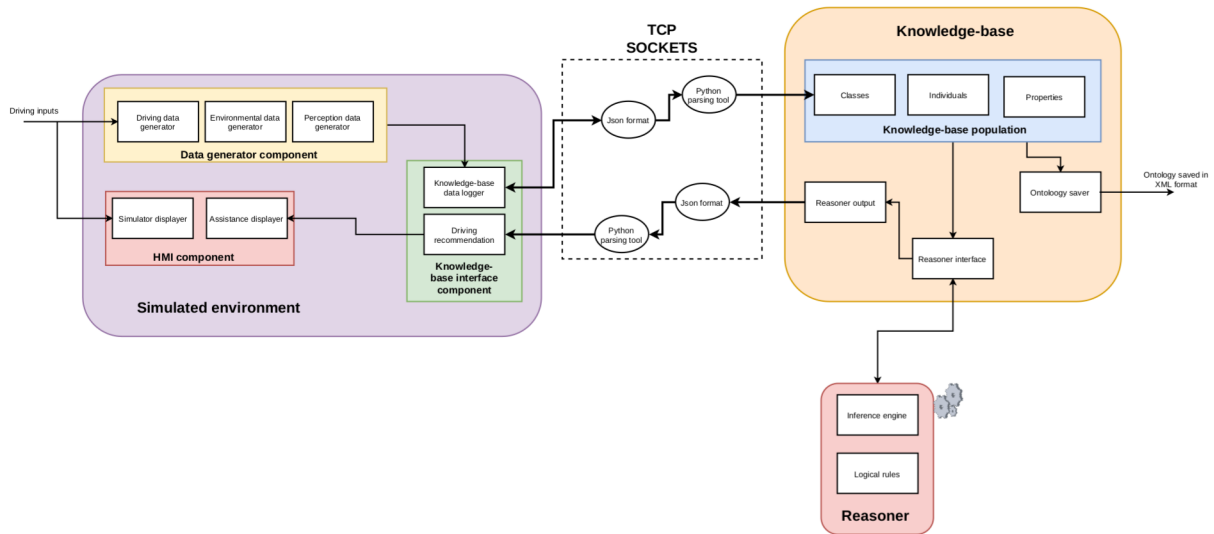


Figure 5.8 – Global architecture of the simulator.

Simulated environment The simulated environment is where the core interaction takes place. It is through it that the test subject can browse the environment, be advised on their next surroundings, and complete the given course. The different elements of the context are initiated here (vehicles, obstacles, hazards, etc.), and all the necessary data are generated periodically during the experiment. It is in constant communication with the knowledge-base via a TCP socket interface, sending data on the surroundings and receiving the inferred results from the reasoner. The simulator runs in the Unity software.

The simulated environment is made of three major components:

- The data generator component manages the virtual data that will be generated and gathered by the system. There are three different types of data:
 - The driving data, related to the ego vehicle controlled by the test subject and influenced by the driving inputs (cf. Figure 5.6). Those are all the information on which the test subject has a direct influence, such as the speed or the steering angle.

- The Environmental data are the data gathered by the Environmental sensors. The simulator allows the manipulation of different variables, for example brightness or rain level, and those can be used for the correct identification of the environment.
- The Perception data are the data gathered by the other perception sensors, for example distance or position sensors. They are generated by the sensors embedded on the vehicles and UAV, and used for the identification of other entities on the road.
- The knowledge-base interface component manages the interfacing of the simulated environment and the knowledge-base. The generated data are encapsulated in a JSON format before being sent to the knowledge-base via a TCP socket, and the same channel is used for the receiving the inference results and driving recommendation.
- The HMI (Human-Machine Interface) component allows the interaction between the user and the simulator. The actions the driver take are shown in the simulator, like the car moving when the driver uses the accelerator, and the assistance widget displays the results of the inference engine, for example warning of the presence of an obstacle ahead of them.

Knowledge-base The knowledge-base is where all the data are stored and processed. It comes in the form of an ontology which needs to be setup beforehand: all the necessary classes are initialized, as well as the logical rules for the reasoner. The different individuals are declared based on the information incoming from the simulator's data, and categorized in the according Class. The ontology is stored in XML format and built thanks to the Protégé software.

The data received from the simulator are used for populating the knowledge-base: the individuals are stored in the right class, and their inner and shared properties are also

declared. Those elements are sent to the reasoner in order to infer some new conclusions from the existing entities. The ontology is also periodically saved in the XML format.

Reasoner The reasoner serves as the intelligence layer of the model. By cross-referencing the knowledge-base individuals with the logical rules, the reasoner can infer new information, such as the environment the vehicle is evolving in, or the action that it should take. Those deductions are also stored in the knowledge-base in specific classes such as the Action or Environment ones. The reasoner used is Pellet, and the logical rules editor is a plug-in to the Protégé software (cf. Figure 5.9).

The reasoner part is made of the inference engine (Pellet), which crosses the knowledge-base population with the pre-declared set of logical rules in order to infer new actions and elements that will be sent back to the ontology and stored in a specific class.

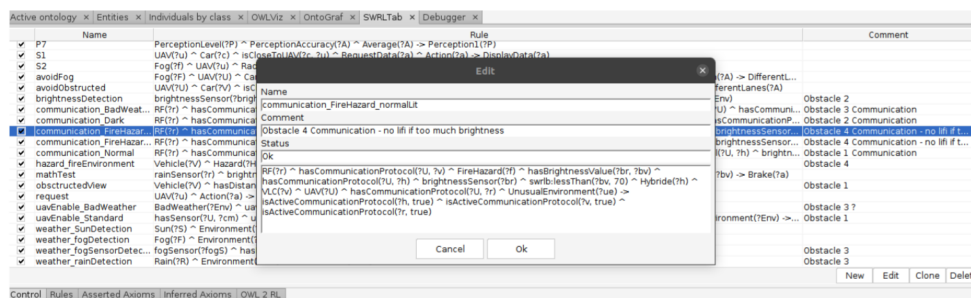


Figure 5.9 – SWRL logical rules editor as it appears in the Protégé software.

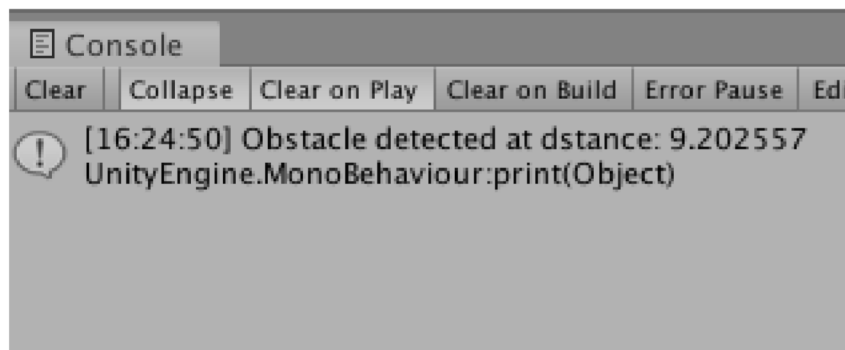


Figure 5.10 – Informing the driver about the distance to an obstacle

5.3 Scenarios

Different test groups were led through various situations and their reactions were logged and evaluated. The scenarios ranged from some single basic situations to complex numerous events.

The following scenarios were arbitrarily chosen because they illustrate different aspects and allow the validation of the model.

- The first scenario is a simple case explaining the basic case and tools of the model.
- The second and third scenarios respectively cover how the system reacts to an inclement weather and a poorly lit situations, and what kind of rules apply in those situations.
- In the fourth scenario, we introduce a rainy situation where the model is not able to perform correctly due to the lack of some parameters. Then we compare the performances with the same scenario and a functioning model
- The last scenario is a complex one with multiple events the test subject needs to cross. There is also an illustration on how the data cleaning and completion process is used for more reliability in the gathered data.

Those different scenario allow us to cover different situations and validate our work in a variety of situations.

5.3.1 Intersection with a moving vehicle

In the scenario of an intersection, a vehicle reaches an intersection where the visibility is limited by a natural element (a rock). Another vehicle is incoming at the same intersection and cannot be detected because of the obstacle. A UAV is stationed at the intersection and transmits additional position data if requested. This experiment does not focus on the scheduling priority of an intersection. An illustration can be found in Figure 5.11.

The different elements of this scenario are represented in Table 5.2, along with their matching expressions in the knowledge-base. Those are the main elements that will be stored in the ontology during the course of the test, but not all of them have to be used by the reasoner: for example a vehicle can be carrying multiple sensors, but only one of them could be relevant or sufficient, depending on the context.

- **myVehicle** is an individual of the **Vehicle** class serving as the main vehicle. **intersectionVehicle** is the vehicle that can be encountered on the road and is hidden by the obstacle
- **UAV_Crossroad** is the UAV stationed in the intersection and covering the blind spot made by the obstacle.
- **LiFi**, **DSRC**, and **HybridProtocol** are the different communication protocols equipping the drone and the main vehicle. The individual names belong to different classes in the ontology: They all belong to the **CommunicationProtocol** class, but LiFi (Light-Fidelity) is a VLC technology, while DSRC (Dedicated Short-Range Communications), is an RF technology.
- **Rock** belongs to the **Obstacle** class, and is identified as so due to its proximity to the main vehicle and the impact it has on the visibility.
- **CameraS1**, **LidarL1**, **LidarL2**, **RadarR1** are the names different sensors embedded on the elements via the **hasSensor** property. The main Vehicle carries the LidarL1 and RadarR1 sensors, while the UAV carries the CameraS1 and LidarL2 sensors. Those elements belong to their respective classes in the ontology, but they are all members of the **Sensor** parent class.
- **NormalEnvironment** and **ObstructedView** are elements used for the identification of the context. The brightness and weather are fine, meaning the natural surroundings are normal and the **Environment** class will stick to the default value,

but there is an obstacle close, so the reasoner will infer the Action class as ObstructedView. (The Action class serves for the storage of the deductions of the reasoner, not only the physical driving recommendations)

Simulator element	Associated logical individual
myVehicle intersectionVehicle	Vehicle(?v)
UAV_crossroad	UAV(?u)
LiFI	VLC(?v)
DSRC	RF(?r)
HybridProtocol	CommunicationProtocol(?c)
Rock	Obstacle(?o)
CameraS1, CameraS2 LidarL1, LidarL2 RadarR1, RadarR2, RadarR3	CameraStereo(?cs), Lidar(?l) Radar(?r) Sensor(?s)
NormalEnvironment ObstructedView	Environment(?e) Action(?a)

Table 5.2 – Elements of the first scenario as they are identified in the simulator and their matching variable in the logical rule

Four drivers took part in the experiment. Two were guided by the reasoning system, and two were not. The users were given some preparation time before in order to get used to the driving commands. Results are summarized in Table 5.3.

The drivers had different approaches for this situation. Some tried to rush their way through the intersection, while others proceeded with extreme caution. The non-assisted test subject who decided to speed their way through the experiment crashed in the upcoming vehicle and was ejected out of the road, resulting in the slowest completion time of more than 40s. The one who decided to go slowly had to proceed at an extremely low speed in order to make sure that they would encounter no incoming vehicle, an action which took 27s.

Thanks to the UAV stationed at the right position, the assisted test groups were informed of the presence of an incoming vehicle ahead of reaching the intersection. They were also notified of the distance between them and the moving vehicle, and depending



Figure 5.11 – Environment test with an obstructed view

	Average speed	Time to complete the experiment	Incident
Non-assisted driver 1	19km/h	45s	Yes
Non-assisted driver 2	11km/h	27s	No
Assisted driver 1	17km/h	15s	No
Assisted driver 2	16km/h	22s	No

Table 5.3 – Results of the obstructed view driving experiment.



Figure 5.12 – Environment test with an obstructed view

on it could either try to go through the intersection while the other vehicle is at a safe enough distance, or wait for it to pass. The test subject who went with the former option took 15s to complete the circuit, which is the fastest time, while the one who waited showed a result of 22s.

In this scenario, the events are happening in a normal environment, which is a Sunny situation where no real perturbation is applied on the sensors. Considering the nature of the situation, a basic sensors, such as a stereoscopic camera, should be enough to improve perception. The set of rules in Figure 5.13 allows for the handling of the situation : An object in front of the vehicle and classified at NearDistance (20m) raises the Obstructed-View property, meaning that the front visibility is reduced. Since there is a UAV nearby, the system ensures it possesses the right sensor, and requests additional data from it.

Here is a breakdown of the rules used in this scenario. More explanations about the rules can be found in Appendix A.

- In the first rule, the element **Vehicle(?V)** refers to the vehicle the driver is control-

ling, known as **myVehicle**

- The Rock blocking the view is an Object on the road, making it **Obstacle(?O)**
- The vehicle is close to the obstacle, which is expressed through the **NearDistance** value of the **hasDistanceFromVehicle** property.
- The previous elements allow to conclude that the vehicle V has the property **hasObstructedView** with the obstacle O.

The second rule includes some of the same elements in addition to new ones.

- In the second rule, the element **UAV(?U)** refers to the drone stationed at the intersection, individually named **UAV_Crossroad**.
- The **cameraStereo(?cs)** element refers to a stereoscopic camera, and the **hasSensor(?U,?cs)** property means that the sensor should be embedded on the UAV. As shown in Table 5.2, those conditions are validated since the drone does carry the adequate camera.
- **NormalEnvironment(?Env)** ensures that there is no specific constraint on the environment, meaning the weather and brightness are adequate.
- **isCloseToUAV(?V,?U)** is a condition ensuring the vehicle and the drone are close enough.
- Once all the conditions are validated, the **isActiveUAV(?U,true)** property is inferred as true: the vehicle requests the assistance of the drone.

The third rule follows a similar approach for the management of the communication protocols.

- **RF(?r)** and **VLC(?v)** respectively designates the RF and VLC communication protocols, while the **Hybride(?h)** refers to the hybrid protocol

```

Vehicle(?V) ^ Object(?O) ^ Environment(?Env) ^
hasDistanceFromVehicle(?O, NearDistance)
-> hasObstructedView(?V, ?O)

UAV(?U) ^ cameraStereo(?cs) ^ hasSensor(?U, ?cs) ^
Vehicle(?V) ^ Object(?O) ^ hasObstructedView(?V, ?O) ^
NormalEnvironment(?Env) ^ isCloseToUAV(?V,?U)->
isActiveUAV(?U, true)

UAV(?U) ^ cameraStereo(?cs) ^ hasSensor(?U, ?cs) ^
isActiveUAV(?U,true) -> isActiveSensor(?cs, true)

```

Figure 5.13 – Avoiding obstacle in fog

- The **hasCommunicationProtocol(?U,?v)** means the UAV is equipped with the tools to establish a VLC communication. The same applies to the RF and Hybrid objects.
- As for the previous rule, the **NormalEnvironment(?E)** indicates there is no real noticeable event in the environment, meaning there is no constraint on the activation of any protocol.
- Since the UAV assistance is already required, as shown by the property **isActiveUAV(?U,true)**, the communication protocols can be activated thanks to the property **isActiveCommunicationProtocol**

Since there is no specific element that could hinder the communication protocols, it is safe to go with the hybrid protocol in order to improve the redundancy of information. In order to do so, both RF and VLC need to be activated too, as shown in the SWRL rules in Figure 5.14.

The variable in the knowledge-base are directly referencing objects and properties in the simulator. For example, *Vehicle(?V)* represents the vehicle the test subject is driving, while *Object(?O)* is the rock blocking the view.

```

UAV(?U) ^ RF(?r) ^ VLC(?v) ^ Hybride(?h)
^hasCommunicationProtocol(?U, ?v)
^hasCommunicationProtocol(?U, ?h)
^hasCommunicationProtocol(?U, ?r)
^NormalEnvironment(?E)
^isActiveUAV(?U,true) ->
isActiveCommunicationProtocol(?r, true) ^
isActiveCommunicationProtocol(?v, true) ^
isActiveCommunicationProtocol(?h, true)

```

Figure 5.14 – Communication in a simple obstructed view scenario

5.3.2 Foggy area

In this situation, a vehicle is driving on a single lane when it enters a foggy area where a UAV is stationed. An immobile vehicle is located on the same road and serves as an obstacle, as well as some pedestrians crossing the road. The speed, weather and distance data are gathered from the vehicle sensors and are transmitted to the knowledge-base. If necessary, and if requested, there is a UAV nearby which can provide information of the present elements.

The different elements of this scenario are represented in Table 5.4, along with their matching expressions in the knowledge-base. Those are the main elements that will be stored in the ontology during the course of the test.

- **myVehicle** is an individual of the `Vehicle` class serving as the main vehicle.
- **UAV_Fog** is the UAV stationed in the foggy area. It would still encounter the same conditions and sensor hindrance than the vehicles, but the real interest here is in the **perception range enhancement**: by being stationed further on the road, the UAV has access on information concerning what the main vehicle will encounter. Even with a limited radius, the advanced positioning still offers potential.
- **LiFi, DSRC, and HybridProtocol** are the different communication protocols equipping the drone and the main vehicle. Their activation depends on the current



Figure 5.15 – Environment test with fog

environment : for example, the light diffraction from the fog can be disabling for the VLC technology.

- **FogF** belongs to the Fog class, itself a sub-class of Weather. It generates the FogV numerical value, which is gathered by a FogSensor object and compared to a threshold value to derive the weather state.
- **FogSensorF1**, **CameraI1**, **LidarL1**, **LidarR1**, **RadarR2** are the names different sensors embedded on the elements via the **hasSensor** property. FogSensorF1 is an EnvironmentalSensor used by the main vehicle for the identification of the environment. The main Vehicle also carries the LidarR1 and RadarR1 sensors, while the UAV carries the CameraI1 and LidarR2 sensors. Those elements belong to their respective classes in the ontology, but they are all members of the Sensor parent class.
- **BadWeather** is from the Environment class and is inferred by the elements of the Weather class.

Simulator element	Associated logical individual
myVehicle movingVehicle	Vehicle(?v)
UAV_Fog	UAV(?u)
LiFI DSRC HybridProtocol	VLC(?v) RF(?r) CommunicationProtocol(?c)
FogF	Fog(?f) Weather(?w)
FogSensorF1, CameraI1, LidarL1, RadarR1, RadarR2	FogSensor(?fs) CameraInfra(?ci) Lidar(?l) Radar(?r) Sensor(?s)
BadWeather	Environment(?e)

Table 5.4 – Elements of the second scenario as they are identified in the simulator and their matching variable in the logical rule

	Average speed	Time to complete the experiment	Max speed	Incident
Non-assisted driver 1	41km/h	17s	69km/h	No
Non-assisted driver 2	28km/h	48s	70km/h	Yes
Assisted driver 1	46km/h	13s	84km/h	No
Assisted driver 2	37km/h	15s	57km/h	No

Table 5.5 – Results of the foggy driving experiment.

Four drivers took part in the experiment. Two were guided by the reasoning system, and two were not. The users were given some preparation time before in order to get used to the driving commands. Results are summarized in Table 5.5.

In general, the assisted drivers performed slightly better than the non-assisted group. In fact, one of the non-assisted test subject crashed into the obstacle and could not recover from the incident, despite going at a relatively low speed (28km/h). The assisted users managed to complete the experiment faster: when informed about the presence of an obstacle, they have enough time to plan for the overtaking process. Also, when asked about their overall feedbacks, they stated that the presence of the assistance system gave them more confidence into going at a faster speed.

With the assisted group, two different logical rules are applied. The rules in Figure 5.17 are the ones managing the detection of the environment and the decision-making, while

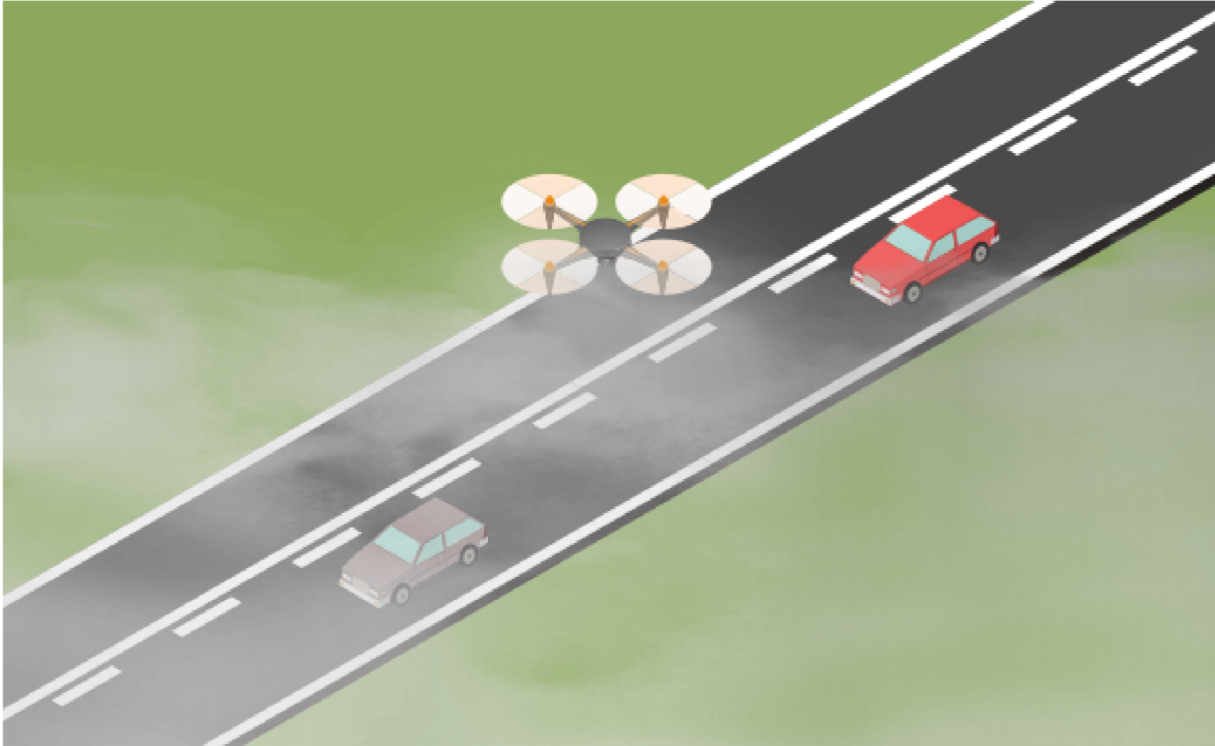


Figure 5.16 – Illustration of the foggy simulated environment

the rule in Figure 5.18 manages the communication protocol. The former uses the virtual sensors of the simulated car in order to evaluate the environment (inferred as "Foggy" weather and "Bad Weather" environment). It will then ensure that the nearby UAV possesses the adequate sensors and requests its assistance. If it exists, the assisted rule will advise the driver on the best action to make.

The second set of rules concerns the communication protocol. The strengths and weaknesses of the different communication protocols are covered in Chapter 4, and in this particular scenario, the bad weather would have an impact on the VLC communication, which makes it best to rely on RF communications instead.

This basic experiment validates the general usefulness of the model. The assisted users showed both a safer behaviour and a faster average speed than the other test subjects. In addition to the technical perception enhancement, the knowledge of being assisted seems to also bring a psychological support to the driver.

```

Vehicle(?v ) ^ FogSensor(?fogS) ^ hasSensor(?C,?fogS)
^ hasFogValue(?fogS, ?fogV) ^ swrlb:greaterThan(?fogV, 50)
^ Weather(?W)
->Fog(?W)

```

```

Fog(?F) ^ Environment(?Env)
-> BadWeather(?Env)

```

```

Fog(?F) ^ UAV(?u) ^ Radar(?ra) ^ cameraInfra(?c)^
hasSensor(?u,?c) ^ hasSensor(?u,?ra)
-> isActiveUAV(?u,true)

```

```

UAV(?u) ^ Radar(?ra) ^ cameraInfra(?c)^
hasSensor(?u,?c) ^ hasSensor(?u,?ra) ^ isActiveUAV(?u,true)
-> isActiveSensor(?ra, true) ^ isActiveSensor(?c, true)

```

Figure 5.17 – Avoiding obstacle in a foggy situation

```

UAV(?U) ^ RF(?r) ^ VLC(?v) ^ Hybride(?h)
^hasCommunicationProtocol(?U, ?v)
^hasCommunicationProtocol(?U, ?h)
^hasCommunicationProtocol(?U, ?r) ^ BadWeather(?b)->
isActiveCommunicationProtocol(?r, true) ^
isActiveCommunicationProtocol(?v, false) ^
isActiveCommunicationProtocol(?h, false)

```

Figure 5.18 – Communication in a foggy situation

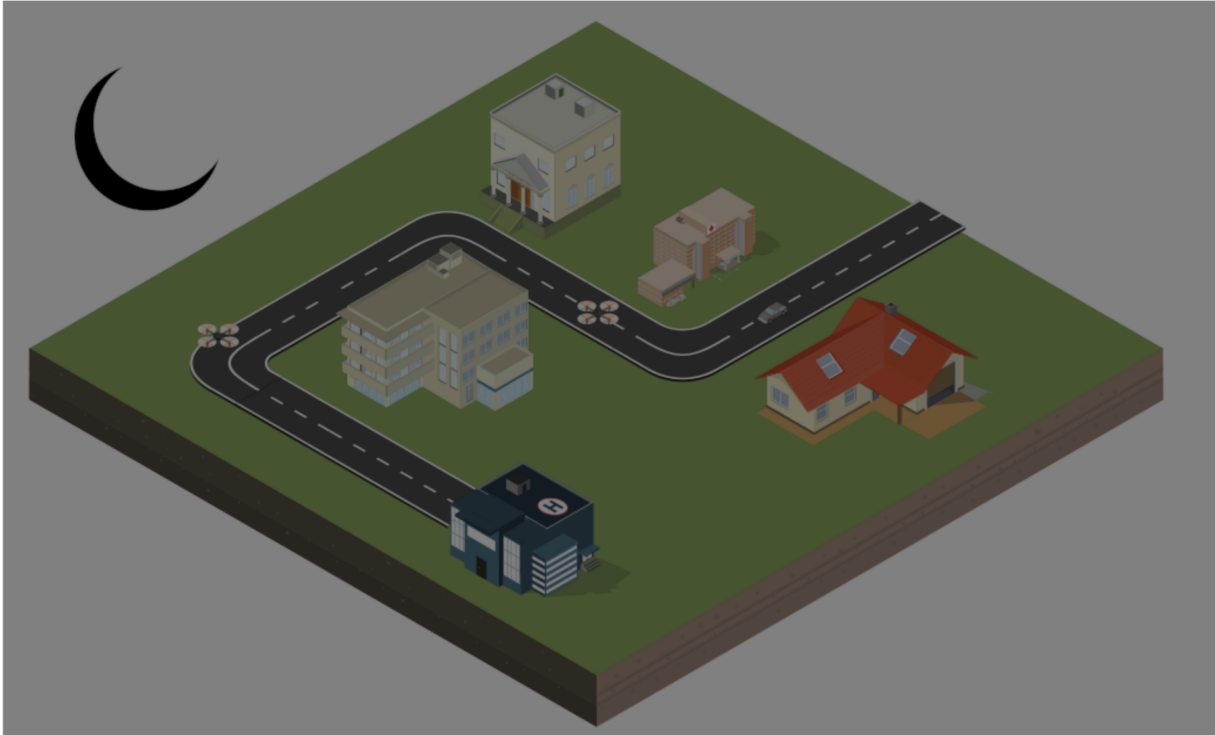


Figure 5.19 – Environment test in a dark area

5.3.3 Dark environment

In the dark environment scenario, the users had to drive through a circuit with poor brightness and extremely reduced visibility. There will be different UAV stationed at some positions which will be providing light, and doubling as VLC-communication equipment. An illustration can be found in Figure 5.19.

The different elements of this scenario are represented in Table 5.6, along with their matching expressions in the knowledge-base. Those are the main elements that will be stored in the ontology during the course of the test.

- **myVehicle** is an individual of the Vehicle class serving as the main vehicle.
- **UAV_Dark** is the UAV stationed in the foggy area. It would still encounter the same conditions and sensor hindrance than the vehicles, but in the similar way to the previous scenario, the interest here is in extending the perception range.

Simulator element	Associated logical individual
myVehicle	Vehicle(?v)
UAV_Dark	UAV(?u)
LiFi DSRC HybridProtocol	VLC(?v) RF(?r) CommunicationProtocol(?c)
BrightnessSensorBS1, CameraI1, LidarL1, RadarR1, RadarR2	BrightnessSensor(?bs) CameraInfra(?ci) Lidar(?l) Radar(?r) Sensor(?s)
Dark	Environment(?e)

Table 5.6 – Elements of the third scenario as they are identified in the simulator and their matching variable in the logical rule

- **LiFi, DSRC, and HybridProtocol** are the different communication protocols equipping the drone and the main vehicle.
- **BrightnessSensorBS1, CameraI1, LidarL1, LidarR1, RadarR2** are the names different sensors embedded on the elements via the **hasSensor** property. BrightnessSensorBS1 is an EnvironmentalSensor used by the main vehicle for the identification of the environment. The main Vehicle also carries the LidarR1 and RadarR1 sensors, while the UAV carries the CameraI1 and LidarR2 sensors. Those elements belong to their respective classes in the ontology, but they are all members of the Sensor parent class.
- **Dark** is from the Environment class and is inferred by the value of the brightness.

The vehicle is maneuvering in a densely populated urban area, with potentially other obstacles. An obstacle is stationed in the pathway, after an intersection with limited perception, but can be avoided with the additional assistance of a UAV. Three test subjects partook in this experiment in three different situations :

- **Absence of UAV** : There was no light source in the circuit and the driver could only see a few meters ahead of them



Figure 5.20 – Environment test in a dark area

- UAV without the necessary sensors : Some sensors, mostly cameras, work poorly on a dark environment. They might not be able to transmit the correct data
- UAV with the necessary sensors : Even in a dark environment, the UAV provide both brightness and information on the vehicle's surroundings

Details of the results can be found on Table 5.7. As expected, the subject with no additional lighting took the longest time to complete the course, with the slowest average speed. Due to the speed constraint imposed by the nature of the experiment, it was still able to avoid the obstacle after having spotted it. The user who had additional illumination thanks to the UAVs went considerably faster, still going slowly in areas that were not correctly brightened. Finally, the user who was assisted by the perception system showed a globally similar speed but was able to foresee the presence of the obstacle, preparing their maneuver in advance.

For the assistance experiment, the logical rules in Figure 5.22 and Figure 5.23 were activated. The former set manages the identification of the environment (classified as Dark) thanks to the returned brightness sensor value, and asks for the assistance of a nearby

	Average speed	Time to complete the experiment	Incident
No UAV	12km/h	94s	No
UAV illumination and no assistance	16km/h	65s	No
UAV assistance	17km/h	57s	No

Table 5.7 – Results of the darkness environment driving experiment.



Figure 5.21 – Presence of the obstacle in a dark environment


```
brightnessSensor(?brightS) ^ hasBrightnessValue(?brightS, ?brightV) ^
swrlb:lessThan(?brightV, 50) ^ Environment(?Env)
-> Dark(?Env)
```

```
UAV(?U) ^ cameraInfra(?ci) ^ Radar(?r) ^ Lidar(?l) hasSensor(?U, ?ci) ^
hasSensor(?U, ?r) ^ hasSensor(?U, ?l) ^ hasSensor(?U, ?r) ^
Vehicle(?V) ^ isCloseToUAV(?V, ?U) ^
Dark(?Env) ->
isActiveUAV(?U, true) ^ isActiveSensor(?ci, true) ^
isActiveSensor(?r, true) ^ isActiveSensor(?l, true)
```

```
UAV(?U) ^ cameraInfra(?ci) ^ Radar(?r) ^ Lidar(?l) hasSensor(?U, ?ci) ^
hasSensor(?U, ?r) ^ hasSensor(?U, ?l) ^ hasSensor(?U, ?r) ^
isActiveUAV(?U, true) ->
isActiveSensor(?ci, true) ^
isActiveSensor(?r, true) ^ isActiveSensor(?l, true)
```

Figure 5.22 – Obstructed view

```
RF(?r) ^ hasCommunicationProtocol(?U, ?v) ^ Dark(?d)
^ hasCommunicationProtocol(?U, ?h) ^ Hybride(?h) ^
VLC(?v) ^ UAV(?U) ^ hasCommunicationProtocol(?U, ?r)
-> isActiveCommunicationProtocol(?h, true) ^
isActiveCommunicationProtocol(?v, true) ^
isActiveCommunicationProtocol(?r, true)
```

Figure 5.23 – Communication in dark environment

UAV if it holds the adequate sensors.

The rule in Figure 5.23 is about the communication process. As stated earlier, VLC can also serve to brighten an environment, making it an excellent choice for a situation where perception is limited. In order to ensure the quality of service and security of the transmitted data, the hybrid approach can even be considered.

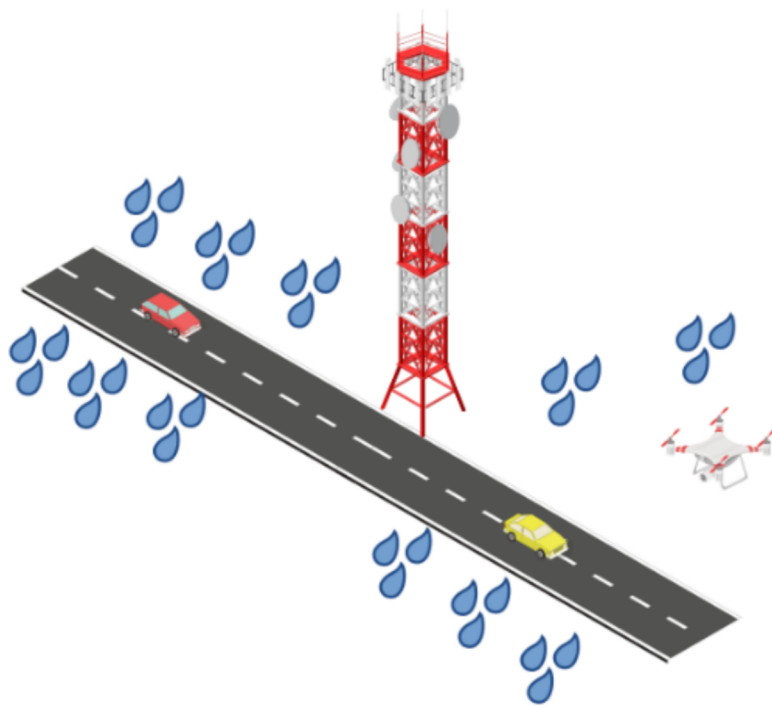


Figure 5.24 – Environment test in a rainy area

5.3.4 Rainy weather with inadequate sensors

5.3.4.1 Without adequate sensors

In the rainy scenario, the test subjects had to go through an area with a rainy situation. There are multiple UAVs stationed, but not all of them have the necessary sensors to navigate in such an environment. The assistance system is then compromised.

An illustration of the process can be found in Figure 5.24.

The different elements of this scenario are represented in Table 5.4, along with their matching expressions in the knowledge-base. Those are the main elements that will be stored in the ontology during the course of the test.

- **myVehicle** is an individual of the **Vehicle** class serving as the main vehicle.
- **UAV_Rain** is the UAV stationed in the rainy area.
- **LiFi**, **DSRC**, and **HybridProtocol** are the different communication protocols

equipping the drone and the main vehicle. Their activation depends on the current environment : for example, the light diffraction from the rain can be disabling for the VLC technology.

- **RainR** belongs to the Rain class, itself a sub-class of Weather. It generates the RAinV numerical value, which is gathered by a RainSensor object and compared to a threshold value to derive the weather state.
- **RainSensorR1, CameraI1, CameraS1, CameraS2, LidarL1, RadarR1** are the names of different sensors embedded on the elements via the **hasSensor** property. RainSensorR1 is an EnvironmentalSensor used by the main vehicle for the identification of the environment. The main Vehicle also carries the CameraS1 and CameraS2 sensors. The first part of this scenario covers the case where the UAV only carries the LidarL1 sensor, which acts poorly in a rainy situation, and the scenario is then completed by adding the CameraI1 and RadarR1 sensors, which perform better in that context.
- **BadWeather** is from the Environment class and is inferred by the elements of the Weather class.

Four users took part in this experiment, divided into 2 groups of assisted and non-assisted drivers. The experiment's results can be found on Table 5.9.

Due to the lack of the appropriate sensors on some of the UAV, the assistance system could not provide perception enhancement in all of the situations. For this reason, assisted and non-assisted users encountered an incidental event. This is also due to the fact that the rain event makes the road slippery and has a impact effect on the controls of the vehicle. The second assisted driver still managed to complete the course with the fastest time and with no incident, but the performances were only slightly better than the non-assisted driver who also managed to do so, albeit going at a slower speed.

Simulator element	Associated logical individual
myVehicle	Vehicle(?v)
UAV_Rain	UAV(?u)
LiFi	VLC(?v)
DSRC	RF(?r)
HybridProtocol	CommunicationProtocol(?c)
RainR	Rain(?r) Weather(?w)
RainSensorR1, CameraI1, CameraS1, CameraS2, LidarL1, RadarR1	RainSensor(?rs) CameraInfra(?ci) CameraSterel(?cs) Lidar(?l) Radar(?r) Sensor(?s)
BadWeather	Environment(?e)

Table 5.8 – Elements of the fourth scenario as they are identified in the simulator and their matching variable in the logical rule

	Average speed	Time to complete the experiment	Max speed	Incident
Non-assisted driver 1	25km/h	72s	53km/h	No
Non-assisted driver 2	30km/h	90s	55km/h	Yes
Assisted driver 1	33km/h	81s	62km/h	Yes
Assisted driver 2	31km/h	59s	59km/h	No

Table 5.9 – Results of the rainy experiment.

```

rainSensor(?rainS) ^ hasRainValue(?rainS, ?rainV) ^
swrlb:greaterThan(?rainV, 50) ^ Weather(?w)
-> Rain(?w)

Rain(?w) ^ Environment(?Env)
-> BadWeather(?Env)

Rain(?R)^ UAV(?u)^ Radar(?ra)^ cameraInfra(?c)^
hasSensor(?u,?c)^ hasSensor(?u,?ra)
-> isActiveUAV(?u,true)

UAV(?u)^ Radar(?ra)^ cameraInfra(?c)^
hasSensor(?u,?c)^ hasSensor(?u,?ra)
^ isActiveUAV(?u,true) ->
isActiveSensor(?ra,true) ^ isActiveSensor(?c,true)

```

Figure 5.25 – Set of rules for the management of a rainy environment.

The logical rules in Figure 5.25 are supposed to ensure that the UAV possesses the minimum adequate sensors (Infrared camera and Radar) in order to function in a bad weather environment. However, as shown in Figure 5.26, the UAV individual does not have those necessary sensors, making it impossible for the rule to be validated. The **isActiveUAV** property remains at the state **false**.

The first two rules of the set are used for the identification of the environment.

- In the first rule, the object **rainSensor(?rainS)** represents the elemental sensor used for the identification of the rain. Through the **hasRainValue** property, it links the gathered value of the sensor to the integer variable **rainV**.
- The gathered value needs to be compared to a threshold, which is done through the **swrlb:greaterThan(?rainV,50)** property: the property is validated as true if the value gathered by the sensor is greater than 50.
- This would allow the **Weather(?w)** object to be inferred as also belonging to the **Rain** class, which would also imply the **Environment** to be of the sub-class **BadWeather**

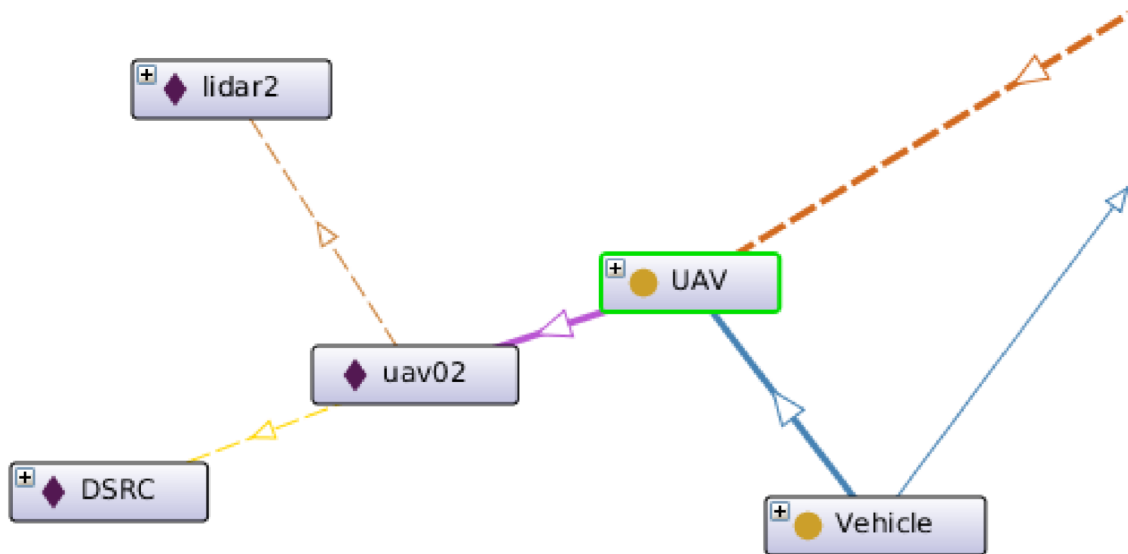


Figure 5.26 – UAV agent as seen in the ontology

The two other rules of Figure 5.25 are related to the sensor and UAV activation.

- In the second rule, the element **UAV(?U)** refers to the drone stationed at the intersection, individually named **UAV_Rain**. The **Rain(?r)** elements refers to a rainy context.
- The **Radar(?ra)** and **cameraInfra(?c)** elements refer to the radar and infrared camera respectively, and the **hasSensor(?u,c)** and **hasSensor(?u,ra)** properties mean that the sensors should be embedded on the UAV. However, as stated earlier and shown in Figure 5.26, in this situation the UAV only carries a Lidar sensor, invalidating this rule and making it impossible to request the UAV assistance through the **isActiveUAV(?u,true)** property.
- The last rule should be about the sensor activations, but since the **isActiveUAV** is not validated, the rule is not executed neither

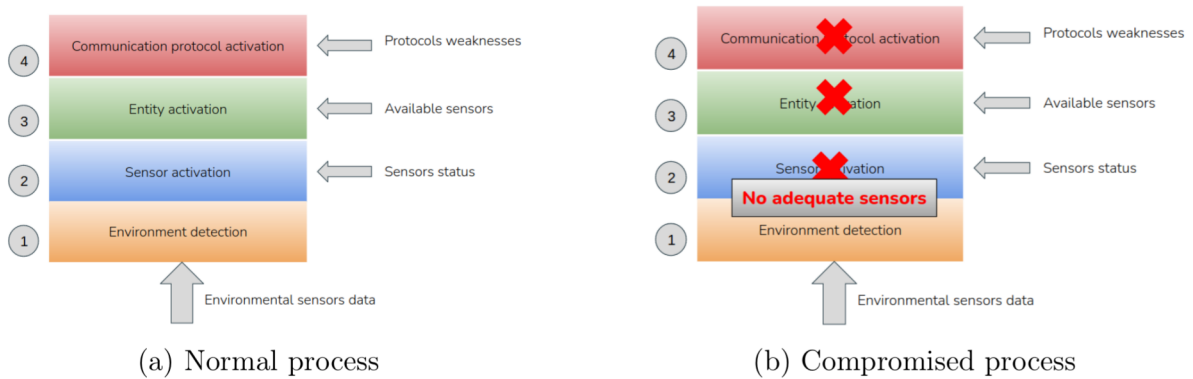
This effectively means the UAV cannot assist the user. In addition, the rain makes the roads slippery and has a direct impact on the vehicle's road handling.

```

UAV(?U) ^ RF(?r) ^ hasCommunicationProtocol(?U, ?v) ^
hasCommunicationProtocol(?U, ?h) ^ Hybride(?h) ^ VLC(?v) ^
BadWeather(?b) ^ hasCommunicationProtocol(?U, ?r)
-> isActiveCommunicationProtocol(?r, true) ^
isActiveCommunicationProtocol(?v, false) ^
isActiveCommunicationProtocol(?h, false)

```

Figure 5.27 – Communication protocol



Rain also has an impact on the communication protocol. The water drops have a refractive effect that make VLC communication extremely difficult, triggering the rule in Figure 5.27. Due to the nature of the environment, both perception and control of the vehicle were tampered. The assistance system performances were notably limited by the rainy environment and the lack of the adequate sensors : Even with a float of UAV deployed, some of them were not really useful since their set of sensors presented weakness to the environment.

In Section 3.3.3 we presented the process through which the inference system validates the assistance request. In this experiment, the lack of adequate sensors cancelled the chain of steps and led to the non-validation of the logical rules, making the assistance system non-operational.

5.3.4.2 With adequate sensors

The same scenario was remade with a UAV carrying the proper sensors. The experiment's results can be seen in Table 5.10.

	Average speed	Time to complete the experiment	Max speed	Incident
Assisted driver 1	38km/h	60s	67km/h	No
Assisted driver 2	33km/h	72s	65km/h	No
Assisted driver 3	29km/h	77s	64km/h	No

Table 5.10 – Results of the rainy experiment.

In this situation, the rules in Figures 5.25 and 5.27 were validated thanks to the addition of the Infrared camera and Radar sensors, making it possible for the perception enhancement process to be implemented. The driving performances of the test subjects are improved compared to the ones with a faulty assistance system.

This scenario covered a situation where one of the entities was lacking the proper sensors and was not able to perform correctly. The assisted and non-assisted sets both perform poorly. The tests that took place in the same environment but with all the correct sensors performed noticeably better.

5.3.5 Complex scenario

This experiment combines multiple different elements in order to make a more complex situation. The test users have to go through a circuit where they encounter an obstructed view event, a foggy area, and a building with a fire hazard. Multiple UAV are stationed in key positions in order to ensure assistance when necessary. An illustration of the process can be found in Figure 5.29.

Section 3.3.1 covered different ways of dealing missing data. In this complex scenario, an additional script was added that would randomly drop some of the numerical values. At a given time T, the returned values of randomly-chosen sensors were dropped, and the data-completion part was called in order to deal with those missing data. The set of data gathered would initially look like in Figure 5.31.

This would mark the object *PerceptionAccuracy* as being **Poor**, resulting to most of the the environment-identifying rules to not function. By using a completion algorithm like KNN on different previous set of data, the system can try to fill the missing data with

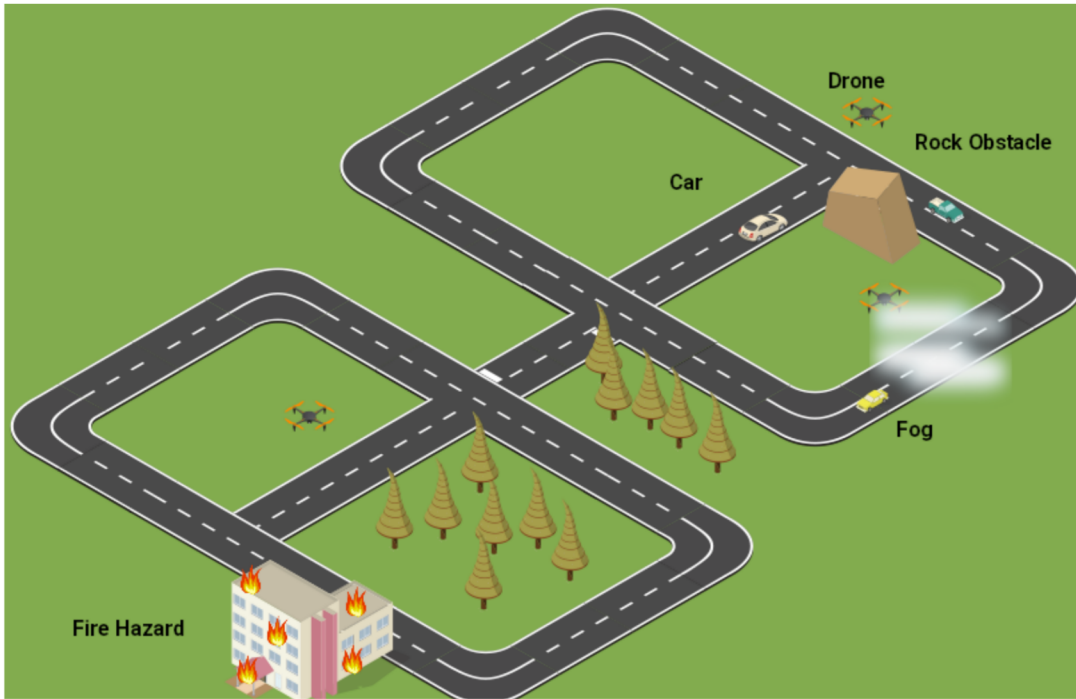


Figure 5.29 – Illustration of the complex scenario.

sensorName	embeddedOn	readValue
rainSensor1	mainVehicle	0
brightnessSensor1	mainVehicle	19
rainSensor2	uav1	X
cameraMono1	uav1	400
lidar1	uav1	X
lidar2	mainVehicle	370
radar1	uav2	700

elementName	position	speed
mainVehicle	27,33	59
uav1	X;41	0
uav2	17,45	X
obstacleVehicle1	29,46	30
uav3	53,X	X
obstacleVehicle2	51,39	X

Figure 5.30 – Some of the gathered data with random values dropped.

sensorName	embeddedOn	readValue
rainSensor1	mainVehicle	0
brightnessSensor1	mainVehicle	19
rainSensor2	uav1	0
cameraMono1	uav1	400
lidar1	uav1	380
lidar2	mainVehicle	370
radar1	uav2	700

elementName	position	speed
mainVehicle	27;33	59
uav1	27;41	0
uav2	17;45	0
obstacleVehicle1	29;46	30
uav3	53;47	0
obstacleVehicle2	51;39	40

Figure 5.31 – Data set after the completion process.

some similar ones. As shown in Chapter 3, this solution is more than 95% accurate when correctly trained, and allows the *PerceptionAccuracy* value to become **Good**.

Due to the complex nature of the scenario, multiple set of rules can be called by the reasoner. They can be grouped depending of the specific situation encountered: rules relative to the first step (obstructed view) can be found in Figure 5.32, the ones related to the second step (foggy area) are in Figure 5.33, and the set of rules for the fire hazard area can be found in Figure 5.34. It is interesting to note that due to the varying intensity of the brightness and the impact it may have on VLC, there are two different communication rules which can apply. For each variable of the rules, the corresponding simulated environment can be found in Table 5.11.

5.3.5.1 Partial assistance

Previous section showed a situation where some of the UAVs were not adequately equipped to deal with a specific environment. This experiment also presents some steps where the the decision-making process cannot be completed. More details about this can be found in Table 5.12.

Global results of the experiment can be found on Table 5.13. There was no noticeable behaviour differences between the drivers, and all showed carefulness in their driving. The unassisted test subject showing the highest completion time also had two driving

Simulator element	Associated logical individual
myVehicle	Vehicle(?v)
UAV_crossroad UAV_road UAV_urban	UAV(?u)
LiFi DSRC HybridProtocol	VLC(?v) RF(?r) Hybrid(?h) CommunicationProtocol(?c)
Fog	Fog(?f) Weather(?w)
Rock	Obstacle(?O)
CameraS1, CameraS2, CameraI1 LidarL1, LidarL2, RadarR1, RadarR2, RadarR3	CameraStereo(?cs), CameraInfra(?ci) Lidar(?l) Radar(?r) Sensors(?s)
FogDetector	FogSensor(?fs) EnvironmentalSensor(?es)
Urban_building1	Building(?b)
FireEvent	FireHazard(?fh) Hazard(?fh)
BadWeather NormalEnvironment UnusualEnvironment ObstructedView	Environment(?e)

Table 5.11 – Elements of the scenario as they are identified in the simulator and their matching variable in the logical rules

Event where the UAV is stationed	Obstructed view	Fog	Fire Hazard
Adequate sensors	Yes	No	Yes
Adequate communication protocol	Yes	Yes	No

Table 5.12 – Summary of the different situations with a faulty assistance.

```

Vehicle(?V) ^ hasDistanceFromVehicle(?O, NearDistance)
^ Environment(?Env) ^ Object(?O) ^
PerceptionAccuracy(?P)^ Good(?P)^
-> hasObstructedView(?V, ?O)

```

```

UAV(?U) ^ cameraStereo(?cs) ^ hasSensor(?U, ?cs) ^
Vehicle(?V) ^ Object(?O) ^ hasObstructedView(?V, ?O) ^
NormalEnvironment(?Env) ^ isCloseToUAV(?V,?U)->
isActiveUAV(?U, true)

```

```

UAV(?U) ^ cameraStereo(?cs) ^ hasSensor(?U, ?cs) ^
isActiveUAV(?U,true) -> isActiveSensor(?cs, true)

```

```

UAV(?U) ^ RF(?r) ^ VLC(?v) ^ Hybride(?h)
^hasCommunicationProtocol(?U, ?v)
^hasCommunicationProtocol(?U, ?h)
^hasCommunicationProtocol(?U, ?r)
^NormalEnvironment(?E)
^isActiveUAV(?U,true) ->
isActiveCommunicationProtocol(?r, true) ^
isActiveCommunicationProtocol(?v, true) ^
isActiveCommunicationProtocol(?h, true)

```

Figure 5.32 – Set of rules for the obstructed section of the scenario

```

Vehicle(?V) ^ FogSensor(?fogS) ^ hasSensor(?C,?fogS)
^ hasFogValue(?fogS, ?fogV) ^ swrlb:greaterThan(?fogV, 50) ^
PerceptionAccuracy(?P) ^ Good(?P) ^
^ Weather(?W)
->Fog(?W)

```

```

Fog(?F) ^ Environment(?Env)
-> BadWeather(?Env)

```

```

Fog(?F) ^ Vehicle(?V) ^ UAV(?u) ^ Radar(?ra) ^
cameraInfra(?c) ^ hasSensor(?u,?c) ^
hasSensor(?u,?ra) ^ isCloseToUAV(?V,?U)
-> isActiveUAV(?u,true)

```

```

UAV(?u) ^ Radar(?ra) ^ cameraInfra(?c) ^
hasSensor(?u,?c) ^ hasSensor(?u,?ra) ^ isActiveUAV(?u,true)
-> isActiveSensor(?ra, true) ^ isActiveSensor(?c, true)

```

```

UAV(?U) ^ RF(?r) ^ hasCommunicationProtocol(?U, ?v) ^
hasCommunicationProtocol(?U, ?h) ^ Hybride(?h) ^ VLC(?v) ^
BadWeather(?b) ^ hasCommunicationProtocol(?U, ?r)
-> isActiveCommunicationProtocol(?r, true) ^
isActiveCommunicationProtocol(?v, false) ^
isActiveCommunicationProtocol(?h, false)

```

Figure 5.33 – Set of rules for the foggy section of the scenario

Building(?B) ^ isOnFire(?B, true) ^ Action(?A) ^ -> FireHazard(?A)

FireHazard(?A) ^ Environment(?E) ^
PerceptionAccuracy(?P) ^ Good(?P) ^
-> UnusualEnvironment(?E)

Vehicle(?V) ^ Hazard(?H) ^ Environment(?Env) -> UnusualEnvironment(?Env)

UnusualEnvironment(?e) ^ FireHazard(?f) ^ Vehicle(?v) ^
UAV(?u) ^ Radar(?ra) ^ Lidar(?l) ^ cameraStereo(?c) ^
hasSensor(?u, ?c) ^ hasSensor(?u, ?l) ^
hasSensor(?u, ?ra) ^ isCloseToUAV(?V, ?U)
-> isActiveUAV(?u, true)

UAV(?U) ^ Radar(?ra) ^ Lidar(?l) ^ cameraStereo(?c) ^
hasSensor(?u, ?c) ^ hasSensor(?u, ?l) ^
hasSensor(?u, ?ra) ^ isActiveUAV(?U, true) ->
isActiveSensor(?c, true) ^ isActiveSensor(?l, true) ^
isActiveSensor(?ra, true)

RF(?r) ^ hasCommunicationProtocol(?U, ?v) ^ FireHazard(?f)
^ hasBrightnessValue(?br, ?bv) ^
hasCommunicationProtocol(?U, ?h) ^ brightnessSensor(?br) ^
swrlb:lessThan(?bv, 70) ^ Hybride(?h) ^ VLC(?v) ^ UAV(?U)
^ hasCommunicationProtocol(?U, ?r) ^
UnusualEnvironment(?ue) ->
isActiveCommunicationProtocol(?h, true) ^
isActiveCommunicationProtocol(?v, true) ^
isActiveCommunicationProtocol(?r, true)

RF(?r) ^ hasCommunicationProtocol(?U, ?v) ^ FireHazard(?f)
^ hasBrightnessValue(?br, ?bv) ^
hasCommunicationProtocol(?U, ?h) ^ brightnessSensor(?br) ^
Hybride(?h) ^ VLC(?v) ^ UAV(?U) ^
hasCommunicationProtocol(?U, ?r) ^ UnusualEnvironment(?ue)
^ swrlb:greaterThan(?bv, 70) ->
isActiveCommunicationProtocol(?r, true) ^
isActiveCommunicationProtocol(?v, false) ^
isActiveCommunicationProtocol(?h, false)

Figure 5.34 – Set of rules for the fire hazard section of the scenario

	Average speed	Time to complete the experiment	Max speed	Incident
Non-assisted driver 1	57km/h	118s	126km/h	0
Non-assisted driver 2	37km/h	161s	69km/h	2
Partially assisted driver 1	55km/h	108s	97km/h	0
Partially assisted driver 2	49km/h	121s	80km/h	1

Table 5.13 – Results of the complex course experiment without assistance

Event where the UAV is stationed	Obstructed view	Fog	Fire Hazard
Adequate sensors	Yes	Yes	Yes
Adequate communication protocol	Yes	Yes	Yes

Table 5.14 – Summary of the different situations with adequate assistance.

incidents, while the other unassisted subject shows performances as good as one of the other assisted driver.

This is due to the fact that because of the lack of adequate gear, the assistance process was not correctly validated (cf. Figure 5.28b). For two out of the three events, the assisted test subjects were in the same situation as the unassisted ones, meaning they only had assistance on the first step.

5.3.5.2 Full assistance

Based on the same circuit, another experiment was made by fixing the faulty sensors and communication tools, allowing for a better perception, as shown in Table 5.14. As shown by the results in Table 5.15, this group performs considerably better than the partially assisted one: the experiment was completed around 25% faster, and the average speed was also higher, showing that the drivers were more confident when assisted. There was only one incident noted, which was due to high-speed driving, as opposed to the three incidents in the non-assisted group.

	Average speed	Time to complete the experiment	Max speed	Incident
Fully assisted driver 1	73km/h	96s	119km/h	0
Fully assisted driver 2	90km/h	89s	130km/h	1
Fully assisted driver 3	79km/h	99s	111km/h	0

Table 5.15 – Results of the complex course experiment with full assistance.

5.3.6 Discussion

The different scenarios described in the previous sections allowed us to cover different situations where our proposed model could be applied and offer the service of perception enhancement to a vehicle through a UAV. When reaching an area where perception is limited, the vehicle would ask for the UAV to gather data and transmit them through a secured channel. There are some interesting points to discuss from there.

In general terms, the model performs positively, with increased driving performances and lesser driving incidents in most of the situations. The average inference time is slightly below 1 second, and both the environment inference and the perception data sharing works correctly.

However, the model showed some limitations. In some situations, the results were not as satisfying as they were expected to be, for example in a rainy situation. Bad weather has a strong impact on both sensors and communication protocols. We did not cover or model the effect of wind, which would have also had a strong impact on the UAV's stability and communication [124], but weather in general was shown to have influence on both driving and perception. On the other hand, obscurity, which is supposed to be the other important factor in perception inhibition, can be controlled thanks to embedding lighting systems on mobile agents. These elements can also double as communication tools, by relying on VLC technology. The experiments being made in a simulated environment also allowed us to bypass other regular issues that could be encountered in a real-life situation. For example, we did not need to worry about the UAV batteries or fuel refills, nor did we have to look for an adequate and large enough space to conduct our experiments.

Another important limitation of the model was due to the very nature of a knowledge-base. Indeed, it is an approach showing great potential for our situation, but it is very rigid and requires a rigorous approach, considering all the eventual elements that can be encountered on a road environment. This applies especially to the logical rules : A

Question	Percentage of positive answers
The driving experience felt realistic	76%
The perception enhancement system was useful	90%
The perception enhancement system acted fast enough	70%

Table 5.16 – Feedbacks of the test subjects on the system

single overlooked element could disable rule made up of various constraints. Hence the importance of thoroughly reviewing the literature to make sure that all the situations are covered. Many of the rules were added after preliminary tests.

Once they were done, the test subjects were asked to give their feedbacks on the general experience and answer to three questions : Did the driving experience felt realistic? Was the perception enhancement system useful? And was it fast enough when providing information?

They were given the choice to answer by Yes or No to each question, and leave an optional commentary. The first question was for the evaluation of the simulator, while the other two referred to the deductions inferred by the model and their relevance. The results can be found in Table 5.16. The 30 users were asked about the simulator and the perception enhancement system. 76% of them found the driving in the simulation realistic enough. The first users gave important feedbacks which allowed to improve the simulator and provide a better experience for the later users. Regarding the perception enhancement system, 90% of the participants found the assistance useful and were satisfied with the assistance it provided them, showing a strong interest in the model in general, and a favorable reaction for UAV applied to vehicular applications. However, 30% of them felt the process to be a little slow, but not enough for it to hinder the experiment.

5.4 Conclusion

This chapter presented the simulator developed for this thesis, as well as the experiments made and their results.

The simulator was made in Unity and represents a driving environment with a realis-

tic physics engine. It can gather environment-generated driving data (speed, position, weather...) and process them with the ontology thanks to a Python tool. Tests were made by having test subjects drive through different situations and monitor their performances with and without assistance of the knowledge-base inferring engine. The scenarios were of different nature and variations were made on the environment, sensors, or general situations in order to cover separate situations.

The results of the experiments showed that the proposed model brings an improvement to the global driving behaviour. Indeed, most of the assisted users demonstrated a safer conduct and a general faster course completion. This is due to the beforehand knowledge of any potential obstacle that would be encountered.

The limitations of the model also emerged from the experiments. There are certain environments where the UAV cannot perform correctly, and it reflects on the results. As a consequence, the model underperformed in extremely bad weather or in unusual environment like a Fire Hazard. It also relies heavily on the set of rules that it is fed, which must be constantly updated according to the different situations that can be potentially encountered.

Chapter 6

General conclusion and future works

Contents

6.1	General conclusion	158
6.2	Future works	159

6.1 General conclusion

Perception plays a major part in the safety of road users. With the advent of smart vehicles and the popularization of smart cities, it is important to find ways to reinforce this process. In the context of this thesis, a collaborative UAV/Vehicle interaction protocol was presented in order to propose an enhancement of the perception.

In Chapter 1, we introduced the motivation and problematic of this work, as well as the security and data management questions arising from them. In Chapter 2, we reviewed the current state of vehicular perception, the ontologies dedicated for road environments, and existing work for UAV and VLC for vehicular applications. We also introduced the general methodology presented in the paper. In Chapter 3, we present the notion of knowledge-base, the ontology developed for this study and its different elements, the logical rules serving as the intelligence layer and managing the system, and the steps of the inferring process. In Chapter 4, we presented the VLC protocol, its potential and limitations, and a proposed hybrid VLC/RF protocol for data security through redundancy of transmitted data. Finally, in Chapter 5, we presented the driving simulator made in Unity, the different scenarios where it was tested and a discussion of the results and the different environments influences.

In this thesis, we focused on a way to enhance the perception of a vehicle. In order to solve this problem, we tackled the challenges of finding an adequate external sensor, having them communicate in a secured way, and merging efficiently the data from different sources. From those challenges, we found some solutions and achieved the following contributions :

- An ontology for the classification and storage of the road users, the road environment, and the different properties characterizing and linking individuals between them.

- A study on driving data completion through a fuzzy logic approach on sensors' reliability and comparison of different algorithms
- A study on different uses of UAV on a road environment, and a proposition of use for perception
- A driving simulator made in Unity with realistic physics that allows for the quick building and testing of scenarios, as well as virtual data gathering and communication with the ontology
- A study on the potential strengths and weaknesses of VLC and RF communications on a road environment
- A VLC/RF Hybrid communication protocol for data protection through transmission redundancy

From the different conclusions drawn from the state of the art, we identified the problems of vehicular perception needing assistance in some environments, and proposed the use of UAV for data gathering, VLC for secured transmission and a knowledge-base for the merging of those data and their processing.

The process is done in 4 steps, mostly managed by logical rules and an inference engine: First, the correct identification of the environment, then picking the sensors working in said environment and looking for the available UAV carrying them, before choosing the communication protocol depending on the environment and the ones available on the UAV.

6.2 Future works

During this thesis, we worked on the management of data completion and came up with a KNN/Fuzzy logic methodology to deal with missing perception data. These elements have been integrated onto the knowledge-base but we believe that this aspect can be utilized

to give better results. This will probably lead to a denser population of the ontology, and numerous other logical rules, which can only be an overall improvement of the system. Naturally, by using a simulator we managed to bypass some of the common constraints we could have encountered, such as the UAV short battery life. We mentioned some possible solutions to this in Chapter 2, but this is a field where multiple works are made, and some studies show promising results [125]. Furthermore, we believe there is a real potential in the use of VLC as a complementary means of communication for security reinforcement. The technology is showing a strong potential in vehicular applications [126]. In this thesis, we presented the idea of using VLC for heavy data transmission and RF for the transmission of the digest for validation purpose. Due to lack of time and the technical complexity of the experiments, we only managed to obtain elemental results, but we would like to go even further with this idea and work on the concept of a fast-switching protocol that would take place on a PHY layer only, ensuring that one protocol takes over the other in case of incident while being transparent to the final user. We also plan to improve the simulator physics and data gathering process in order to improve the inference time and driving experience of the test users

Publications

Khezaz, A., Hina, M. D., Guan, H., & Ramdane-Cherif, A. (2020, December). Hybrid Machine Learning Model for Traffic Forecasting. In International Summit Smart City 360° (pp. 188-199). Springer, Cham. https://doi.org/10.1007/978-3-030-76063-2_14

Khezaz, A., Hina, M. D., Guan, H., & Ramdane-Cherif, A. (2020). Driving Context Detection and Validation using Knowledge-based Reasoning. In KEOD (pp. 219-226). <https://doi.org/10.5220/0010135802190226>

Khezaz, A., Hina, M. D., Guan, H., & Ramdane-Cherif, A. (2021). Perception Enhancement of a Vehicle in a Bad Weather Environment. In EAI CCom 2021.

Khezaz, A., Hina, M. D., Guan, H., & Ramdane-Cherif, A. (2021). Knowledge-Based Approach for the Perception Enhancement of a Vehicle. *Journal of Sensor and Actuator Networks*, 10(4), 66. <https://doi.org/10.3390/jsan10040066>

Appendix A

Presentation of the knowledge-base

Definition of a knowledge-base

A knowledge-base is a formal way of representing knowledge in a specific context. In this thesis, it took the form of an ontology dedicated to the driving environment.

A knowledge-base allows the representation of the various actors making up a given situation, as well as their inner properties and relationship linking them to each others. Those actors are categorized in *Classes*, and each instance of an entity of a class is referred to as an *Individual*. Individuals can have *Data properties* or *Object properties*, respectively their personal or shared properties.

Class	Individual	Data property	Object property
Road	Road_North; Road_West	hasMaxSpeed	
Vehicle	car_A; bike_A; car_B	hasSpeed	isOnRoad

Table A.1 – A representative example of an ontology components

An example is given in Table A.1. In a driving environment, we can find elements like **Roads** and **Vehicles**. There can be different elements of each, for example two different roads. Classes can contain individuals of different **Sub-classes**, as long as they belong to the same parent class: in this example, we can assume the existence of the **Car** and **Bike** classes, which are both sub-classes of **Vehicle**.

The elements also have their properties: for the Road objects, the **hasMaxSpeed** property indicates a numerical value relative to the maximum speed allowed on said road, for example 50 or 80 km/h. It can be related to the **hasSpeed** property of the Vehicles, which is another numerical value depending of the current speed of the vehicle. Also, as stated above, properties can be of the Object type and link objects between them. In this scenario, a Vehicle can have the **isOnRoad** property, linking it to a Road individual : for example **car_A isOnRoad Road_West**, stating the relationship between **car_A** and **Road_West**.

It is interesting to note that an object can belong to multiple classes, as long they are compatible. Indeed, in some cases, it is necessary to declare classes as **Disjoint**. For example, the **Vehicle** and **Pedestrian** classes are not compatible, since an individual can not be a vehicle and a pedestrian at the same time.

The knowledge-base is populated periodically every second. The new detected elements are added, while the ones not detected anymore are deleted from it.

Main Classes of the ontology

This thesis revolves around enhancing the perception ability of an autonomous vehicle. A drone is used for the gathering of additional perception a data, which should be fused with the information the vehicle can generate. A knowledge-base is an adequate tool for this task.

The ontology of this work was made with Protégé [127], an open-source ontology editor developed by the Stanford University and that has been established as an important tool for knowledge management. The important components of our knowledge-base will be presented in this section.

Class	Sub-classes	Properties
Vehicle	Car	hasPhysics ->hasSpeed
	UAV	hasSensor
	Bike	hasCommunicationProtocol
	Bus	hasDriver

Table A.2 – The Vehicle Class as it appears in the ontology

Vehicle

This thesis focuses on the the road environment, making it natural to have some obvious classes, like the Vehicle class. It is represented in Table A.2

The parent class Vehicle covers all the different type of vehicles that can be found in the environment, including Cars, Bikes, Buses but also UAV. The individuals of those sub-classes can have different properties:

- **hasSpeed (Object Property):** represents the speed at which the vehicle is given. In the knowledge-base, this value is not represented as an integer, but as a member of the **Speed** class (itself a sub-class of the **Physics** class) and its sub-classes: **NoSpeed**, **ExtraSlowSpeed**, **LowSpeed**, **NormalSpeed**, **HighSpeed**, **OverSpeed**. The conversion from a numerical value to a class object is done during the gathering process in order to facilitate the inferring process, hence making this property an Object one instead of a Data one.
- **hasSensor (Object Property):** represents all the different sensors embedded on the vehicle. It links the Vehicle object to the **Sensors** class and provides information on what sensors are available for the gathering of data.
- **hasCommunicationProtocol (Object Property):** represents the different communication protocols the vehicle can use, and is linked to the **Communication-Protocol** class.
- **hasDriver (Object Property):** is a property primarily used for the identification of the ego vehicle (the central vehicle of the experiment, from which knowledge is

Class	Sub-classes	Sub-classes
Sensors	ActivSensor	Lidar
		Sonar
		Radar
	PassiveSensor	cameraInfra
		cameraMono
		cameraStereo
	EnvironmentalSensor	brightnessSensor
		fogSensor
		rainSensor

Table A.3 – The Sensors class as it appears in the ontology, and the different sub-classes and sub-sub-classes

Class	Properties
Sensors	isActiveSensor
	hasMinRange
	hasMaxRange
	isWeakToDark
	isWeakToRain
	hasRainValue
	hasFogValue
	hasBrightnessValue

Table A.4 – Properties of the Sensors class

built). It is linked to the **Driver** class, which has only one interesting individual known as *MainDriver*.

Sensors

Sensors are a critical part of any autonomous entity. They allow the perception and understanding of the surroundings for the purpose of making the right decision. There are a variety of them, and both drones and intelligent cars have them embedded on in order to maximize and diversify the data about the environment.

The sensors chosen for this study were based on the recurring ones appearing in the state of the art from Chapter 2.

The different sub-classes of the Sensors class can be found in Table A.3. They are themselves split into different sub-classes.

- **ActiveSensors:** are sensors requiring an original impulse in order to measure a value. They are mainly used for distance measurement.
 - Sonar: are sensors using ultra-sound waves in order to measure a distance
 - Lidar: are sensors using a laser in order to measure a distance
 - Radar: are sensors using electromagnetic waves in order to measure a distance

- **PassiveSensors:** are sensors able to gather data without any form of impulse. It mainly refers to Cameras.
 - cameraMono: is a monoscopic camera. It can be used for image acquisition allowing the detection of objects or patterns.
 - cameraStereo: represents stereoscopic cameras. By using two different cameras, we can also infer informations on the depth and distance of objects.
 - cameraInfra: is an infrared camera. By using infrared technology, this sensor can work in a bad weather or bad brightness conditions.

- **EnvironmentalSensors:** are sensors used for the identification of the environment in which the vehicle is evolving. They are usually embedded on the ego vehicle (the main vehicle)
 - brightnessSensor: is used for the measurement of brightness
 - fogSensor: is used for the measurement of fog
 - rainSensor: is used for the measurement of rain

There are also different properties for the sensors, which can be found in Table A.4. Due to the nature of the properties, not all of them can be applied to all the sensors.

- **isActiveSensor (Data Property):** is a a boolean property stating if a sensor should be activated or not, depending of the environment and how well it can perform in it.

- **hasMinRange (Data Property):** is an integer property stating the minimum distance it needs in order to function properly
- **hasMaxRange (Data Property):** is an integer property stating the maximum distance it needs in order to function properly
- **isWeakToDark (Data Property):** is a boolean property informing if a specific sensor functions poorly in a dark environment
- **isWeakToRain (Data Property):** is a boolean property informing if a specific sensor functions poorly in a rainy environment
- **hasRainValue (Data Property):** is an integer property informing of the current rain value being measure by a **rainSensor** individual.
- **hasFogValue (Data Property):** is an integer property informing of the current fog value being measure by a **fogSensor** individual.
- **hasBrightnessValue (Data Property):** is an integer property informing of the current brightness value being measure by a **brightnessSensor** individual.

CommunicationProtocol

In addition to be able to perceive, an intelligent transportation system must also be able to communicate with its surroundings. There are different communication protocols, but this thesis focuses on three possibilities (detailed in Chapter 4.

Class	Sub-class	Properties
CommunicationProtocol	RF	isActiveCommunicationProtocol
	VLC	
	Hybrid	

Table A.5 – The Vehicle Class as it appears in the ontology

The details of the class can be found in Table A.5. It is a relatively straightforward class, with only three subclasses: one for VLC communication, one for RF communication, and

Class	Sub-class
Environment	Accelerate
	Brake
	ChangeLane
	RemainInTheSameLane
	SlowDown
	BadWeather
	FireHazard

Table A.6 – The Action Class as it appears in the ontology

one for the VLC/RF hybrid approach. The only property it has **isActiveCommunicationProtocol** is a data property of boolean type indicating what communication protocol is currently being used.

Action

The different elements of the knowledge-base are processed by the reasoner in order to infer informations about the environmental context and the optimal actions to make. For better readability, most of the potential outputs of the reasoner are grouped in a class named **Action**.

By using the informations store din the knowledge-base, the inference engine can take decisions on what the next move should be. It can be a direct instruction in the case of an autonomous vehicle, or an ADAS recommendation for a human driver.

- **Accelerate:** indicates that the vehicle should speed up.
- **Brake:** indicates that the vehicle should brake. There are some sub-classes depending on the origin of the instruction (BrakeForRedLight, BrakeForStop, BrakeForObstacle)
- **ChangeLane:** indicates that the vehicle should change lane.
- **RemainInTheSameLane:** indicates that the vehicle should stay in the same lane.
- **SlowDown:** indicates that the vehicle should slow down.

- **BadWeather:** indicates that a Bad Weather event has been detected, for example rain.
- **FireHazard:** indicates that a Fire Hazard event has been detected.
- **ObstructedView:** indicates that the view in front of the main vehicle is obstructed

Environment

The correct identification of the context where the vehicle is evolving is a critical step because it affects the performances of the sensors and communication protocols. The potential environments can be found in the **Environment** Class, and presented in Table A.7.

Class	Sub-class
Environment	NormalEnvironment
	BadWeather
	Dark
	UnusualEnvironment

Table A.7 – The Environment Class as it appears in the ontology

- **NormalEnvironment:** is the typical environment, with an acceptable level of brightness and weather
- **BadWeather:** represents an environment where the weather is inclement, for example in case of rain or fog.
- **Dark:** is an environment where the brightness is really poor, and alters the proper functioning of sensors
- **UnusualEnvironment:** is an environment different from the previous ones, but which can still affect the sensors and communication protocols (e.g. a fire hazard)

Other notable classes

There are multiple other objects used in the reasoning process, with a less complex hierarchical declaration. They will be listed in this section.

- **PerceptionAccuracy:** in order to strengthen the perception process, an accuracy value is defined, based on elements such as the sensors' states or the environment. It is attached to sensors through the **hasPerceptionAccuracy** property. The different values in this class depend on the output of the data cleaning and fuzzy logic process (cf. Section 3.3.1). When the perception is considered as "Good", the model knows that the incoming data are reliable
- **Object:** are other objects that can be encountered on the road and potentially serve as obstacles, for example Rocks or Traffic Lights.
- **Hazard:** is a class used when detecting an unusual event, such as a Fire Hazard
- **Weather:** is a class where the different types of weather are stored, including Sunny, Rain, Fog and Snow.

Logical rules

Logical rules are rules that are fed to a reasoner engine in order to infer new classes and properties of the existing individuals. They need to be declared before the knowledge-base starts running. The reasoner will go through the knowledge-base population and the logical rules in order to reclassify the objects and properties.

There are different rule languages, and this thesis uses the SWRL (Semantic Web Rule Language) language. The official documentation of the language gives different examples.

`Student(?X1) -> Person(?X1)`

Figure A.1 – Example of a logical rule.

Figure A.1 illustrates one of the simplest syllogisms: if an individual is a student, then

they are naturally a person. The ? operator means **Any individual**, meaning if there is at least **one** individual belonging to that class, the condition is validated.



Figure A.2 – Individual reclassification as inferred by the reasoner. The highlighted part represent the output of the reasoner.

For example, if there is a Student individual declared as *Steven*, then the reasoner will automatically identify him as a Person too.

`Student(Steven) -> Person(Steven)`

Figure A.3 – Example of a logical rule targeted to a specific individual.

It is also possible to nominally identify an individual in logical rule, as shown in Figure A.3. By omitting the ? operator and explicitly declaring a name, only individuals bearing that name will be concerned. Based on that rule, a student named *Bobby* would then not be classified as a Person by the reasoner.

Rules can also rely on Properties between objects. This is covered in the example of Figure A.4.

`hasParent(?x1,?x2) & hasBrother(?x2,?x3) -> hasUncle(?x1,?x3)`

Figure A.4 – Example of a logical rule targeted to a specific individual.

In that situation, some properties are declared **hasBrother**, **hasParent** and **hasUncle**. Those are all Object properties linking individuals from the **Person** class to others of the same class. The rule states that if any individual has a **hasParent** property to an individual with the **hasBrother** to property to a third individual, then the first and the third are linked by the **hasUncle** property. In Figure A.5, *George* is declared as having *Bobby* as a parent, who in turn has *Steven* as a brother, making *Steven* the Uncle of *George*.

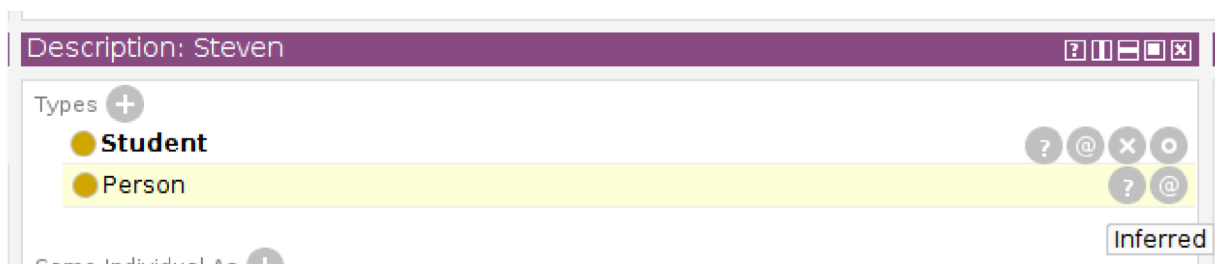


Figure A.5 – New property inferred by the reasoner. The highlighted part represent the output of the reasoner.

Appendix B

Résumé de thèse

La miniaturisation poussée des composants électroniques ainsi que leurs prix de plus en plus abordables permettent d'implémenter une couche "d'intelligence" à de nombreux objets du quotidien, ce qui leur permet de communiquer et de traiter des informations de natures diverses afin d'améliorer la qualité de vie générale des humains.

L'environnement routier n'est pas en reste. En effet, il existe de plus en plus de véhicules connectés et autonomes sur les routes, et une croissance de leur nombre est encore prévue sur les années à venir. Il s'agit de véhicules qui ont la capacité de communiquer avec d'autres entités intelligentes et de se déplacer avec une assistance humaine réduite.

Le processus de navigation d'un véhicule autonome repose sur quatre étapes : Percevoir, Identifier, Décider et Agir. Il faut tout d'abord récupérer des informations sur l'environnement, puis les interpréter afin d'identifier les éléments environnants et de prendre une décision à partir de ses éléments avant de réaliser une action physique grâce aux actionneurs du véhicule.

Face au nombre croissant de véhicules autonomes, il est important de continuer de garantir la sécurité des usagers de la route. Pour un conducteur humain, la sécurité est garantie par divers éléments, notamment par une bonne visibilité de l'environnement. Une approche

similaire peut être prise pour les véhicules autonomes : en augmentant la capacité et le rayon de perception initiale du véhicule, nous pouvons renforcer l'ensemble du processus de navigation.

Cette thèse propose de réaliser cela grâce à l'utilisation de drones. La plupart des travaux sur les drones dans un milieu routier proposent de les utiliser à des fins de communication, néanmoins du fait de leur capacité à voler, leur légèreté et leur mobilité inhérente, ces derniers peuvent être déployés dans certaines zones où la visibilité est réduite afin de récolter des données de perception grâce à divers capteurs embarqués.

Naturellement, ces données doivent ensuite être transmises au véhicule de manière sécurisée. À cet effet, nous proposons l'utilisation d'un nouveau protocole de communication hybride basé sur l'utilisation de la technologie VLC (Visible Light Communication). Cette technique consiste à moduler de la lumière visible au delà d'une certaine fréquence afin de pouvoir transmettre des informations tout en gardant un éclairage adéquat pour l'œil humain. Cette thèse propose de coupler cette technologie avec des méthodes de communication radio déjà existantes afin de renforcer la sécurité et l'intégrité des données transmises: les données seraient transmises par paquets sur un canal (VLC), tandis que leur *digests* seraient communiqués sur un deuxième canal (RF). Cela garantirait une protection contre l'éventuelle corruption des données, et permettrait de détecter les paquets défectueux et de demander un renvoi si nécessaire.

Les données de perception générées par le drone seraient alors transmises au véhicule afin d'être fusionnées avec celles récoltées localement. Ceci se fait via le biais d'une base de connaissance, une méthode formelle pour la représentation des connaissances d'un contexte spécifique : les éléments détectés sont classifiés selon différents types, et leurs propriétés spécifiques ainsi que leurs relations internes sont également inscrites dans la base de connaissance.

Un raisonneur est également rajouté en aval de la base de connaissance et plusieurs règles logiques lui sont fournis. Le traitement des différentes informations stockées dans la base de connaissance permet d'identifier l'environnement dans lequel le véhicule avance, ainsi que d'évaluer le niveau de perception global et de requérir l'utilisation de données supplémentaires si besoin afin de prendre une décision adéquate.

Le modèle proposé dans cette thèse a été validé grâce à un simulateur de conduite virtuel développé pour l'occasion : des données de capteurs de conduite sont générées au fil de l'expérience, et sont stockées dans la base de connaissance avant d'être traitées. Plusieurs scénarios ont été développés et testés par des conducteurs volontaires, et il en résulte une amélioration du comportement de conduite général, ainsi qu'un environnement plus sécurisé.

References

1. *List of countries by vehicles per capita* Dec. 2021. https://en.wikipedia.org/wiki/List_of_countries_by_vehicles_per_capita.
2. Fayyad, J., Jaradat, M. A., Gruyer, D. & Najjaran, H. Deep Learning Sensor Fusion for Autonomous Vehicle Perception and Localization: A Review. en. *Sensors* **20**, 4220. ISSN: 1424-8220. <https://www.mdpi.com/1424-8220/20/15/4220> (2021) (July 2020).
3. Abderraouf, K., Hina, M. D., Guan, H. & Ramdane-Cherif, A. *Hybrid Machine Learning Model for Traffic Forecasting in International Summit Smart City 360°* (2020), 188–199.
4. Khezaz, A., Hina, M. D., Guan, H. & Ramdane-Cherif, A. Knowledge-Based Approach for the Perception Enhancement of a Vehicle. *Journal of Sensor and Actuator Networks* **10**, 66 (2021).
5. Khezaz, A., Hina, M. D., Guan, H. & Ramdane-Cherif, A. *Driving Context Detection and Validation using Knowledge-based Reasoning*. in *KEOD* (2020), 219–226.
6. Das, S., Brimley, B. K., Lindheimer, T. E. & Zupancich, M. Association of reduced visibility with crash outcomes. en. *IATSS Research* **42**, 143–151. ISSN: 03861112. <https://linkinghub.elsevier.com/retrieve/pii/S0386111216300681> (Oct. 2018).
7. World Health Organization. *Save LIVES: a road safety technical package* en. OCLC: 1131688778. ISBN: 978-92-4-151170-4 (World Health Organization, Geneva, 2017).
8. Bengler, K. *et al.* Three Decades of Driver Assistance Systems: Review and Future Perspectives. en. *IEEE Intelligent Transportation Systems Magazine* **6**, 6–22. ISSN: 1939-1390. <http://ieeexplore.ieee.org/document/6936444/> (2019) (2014).

9. Van Brummelen, J., O'Brien, M., Gruyer, D. & Najjaran, H. Autonomous vehicle perception: The technology of today and tomorrow. en. *Transportation Research Part C: Emerging Technologies* **89**, 384–406. ISSN: 0968090X. <https://linkinghub.elsevier.com/retrieve/pii/S0968090X18302134> (2020) (Apr. 2018).
10. Campbell, M., Egerstedt, M., How, J. P. & Murray, R. M. Autonomous driving in urban environments: approaches, lessons and challenges. en. *Phil. Trans. R. Soc. A.* **368**, 4649–4672. ISSN: 1364-503X, 1471-2962. <https://royalsocietypublishing.org/doi/10.1098/rsta.2010.0110> (2020) (Oct. 2010).
11. Vanholme, B., Gruyer, D., Lusetti, B., Glaser, S. & Mammar, S. Highly Automated Driving on Highways Based on Legal Safety. en. *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS* **14**, 15 (2013).
12. Woll, J. *Monopulse Doppler radar for vehicle applications* en. in *Proceedings of the Intelligent Vehicles '95. Symposium* (IEEE, Detroit, MI, USA, 1995), 42–47. ISBN: 978-0-7803-2983-6. <http://ieeexplore.ieee.org/document/528255/> (2020).
13. Mayhan, R. J. & Bishel, R. A. A two-frequency radar for vehicle automatic lateral control. *IEEE Transactions on Vehicular Technology* **31**, 32–39 (1982).
14. Rasshofer, R. H. & Gresser, K. *Advances in Radio Science*. en, 5.
15. Sivaraman, S. & Trivedi, M. M. Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis. en. *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS* **14**, 23 (2013).
16. Sahin, F. E. Long-Range, High-Resolution Camera Optical Design for Assisted and Autonomous Driving. en. *Photonics* **6**, 73. ISSN: 2304-6732. <https://www.mdpi.com/2304-6732/6/2/73> (2020) (June 2019).
17. Smith, M. Light detection and ranging (LIDAR), volume 2. A bibliography with abstracts. *ntis* (1978).

18. Li, B., Zhang, T. & Xia, T. Vehicle Detection from 3D Lidar Using Fully Convolutional Network. en. *arXiv:1608.07916 [cs]*. arXiv: 1608.07916. <http://arxiv.org/abs/1608.07916> (2020) (Aug. 2016).
19. Mahlisch, M., Schweiger, R., Ritter, W. & Dietmayer, K. *Sensorfusion Using Spatio-Temporal Aligned Video and Lidar for Improved Vehicle Detection* en. in *2006 IEEE Intelligent Vehicles Symposium* (IEEE, Meguro-Ku, Japan, 2006), 424–429. ISBN: 978-4-901122-86-3. <http://ieeexplore.ieee.org/document/1689665/> (2020).
20. Sentinel, M. Phantom Auto'will tour city. *The Milwaukee Sentinel*, 4 (1926).
21. Skeete, J.-P. Level 5 autonomy: The new face of disruption in road transport. en. *Technological Forecasting and Social Change* **134**, 22–34. ISSN: 00401625. <https://linkinghub.elsevier.com/retrieve/pii/S0040162517314737> (2020) (Sept. 2018).
22. Menouar, H. *et al.* UAV-Enabled Intelligent Transportation Systems for the Smart City: Applications and Challenges. en. *IEEE Communications Magazine* (2017).
23. *Starship Technologies* <https://www.starship.xyz>.
24. *Google X Wing* <https://wing.com/>.
25. *Amazon Prime Air* www.amazon.com/primeair.
26. Sakiyama, M. *The balance between privacy and safety in police UAV use: The power of treat and its effect on people's receptivity* PhD thesis (2017).
27. Galkin, B. & DaSilva, L. A. UAVs as Mobile Infrastructure: Addressing Battery Lifetime (July 3, 2018).
28. Xu, Y., Yu, G., Wang, Y., Wu, X. & Ma, Y. Car Detection from Low-Altitude UAV Imagery with the Faster R-CNN. en. *Journal of Advanced Transportation* **2017**, 1–10. ISSN: 0197-6729, 2042-3195. <https://www.hindawi.com/journals/jat/2017/2823617/> (2019) (2017).

29. Hadiwardoyo, S. A., Hernández-Orallo, E., Calafate, C. T., Cano, J. C. & Manzoni, P. Experimental characterization of UAV-to-car communications. *Computer Networks* (May 2018).
30. Menouar, H. *et al.* UAV-Enabled Intelligent Transportation Systems for the Smart City: Applications and Challenges (2017).
31. Shi, W. *et al.* Drone Assisted Vehicular Networks: Architecture, Challenges and Opportunities. *IEEE Network* (May 2018).
32. Bendadouche, R., Roussey, C., de Sousa, G., Chanet, J.-P. & Hou, K. M. Etat de l'art sur les ontologies de capteurs pour une intégration intelligente des données. fr, 16.
33. Calder, M., Morris, R. A. & Peri, F. Machine reasoning about anomalous sensor data. en. *Ecological Informatics* **5**, 9–18. ISSN: 15749541. <https://linkinghub.elsevier.com/retrieve/pii/S1574954109000715> (Jan. 2010).
34. Compton, M., Neuhaus, H., Taylor, K. & Tran, K.-N. Reasoning about Sensors and Compositions. en, 16.
35. Lefort, L. *et al.* Semantic sensor network xg final report (2011).
36. *Mechanism of VICS* <https://www.vics.or.jp/en/know/structure/index.html>.
37. Alam, M., Ferreira, J. & Fonseca, J. *Intelligent transportation systems* (Springer Berlin Heidelberg, 2016).
38. Akanegawa, M., Tanaka, Y. & Nakagawa, M. Basic study on traffic information system using LED traffic lights. *IEEE Transactions on Intelligent Transportation Systems* (2001).
39. B.Bechadergue. *Visible Light Range-Finding and Communication Using the Automotive LED Lighting* PhD thesis (Université Paris Saclay, 2017).

40. Goto, Y. *et al.* A New Automotive VLC System Using Optical Communication Image Sensor. *IEEE Photonics Journal* (June 2016).
41. Yamazato, T. V2X communications with an image sensor. *Journal of Communications and Information Networks* (Dec. 2017).
42. Jin, S. Y., Choi, D. & Kim, B. W. Optical Vehicle to Vehicle Communications for Autonomous Mirrorless Cars. *Journal of Multimedia Information System* (2018).
43. Yu, S.-H., Shih, O., Tsai, H.-M., Wisitpongphan, N. & Roberts, R. D. Smart automotive lighting for vehicle safety. *IEEE Communications Magazine* (2013).
44. Cui, K., Chen, G., Xu, Z. & Roberts, R. D. Traffic light to vehicle visible light communication channel characterization. *Applied Optics* (2012).
45. Kim, Y. H., Cahyadi, W. A. & Chung, Y. H. Experimental Demonstration of VLC-Based Vehicle-to-Vehicle Communications Under Fog Conditions. *IEEE Photonics Journal*.
46. Fuada, S., Putra, A. P., Aska, Y. & Adiono, T. A First Approach to Design Mobility Function and Noise Filter in VLC System Utilizing Low-cost Analog Circuits (2017).
47. Adiono, T. & Fuada, S. Optical Interference Noise Filtering over Visible Light Communication System utilizing Analog High-Pass Filter Circuit, 4.
48. Haigh, P. A. *Using Equalizers to Increase Data Rates in Organic Photonic Devices for Visible Light Communications Systems* PhD thesis ().
49. Haigh, P. A. *et al.* A 20-Mb/s VLC Link With a Polymer LED and a Multilayer Perceptron Equalizer. *IEEE Photonics Technology Letters*.
50. Haigh, P. A., Ghassemlooy, Z., Papanikolaou, I. & Le Minh, H. 2.7 Mb/s With a 93-kHz White Organic Light Emitting Diode and Real Time ANN Equalizer. *IEEE Photonics Technology Letters*.

51. Ghassemlooy, Z. *et al.* Visible light communications: 375 Mbits/s data rate with a 160 kHz bandwidth organic photodetector and artificial neural network equalization [Invited]. *Photonics Research*.
52. Rajbhandari, S., Haigh, P. A., Ghassemlooy, Z. & Popoola, W. Wavelet-Neural Network VLC Receiver in the Presence of Artificial Light Interference. *IEEE Photonics Technology Letters*.
53. Zagzebski, L. What is knowledge? *The Blackwell guide to epistemology*, 92–116 (2017).
54. Nickols, F. The knowledge in knowledge management. *The Knowledge Management Yearbook, 2000–2001* **12**, 21 (2000).
55. Endsley, M. R. in *Situational awareness* 9–42 (Routledge, 2017).
56. Baumann, M. R., Petzoldt, T. & Krems, J. F. Situation Awareness beim Autofahren als Verstehensprozess. *MMI Interaktiv-Aufmerksamkeit und Situationawareness beim Autofahren: Vol. 1, No. 11* (2006).
57. Schilit, B., Adams, N. & Want, R. *Context-aware computing applications in 1994 First Workshop on Mobile Computing Systems and Applications* (1994), 85–90.
58. Chen, G. & Kotz, D. A survey of context-aware mobile computing research (2000).
59. Hofer, T. *et al.* *Context-awareness on mobile devices-the hydrogen approach in 36th annual Hawaii international conference on system sciences, 2003. Proceedings of the* (2003), 10–pp.
60. Fuchs, S., Rass, S. & Kyamakya, K. Integration of Ontological Scene Representation and Logic-Based Reasoning for Context-Aware Driver Assistance Systems. en. *Electronic Communications of the EASST*. Publisher: European Association of Software Science and Technology, Volume 11: Contextaware Adaption Mechanisms for Pervasive and Ubiquitous Services. <http://journal.ub.tu-berlin.de/eceasst/article/view/127> (2020) (June 2008).

61. Starfield, T. Principles of Modeling: Real World - Model World. *University of Vermont Lectures* (2005).
62. Balci, O. & Smith, E. P. *Validation of expert system performance* tech. rep. (Department of Computer Science, Virginia Polytechnic Institute & State ..., 1986).
63. Jaffar, J. & Maher, M. J. Constraint logic programming: a survey. *The Journal of Logic Programming* **19-20**, 503–581 (1994).
64. Paulheim, H. Knowledge graph refinement: A survey of approaches and evaluation methods. en. *SW* **8** (ed Cimiano, P.) 489–508. ISSN: 22104968, 15700844. <https://www.medra.org/servlet/aliasResolver?alias=iospress&doi=10.3233/SW-160218> (2020) (Dec. 2016).
65. Noy, N. & McGuinness, D. Ontology Development 101: A Guide to Creating Your First Ontology. *Knowledge Systems Laboratory* **32** (Jan. 2001).
66. Horridge, M. & Bechhofer, S. The OWL API: A Java API for OWL ontologies. en. *Semantic Web* **2**, 11–21. ISSN: 15700844. <https://www.medra.org/servlet/aliasResolver?alias=iospress&doi=10.3233/SW-2011-0025> (2020) (2011).
67. Lahat, D., Adali, T. & Jutten, C. Multimodal data fusion: an overview of methods, challenges, and prospects. *Proceedings of the IEEE* **103**, 1449–1477 (2015).
68. Hall, D. L. & Llinas, J. An introduction to multisensor data fusion. *Proceedings of the IEEE* **85**, 6–23 (1997).
69. Hina, M. D., Thierry, C., Soukane, A. & Ramdane-Cherif, A. Cognition of Driving Context for Driving Assistance. en. **12**, 11 (2018).
70. Bobzien, S. in *The Stanford Encyclopedia of Philosophy* (ed Zalta, E. N.) Summer 2020 (Metaphysics Research Lab, Stanford University, 2020).
71. O'Connor, M., Nyulas, C., Shankar, R., Das, A. & Musen, M. The SWRLAPI: A Development Environment for Working with SWRL Rules. en, 5.

72. Kang, H. The prevention and handling of the missing data. *Korean journal of anesthesiology* **64**, 402 (2013).
73. Gleason, T. C. & Staelin, R. A proposal for handling missing data. *Psychometrika* **40**, 229–252 (1975).
74. Davis, G. A. & Nihan, N. L. Nonparametric Regression and Short-Term Freeway Traffic Forecasting. en. *Journal of Transportation Engineering* **117**, 178–188. ISSN: 0733-947X, 1943-5436. <http://ascelibrary.org/doi/10.1061/%28ASCE%290733-947X%281991%29117%3A2%28178%29> (2020) (Mar. 1991).
75. Clark, S. Traffic Prediction Using Multivariate Nonparametric Regression. en. *Journal of Transportation Engineering* **129**, 161–168. ISSN: 0733-947X, 1943-5436. <http://ascelibrary.org/doi/10.1061/%28ASCE%290733-947X%282003%29129%3A2%28161%29> (2020) (Mar. 2003).
76. Vlahogianni, E. I., Karlaftis, M. G. & Golias, J. C. Short-term traffic forecasting: Where we are and where we're going. en. *Transportation Research Part C: Emerging Technologies* **43**, 3–19. ISSN: 0968090X. <https://linkinghub.elsevier.com/retrieve/pii/S0968090X14000096> (2020) (June 2014).
77. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. en. *Nature* **521**, 436–444. ISSN: 0028-0836, 1476-4687. <http://www.nature.com/articles/nature14539> (2019) (May 2015).
78. Ma, X., Tao, Z., Wang, Y., Yu, H. & Wang, Y. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. en. *Transportation Research Part C: Emerging Technologies* **54**, 187–197. ISSN: 0968090X. <https://linkinghub.elsevier.com/retrieve/pii/S0968090X15000935> (2020) (May 2015).
79. Min, W. & Wynter, L. Real-time road traffic prediction with spatio-temporal correlations. en. *Transportation Research Part C: Emerging Technologies* **19**, 606–

616. ISSN: 0968090X. <https://linkinghub.elsevier.com/retrieve/pii/S0968090X10001592> (2020) (Aug. 2011).
80. Mazumder, R., Hastie, T. & Tibshirani, R. Spectral Regularization Algorithms for Learning Large Incomplete Matrices. en, 36.
81. Scheffer, J. Dealing with Missing Data. en. **3**, 8 (2002).
82. Hall, P., Park, B. U. & Samworth, R. J. Choice of neighbor order in nearest-neighbor classification. en. *Ann. Statist.* **36**. arXiv: 0810.5276, 2135–2152. ISSN: 0090-5364. <http://arxiv.org/abs/0810.5276> (2020) (Oct. 2008).
83. Tibshirani, S. & Friedman, H. Valerie and Patrick Hastie. en, 764.
84. Zadeh, L. A. Is there a need for fuzzy logic? *Information sciences* **178**, 2751–2779 (2008).
85. Horridge, M. A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools Edition 1.3. en, 108.
86. Chen, V. C., Li, F., Ho, S.-S. & Wechsler, H. Micro-Doppler effect in radar: phenomenon, model, and simulation study. *IEEE Transactions on Aerospace and electronic systems* **42**, 2–21 (2006).
87. Luciano, D. & Prichett, G. Cryptology: From Caesar ciphers to public-key cryptosystems. *The College Mathematics Journal* **18**, 2–17 (1987).
88. Deng, F.-G. & Long, G. L. Secure direct communication with a quantum one-time pad. *Physical Review A* **69**, 052319 (2004).
89. Borisov, N., Goldberg, I. & Brewer, E. *Off-the-record communication, or, why not to use PGP* in *Proceedings of the 2004 ACM workshop on Privacy in the electronic society* (2004), 77–84.
90. Cocks, C. *An identity based encryption scheme based on quadratic residues* in *IMA international conference on cryptography and coding* (2001), 360–363.

91. Kim, S.-J. & Joe, I.-J. Management Method for Private Key File of PKI using Container ID of USB memory. *The Journal of the Korea Contents Association* **15**, 607–615 (2015).
92. Pérez-Resca, A., Garcia-Bosque, M., Sánchez-Azqueta, C. & Celma, S. Physical layer encryption for industrial ethernet in gigabit optical links. *IEEE Transactions on Industrial Electronics* **66**, 3287–3295 (2018).
93. Lee, J.-S., Su, Y.-W. & Shen, C.-C. *A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi* in *IECON 2007-33rd Annual Conference of the IEEE Industrial Electronics Society* (2007), 46–51.
94. Çengel, Y. A. & Ghajar, A. J. *Heat and Mass Transfer: Fundamentals [and] Applications* (McGraw-Hill Education, 2020).
95. Van Herwaarden, A. & Sarro, P. Thermal sensors based on the Seebeck effect. *Sensors and Actuators* **10**, 321–346 (1986).
96. Jouhari, M., Ibrahim, K., Tembine, H. & Ben-Othman, J. Underwater wireless sensor networks: A survey on enabling technologies, localization protocols, and internet of underwater things. *IEEE Access* **7**, 96879–96899 (2019).
97. Lanzagorta, M. Underwater communications. *Synthesis lectures on communications* **5**, 1–129 (2012).
98. Francis F, C. *Introduction to plasma physics and controlled fusion* 2016.
99. Galisteo, A., Juara, D. & Giustiniano, D. *Research in visible light communication systems with OpenVLC1. 3* in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)* (2019), 539–544.
100. Tirumala, A. Iperf: The TCP/UDP bandwidth measurement tool. <http://dast.nlanr.net/Projects/Iperf/> (1999).

101. Nauryzbayev, G., Abdallah, M. & Al-Dhahir, N. Outage Analysis of Cognitive Electric Vehicular Networks over Mixed RF/VLC Channels. en. *arXiv:2004.11143 [cs, math]*. arXiv: 2004.11143. <http://arxiv.org/abs/2004.11143> (2020) (Apr. 2020).
102. Marabissi, D. *et al.* Experimental Measurements of a Joint 5G-VLC Communication for Future Vehicular Networks. en. *JSAN* **9**, 32. ISSN: 2224-2708. <https://www.mdpi.com/2224-2708/9/3/32> (2020) (July 2020).
103. Rahaim, M. B., Vegni, A. M. & Little, T. D. C. *A hybrid Radio Frequency and broadcast Visible Light Communication system* en. in *2011 IEEE GLOBECOM Workshops (GC Wkshps)* (IEEE, Houston, TX, USA, Dec. 2011), 792–796. ISBN: 978-1-4673-0040-7 978-1-4673-0039-1 978-1-4673-0038-4. <http://ieeexplore.ieee.org/document/6162563/>.
104. Rakia, T., Yang, H.-C., Gebali, F. & Alouini, M.-S. Optimal Design of Dual-Hop VLC/RF Communication System With Energy Harvesting. en. *IEEE Commun. Lett.* **20**, 1979–1982. ISSN: 1089-7798. <http://ieeexplore.ieee.org/document/7524747/> (2021) (Oct. 2016).
105. Pan, G., Ye, J. & Ding, Z. Secure Hybrid VLC-RF Systems with Light Energy Harvesting. en. *IEEE Trans. Commun.*, 1–1. ISSN: 0090-6778. <http://ieeexplore.ieee.org/document/7935472/> (2021) (2017).
106. Tonguz, O., Wisitpongphan, N., Bai, F., Mudalige, P. & Sadekar, V. Broadcasting in VANET, 7–12 (2007).
107. Rivest, R. & Dusse, S. *The MD5 message-digest algorithm* 1992.
108. Pamuła, D. & Zi, A. Securing video stream captured in real time. en, 3.
109. Webb, W. T. & Hanzo, L. *Modern Quadrature Amplitude Modulation: Principles and applications for fixed and wireless channels: one* (IEEE Press-John Wiley, 1994).

110. Haigh, P. A., Ghassemlooy, Z., Rajbhandari, S. & Papakonstantinou, I. Visible light communications using organic light emitting diodes. en. *IEEE Commun. Mag.* **51**, 148–154. ISSN: 0163-6804. <http://ieeexplore.ieee.org/document/6576353/> (2021) (Aug. 2013).
111. Haigh, P. A., Ghassemlooy, Z., Rajbhandari, S., Papakonstantinou, I. & Popoola, W. Visible Light Communications: 170 Mb/s Using an Artificial Neural Network Equalizer in a Low Bandwidth White Light Configuration. en. *J. Lightwave Technol.* **32**, 1807–1813. ISSN: 0733-8724, 1558-2213. <http://ieeexplore.ieee.org/document/6781659/> (2019) (May 2014).
112. Shi, M. *et al.* A 5Gb/s 2×2 MIMO Real-time Visible Light Communication System based on silicon substrate. en, 5.
113. Ahamed, S. *Visible Light Communication in Railways* en. in *International Conference on Railway Engineering (ICRE 2016)* (Institution of Engineering and Technology, Brussels, Belgium, 2016), 7 (5 .)–7 (5 .) ISBN: 978-1-78561-292-3. <https://digital-library.theiet.org/content/conferences/10.1049/cp.2016.0516> (2021).
114. Group, R. P. *bibinitperiod S. I. NATIONAL RAIL & UNDERGROUND CLOSED CIRCUIT TELEVISION (CCTV) GUIDANCE DOCUMENT* Oct. 2015.
115. Perry, V. H. & Zakariasen, D. First Use of Ultra-High Performance Concrete for an Innovative Train Station Canopy. en, 8 (2004).
116. Jun Horikoshi, Tanaka, K. & Morinaga, T. 1.2 GHz band wave propagation measurements in concrete building for indoor radio communications. en. *IEEE Trans. Veh. Technol.* **35**, 146–152. ISSN: 0018-9545, 1939-9359. <http://ieeexplore.ieee.org/document/1623397/> (2021) (Nov. 1986).
117. IT-CCTV-RP, A. Selection of Cameras, Digital Recording Systems, Digital High-Speed Networks and Trainlines for Use in Transit-Related CCTV Systems (2011).

118. Wenger, S. H. 264/avc over ip. *IEEE transactions on circuits and systems for video technology* **13**, 645–656 (2003).
119. Udacity. *Udacity Self-Driving car project* <https://github.com/udacity/self-driving-car-sim>. 2017.
120. Haas, J. K. A history of the unity game engine (2014).
121. Heng, L. *et al.* *Project autovision: Localization and 3d scene perception for an autonomous vehicle with a multi-camera system* in *2019 International Conference on Robotics and Automation (ICRA)* (2019), 4695–4702.
122. Gao, H. *et al.* Object classification using CNN-based fusion of vision and LIDAR in autonomous vehicle environment. *IEEE Transactions on Industrial Informatics* **14**, 4224–4231 (2018).
123. Steinbaeck, J., Steger, C., Holweg, G. & Druml, N. *Next generation radar sensors in automotive sensor fusion systems* in *2017 Sensor Data Fusion: Trends, Solutions, Applications (SDF)* (2017), 1–6.
124. Pflimlin, J. M., Soueres, P. & Hamel, T. *Hovering flight stabilization in wind gusts for ducted fan UAV* in *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)* **4** (2004), 3491–3496.
125. Li, N. *et al.* Study on the environmental adaptability of lithium-ion battery powered UAV under extreme temperature conditions. *Energy* **219**, 119481 (2021).
126. Plascencia, E., Shagdar, O., Guan, H., Barrois, O. & Chassagne, L. Optical CDMA MAC Evaluation in Vehicle-to-Vehicle Visible Light Communications. *Electronics* **11**, 1454 (2022).
127. Musen, M. A. The protégé project: a look back and a look forward. en. *AI Matters* **1**, 4–12. ISSN: 2372-3483, 2372-3483. <https://dl.acm.org/doi/10.1145/2757001.2757003> (2020) (June 2015).

Titre: Amélioration de la capacité de perception d'un véhicule autonome grâce à l'utilisation de drones

Mots clés: Véhicules autonomes, Drones, VLC, Base de connaissances

Résumé: Du fait de l'évolution des méthodes électronique de calcul et de communication, le concept de ville intelligente prend rapidement forme. Les nombreux capteurs et actionneurs positionnés dans les espaces habités permettent un meilleur contrôle des données qui transitent entre les usagers, et une optimisation du confort de vie.

L'utilisation de véhicules reste l'un des moyens de déplacement les plus utilisés dans le monde. Là-aussi, les avancées technologiques ont permis des évolutions permettant d'avoir des voitures qui se déplacent par elle-même, avec pas ou peu d'assistance humaine. Il est cependant

important de considérer la sécurité des usagers de la route dans cet environnement de plus en plus peuplé.

Cette thèse propose d'améliorer la capacité de perception d'un véhicule autonome grâce à l'utilisation de drones. Du fait de leur positionnement avantageux et de leur taille réduite, ces derniers peuvent récolter des données de perception et les transmettre au véhicule grâce à une méthode sécurisée, par exemple une communication VLC. Les données des différentes sources sont ensuite fédérées et traitées grâce à une base de connaissance et un jeu de règles logiques.

Title: Perception Enhancement of an Autonomous Vehicle through the use of UAV

Keywords: ITS, UAV, VLC, Knowledge-base

Abstract: Due to the evolution of electronic calculation and communication methods, the concept of the intelligent city is rapidly taking shape. The numerous sensors and actuators positioned in inhabited spaces allow a better monitoring of the data that passes between users, and an optimization of living comfort.

The use of vehicles remains one of the most widely used means of travel in the world. Here too, technological advances have allowed evolution enabling cars to drive themselves, with little to no human assistance. However, it is impor-

tant to consider the safety of road users in this increasingly populated environment. This thesis proposes to improve the perception capability of an autonomous vehicle through the use of drones. Due to their advantageous positioning and small size, they can collect perception data and transmit them to the vehicle using a secure method, such as VLC communication. The data from the different sources are then federated and processed thanks to a knowledge base and a set of logical rules.

