



HAL
open science

Knowledge-based approach for the perception enhancement of a vehicle

Abderraouf Khezaz, Manolo Dulva Hina, Hongyu Guan, Amar Ramdane-Cherif

► To cite this version:

Abderraouf Khezaz, Manolo Dulva Hina, Hongyu Guan, Amar Ramdane-Cherif. Knowledge-based approach for the perception enhancement of a vehicle. *Journal of sensor and actuator networks*, 2021, 10 (4), <10.3390/jsan10040066>. <hal-03487758>

HAL Id: hal-03487758

<https://hal.science/hal-03487758v1>

Submitted on 2 Jun 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

Article

Knowledge-Based Approach for the Perception Enhancement of a Vehicle[†]

Abderraouf Khezaz^{1,2,*}, Manolo Dulva Hina¹ , Hongyu Guan² and Amar Ramdane-Cherif²

¹ ECE Paris Engineering School, 75015 Paris, France; manolo-dulva.hina@ece.fr

² Laboratoire LISV, Université Versailles Saint-Quentin, 78000 Versailles, France; hongyu.guan@lisv.uvsq.fr (H.G.); amar.ramdane-cherif@uvsq.fr (A.R.-C.)

* Correspondence: abderraouf.khezaz@ece.fr

† This paper is an extended version of our paper published in EAI CCom 2021—Perception Enhancement of a Vehicle in a Bad Weather Environment, Versailles, France, 18–19 November 2021.

Abstract: An autonomous vehicle relies on sensors in order to perceive its surroundings. However, there are multiple causes that would hinder a sensor's proper functioning, such as bad weather or lighting conditions. Studies have shown that rainfall and fog lead to a reduced visibility, which is one of the main causes of accidents. This work proposes the use of a drone in order to enhance the vehicle's perception, making use of both embedded sensors and its advantageous 3D positioning. The environment perception and vehicle/Unmanned Aerial Vehicle (UAV) interactions are managed by a knowledge base in the form of an ontology, and logical rules are used in order to detect and infer the environmental context and UAV management. The model was tested and validated in a simulation made on Unity.

Keywords: knowledge base; ontology; UAV; autonomous vehicle



Citation: Khezaz, A.; Hina, M.D.; Guan, H.; Ramdane-Cherif, A. Knowledge-Based Approach for the Perception Enhancement of a Vehicle. *J. Sens. Actuator Netw.* **2021**, *10*, 66. <https://doi.org/10.3390/jsan10040066>

Academic Editor: Qing Yang

Received: 1 October 2021

Accepted: 9 November 2021

Published: 18 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

An autonomous vehicle is legally defined as a “vehicle that uses artificial intelligence, sensors, global positioning system coordinates, or any other technology to carry out the mechanical operations of driving without the active control and continuous monitoring of a human operator” [1,2], meaning that it uses both software and hardware elements in order to perceive its surroundings and safely navigate them. It also implies that sensors are very important added component of an autonomous vehicle, since they act as a perception tool to the vehicle.

That being said, sensors remain electronic components, and they can only operate under the right conditions. By its nature, an autonomous vehicle is required to navigate into different environments, and the variations of brightness and weather might have an impact on the sensors' efficiency.

On the other hand, an autonomous vehicle is also evolving in an environment which is becoming more intelligent and more connected, and the concept of “*smart city*” is slowly taking shape [3]. It relies on different entities' ability to communicate in order to exchange information and ensure security. It would be interesting to use this in order to enhance the vehicle's perception in a situation where it is needed. It has been shown, for example, that vehicular sensors do not work well in a bad-weather or bad-lighting situations, but there could be ways to solve this issue using new technologies, such as UAVs (Unmanned Aerial Vehicle, also known as drones).

In addition to the perception process, data should be processed and calculated through a layer of intelligence in order to guarantee the road users' safety: Decision-making can be implemented in many different ways, such as by using machine-learning approaches [4], statistical methods [5] or logical rules [6].

In 2007, Fuchs et al. [7] submitted the idea of exploiting the vehicle's context in order to offer a better Driver Assistance System (DAS). "Context" being an ambiguous word, the definition chosen was the one from Endsley in 1995 [8,9] and it is characterized 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". They notably proposed to hierarchize the context and sub-contexts in order to efficiently exploit the different information coming from them, identifying in the process the different levels of contexts (spatial context, local context, traffic objects, participants, road conditions, etc.). By fusing the data from those different but parallel situations, the intelligent system would then be able to advise the driver on the best action to take. In addition, they also made two other interesting propositions that will be considered in this work. The first one was to have vehicles communicate and exchange information between them in order to have a better grasp of the surrounding context and optimizes the decision-making process. The other interesting proposition was that due to the increasingly complex driving situations that a vehicle would encounter, a knowledge-based approach seems to be optimal in order to store and manage all the different information.

This paper presents a model based on a knowledge base that detects a vehicular context and uses a UAV and logical rules in order to enhance vehicular perception. The work is then validated using a driving simulator.

The structure of the paper is as follows: Section 2 is dedicated to Related Works. Section 3 describes the ontology developed for this work as well as the proposed communication protocol. Section 4 introduces the simulator and the use cases chosen for testing. The paper is finally concluded with an analysis and a perspective of our future works.

2. Related Works

2.1. Vehicular Perception

Weather is an external variable that cannot be controlled. A driver's visibility can be heavily hindered by the quality of the weather, and this phenomena has been intensely studied in the past, some studies dating back as far as the 1970s [10]. In their 2019 work, Harith et al. [11] identified 45 different works made in 21 countries and covering more than 500,000 accident cases caused by adverse weather, mostly rain and fog conditions. They concluded that a driver's vigilance is key to keeping the road safe.

This concern is shared by Das et al. [12], who agree that rainy weather is one of the most hazardous driving conditions, causing up to 25% of crashes in some areas. They assembled a dataset based on the crash records in the state of Florida, and managed to prove that there is a relationship between road accidents and poor visibility due to bad weather.

The same team used data mining on extensive data sets and aimed to isolate aggravating circumstances that can increase the accident rate when coupled with rainfalls [13]. The patterns that emerge can provide valuable insight for safety professionals, but at the time of the study, a comparison with a clear-weather similar dataset was lacking.

Andrey et al. [14] point out that despite the accident rate increasing by up to 70% in case of rain, it returns to a normal value when the rain stops, despite the lingering effect of wet roads. This would mean that the main reason behind those accidents might actually be the poor visibility conditions rather than the slippery roads.

Visibility is also affected by lighting conditions. Cameras are among the most present sensors in a vehicle, and they perform poorly if the lighting is not adequate. Carlevaris-Bianco et al. [15] built a dataset containing similar images with different illuminations, and underlined the difficulty an intelligent unit could have in processing the same situation if the lighting is different.

Most of the causes of illumination variations are natural and independent of humans, for example night time or weather situations. The Gade et Moeslund [16] study says that lighting variation impacts many parameters of an image (intensity, color balance, etc.), but also proposes the use of thermal camera instead, despite having their own sets of drawbacks, such as the inability to classify the detected objects. Visibility can also be

improved in software via algorithms, as shown by Tarel et al. [17,18]. They focused their works on foggy roads and showed great results for their filtering algorithm.

Considering the numerous works made on the topic, it is then clear that inclement weather and illumination have a negative impact on road safety. For a human driver, this requires an increased vigilance and trying to improve the global visibility (fog lamps, windshield wipers, etc.) [19], but in an Intelligent Transport System (ITS), it is more about a general improvement of the perception.

As with any other robotic body, an autonomous vehicle relies on sensors in order to perceive its surroundings. In their survey work, Yurtsever et al. [20] pointed out the importance of perception in autonomous driving and included types of sensors (camera, lidar, radar, etc.) and their corresponding algorithms (object detection, event detection, semantic segmentation, etc.) in their review. They also classified the sensors in two different categories:

Exteroceptive sensors for perceiving the environment and Proprioceptive for internal vehicle state monitoring tasks. Vehicular sensors are ultimately the same as a human driver using their eyes as sensors to assess the situation on the road. Table A1 offers a comparison of the human eye to the other most common vehicular sensors, based on some parameters including performances in bad-weather and bad-illumination situations. Extensive tests have been made and results were documented in the work of Shoettle [20,21].

In 2018, Van Brummelen et al. [22] made an extensive review of the current state of vehicular perception. They define autonomous vehicle navigation with five main components: Perception, Localization and Mapping, Path Planning, Decision Making, and Vehicle Control, with Perception being 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 amount of different sensors can be considered [22–24]:

- Radars have been used for decades for vehicular applications [25,26]. 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 [27]. It is still heavily present in vehicles but has a small Field Of View (FOV) and shows poor results in near-distance measurement and static object detection. There is also the problem 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 [24]. They allow a quick classification of the obstacle and a potential 3D mapping of the area. Stereoscopia in particular shows very good results in detecting forms, depth, colors and velocity, although it requires substantial computational power [28]. The most advanced models can also be used for long-range precise detection, but they have a more important cost [29]. However, the performance highly depends on the weather and brightness [27], and the required computational 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 1980s [30] but it is only in early 2000 that it has found its way in vehicular application [31,32]. It is a useful tool for 3D mapping and localization, and can be used on a large FOV [27], but it relies heavily on good-weather conditions and is not efficient outside a defined range.
- An infrared camera measures temperature radiations in order to detect moving objects, and it shows great results in both bad weather (rain, snow, fog), and lack of brightness [16,33]. However, they cannot be used for the classification of an image’s object and cannot inform on their distance.

Other types of sensor can be found on vehicles, such as ultrasounds, and some of their performances can be found in Table A1 in Appendix A. However, there is no unique “ideal” sensor that allows a perfect perception on a bad weather.

In their 2015 ADAS [34] review, Bengler et al. [35] said that vehicular perception has evolved from being centered on the vehicle toward its surroundings (exteroceptive to

proprioceptive), and argued that the next natural step will be to fuse data from multiple sensors in order to obtain a better reliability.

2.2. UAV for Vehicular Applications

It would then be interesting to consider the use of a sensor which is external to the vehicle. One such tool would be the Unmanned Aerial Vehicle (UAV), also known as a drone. In their 2017 paper, Menouar et al. [36] initiated the idea of using UAV as supporting items of the Intelligent Transport System (ITS), proposing multiple possible uses. Indeed, their ability to move in a 3D space at high-speed, as well as their size that allows package transportation while remaining smaller than cars, give them important benefit in a world where transportation is mainly 2D-oriented.

Shi et al. [37] made a study focusing on the data throughput in a UAV–Vehicle case and reached a speed of 2 MB/s in their simulations comparing a regular 802.11p car-only communications with a 2.4 GHz Dedicated Short Range Communication (DSRC) 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.

The majority of works around UAVs focus on their use as a network node. There are, however, a few works on UAV–Vehicle communications, such as Hadiwardoyo et al. [38] which tested the impact of land topology on UAV–Vehicle communication and showed great results in long-distance communication, reaching a distance of over 1 km.

There are also cases of UAVs being used for Wireless Sensors Network (WSN), such as in Zhan et al. [39], who proposed an energy-efficient data collection in a UAV-enabled WSN. Being mobile, a UAV's position can be deployed in an optimized way in order to gather data from a specific source.

2.3. Data Fusion

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

There are a multitude of reasons that could lead to performance issues, for example error accumulation over time [40]. 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*” [41]. 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 ontology-approach to this solution, as shown by the review work of Bendadouche et al. [42]. For example, Calder et al. [43] 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 sensor-dedicated ontology [44]. They aimed to make a model abstracted enough to allow that. This consequent work would then allow an easy way of both adding new sensors, and reading the gathered value. This work, as well as [43] was later on merged in the Semantic Sensor Network (SNN) ontology [45], 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*”.

2.4. Secured Communication

The use of a perception agent external to the vehicle (the UAV) means that there will be some need of a communication between both entities, which implies there will need to be some form of security in order to ensure the integrity of data. There are many possible approaches, such as the RSA encryption [46] or the MD5 hashing method [47]. A more detailed review of cryptographic functions can be found in Fattahi's work [48].

Data security usually takes place in the higher levels of communication (higher layers of the OSI model [49]). This work proposes an additional security approach which would also take place on the Physical Layer. Indeed, the Visible Light Communication (VLC) relies on the modulation of light generated through Light Emitting Diode (LED) in order to transmit data. The necessary Line Of Sight (LOS) condition makes it difficult for a malicious agent to manipulate or intercept the data flow in a vehicular environment. However, VLC does have its own set of disadvantages and performs poorly in some environment, which is why it is more interesting to use it as a complementary protocol. VLC in hybrid works has already been considered in the past. In [50], 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 [51], 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. [52], where VLC would act as a complementary protocol that would take over when WiFi reaches maximum capacity. In 2016, Rakia et al. [53] 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. [54] 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.

We can outline a few deductions from this literature review:

- An inclement weather or poor illumination leads to a weakened visibility which is a main factor of car crashes;
- In an environment that is more and more connected, there are some intelligent tools that can be requested to provide additional data to improve perception and visibility, such as UAV;
- Having data from various sensors raises the question of having a mean of fusing them. Knowledge-based approaches, especially ontologies, have shown great potential for multi-sensors management;
- When using an external entity, the communication must be guaranteed to be secured. In that regard, VLC technology offers a strong potential.

The methodology presented in this paper is based on those deductions: In a situation where the vehicle's sensors are hindered, there is the possibility to request data from a UAV stationed nearby (as illustrated in Figure 1). The gathered information are transmitted through a secured channel, and the data are logged and federated in an ontology, where they are then processed.

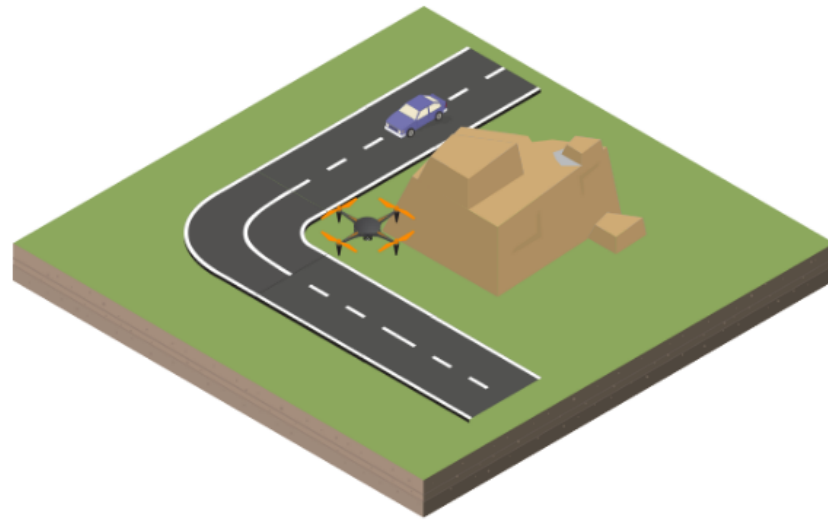


Figure 1. An illustration of a simple scenario. The vehicle’s perception is limited by the environment, and it can then rely on the drone to gather data instead.

3. Proposed Methodology

3.1. Knowledge Base

Using multiple sources of information require a means of federating and managing them. There are multiple ways of doing so, one of them being as a knowledge base [55]. A knowledge base is an object-model way of representing data. As an expert system [56], it contains not only all the different actors and entities of a given situation, but also the abstract concepts, the properties and relationships between the 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 commands, 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 [57] or a knowledge-graph [58]. An ideal representation of a knowledge base is as an ontology. The Stanford 101 Guide defines Ontology [59] as “a formal explicit description of concepts in a domain of discourse, properties of each concept describing various features and the 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. A few of the main components of an ontology are listed in Table 1 [60].

Table 1. Main components of an ontology.

Component	Description
Class	Object describing the concepts in the domain, whether they are abstract ideas or physical actors. Classes can be hierarchized by levels, for example having a Vehicle as a top-class containing Car, Bus and Bike as sub-classes
Individuals	Real instances belonging to Classes and representing the actual elements stored in the knowledge base
Properties	The specific information relative to classes. They can be intrinsic to an object, or extrinsic, representing the interconnections between different concepts and allow to link two individuals together.

In addition to being able to represent all the elements of a situation, it is 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.

An ontology allows the link of different individuals and their properties. This permits the use of those elements for a decision-making process. They can be solely dedicated to the building of a decision-making process, as well as the storage of the previous solutions [61]. For a more complex situation, some reasoners can be supplied by Semantic Web Rule Language (SWRL) rules [62]. 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.

The ontology used for this study focuses on both the vehicle and its surroundings. There are many different interlinked classes, but only a few of them make up the core of the application:

- **Vehicle** represents the different; vehicles detected in the environment. The class encompasses both the Car and the UAV entities
- **Weather** lists all the possible type of weathers that can be encountered. In this case, it covers [Sunny, Fog, Rain, Snow];
- **Environment** describes the context in which the vehicle it evolves, one amongst [NormalEnv, DarkEnv, BadWeatherEnv, UnusualEnv];
- **Sensors** covers the sensors that are used for the perception on a vehicle, as detailed in [22]. The main ones are [cameraMono, cameraStereo, cameraInfra, Lidar, Radar, Sonar]. In addition, there are also environmental sensors used to determine the environment status, [rainSensor, brightnessSensor, fogSensor]. It is illustrated in Figure 2 and detailed Table A2 in Appendix A.

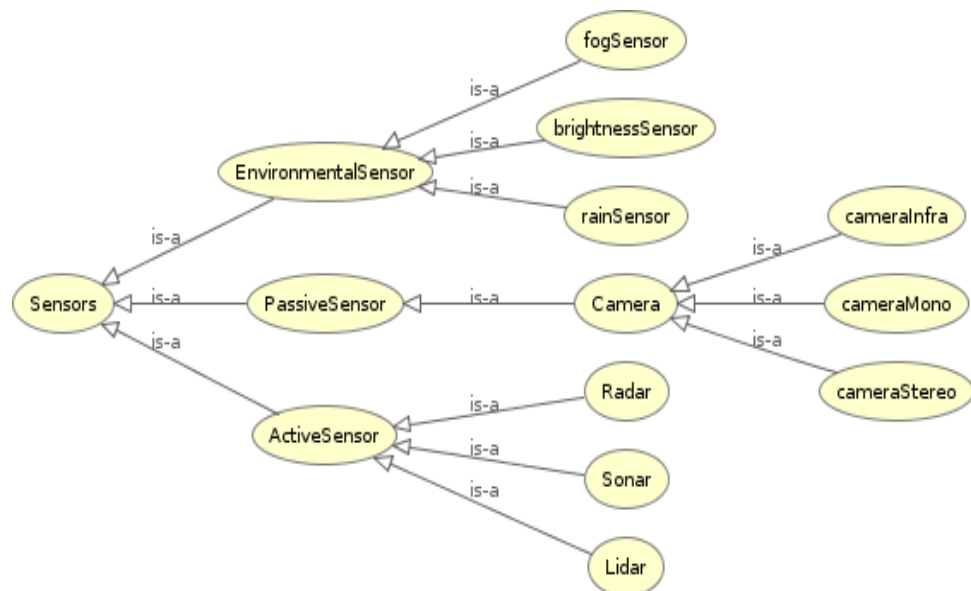


Figure 2. Class and subclasses of sensors.

The use of ontologies for vehicular applications has already been considered in the past. Armand et al. [63] proposed an ontology allowing for a coherent understanding of a driving environment and through the use of the adequate rules, properties and entities. Their works were encouraging for knowledge-based ADAS, despite the meticulousness required for declaring all the rules, and the inference time considered too long [6].

In addition to the population of the knowledge base, a layer of intelligence is added through the use of logical rules. The environmental sensors are permanently logging the gathered value. Upon reaching a certain threshold, the reasoner engine infers a new Environment class, and the necessary Sensors are activated accordingly.

3.2. VLC Communication

The communication protocol management would still need to be a part of the knowledge base. There are indeed three possible choices which depends of the environment.

- **Radiofrequency:** The DSRC (Dedicated Short (RF) [64] is the communication protocol designated for automotive V2X use. It is the standard protocol, but the strength of the signal depends of the land form and its sensitivity to electromagnetic interference.
- **VLC:** The VLC protocol performs poorly in some weather conditions, but it can also improve the lighting in darker areas. It can be an interesting choice.
- **Hybrid RF/VLC:** This protocol allows a better Quality of Service via the redundancy of information. If the context allows it, this should be the preferred choice.

The VLC/RF hybrid approach is suitable in this situation due to the following reasons:

- Intelligent Transport Systems can natively communicate with their surroundings [65]
- 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 paper is to hauseve VLC and RF as a hybrid communication protocol in order to ensure that no information is lost during data transmission between two agents: The “heavy” data are 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 3.

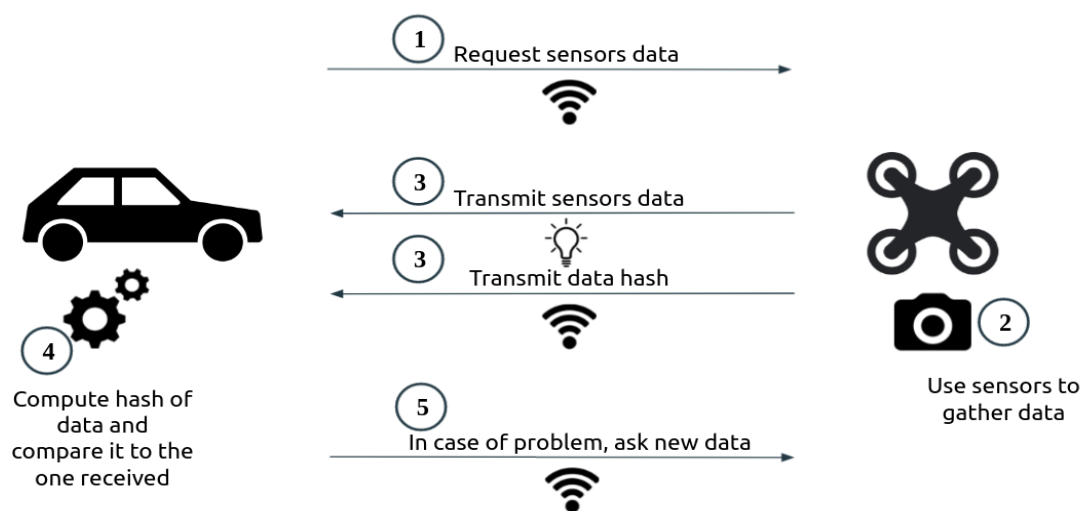


Figure 3. Illustration of the proposed communication protocol.

3.2.1. Hashing Algorithms

“Hashing” refers to the process of passing data 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 paper focuses on the protection against data loss during transmission and speed transmission, and to that end a basic hashing algorithm

such as MD5 is acceptable [66]. Pamula et Ziebinski [67] proposed the real-time hashing of a buffered live video stream by using Field Programmable Gate Arrays (FPGA). That study managed to reach a hashing throughput of more than 400 Mb/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 1 Gb of data in about 65 ms, and quickly ask for a resend if an error is detected.

3.2.2. VLC Transmission Speed

As stated in the previous section, 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 data.

Depending on the type of LED and receivers used, the transmission rate can greatly vary: Table 2 offers an illustration of some of the throughputs reached by other works, as well as the average unloading time of 200 Gb of data. The first study made use of OLED in order to reach a speed of 3 Mb/s, and the same team made an Artificial Neural Network as an equalizer and high-speed receivers in order to improve the speed to 170 Mb/s. Another study made use of FPGA and 64 Quadrature Amplitude Modulation [68] (QAM) to reach a VLC throughput of 5 Gb/s.

Table 2. Transmission speed of some VLC studies.

Study	Speed	Transmission Time (for 200 Gb)
Haigh et al., 2013 [69]	3 Mb/s	18 h
Haigh et al., 2016 [70]	170 Mb/s	19 min
Shi et al., 2019 [71]	5 Gb/s	40 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 functions are only a few hundred bits long, making their fast 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. Simulation and Results

4.1. Simulated Environment

The model was tested and validated in a simulated environment. The interface was based on the Udacity [72] project, a car simulator built with the Unity engine [73]. It allows the building of driving surroundings (roads, obstacles), driving conditions (rain, fog, physics constraints, etc.), and the manual control of the vehicle.

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 to infer the environmental status.

The communication between the simulator and the knowledge base is done through a socket connection. We considered two different software tools to do so: the Java OWL API [60] and the Owlready Python library. We compared both of those approaches in order to choose the optimal one for our study.

In Figures 4 and 5, we compared both engines execution times. We executed both of them in a similar environment where the vehicle encounters some heavy-processing events. As we can see, and depending on the situation, the Python tool offers steady performances with a processing time of around 1 s. 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 6, 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 [74], LIDAR technology supporting this process with depth-computation [75] or radar sensors for the detection of close elements [76]. The multiple type 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 and Appendix A.

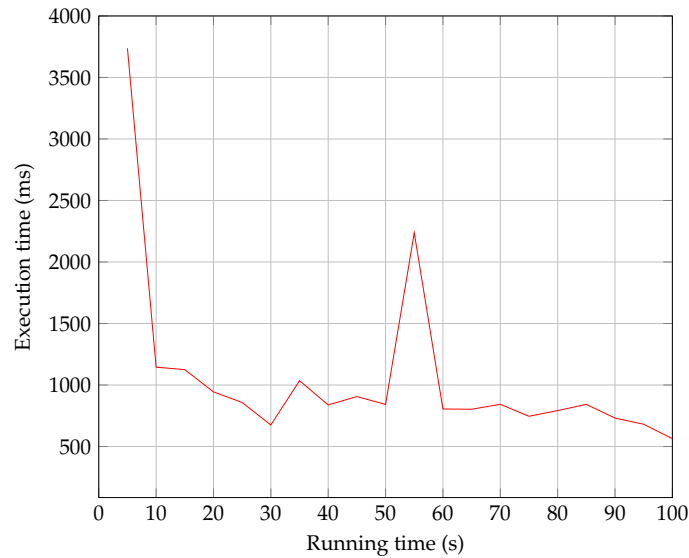


Figure 4. Computed inferring time for the Java tool.

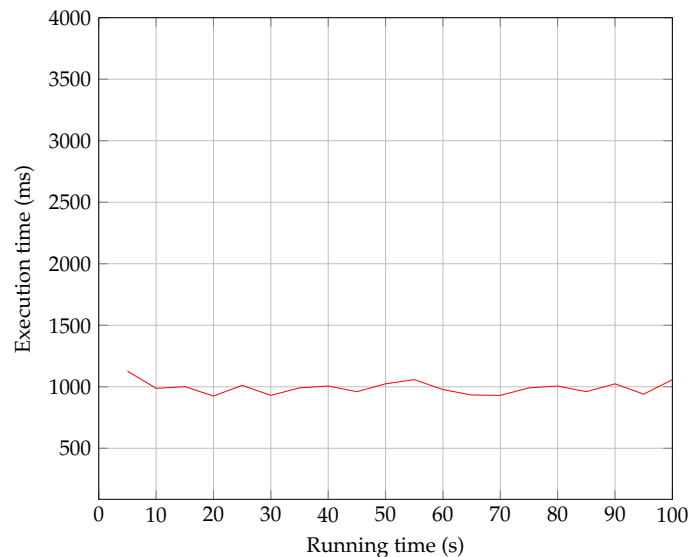


Figure 5. Computed inferring time for the Python tool.

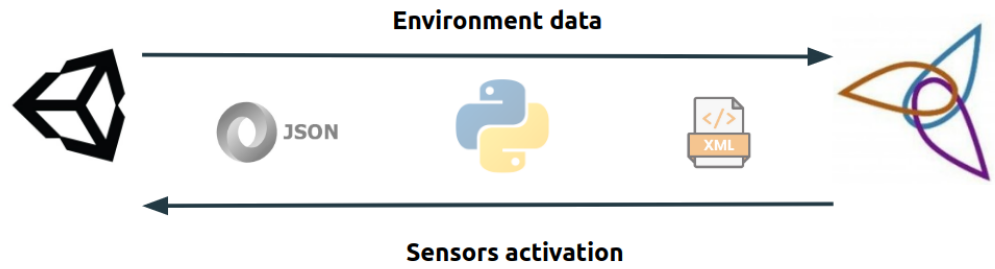


Figure 6. Illustration of the technical implementation.

4.2. Logical Rules

A set of logical rules has been implemented for the context detection, and the Pellet reasoner was used for this work [77]. Figure 7 illustrates an example of a reasoning process for a foggy situation detection and sensors activation. UAV can carry a multitude of sensors, such as cameras, lidar, radar or ultrasound. Due to weight constraints and considering the known energy consumption issues of UAV [78], it is difficult to embed all the sensors on a single drone, and it is more efficient to use combinations of sensors.

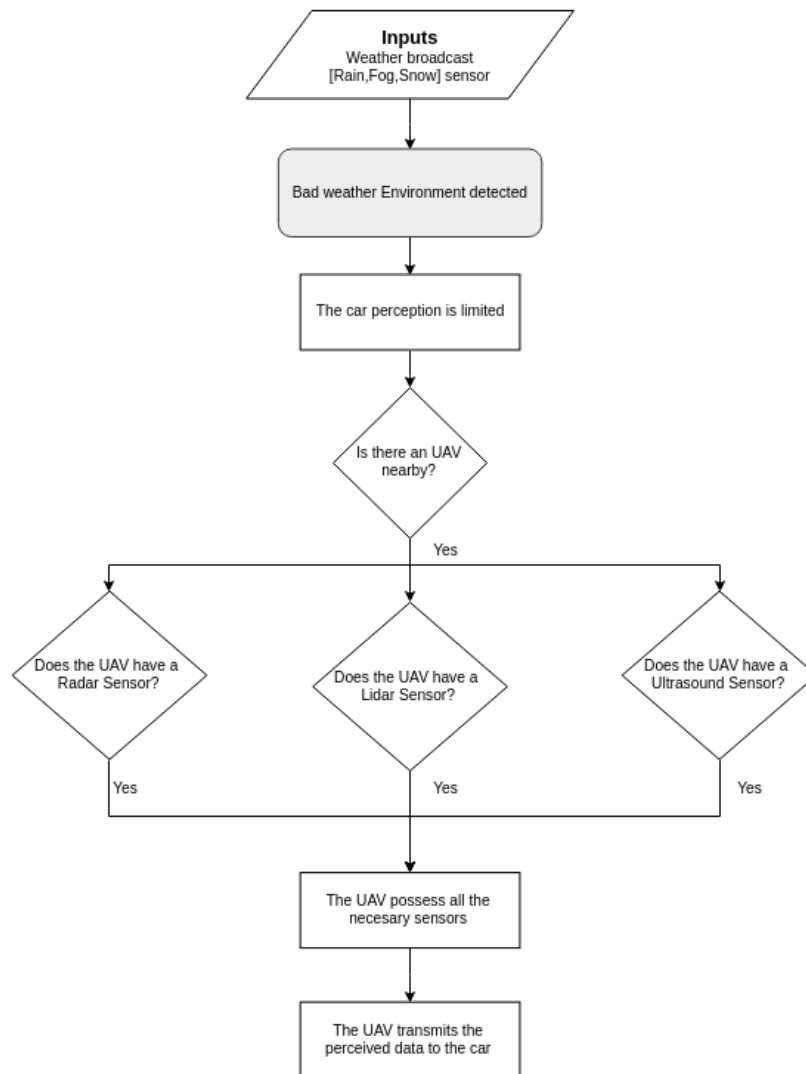


Figure 7. Flowchart of the reasoning process.

When a bad weather situation is detected, the system looks for a UAV nearby and checks if the correct sensors are embedded on it. It is interesting to note that because of the strict and rigorous syntax of the SWRL language, when an “If” condition encounters a “No” result, the whole rule is dropped, meaning that in some cases an important number of rules must be developed for covering a single situation.

Figure 8 gives an in-depth look of the process with the logical rules applied to the foggy situation.

```

1.          Car(?C) & FogSensor(?fogS) & hasSensor(?C,?fogS)
& hasFogValue(?fogS, ?fogV) & swrlb:greaterThan(?fogV, 50)
& Weather(?W)
->Fog(?W)

2.          Fog(?F) & Environment(?Env)
-> BadWeather(?Env)

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

4.          UAV(?u) & Car(?c) & RF(?r) & isCloseToUAV(?c,?u)&
& Action(?a) &
-> RequestData(?a) &
isActiveCommunicationProtocol(?r,true)

```

Figure 8. Set of rules for the management of a foggy environment.

A quick introduction to the SWRL syntax: Car(?C) represents an individual named C that belongs to the Class Car. The operator & (or \wedge) represents an AND logical operation, and the operator -> represents the logical operation THEN. The first rule then reads: IF THERE IS an individual C of class Car AND an individual fogS of class FogSensor AND the individual C has the property hasSensor fogS AND the individual fogS has the property hasFogValue fogV AND the value of fogV is greater than the numerical value 50 AND there exists an individual W of class Weather THEN the weather W is inferred to be of the class Fog.

Here is a breakdown of this set of rules:

1. The environmental sensors embedded on the main vehicle will send back some data. If it is above a certain threshold, the environment is inferred as *Foggy*;
2. A *Foggy* environment is considered as a Bad Weather environment, same as *Rainy* or *Snowy*;
3. The model looks up for a UAV carrying sensors that works well in this environment and is within reach. If it is deemed acceptable, the UAV is considered for potential data transmission;
4. If the proximity condition is fulfilled, a data request is made. Due to VLC performing poorly in heavy fog [79], an RF-communication protocol is chosen.

The general decision process methodology is summarized in Figure 9: by considering the environmental elements (brightness, weather) and the state and efficiency of the sensors in the said environment, the system chooses the appropriate sensors, entities and communication protocols and requests their activation.

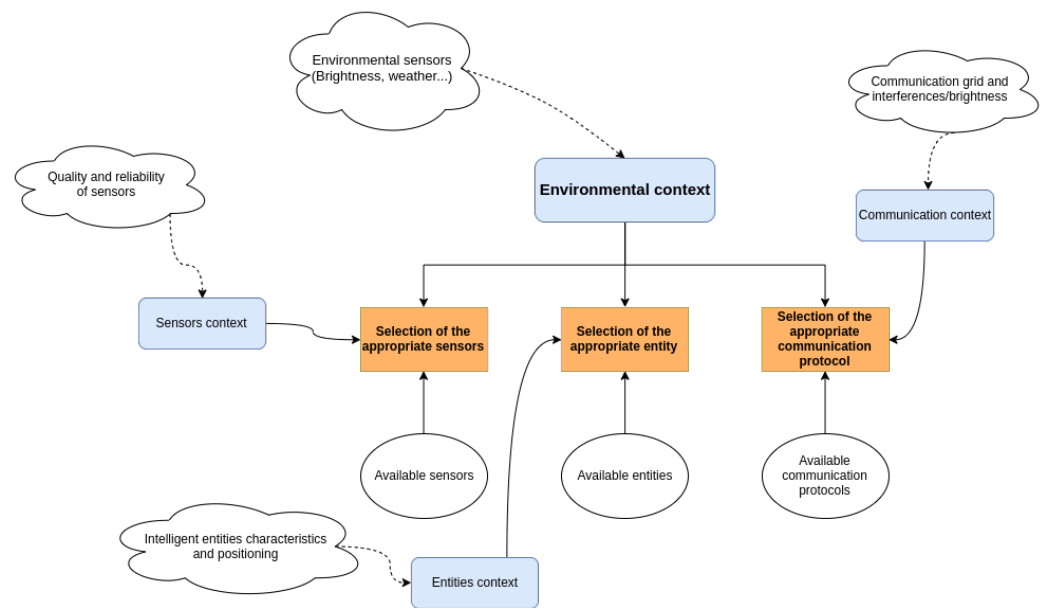


Figure 9. Influence of the environment on the inference process.

4.3. Experimentation Description

The experimentation using the case description was made by having different human agents drive a determined course. The drivers control the vehicle with a keyboard and the simulator, acting as the vehicle’s sensors, logs the environmental driving data into the knowledge base. Those 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 those data.

For example, 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 fixed at 50 for fog-detection and the *Weather* individual will be classified as *Fog*. An illustration of this process is shown in Figure 10. The raw data are sent from the simulator to the ontology through the Python pipeline. Details of the transmitted data can be found in Table A3.

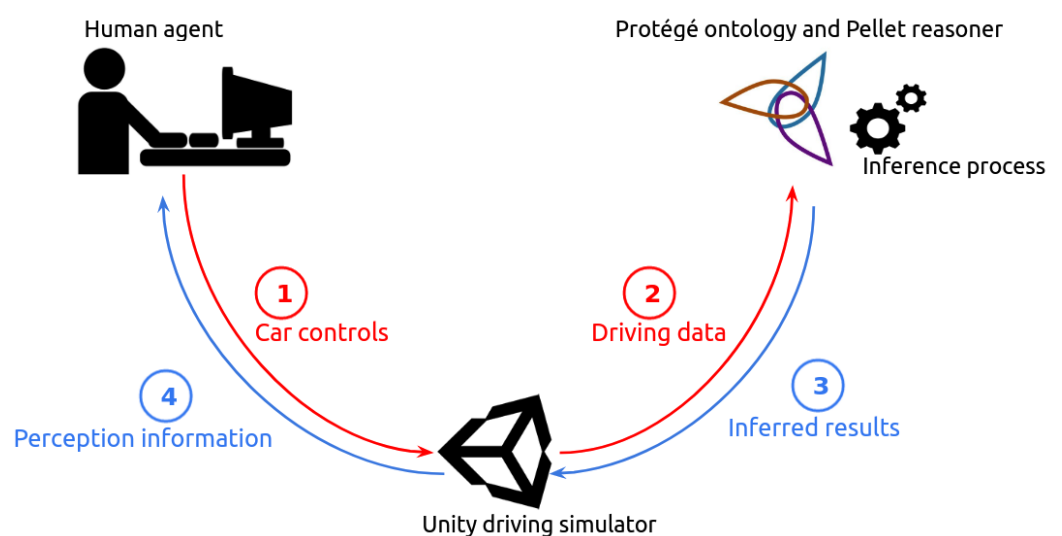


Figure 10. Interactions between the different agents. The human agent controls the car, the driving data generated are stored in the knowledge base, and the eventual important information are displayed to the user in a widget, as shown in Figure 11.

Once logged in the knowledge base, they are processed by the set of logical rules and the tools described in Section 4.1. If the system inference requires a perception enhancement, and if the conditions are verified (i.e., available UAV in reach with all the correct sensors), the additional information will be displayed on screen, as shown in Figure 11: 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 drivers are to encounter the following situations, as illustrated in Figure 12:

- The driver needs to take a turn in an intersection with limited visibility.
- The driver goes through a foggy area.
- The driver needs to go through a certain area where one of the buildings is on fire.

The limited-visibility areas are built so that the controlled vehicle would crash into stationary or moving obstacles if they are 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 11. In this way, the ego vehicle gets knowledge of obstacles in a specific area in advance.

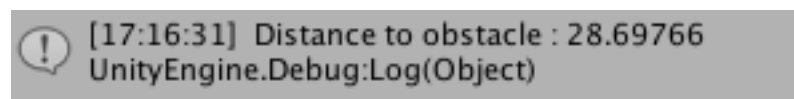


Figure 11. Information of the distance between the vehicle and an obstacle. This information is displayed when the vehicle makes a request to a UAV.

In this experiment, specimen drivers are grouped into two sets: The first group drives the circuit without any warning indications, while the second is guided with an interface warning them on potential obstacles, even with a limited visibility. The different sets are then evaluated on mainly three criteria: average speed, speed variation in a difficult situation, and the number of crashes or bad decisions made.



Figure 12. Illustration of the driving environment in the simulator.

The logical rules associated to the different situations can be found in Figures 13 and 14 (the rules relative to a foggy environment can be found in Figure 8):

```

1. Vehicle(?V) & hasDistanceFromVehicle(?O, NearDistance)
&Environment(?Env) & Object(?O)
-> hasObstructedView(?V, ?O)

2. Vehicle(?V) & UAV(?U) & cameraStereo(?cm) & hasSensor(?U, ?cm) &
RF(?r) &HybridCommunication(?h) &Object(?O) & hasObstructedView(?V, ?O) &
NormalEnvironment(?Env) & isCloseToUAV(?V,?u) & Action(?a)
-> RequestData(?a) & isActiveSensor(?cm, true)
& isActiveCommunicationProtocol(?r, true)
& isActiveCommunicationProtocol(?h,true)

```

Figure 13. Set of rules for the management of an obstructed view.

```

1. Vehicle(?V) & FireHazard(?H) & Environment(?Env)
-> FireHazardEnvironment(?Env)

2. Vehicle(?V) & UAV(?U) & isCloseToUAV(?V, ?U) & cameraStereo(?cm) & Radar(?r)
& cameraInfra(?ci) & hasSensor(?U, ?ci) & hasSensor(?U, ?r) &
hasSensor(?U, ?cm) & FireHazardEnvironment(?Env) & Action(?a)
-> RequestData(?a)

3a. RF(?r) & hasCommunicationProtocol(?U, ?v) & FireHazardEnvironment(?f) &
hasBrightnessValue(?br, ?bv) & hasCommunicationProtocol(?U, ?h) &
brightnessSensor(?br) & swrlb:lessThan(?bv, 70) ~ Hybride(?h)
& VLC(?v) & UAV(?U) & hasCommunicationProtocol(?U, ?r) &
-> isActiveCommunicationProtocol(?h, true)
& isActiveCommunicationProtocol(?v, true)
& isActiveCommunicationProtocol(?r, true)

3b. RF(?r) & hasCommunicationProtocol(?U, ?v) & FireHazardEnvironment(?f) &
hasBrightnessValue(?br, ?bv) & hasCommunicationProtocol(?U, ?h) &
brightnessSensor(?br) &
Hybride(?h) & VLC(?v) & UAV(?U) & hasCommunicationProtocol(?U, ?r) &
FireHazardEnvironment(?ue) & swrlb:greaterThan(?bv, 70)
-> isActiveCommunicationProtocol(?r, true)
& isActiveCommunicationProtocol(?v, false)
& isActiveCommunicationProtocol(?h, false)

```

Figure 14. Set of rules for the management of a Fire Hazard. It is interesting to notice that due to the different possible states of brightness and the VLC efficiency threshold, there are two mutually exclusive rules concerning the choice of the communication protocol.

Figures 15–17 illustrate views from the simulated environment:



Figure 15. The first obstacle with limited visibility.



Figure 16. The second obstacle with a foggy area.



Figure 17. The third obstacle with a fire hazard.

4.4. Results and Discussion

The first set of participants was made up of three different profiles: A normal driver, a careless driver and an overly-cautious driver. They were asked to complete the course with no prior knowledge of the circuit or any form of driving assistance. Their overall results can be found in Table 3.

The second set of participants also consists of the same three different profiles but with driving assistance added, and their results are shown in Table 4.

Some interesting results can be extracted by comparing the two sets. Thanks to the assistance provided by the inferred rules, the second set of subjects actually manage to finish the circuit faster than the controlled subjects. Their trust in the assistance system allows them a better management of their speed and the overall driving quality. There are still some incidents to deplore, especially for the Careless Driver profile, but the high speed they adopt makes it difficult for them to react on time. The assistance still allows for an lesser number of road incidents.

Table 3. Results of the tests with no additional guidance.

	Average Speed	Time to Complete the Circuit	Max Speed	Number of Incidents
Normal driver	27 km/h	67 s	69 km/h	1
Cautious driver	16 km/h	92 s	37 km/h	0
Careless driver	33 km/h	54 s	78 km/h	3

Table 4. Results of the tests with assistance from knowledge base and UAV.

	Average Speed	Time to Complete the Circuit	Max Speed	Number of Incidents
Normal driver	31 km/h	50 s	78 km/h	0
Cautious driver	22 km/h	65 s	46 km/h	0
Careless driver	44 km/h	41 s	120 km/h	2

Overall, the system performs to allow a better and safer driving. However, it still has its own set of drawbacks, for example the execution speed. As stated in Section 2, the inference time takes some time, reaching around 1.5 s in our studies, and seeming to increase proportionally to the knowledge base population size. This is somehow improved by alternating the data gathering process and the inference process in order to optimize the time management, but due to the straightforward and brute approach of a rule-based system, it will ultimately be tied to the hardware power.

The same rule-based approach requires a rigorous approach and consideration of every possible situation that the vehicle can encounter. To the contrary of a Neural Network approach, the complexity of our system is not in its technicality but is more appropriate in considering the multiple situations that can happen in different contexts. Having a greater set of rules and elements is the key to ensuring a better operation of the model.

5. Conclusions

In this paper, a vehicular perception-enhancement knowledge base system is presented. It gathers the data from a vehicle's surroundings in order to determine the environmental context thanks to a set of logical rules. The system relies on the use of drones (UAV) and their embedded sensors for the collection of additional perception data. The system was tested on a driving simulator with a realistic physics engine. Driving data are saved and logged in an ontology, where they are accordingly stored and processed. In situations where the perception is limited, for example, in bad weather or a fire hazard, the model can request additional data from the drones. The drones are equipped with a set of various sensors allowing them to cover multiple situations. In addition to the UAV activation, the knowledge base can also use the environmental context and the available set of sensors to decide which ones are unreliable. Hence, through the use of the drone's sensors, the vehicle can enhance its perception and detect obstacles in a poor-visibility environment. The model was tested by analyzing the performances of two sets of drivers in a simulation experiment. The experiment consists of having the subjects follow a specific circuit where they will encounter same situations with limited visibility. The first set of drivers received no additional guidance, while the second was assisted by the knowledge base and UAV.

The latter performed better than the former; the participants were able to finish the circuit on average 20 s faster and with 50% less incident.

The main contributions of this paper revolves around the management of multiple types of sensors. Indeed, an optimal perception is obtained through the use of a variety of sensors, each having its own characteristics. There are multiple studies on the use of UAV for communication purposes, but only recently have they been considered for data fusion of sensors. Furthermore, a knowledge base allows an effective storing and management of sensors, whether they are directly embedded on the vehicle or located on a different unit, such as a UAV.

The work presented in this paper is a first step in the UAV–Vehicular perception-enhancement process. Perception is a key factor in autonomous vehicle and there are multiple recent works focusing on improving it [80–82]. Due to the advantageous positioning of UAV, we believe that there can be a real interest in using drones for vehicular applications. The use of a knowledge base for the federation of data from multiple sources is also a promising concept, and can be generalized to other major aspects of vehicular applications, such as V2X or communication protocol management.

Author Contributions: Investigation, A.K.; Methodology, A.K.; Software, A.K.; Supervision, M.D.H., H.G. and A.R.-C.; Validation, A.K.; Visualization, A.K.; Writing—original draft, A.K.; Writing—review & editing, A.K. and M.D.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Comparison of the human eye performances to other vehicular sensors. Green means the sensor works ideally for the given situation, yellow that it is of acceptable range, and red that it works poorly for a specific role.

Parameter	Human Eye	Monoscopic Camera	Stereoscopic Camera	Thermal Camera
Object detection	Very Good	Good	Very Good	Good for shape detection
Object recognition	Very Good	Good	Good	Poor
Range detection	Up to 300 m	Poor	Good	Poor
Poor weather performance	Poor in snow, fog and heavy rain	Poor in snow, fog and rain	Poor in snow, fog and rain	Good in snow, fog and rain
Poor illumination performance	Poor	Poor	Poor	Good
Parameter	Human Eye	Lidar	Radar	Ultrasound
Object detection	Very Good	Very Good	Very Good for distance measurement	Very Good
Object recognition	Very Good	Good	Poor	Poor
Range detection	Up to 300 m	Up to 200 m	Up to 200 m	Very Good
Poor weather performance	Poor in snow, fog and heavy rain	Poor in snow, fog and rain	Good	Good
Poor illumination performance	Poor	Good	Indepent of illumination	Independent of illumination

Table A2. More details on the Sensors class.

Class	Description
Active Sensors	Lidar Radar Ultrasound
Passive Sensors	Monoscopic Camera Stereoscopic Camera Thermal Camera
Environmental Sensors	Rain Sensor Fog Sensor Brightness Sensor

Table A3. 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.

Variable	Ontology Class Values and/or Linking Property	Associated Simulator Value	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	
Environmental Status	[Normal,Dark,BadWeather,Hazardous]	-	Inferred from other elements
Hazard	[FireHazard]	(int,int) X & Y Position of the hazard	Not declared if there is no 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

References

1. New Jersey Bill A2757. Senate and General Assembly of the State of New Jersey. 2012. Available online: <https://www.njleg.state.nj.us/bills/BillView.asp?BillNumber=A2757> (accessed on 8 November 2021)
2. Oklahoma Bill HB3007. Oklahoma House of Congress. 2012. Available online: <http://www.oklegislature.gov/BillInfo.aspx?Bill=hb3007&Session=1800> (accessed on 8 november 2021)
3. Su, K.; Li, J.; Fu, H. Smart city and the applications. In Proceedings of the 2011 International Conference on Electronics, Communications and Control (ICECC), Ningbo, China, 9–11 September 2011; pp. 1028–1031.
4. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)] [[PubMed](#)]
5. Minkoff, A.S. A Markov decision model and decomposition heuristic for dynamic vehicle dispatching. *Oper. Res.* **1993**, *41*, 77–90. [[CrossRef](#)]
6. Khezaz, A.; Hina, M.D.; Guan, H.; Ramdane-Cherif, A. Driving Context Detection and Validation using Knowledge-based Reasoning. In Proceedings of the 12th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, Budapest, Hungary, 2–4 November 2020; Volume 2.
7. Fuchs, S.; Rass, S.; Kyamakya, K. Integration of Ontological Scene Representation and Logic-Based Reasoning for Context-Aware Driver Assistance Systems. *Electron. Commun. EASST* **2008**, *11*, 1–12
8. Endsley, M.R. Toward a theory of situation awareness in dynamic systems. In *Situational Awareness*; Routledge: London, UK, 2017; pp. 9–42.
9. Baumann, M.R.; Petzoldt, T.; Krems, J.F. Situation Awareness beim Autofahren als Verstehensprozess. In *MMI Interaktiv-Aufmerksamkeit und Situationawareness beim Autofahren*; Gesellschaft für Informatik e.V.: Bonn, Germany, 2006; Volume 1.
10. Campbell, M. The wet-pavement accident problem: Breaking through. *Traffic Q.* **1971**, *25*, 209–214.
11. Harith, S.H.; Mahmud, N.; Doulatbadi, M. Environmental Factor and Road Accident: A Review Paper. In Proceedings of the International Conference on Industrial Engineering and Operations Management, Bangkok, Thailand, 5–7 March 2019; p. 10.
12. Das, S.; Brimley, B.K.; Lindheimer, T.E.; Zupancich, M. Association of reduced visibility with crash outcomes. *IATSS Res.* **2018**, *42*, 143–151. [[CrossRef](#)]
13. Das, S.; Dutta, A.; Sun, X. Patterns of rainy weather crashes: Applying rules mining. *J. Transp. Saf. Secur.* **2020**, *12*, 1083–1105. [[CrossRef](#)]
14. Andrey, J.; Yagar, S. A temporal analysis of rain-related crash risk. *Accid. Anal. Prev.* **1993**, *25*, 465–472. [[CrossRef](#)]
15. Carlevaris-Bianco, N.; Eustice, R.M. Learning visual feature descriptors for dynamic lighting conditions. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 2769–2776.
16. Gade, R.; Moeslund, T.B. Thermal cameras and applications: A survey. *Mach. Vis. Appl.* **2014**, *25*, 245–262. [[CrossRef](#)]
17. Tarel, J.P.; Hautiere, N. Fast visibility restoration from a single color or gray level image. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009; pp. 2201–2208. [[CrossRef](#)]
18. Tarel, J.P.; Hautiere, N.; Caraffa, L.; Cord, A.; Halmaoui, H.; Gruyer, D. Vision Enhancement in Homogeneous and Heterogeneous Fog. *IEEE Intell. Transport. Syst. Mag.* **2012**, *4*, 6–20. [[CrossRef](#)]
19. World Health Organization. *Save LIVES: A Road Safety Technical Package*; World Health Organization: Geneva, Switzerland, 2017;
20. Yurtsever, E.; Lambert, J.; Carballo, A.; Takeda, K. A Survey of Autonomous Driving: Common Practices and Emerging Technologies. *IEEE Access* **2020**, *8*, 58443–58469. [[CrossRef](#)]
21. Schoettle, B. *Sensor Fusion: A Comparison of Sensing Capabilities of Human Drivers and Highly Automated Vehicles*; University of Michigan: Ann Arbor, MI, USA, 2017.
22. Van Brummelen, J.; O'Brien, M.; Gruyer, D.; Najjaran, H. Autonomous vehicle perception: The technology of today and tomorrow. *Transp. Res. Part C Emerg. Technol.* **2018**, *89*, 384–406. [[CrossRef](#)]
23. Campbell, M.; Egerstedt, M.; How, J.P.; Murray, R.M. Autonomous driving in urban environments: Approaches, lessons and challenges. *Philos. Trans. R. Soc. A* **2010**, *368*, 4649–4672. [[CrossRef](#)]
24. Vanholme, B.; Gruyer, D.; Lusetti, B.; Glaser, S.; Mammari, S. Highly Automated Driving on Highways Based on Legal Safety. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 333–347. [[CrossRef](#)]
25. Woll, J. Monopulse Doppler radar for vehicle applications. In Proceedings of the Intelligent Vehicles '95. Symposium, Detroit, MI, USA, 25–26 September 1995; pp. 42–47. [[CrossRef](#)]
26. Mayhan, R.J.; Bishel, R.A. A two-frequency radar for vehicle automatic lateral control. *IEEE Trans. Veh. Technol.* **1982**, *31*, 32–39. [[CrossRef](#)]
27. Rasshofer, R.H.; Gresser, K. Automotive radar and lidar systems for next generation driver assistance functions. *Adv. Radio Sci.* **2005**, *3*, 205–209. [[CrossRef](#)]
28. Sivaraman, S.; Trivedi, M.M. Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 1773–1795. [[CrossRef](#)]
29. Sahin, F.E. Long-Range, High-Resolution Camera Optical Design for Assisted and Autonomous Driving. *Photonics* **2019**, *6*, 73. [[CrossRef](#)]

30. Smith, M. *Light Detection and Ranging (LIDAR), Volume 2. A Bibliography with Abstracts*; National Technical Information Service: Springfield, VA, USA, 1978.
31. Li, B.; Zhang, T.; Xia, T. Vehicle detection from 3D lidar using fully convolutional network. *arXiv* **2016**, arXiv:1608.07916.
32. Mahlich, M.; Schweiger, R.; Ritter, W.; Dietmayer, K. Sensorfusion Using Spatio-Temporal Aligned Video and Lidar for Improved Vehicle Detection. In Proceedings of the 2006 IEEE Intelligent Vehicles Symposium, Meguro-Ku, Japan, 13–15 June 2006; pp. 424–429. [[CrossRef](#)]
33. Iwasaki, Y. A method of robust moving vehicle detection for bad weather using an infrared thermography camera. In Proceedings of the 2008 International Conference on Wavelet Analysis and Pattern Recognition, Hong Kong, China, 30–31 August 2008; Volume 1, pp. 86–90.
34. Hina, M.D.; Guan, H.; Soukane, A.; Ramdane-Cherif, A. CASA: An Alternative Smartphone-Based ADAS. *Int. J. Inf. Technol. Decis. Mak.* **2021**, *20*, 1–41. [[CrossRef](#)]
35. Bengler, K.; Dietmayer, K.; Farber, B.; Maurer, M.; Stiller, C.; Winner, H. Three Decades of Driver Assistance Systems: Review and Future Perspectives. *IEEE Intell. Transp. Syst. Mag.* **2014**, *6*, 6–22. [[CrossRef](#)]
36. Menouar, H.; Guvenc, I.; Akkaya, K.; Uluagac, A.S.; Kadri, A.; Tuncer, A. UAV-Enabled Intelligent Transportation Systems for the Smart City: Applications and Challenges. *IEEE Commun. Mag.* **2017**, *55*, 22–28. [[CrossRef](#)]
37. Shi, W.; Zhou, H.; Li, J.; Xu, W.; Zhang, N.; Shen, X. Drone assisted vehicular networks: Architecture, challenges and opportunities. *IEEE Netw.* **2018**, *32*, 130–137. [[CrossRef](#)]
38. Hadiwardoyo, S.A.; Hernández-Orallo, E.; Calafate, C.T.; Cano, J.C.; Manzoni, P. Experimental characterization of UAV-to-car communications. *Comput. Netw.* **2018**, *136*, 105–118. [[CrossRef](#)]
39. Zhan, C.; Zeng, Y.; Zhang, R. Energy-Efficient Data Collection in UAV Enabled Wireless Sensor Network. *arXiv* **2017**, arXiv:1708.00221.
40. Božek, P.; Bezák, P.; Nikitin, Y.; Fedorko, G.; Fabian, M. Increasing the production system productivity using inertial navigation. *Manuf. Technol.* **2015**, *15*, 274–278. [[CrossRef](#)]
41. Fayyad, J.; Jaradat, M.A.; Gruyer, D.; Najjaran, H. Deep learning sensor fusion for autonomous vehicle perception and localization: A review. *Sensors* **2020**, *20*, 4220. [[CrossRef](#)]
42. 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. In Proceedings of the INFORSID 2012, Montpellier, France, 29–31 May 2012; pp. 89–104.
43. Calder, M.; Morris, R.A.; Peri, F. Machine reasoning about anomalous sensor data. *Ecol. Inform.* **2010**, *5*, 9–18. [[CrossRef](#)]
44. Compton, M.; Neuhaus, H.; Taylor, K.; Tran, K.N. Reasoning about sensors and compositions. *SSN* **2009**, *522*, 33–48.
45. Henson, L.; Barnaghi, T.; Corcho, C.; Castro, G.; Herzog, G.; Janowicz, K. Semantic Sensor Network xg Final Report. 2011. Available online: <https://www.w3.org/2005/Incubator/ssn/XGR-ssn-20110628/> (accessed on 8 November 2021)
46. Rivest, R.L.; Shamir, A.; Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* **1978**, *21*, 120–126. [[CrossRef](#)]
47. Wang, X.; Feng, D.; Lai, X.; Yu, H. Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD. *IACR Cryptol. ePrint Arch.* **2004**, *2004*, 199.
48. Fattahi, J. Analyse des Protocoles Cryptographiques par les Fonctions témoins. Ph.D. Thesis, Université Laval, Quebec City, QC, Canada, 2016.
49. Zimmermann, H. OSI reference model-the ISO model of architecture for open systems interconnection. *IEEE Trans. Commun.* **1980**, *28*, 425–432. [[CrossRef](#)]
50. Nauryzbayev, G.; Abdallah, M.; Al-Dhahir, N. Outage Analysis of Cognitive Electric Vehicular Networks over Mixed RF/VLC Channels. *arXiv* **2020**, arXiv: 2004.11143.
51. Marabissi, D.; Mucchi, L.; Caputo, S.; Nizzi, F.; Pecorella, T.; Fantacci, R.; Nawaz, T.; Seminara, M.; Catani, J. Experimental Measurements of a Joint 5G-VLC Communication for Future Vehicular Networks. *J. Sens. Actuator Netw.* **2020**, *9*, 32. [[CrossRef](#)]
52. Rahaim, M.B.; Vegni, A.M.; Little, T.D.C. A hybrid Radio Frequency and broadcast Visible Light Communication system. In Proceedings of the 2011 IEEE GLOBECOM Workshops (GC Wkshps), Houston, TX, USA, 5–9 December 2011; pp. 792–796. [[CrossRef](#)]
53. Rakia, T.; Yang, H.C.; Gebali, F.; Alouini, M.S. Optimal Design of Dual-Hop VLC/RF Communication System with Energy Harvesting. *IEEE Commun. Lett.* **2016**, *20*, 1979–1982. [[CrossRef](#)]
54. Pan, G.; Ye, J.; Ding, Z. Secure Hybrid VLC-RF Systems with Light Energy Harvesting. *IEEE Trans. Commun.* **2017**, *65*, 4348–4359. [[CrossRef](#)]
55. Trochim, W.M.; Donnelly, J.P. Research Methods Knowledge Base. 2001. Available online: <https://conjointly.com/kb/> (accessed on 8 November 2021)
56. Balci, O.; Smith, E.P. *Validation of Expert System Performance*; Technical Report; Department of Computer Science, Virginia Polytechnic Institute & State University: Blacksburg, VA, USA, 1986.
57. Jaffar, J.; Maher, M.J. Constraint logic programming: A survey. *J. Log. Program.* **1994**, *19–20*, 503–581. [[CrossRef](#)]
58. Paulheim, H. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semant. Web* **2016**, *8*, 489–508. [[CrossRef](#)]
59. Noy, N.F.; McGuinness, D.L. *Ontology Development 101: A Guide to Creating Your First Ontology*; Knowledge Systems Laboratory: Stanford, CA, USA, 2001.

60. Horridge, M.; Bechhofer, S. The owl api: A java api for owl ontologies. *Semant. Web* **2011**, *2*, 11–21. [[CrossRef](#)]
61. Kornysheva, E.; Deneckère, R. Decision-making ontology for information system engineering. In *International Conference on Conceptual Modeling*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 104–117.
62. O'Connor, M.J.; Shankar, R.D.; Musen, M.A.; Das, A.K.; Nyulas, C. The SWRLAPI: A Development Environment for Working with SWRL Rules. In *Proceedings of the Fifth OWLED Workshop on OWL: Experiences and Directions, Collocated with the 7th International Semantic Web Conference (ISWC-2008)*, Karlsruhe, Germany, 26–27 October 2008.
63. Armand, A.; Filliat, D.; Ibanez-Guzman, J. Ontology-based context awareness for driving assistance systems. In *Proceedings of the 2014 IEEE Intelligent Vehicles Symposium Proceedings*, Dearborn, MI, USA, 8–11 June 2014; pp. 227–233. [[CrossRef](#)]
64. Kenney, J.B. Dedicated short-range communications (DSRC) standards in the United States. *Proc. IEEE* **2011**, *99*, 1162–1182. [[CrossRef](#)]
65. Tonguz, O.; Wisitpongphan, N.; Bai, F.; Mudalige, P.; Sadekar, V. Broadcasting in VANET. In *Proceedings of the 2007 Mobile Networking for Vehicular Environments*, Anchorage, AK, USA, 11 May 2007; pp. 7–12.
66. Rivest, R.; Dusse, S. *The MD5 Message-Digest Algorithm*; Internet Activities Board, Internet Privacy Task Force: Fremont, CA, USA, 1992. [[CrossRef](#)]
67. Pamuła, D.; Zi, A. Securing video stream captured in real time. *Przełąd Elektrotechniczny* **2010**, *86*, 167–169.
68. Webb, W.T.; Hanzo, L. *Modern Quadrature Amplitude Modulation: Principles and Applications for Fixed and Wireless Channels: One*; IEEE Press-John Wiley: Hoboken, NJ, USA, 1994.
69. Haigh, P.A.; Ghassemlooy, Z.; Rajbhandari, S.; Papakonstantinou, I. Visible light communications using organic light emitting diodes. *IEEE Commun. Mag.* **2013**, *51*, 148–154. [[CrossRef](#)]
70. 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. *J. Light. Technol.* **2014**, *32*, 1807–1813. [[CrossRef](#)]
71. Shi, M.; Wang, C.; Li, G.; Liu, Y.; Wang, K.; Chi, N. A 5Gb/s 2×2 MIMO Real-time Visible Light Communication System based on silicon substrate LEDs. In *Proceedings of the 2019 Global LIFI Congress (GLC) Paris, France, 12–13 June 2019*; p. 5.
72. Udacity. Udacity Self-Driving Car Project. 2017. Available online: <https://github.com/udacity/self-driving-car-sim> (accessed on 8 november 2021).
73. Haas, J.K. A History of the Unity Game Engine. Ph.D. Thesis, Worcester Polytechnic Institute Worcester, MA, USA, 2014.
74. Heng, L.; Choi, B.; Cui, Z.; Geppert, M.; Hu, S.; Kuan, B.; Liu, P.; Nguyen, R.; Yeo, Y.C.; Geiger, A.; et al. Project autovision: Localization and 3d scene perception for an autonomous vehicle with a multi-camera system. In *Proceedings of the 2019 International Conference on Robotics and Automation (ICRA)*, Montreal, QC, Canada, 20–24 May 2019; pp. 4695–4702.
75. Gao, H.; Cheng, B.; Wang, J.; Li, K.; Zhao, J.; Li, D. Object classification using CNN-based fusion of vision and LIDAR in autonomous vehicle environment. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4224–4231. [[CrossRef](#)]
76. Steinbaeck, J.; Steger, C.; Holweg, G.; Druml, N. Next generation radar sensors in automotive sensor fusion systems. In *Proceedings of the 2017 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, Bonn, Germany, 10–12 October 2017; pp. 1–6.
77. Sirin, E.; Parsia, B.; Grau, B.C.; Kalyanpur, A.; Katz, Y. Pellet: A practical owl-dl reasoner. *J. Web Semant.* **2007**, *5*, 51–53. [[CrossRef](#)]
78. Aleksandrov, D.; Penkov, I. Energy consumption of mini UAV helicopters with different number of rotors. In *Proceedings of the 11th International Symposium “Topical Problems in the Field of Electrical and Power Engineering”*, Pärnu, Estonia, 16–21 January 2012; pp. 259–262.
79. Kim, Y.H.; Cahyadi, W.A.; Chung, Y.H. Experimental demonstration of VLC-based vehicle-to-vehicle communications under fog conditions. *IEEE Photonics J.* **2015**, *7*, 1–9. [[CrossRef](#)]
80. Chen, Q.; Xie, Y.; Guo, S.; Bai, J.; Shu, Q. Sensing system of environmental perception technologies for driverless vehicle: A review of state of the art and challenges. *Sens. Actuators A Phys.* **2021**, *319*, 112566. [[CrossRef](#)]
81. Mohammed, A.S.; Amamou, A.; Ayevide, F.K.; Kelouwani, S.; Agbossou, K.; Zioui, N. The perception system of intelligent ground vehicles in all weather conditions: A systematic literature review. *Sensors* **2020**, *20*, 6532. [[CrossRef](#)] [[PubMed](#)]
82. Gruyer, D.; Magnier, V.; Hamdi, K.; Claussmann, L.; Orfila, O.; Rakotonirainy, A. Perception, information processing and modeling: Critical stages for autonomous driving applications. *Annu. Rev. Control* **2017**, *44*, 323–341. [[CrossRef](#)]