Silesian University of Technology
Gliwice

Faculty of Automatic Control, Electronics and Computer Science

Politechnika Śląska

PhD dissertation

mgr inż. Paweł Markiewicz

Research and Development of Occupancy Grid Fusion for Automotive Applications

Supervisor:

dr hab. inż. Roman Starosolski

Industrial Supervisor:

dr hab. inż. Paweł Skruch

Gliwice 24/08/2022

Politechnika Śląska

Wydział Automatyki, Elektroniki i Informatyki

Rozprawa doktorska

mgr inż. Paweł Markiewicz

Badanie i Rozwój Metod Oceny Siatek Zajętości Otoczenia do Zastosowania w Systemach Percepcji Samochodowej

Promotor:

dr hab. inż. Roman Starosolski

Opiekun Przemysłowy:

dr hab. inż. Paweł Skruch

Gliwice 24/08/2022

Abstract

This study focuses on researching machine perception methods for automated driving and advanced driver assistance systems. The developed solution is based on occupancy grid map generation using sensor data fusion. The research goes through the state of the art data fusion and inverse sensor methods. In the research, a novel process for performance assessment of grid-based perception has been presented. The process is based on the commercial automotive virtual validation tooling, that has been extended with means for the generation of sensors and reference data. The process uses generated sensor data to execute prototypes of algorithms as well as the reference data to accomplish assessment of their performance. This work also covers new architectural variants by combining the existing methods. The experimental validation has also been carried out in real-world experiments, utilizing data from real sensors installed on the test vehicle. Real-world data has been reprocessed in Hardware in The Loop (HIL) system built for this research purposes. For the selected methods in the assessment, the execution time has been measured on the prototyping platform.

Streszczenie

Praca skupia się na badaniu rozwiązań percepcji maszynowej do wykorzystania w pojazdach autonomicznych oraz samochodowych systemach bezpieczeństwa. Opracowywane i badane rozwiązania opierają się o metody generacji siatek zajętości przez fuzję danych pochodzących z wielu czujników (takich jak radary oraz lidary). W pracy wykonany został przegląd istniejących metod fuzji i szczegółowo opisany sposób modelowania odwrotnego sensorów. Zaproponowany został nowy proces ewaluacji istniejących metod. Proces ewaluacji zaproponowany w ramach pracy ma na celu pozyskanie wskaźników jakości działania wybranych metod fuzji danych. Proces ten oparty został o komercyjne narzędzie do symulacji scenariuszy drogowych, które zostało w ramach prac rozbudowane o narzędzia do generacji symulowanych danych sensorycznych oraz danych kontrolnych. Zaprezentowane zostały także nowe architektury zakładające łączenie znanych metod fuzji. W ramach pracy wykonana została także analiza eksperymentalna w oparciu o rzeczywiste dane z czujników umieszczonych na samochodzie. Dane pozyskane z eksperymentów w rzeczywistym środowisku przetwarzane były na stanowisku Hardware in The Loop (HIL) zbudowanym na potrzeby projektu. Dla wybranych metod została przeprowadzona ocena czasu wykonania algorytmów na platformie prototypowej.

Contents

1. Introduction

The level of complexity of automotive systems has extended beyond the technology just enabling a simple task of transportation of people and goods. Modern cars are equipped with multitude of technologies changing them into mobile data centers for safety and infotainment. Safety and infotainment branches of automotive products rely heavily on computational power and external communication stretching the automotive system to the outside of the vehicle to other traffic participants and the infrastructure [41]. The crucial component of any of those systems relies on means of sensors and machine perception to digitize the environment. The digitized representation of the environment (be it in the cabin or on the outside of the vehicle) finds application in increasing the safety, comfort and efficiency of transportation.

Advanced Driver Assistance Systems (ADAS) are built around redundancy on the sensor level and address different physical characteristics of those sensors. The review of those sensors is covered in further part of this thesis. This architectural diversification propagates to perception algorithms used to process the raw sensor data [16]. The reasons for redundancy, i.e., utilization of independent sensor technologies (like lidar or radar) comes not only from perception performance needs but also from diversification of failure modes. Similar failure mode diversification applies to the algorithms used for processing the sensor data to create environmental models. An example of such redundant processing is presented in Figure 2.8. The dominating algorithms on the market focus on object tracking and object level fusion - in a bounding box form of representation [94]. The alternative form of this representation is the occupancy grid - rasterized representation of the surroundings.

The scope of this work focuses on assessing and proposing new methods for perception and fusion methods used for occupancy grid based environment perception for automotive safety applications. The main goal of the research is to verify the capability of doing comparative assessment and conducting prototyping implementation using the means of virtual validation. The thesis of this dissertation is that it is possible to prototype, develop and comparatively assess the state-of-the-art occupancy grid algorithms using virtual validation methods. The proposed new methods of assessment and combination of new fusion methods is realized in the scope of the research.

Chapter 2. Background and Methods presents a review and walk-through underlying technologies for occupancy grid creation as well as functions that may find potential applications in this type of perception. Chapter discusses what type of feature functions are available and mandatory on the market and walks through types and technology used in the sensors. A review of the state of the art occupancy grid architecture models is discussed along with Inverse Sensor Models used for research. An overview of virtual validation capabilities is presented along with the real world testing capability requirements. This chapter also presents original contribution of engineering vehicle instrumentation performed for this research. An overview of methods to be used for performance assessment is presented in conjunction with verification setup and test case scenarios. Chapter 3. Architectures and Novel Process Combinations presents the main original contribution to the field, composed of new proposed method for Cell-Wise Accuracy estimation based on the Virtual Validation tooling, the framework architecture, underlying fusion methods and their combinations. This chapter covers the new proposed form of simplification of the inverse sensor modeling to 1D for optimization purposes, this method has been patented as a result of the research work under the patent number:FR3097972B1 [52]. Moreover an application of the real world test to perform parking space assessment is presented. The executed virtual and real world tests contribute to the original work performed for this research. Chapter 4 Experimental Results addresses the results obtained from virtual validation and real world tests. Chapter 5 Summary and Conclusions wraps up the findings and conclusions resulting from the performed research. Final chapters gather bibliography, list of abbreviations, symbols, figures and tables.

1.1. Original Contribution - Recap

This subsection summarizes the parts of this work that constitute the original engineering and scientific contribution performed in the scope of this work. Subsection 2.2.2 consists of a Synthetic Comparison performed based on existing sensors based on the implementation experience in the research. In Section 2.5 an overview of the vehicle hardware and software instrumentation of development vehicle is presented. The discussed vehicle instrumentation constitutes an original development work performed for the research purposes. Subsection 2.6 presents the HIL setup built both as hardware and software test harness for the research purposes. Subsection 2.8.3 overviews virtual validation scenarios implemented for the research purposes in the commercial simulation framework. Subsection 2.8.4 presents the real testing executed for the research purposes in real world environment to allow data collection for offline validation. In Subsection 3.1 a novel process for cell wise accuracy estimation is presented, this method was followed by original implementation of the engineering tool used to perform the evaluation in the newly implemented framework presented in Subsection 3.1.1. Subsection 3.1.2 presents sensor models that were custom implemented on top of the existing software to generate sufficient realism of the synthetic data used in the research. The original contribution for optimization of the Inverse Sensor Model (ISM) is presented in Section 3.2. In Section 3.3 a novel proposal of architecture optimization has been proposed. Next in Section 3.4 original combinations of fusion algorithms are presented. Subsection 3.4 contains proposal of newly implemented method for reference data generation that was implemented and used for the purpose of this research. In Section 3.5 the real world testing performed to assign free parking spaces for the research purposes is described. Chapter 4 consists of the results of experiments performed in both virtual validation and real world environment, the presented results were obtained from engineering tooling implemented during the course of the research for the purpose of comparative assessment of fusion methods. Core results and conclusions are presented in Chapter 5.

2. Background and Methods

This chapter presents a review of assessed sensors and methods used in the research. Chapter walks through the common types of perception technologies used on the market. Further, a description of fusion and inverse sensor methods is presented, followed by an overview of virtual simulation technology employed in the research. Chapter also presents the real-world testing setup and the offline reprocessing tools.

2.1. Modern Advanced Driver Assistance Features

To put the sensing function into perspective it is crucial to understand what type of functions on the vehicle system level are realized and how the sub-components of those systems interact with each other. The current focus of the European market is driven towards components described in Euro NCAP 2025 Roadmap [55]. The legislation direction flowing from the European market creates a demand for the widespread adoption of the technology obligatory for all new cars. The road map paves a plan for longer-term, more advanced solutions related to comfort features. All the anticipated vehicle level functions described in the road map need to rely on the perception sensors and algorithms allowing the implementation of safe system-level architectures. The safety of the architectures is achieved by diversification of sensor types and methods used for reasoning from the sensor measurements. The wide scope of the NCAP adoption road map emphasizes opportunities to exploit virtual testing to add more robustness to the assessment [55]. Apart from the adoption of mandatory safety systems driven by the Euro NCAP, car manufacturers provide a wide portfolio of safety and comfort features that require extensive exterior sensing features. Those functions are described further in this section.

2.1.1. Euro NCAP 2025 Roadmap - Safety Assist

Autonomous Emergency Braking (AEB)

AEB is the staple functionality that has been integrated into NCAP test portfolio used for safety ratings in 2014. The simplest assessment of the function assumes rear-end car-to-car collision mitigation [75]. The example of the AEB test scenario is shown in Figure 2.1. The evolution of this function extends into the mitigation of pedestrian crossing accidents. This function due to various commercial levels in existence requires either a simple radar sensor or extended perception fusion methods for more complex variations [93]. The primary goal of the function is to mitigate or reduce the severity of the crash with the target in the case when the driver fails to react or support the breaking in case of an already initiated breaking. Various supporting functionalities having a similar goal to the AEB are implemented in the vehicles. Functions such as Forward Collision Warning or pre-charging brake pressure to reduce the breaking response time are going to be discussed in further subsections. Those functions allow the increase of the safety of the complete system however is not systematically separated in the NCAP road-map[55].

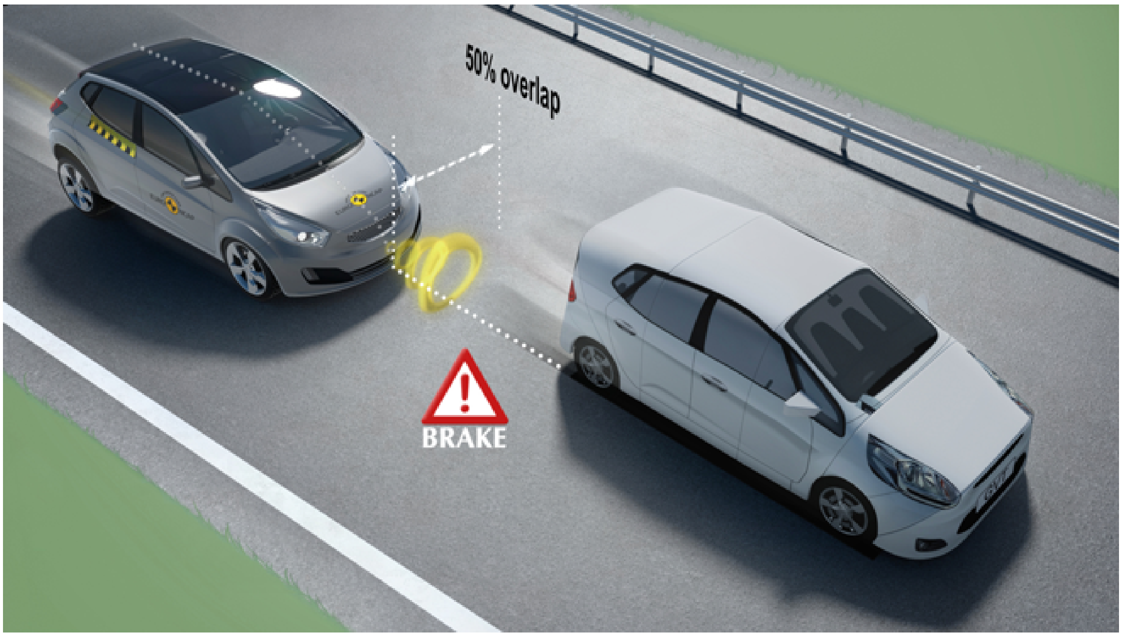


Figure 2.1: Overview of the AEB Car-to-Car scenario in NCAP Safety Assist scenario source: www.euroncap.com

The important variation of the AEB function is addressing back-over reversing crashes as those constitute a substantial amount of personal injuries. This type of mitigation system imposes a need to provide perception features in the rear part of the vehicle. An additional potential requirement for a perception system working on the reverse support system is the capability to distinguish static obstacles from Vulnerable Road Users (VRU) (pedestrians, cyclists, mobility scooters, scooters, mopeds, motorbikes etc.) [55].

The most complex portion of perception and reasoning needs to be employed to assess crossing and turning maneuvers at junctions. The risk of those accidents involves mostly vehicle to vehicle and vehicle to VRU. Those features can be supported by forward-looking radars and cameras capable of detection and classification of targets [34] as well as augmented by perceiving the surrounding infrastructure to help with host vehicle trajectory prediction and localization [33].

The development steps according to include the bring-up of such systems starting from driver initiated within-lane steering support which will require the driver to actually interact with the vehicle to start such lane change, where the system will takeover augmentation of these activities. The more advanced step anticipated in the legislation assumes a merge of the two functions to perform collision mitigation by braking and steering. To achieve these functions a minimum required sensor set is presented in Figure 2.2.

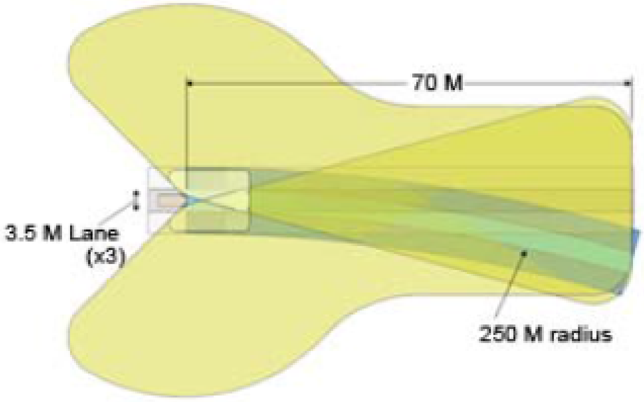


Figure 2.2: An example of sensors envelope used for perception focusing on forward looking applications with a curved segment of a road

Automatic Emergency Steering (AES)

The AES functionality in terms of its perception needs is a more challenging function than the AEB. The main type of crash this function is trying to mitigate is a head-on/ small overlap crashes involving VRUs [76]. As per STATS [19] research most of those accidents result from loss of control or lane/road departure [55]. AES needs the actuator end (automated steering) in the system. Those needs are partially satisfied by other safety/comfort functions such as automated parking, lane centering) that will be discussed further in this section.

2.1.2. Other Comfort and Safety Features

This chapter presents a review of other functions offered in consumer vehicles as parts of safety and comfort packages that can benefit from the perception methods evaluated in this study.

Forward Collision Warning (FCW)

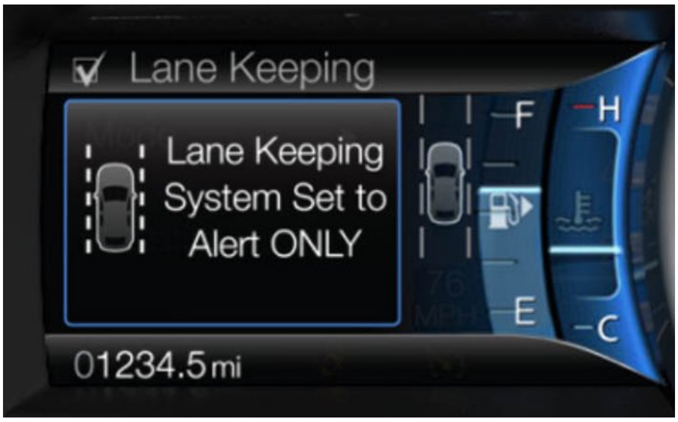
The system is based on the same principles as the Autonomous Emergency Braking, relying on a similar senor suite however limited to providing an audio or visual warning to the driver in case of the potential risk of a crash. Depending on the commercial implementation these types of systems can have different names. In the majority of cases, the FCW is integrated as a subset of suite functions [35].

Emergency Brake Assist (EBA)

One of the systems proposed for standardization by the UNECE defines the goal of reaching the capability to increase road safety by providing a capability to minimize the brake response time in case of collision risk[86]. The system's capability is defined as having functions allowing to detect a dangerous situation using the perception systems onboard the vehicle to provide a decision to the braking system to increase the pressure in the braking system allowing the brake calipers to minimize the travel time in case of required emergency braking situation. This type of system operates on a similar principle to AEB and is a supporting function, however since it is a much simpler method, can rely only on a simple sensor used to assess the traffic conditions.

Lane Keeping Aid (LKA)

The supporting function allows the system to support the driver in maintaining the position of the vehicle between the lanes on the road. The system finds multiple implementations in vehicles depending on manufacturers and ranges from the capability to provide an audible warning to the driver in case of unintended drift to the side of the road without enabling the indicators, to closed-loop feedback assisting with the automated steering to mitigate such unintentional drift. Usually, the system is enabled and disabled using steering wheel buttons as shown in Figure 2.4. The lane supporting functions require sensors such as cameras to interpret the lane markings, however can benefit in road shape estimation with perception methods allowing to build a road shape from other data [61]. An example of user interface information on the function availability is shown in Figure 2.3.



☑ Lane Keeping

Lane Keeping System Set to Alert ONLY

01234.5 mi

Figure 2.3: Ford graphical user interface showing that lane keeping system works in alert mode only. Image source: www.autoguide.com

Adaptive/Active Cruise Control (ACC)

The Adaptive Cruise Control function also is available with different commercial naming depending on the car manufacturer, however, the overall principle of operation is the same. The key goal of the system is to support the driver with maintaining a safe, preset distance in time to the preceding vehicle while maintaining velocity control [30]. Usually the system is enabled and disabled using steering wheel buttons as

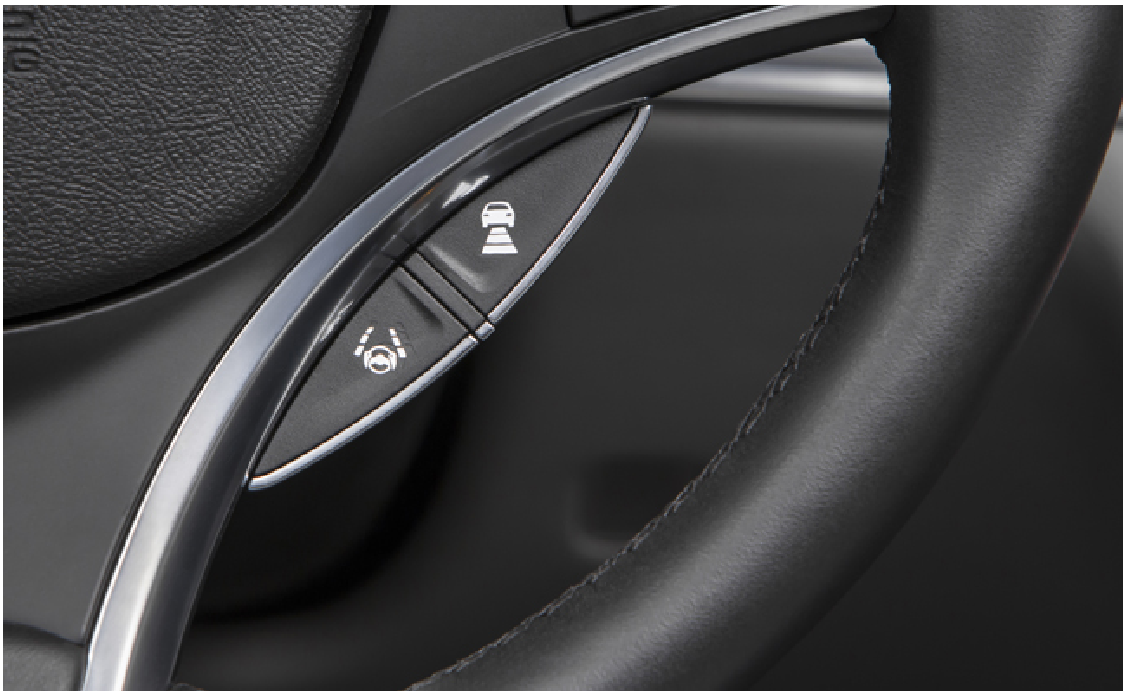


Figure 2.4: Acura's MDX steering wheel buttons for enabling and disabling the LKA and ACC functions Image source: www.autoguide.com

shown in Figure 2.4. The system primarily relies on forward-looking sensors to gather information on dynamic objects in front of the vehicle, however, some variations can rely on additional data about the infrastructure such as lanes or barriers shape to allow lane assignment and threat assessment of the vehicles in front. Lane shape and host trajectory estimation can be supported by the utilization of occupancy grids in a similar way as LKA. Potential application of the occupancy grid for those features would require post-processing methods applied to the occupancy grid to extract the road shape to increase the reliability of the estimated model of the road to increase the performance of lane assignment.

Intersection Assistant

The intersection assistant is a function that focuses on support on low-speed scenarios (such as exiting the parking lot onto a busy street). The function can be split into two variations in a similar manner as in the case of the AEB/FCW. The two types of responses can be provided by the vehicle system - warning in case of lower-risk scenario and actual braking in case of an imminent threat [48]. The direct focus of the perception system in the basic version of the function utilizes the front radar. However, variations of the function for use in reverse gear also exist. The variation of the feature can be commercially found under a name of "rear cross-traffic alert". It is more common to utilize rear corner radars for such functions. The example of a use case scenario for intersection assistant is presented in Figure 2.5.

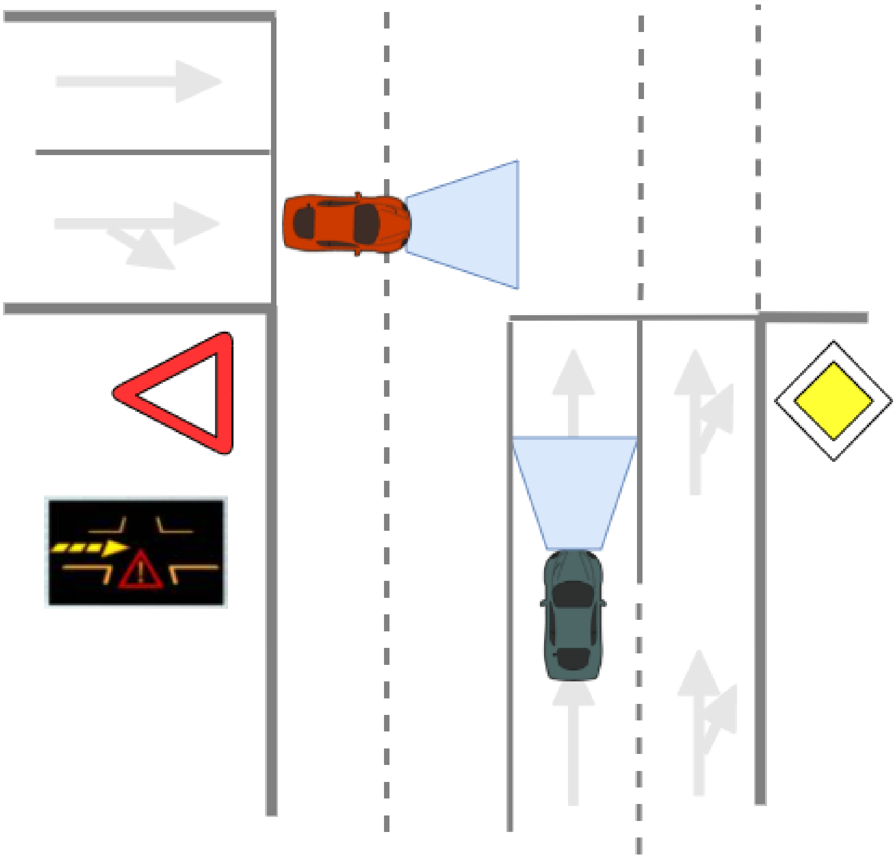


Figure 2.5: Intersection Assistant function scenario schematics with view of the warning user interface icon.

Automated Parking

Another use case for safety and comfort supporting systems focuses on automated parking (both in terms of finding parking spots and execution of parking maneuvers). Perception for this type for application is explored in this research in Chapter on experimental assessment of the occupancy grid methods.

The general parking use case requires surround perception which can utilize corner sensors as well as forward and rear looking ones [83]. Optimization of the sensor suite and perception methods could be categorized depending on exact applications. The following systematization of the applications is proposed:

- Parallel parking with the front of the vehicle - where the parking space is approached by the vehicle moving forward and parking maneuver is executed by driving in the spot with the front of the vehicle [18]. A schematics of that situation is described in Figure 2.6.
- Parallel parking in reverse - similar to the previous method however the vehicle drives past the spot and then reverses into the free parking space [18].
- Angled parking with the front of the vehicle - where the vehicle drives into a spot with its front. The parking space is positioned at an angle to the road which the vehicle is driving [46]. An example of such a parking application is shown in Figure 2.7.
- Angled parking with the rear of the vehicle - similar as previous with the vehicle reversing into the parking spot [80].
- Perpendicular parking with the front of the vehicle - it is a specific subset of the angled parking with the front of the vehicle but the parking spot is at the 90° angle to the road [46].
- Perpendicular parking with the rear of the vehicle - it is a specific subset of the angled parking with the rear of the vehicle but the parking spot is at the 90° angle to the road [80].

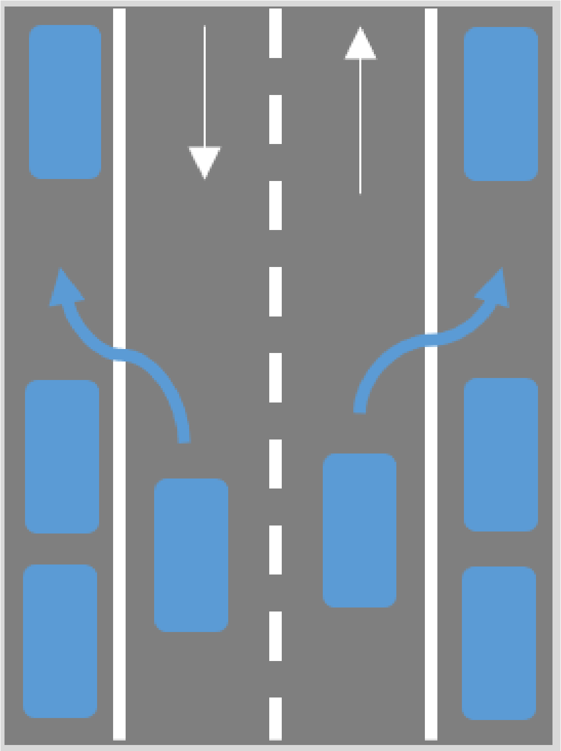


Figure 2.6: Parallel parking scenario as described in https://wiki.unece.org/ ADAS 03-13 AVERE Input With respect to parking use cases as addition to ADAS 02-17.

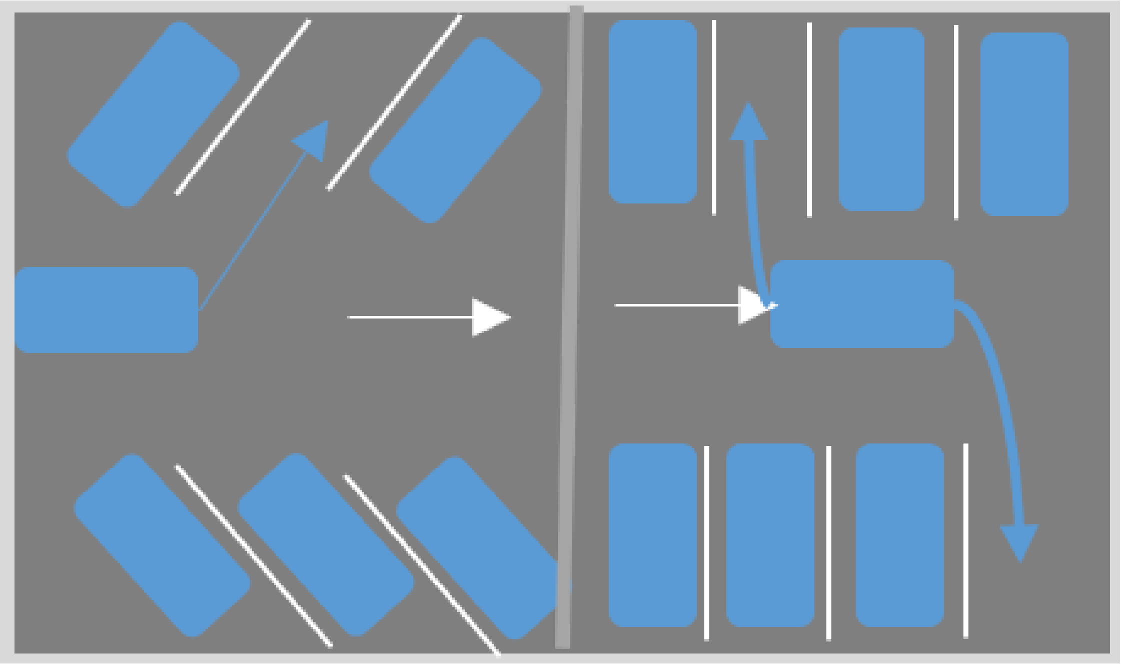


Figure 2.7: Angled and perpendicular parking scenario as described in https://wiki.unece.org/ ADAS 03-13 AVERE Input With respect to parking use cases as addition to ADAS 02-17.

Primary diversification of sensor suites location follows similar rules as in case of collision warning, braking, and intersection assists - that maneuvers requiring forward motion require preliminary focus with perception in corresponding parts of the vehicle[56]. Parallel parking in some cases benefits from radars mounted on the side of the vehicle i.e. in the vehicle doors or sills.

The problematic topic that impacts sensors installation in vehicle doors is the vulnerability of the system to misalignment within the life-cycle of the vehicle as to the alignment of the doors changes due to wear and can cause problems when the vehicle doors are suddenly opened. An additional problem with this approach is present with vehicles equipped with sliding doors (such as vans or buses or trucks), due to impact forces exerted on the mechanical components due to the shock of the doors being closed - that type of application influences the mechanical construction of the Printed Circuit Boards (PCB) as they have to be more resistant to the shear forces increasing the design cost. The additional benefit of placing the sensors in the sills improves the capability for curbs detection which allows more accuracy for localization applications. Low mounting height however is not desired for long-range perception as the road surface enters the sensor field of view at a low range resulting in the case of radars with ground reflections impacting mid to long range performance.

The use-case for parking applications needs more capability to detect objects at closer ranges where ultrasonic sensors play a great role. The specific characteristics of modern Parking Distance Control (PDC) sensors need to be accounted for in perception systems. Two primary features that need to be accounted for is the fact that those sensors incorporate long delays due to slow propagation of ultrasonic waves and lack angular discrimination in their function[20]. Angular discrimination is obtained in such cases with the use of post-processing algorithms to find the reflection points or accumulate probability on occupancy grid maps. More detailed discussion on PDCs can be found in Section focusing on the sensors and their capabilities.

Steering Assistants

Various optional functions available in modern vehicles provide the capability to assist the driver in the task of driving. Commercial names of the systems vary from safety assistants to traffic jam assistants, however, in terms of provided closed-loop capabilities they focus on helping the driver by incorporating more advanced versions of lane assistance (staying in the lane) and adaptive cruise control (maintaining a preset distance to receding vehicles)[10] [2]. The Operational Driving Domain (ODD) of those systems varies in terms of an actual implementation. The ODD is a definition of the set of environmental and situational conditions in which the system is allowed to be enabled by the driver[74]. The conditions include speed ranges (i.e. for lower speeds for driving in traffic jams to support cruising velocities on highways) weather conditions (predominantly good weather), driver engagement (since the system requires the primary task of driving to be carried out by the driver - in many cases the solutions are coupled with Driver Monitoring Systems (DMS) to ensure drivers presence and focus on the task) and geographical location (use of the system in certain geo-fenced areas). Systems like this rely in their implementation on surround sensors allowing them to monitor areas in front, on the sides, and in the rear of the host vehicle. The senor-suites require multiple domains of sensors (such as radars, cameras, and lidars to operate - the detailed discussion on those sensors is covered in further chapters). Similar to the ACC and LKA functions static environment perception is a key for building a reliable model of the surroundings to realize the tactical planning of the control operations. Due to the need for estimation of road shapes and obstacles that can be defined in a probabilistic manner these functions can benefit from the use of occupancy grids.

Jam Pilots

Similar to the Steering Assistants the Pilot functions are bound with similar ODD frameworks. The particular parameters for the ODD for those functions are more restrictive as these features in most cases can be only active in geo-fenced areas[74]. The functions of this type in the ODD require a driver that needs to be able to take over the request of the system to take over full control over the system. Most of the commercial systems deployed on the market are bound to operate in traffic jam scenarios - low-speed stop-and-go situations. The systems design requires the vehicle to be brought to a full, safe stop in case of driver's lack of takeover[17]. Due to these characteristics, the interesting use-case for occupancy grids is to deliver the perception of the obstacles [23].

Potential uses of the perception are to estimate the safe stop areas i.e. in the driving lane or on the road shoulder. An additional important feature for occupancy grid usage in those systems is to support the positioning activities with regards to matching to detailed map models. This localization application is intended to support a demand for a high accuracy position in conjunction with global positioning systems [68]. The localization process can be carried out by extracting characteristic features of the surroundings and matching them with already known and mapped locations [89]. In the case of those systems, demand for methods alternative to tracking is not only driven by the functionality needs but also from the redundant and safe architectures [29]. The example architecture of redundant safety perception algorithms is shown in Figure 2.8.

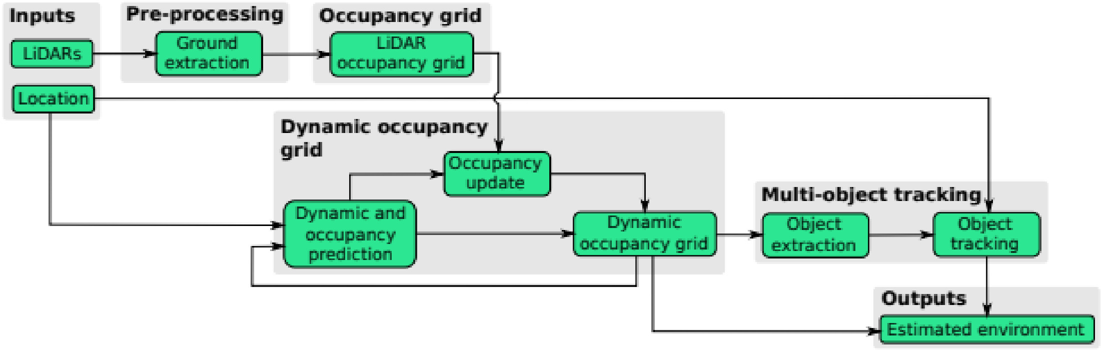


Figure 2.8: An example of perception algorithms architecture proposed by Godoy for application in pilot applications, image sourced from the cited paper[29].

To achieve the desired level of functionalities multiple sensors and devices need to be equipped on boards of the development and test vehicles. The example of such a systems suite is presented in Figure 2.9

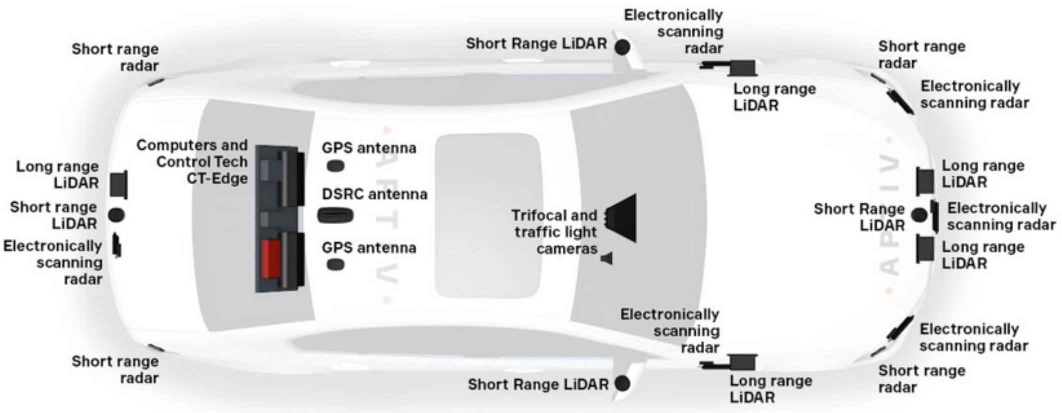


Figure 2.9: A proposed sensor and test device suite for pilot type safety system applications proposed by Aptiv at CES2019.

Trucks Safety

Since the discussed functions that can employ occupancy grids mainly focus on passenger vehicles, there are variants existing for the commercial and Heavy Commercial Vehicle (HCV) segments. The mandatory functions such as blind-spot monitoring for HCV provide technical challenges for perception in terms of vehicle geometry and the presence of trailers. Similarly to the passenger vehicles, the mounting height and the location of the sensors need to comply with the overall functional vehicle design exactly the same perception limitations apply to the system as the field of view (one of the big challenges is the high vertical mounting location influencing the properties of both radars and cameras) as well as more complex control dynamics in case of multi-trailer systems [22]. The HCV applications also underline a good use case for virtual simulations for preliminary optimization of the design. The primary driver for the truck systems are the safety regulations with a lower focus on the comfort functions, however, the mandated functions guarantee a need for big market coverage and a need for low-cost system components.

2.2. Automotive Perception Sensors

To understand the challenges behind the perception systems used in Advanced Driver Assistance, we need to have an outline of what exactly the vehicle has to "see". From the point of view of the end-user, we only see features discussed in previous sections such as Lane Keeping Aid, Highway Pilot, Intersection Assist, etc. All the discussed features require information about the surrounding environment with the precise distinction between object types and parameters describing the surrounding environment. As the current state of technology allows us to differentiate between stationary and moving objects with property assignment in abstract type definitions covering most of the types of obstacles that could be encountered in traffic (vehicles - trucks, passenger cars, lane markings, road signs, barriers, pedestrians, motorcyclists, etc.) [57] [49].

From the point of view of automotive standards, one of the most critical aspects considered in designing any automotive system is safety. This area of expertise over time evolved into its own domain called Functional Safety, which is an independent factor driving requirements and architecture of implementations towards compliance with ISO 26262 and its derivatives[40]. The second most significant challenge that is encountered in automotive design is cost efficiency, which imposes certain limitations directly impacting hardware design, this influence on hardware design [57] indirectly impacts the design of software and algorithms. The major way in which cost efficiency has an impact on algorithmic and software solutions is in available computational resources which can be split into two major factors: computational power and available memory, thus requiring thoughtful design as well as optimization of implementations.

As a result of the above-mentioned conditions and advancing regulations, demand for the low-cost reliable system has been created as more and more segments of vehicles will require a wide range of perception capabilities. Increased demand in sensor performance (accuracy and resolution) is required for higher segments with a focus on more advanced features. This section covers a review of existing systems and technical principles standing behind various types of sensors common for the automotive market. As a result of the review, a comparison of different methods is presented with a focus on detection capability, and safety levels allowed by each of the systems.

2.2.1. Sensing Technology Overview

As a result of the presented constraints, several different approaches have been formed to tackle the problem of detection and tracking of traffic participants and the environment. In the course of history the first implementations of detection and ranging systems were not providing any abstract classification of objects, those systems were capable to determine the distance to whatever obstacle or object was preceding it, without angular discrimination[64]. The first type of object that has been classified in detection methods for commercially available systems was in general form a "vehicle". Early-stage systems are mainly based on ultrasonic sensors (further called PDCs - Parking Distance Sensors) and radars - evolving from single to dual-beam and further to scanning radars. In parallel, the development of vision-based systems was ongoing, in the early stages the use of vision systems was mainly limited to lane detection to precisely locate the host inside the driving lane. Along with regular vision systems more specialized implementations have been presented to the market such as Near Infrared and Far Infrared cameras.

Radar Based Technology

One of the most widespread types of sensors that is currently in use for commercial systems is a microwave radar, the principle of operation of these devices mainly rely on phased array transducers capable of generating high frequency electromagnetic modulated pulses. In typical solutions, the frequencies of those pulses are either in 24GHz or 77GHz. A systemic diversification has been introduced for separation into the 3 major groups of such sensors, with respect to the range of operation. Most of the automotive radars are continuous wave electronic scanning radars, that emit the series of frequency swept pulses, and after performing multiple Fast Fourier Transform (FFT) extract the range, range rate, and azimuth information which further is processed to extract the target information[62].

Long Range Typically Long-range radars operate in ranges up to 250 m in distance[64], the usual field of view of this type of device is narrow 10° to 15°. This type of device is mainly used for features such as ACC for identification of large targets mainly vehicles - at a greater distance up to 250m [70]. In many implementation cases, long-range radars are integrated with medium-range radars with a switchable mode of operation allowing performing different types of scanning. An example of an electronically scanning radar is shown in Figure 2.10.

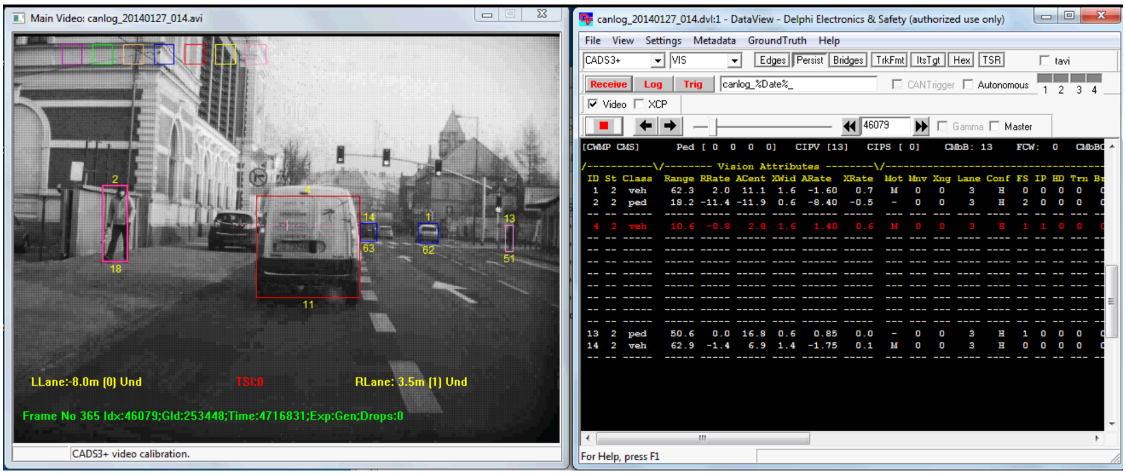
Medium Range Devices of this type as mentioned before are usually fused with long-range radars and used for front sensing - this means mounted on vehicles front end, usually the front grill or behind the front bumper. Medium range radars are among the most commonly used sensors for the ADAS systems, providing the capability of detection of much more diverse types of objects, such as vehicles, barriers road edges, pedestrians, and motorcyclists. For those objects to be used in consumer functions with such safety levels (such as Autonomous Emergency Braking or Collision Avoidance with vehicle Path Change) high levels of functional safety are required. Those devices provide the field of view of from 65° to 70° and effective range depending on the target type.

Figure 2.10: Example of Electronically Scanning Radar - produced by Aptiv.

Short Range Recently systems of such type gained more popularity in safety systems mainly for environment sensing; the output of the devices is applied to support the vehicle position and orientation estimation but is not only limited to that task. They are in most use cases used in pairs mounted on corners of the host vehicle - providing a view to the sides of the car. The fields of view in some areas overlap in front of the vehicles. Pairs are usually mounted in front and rear of the vehicle [71]. These devices in many cases are providing the regular object information feature like free space detection, which later can be consumed for example by Automated Parking Assistance or Lane Change Merge Aid functions. The maximal range of those devices is up to 80m and operates in the same frequencies as allocated for all other automotive radars, with typical optimal detection distance ranging usually up to 50m [28].

Vision Based Technology

The next very heavily used family of sensors that find application in all solutions ranging from driver support systems to automated driving solutions, are based on image processing. Two main types can be distinguished from the main principle of operation point of view, systems that base on the visible light spectrum come mainly in two forms–single vision cameras and stereo vision[87]. Those types of systems gained much popularity due to the easy implementation of image processing methods.


Main Video: canlog_20140127_014.avi
2
18
4
11
14
63
1
62
13
51
CADS3+ video calibration.
canlog_20140127_014.dvl:1 - DataView - Delphi Electronics & Safety (authorized use only)
File View Settings Metadata GroundTruth Help
CADS3+
VIS
Edges
Persist
Bridges
TrkFmt
ItsTgt
Hex
TSR
☐ tavi
Receive
Log
Trig
canlog_%Date%_
☐ CANTrigger
☐ Autonomous
1 2 3 4
☑ Video
☐ XCP
46079
☐ Gamma
☐ Master
For Help, press F1

```
LLane:-8.0m (0) Und          TSI:0          RLane: 3.5m (1) Und

Frame No 365 Idx:46079;GId:253448;Time:4716831;Exp:Gen;Drops:0
```

```
[CWMP CMS]      Ped [ 0   0   0   0]   CIPV [13]   CIPS [ 0]    CMbB: 13    FCW:   0    CMbBC
/----------\/-------- Vision Attributes --------\/----------------------------------------
 ID St Class  Range RRate ACent XWid ARate  XRate  Mot Mnv Xng Lane Conf FS IP HD Trn Br
  1  2  veh    62.3   2.0  11.1  1.6 -1.60   0.7   M    0   0    3    H   0  0  0  0   0
  2  2  ped    18.2 -11.4 -11.9  0.6 -8.40  -0.5   -    0   0    3    H   2  0  0  0   0
 -- -- -----  ----- ----- ----- ---- ------ -----  --- --- --- ---- ---- -- -- -- --- --
  4  2  veh    10.6  -0.8   2.8  1.6  1.40   0.6   M    0   0    3    H   1  1  0  0   0
 -- -- -----  ----- ----- ----- ---- ------ -----  --- --- --- ---- ---- -- -- -- --- --
 -- -- -----  ----- ----- ----- ---- ------ -----  --- --- --- ---- ---- -- -- -- --- --
 -- -- -----  ----- ----- ----- ---- ------ -----  --- --- --- ---- ---- -- -- -- --- --
 -- -- -----  ----- ----- ----- ---- ------ -----  --- --- --- ---- ---- -- -- -- --- --
 -- -- -----  ----- ----- ----- ---- ------ -----  --- --- --- ---- ---- -- -- -- --- --
 -- -- -----  ----- ----- ----- ---- ------ -----  --- --- --- ---- ---- -- -- -- --- --
 -- -- -----  ----- ----- ----- ---- ------ -----  --- --- --- ---- ---- -- -- -- --- --
 -- -- -----  ----- ----- ----- ---- ------ -----  --- --- --- ---- ---- -- -- -- --- --
 13  2  ped    50.6   0.0  16.8  0.6  0.85   0.0   -    0   0    3    H   1  0  0  0   0
 14  2  veh    62.9  -1.4   6.9  1.4 -1.75   0.1   M    0   0    3    H   0  0  0  0   0
 -- -- -----  ----- ----- ----- ---- ------ -----  --- --- --- ---- ---- -- -- -- --- --
```

Figure 2.11: Screenshot of the user interface of data recording and visualization tooling used to register object data information derived from a mono vision camera. The image presents bounding boxes around vehicles and pedestrians, along with distance information and lane marking data.

Mono Vision The most common approach in the application of the vision sensing systems comes in a form of a single lens and single image matrix solution allowing capturing of image data later converted to electronic digital signal [54] in resolutions spreading from VGA up to 8k resolutions, providing the effective field of view of 70°. In the early stages of commercial implementations the systems were used mainly for road edge and lane detection, in most recent systems a widespread set of objects can be safely identified. The main advantage of camera systems is the ability to distinguish various types of objects, on the other hand, the ranging and velocity measurements are not the strongest attribute [72]. As most of the cameras are ambient sensors they have to be able to cope with many different factors influencing their ability to "see", such as weather conditions, occlusion, or darkness. The cameras utilized in those applications can be equipped with non-standard RGB patterns of pixels, including more sub-pixels for red light detection used for efficient detection and classification of stoplights. The infrared (IR) spectrum sensitivity is also commonly utilized and it is described in the further subsections. An example of data recorded form an automotive camera is presented in Figure 2.11.

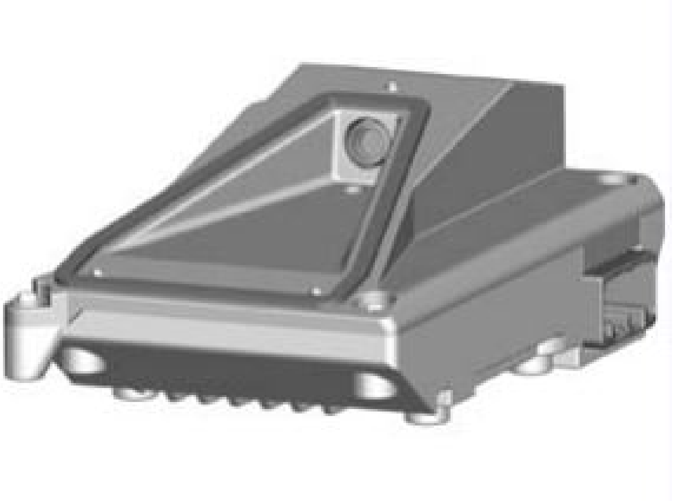


Figure 2.12: Example of Mono Vision Camera - Aptiv IFV200.

Stereo Vision and Multi-Imager Systems Systems equipped with two cameras are much more capable of acquiring geometrical parameters from the environment, such as the distance and height of the object. The ability to extract distance is based on the capability to determine the perspective. Those sensors in commercial segments find their main application in front sensing [27]. Similarly to monovision systems this solution is prone to problems with ambient conditions yet due to redundancy provides more reliable information than optical systems with one optical path. Various different implementations exist with dual or triple optical paths solutions, in which there are up to three images and lenses with different focal lengths feeding alternating images or utilizing multiple sensors simultaneously for perception. Similar to radar solutions multiple cameras can be mounted around the vehicle including locations such as side mirrors, pillars, windshields, and windows. A 3D visualization of discussed mono-vision camera is presented in Figure 2.12.

Near Infrared (NIR) Due to the high dependence on ambient conditions of visible light their infrared counterparts are also employed for automotive purposes, along with the integration of such systems for image processing (1 in every 4 pixels in sensors used to provide intensity in NIR spectrum) auxiliary systems were provided for the driver in form of head-up displays with live streaming of superimposed images providing a better view in bad weather conditions and in darkness due to active illumination[85]. Integration of NIR sensing into monovision sensors is a common practice in modern ADAS sensing systems.

Far Infrared (FIR) The extension of spectrum in the cameras is also employed for some applications. Most of the FIR sensors do not include a combination of visible light. However such solutions are found to be less effective in terms of classification purposes and tracking stability due to lack of color parameters. However, this system can be more effective in identifying pedestrians and live obstacles due to body heat emitted by those targets and the sensor's sensitivity to this data[63].

Red Green Blue and IR (RGB-IR) A common practice in automotive applications relies also on cameras that include both full IR and visible light sensitivity. This type of imager can be used for both interior and exterior sensing. The interior sensing variant of this camera is shown in Figure 2.13.

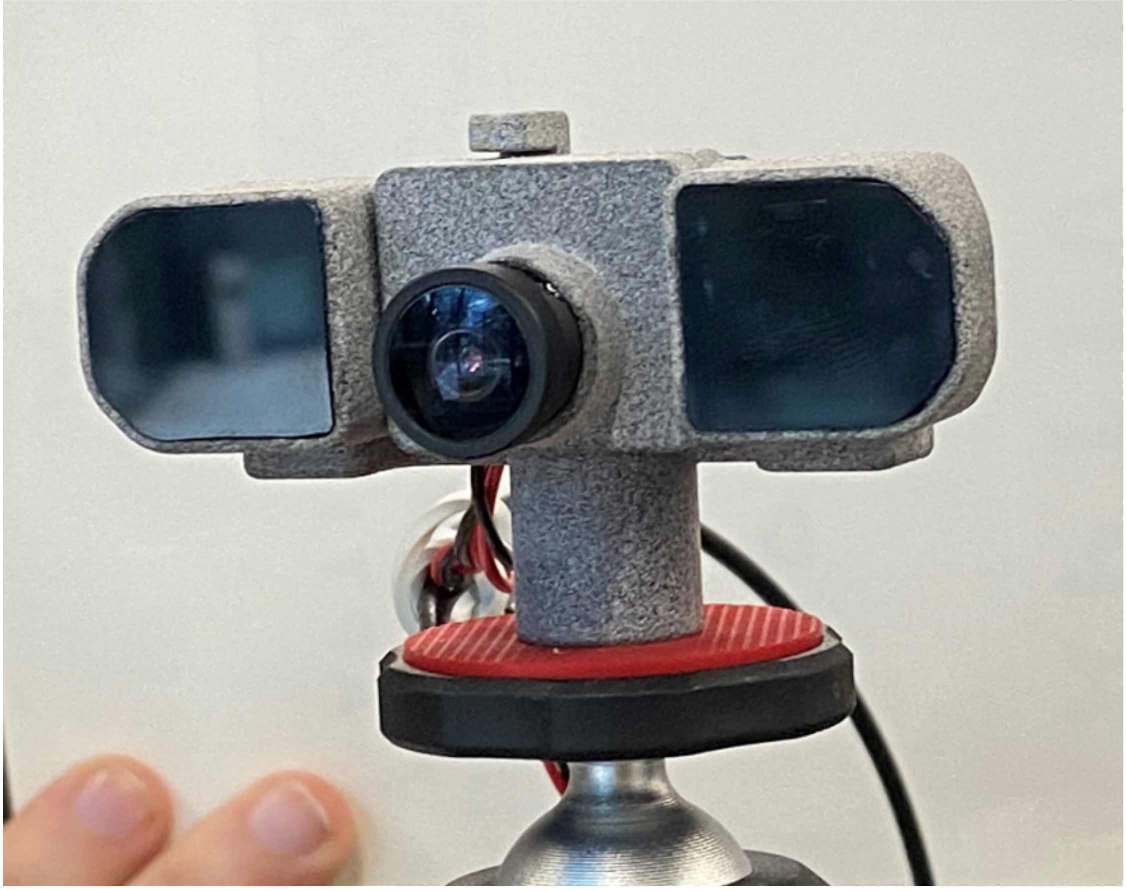


Figure 2.13: Example of RGB-IR interior sensing camera prototype with visible lens and active illumination concealed behind a covering glass.

Surrounding Cameras Image-based sensors find application in sensing not only the area in front of the car but as a source of information on the area behind and on the sides of the car. In commercial applications, the main features that consume data from these sensors can be split into two modes of operations. In low speeds, the system composed of 4 cameras building a 360° field of view is sometimes called birds eye's view [95] or top view [84] - this type of solution usually employs a Graphical Processing Unit (GPU) to assemble together the images to assemble a virtual representation of the top view of the environment[84]. Usually, the data from this kind of sensor is merged in an external processing unit and then processed by a sensing algorithm that determines obstacle's position mainly for automated parking. Another variant of the implementation is based on 2 cameras mounted underside mirrors providing data on lane markers and surroundings along with object data for features warning the driver on the presence of objects in so-called blind spot[65].

Laser

Ever since the invention of the laser, the system has found application in ranging applications [26], the described systems are based on a similar principle yet imply two different approaches for beamforming and apply to different philosophies of sensor application.

Laser Rangefinders Simple range finders can be found in many commercially available systems as a low-cost ADAS sensor for autonomous emergency breaking[36], These sensors usually do not provide a capability to distinguish between object types in front of the vehicle yet provide information on distance to preceding and stationary object in a narrow field of view. Due to low cost, these systems have been broadly implemented in various segments of passenger cars providing a reasonable impact on a decrease in the number of accidents. In many cases, this type of sensor relies on a fixed number of multiple laser beams that are separated by fixed azimuth discrimination.

Lidar Lidar is the most known sensor applied currently in research and development of autonomous driving systems[64]. In Laser Imaging, Ranging and Detection systems the most common approach in the generation of 3D meshes is to measure range to objects using laser beam deflected by a moving mirror, during a rotation and tilting of the mirror several measurements is carried out to estimate distances and in some applications velocities of points reached by a laser beam. Commercial application of currently existing solution is limited due to still high unit cost of production of those devices [3]. These systems, on the other hand, provide a very detailed representation of the environment which imposes also high demands on computational power for processing the data obtained from such sensors. The example of the lidars used for reference data generation is presented in Figure 2.14.

Processing is required for the extraction of the abstract definition of objects for further processing in control algorithms for automated driving purposes. Variation of the technology in this domain is solid-state LIDAR, which is more compact and of lower cost than their industrial derivatives[6]. There are limitations beyond the cost of the systems related to the lifespan of the sensors that need to be accounted for in system-level design when equipping such solutions.

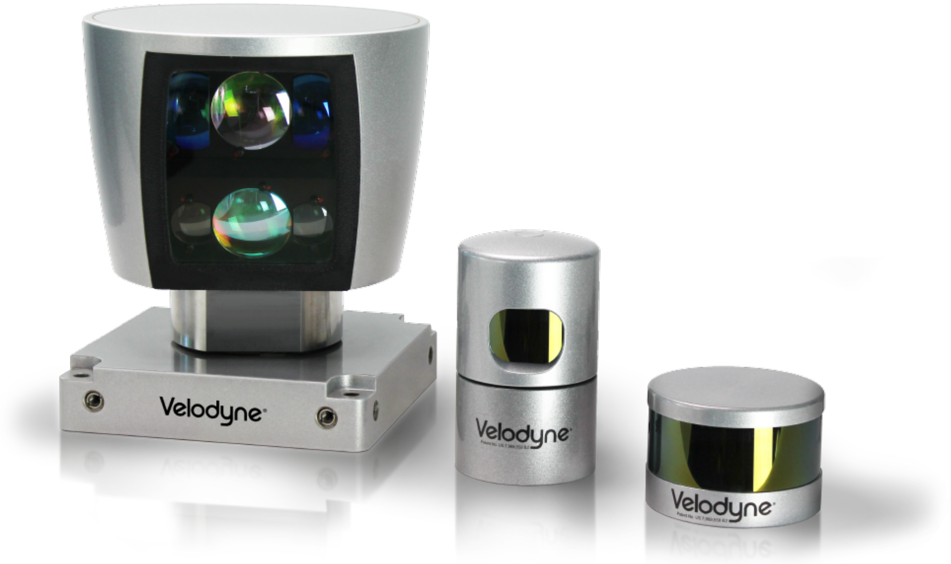


Figure 2.14: Examples of Automotive LIDARs - Velodyne LIDAR Family (http://velodynelidar.com/)

Other

Apart from the systems presented above, there are sensors that don't fall exactly into the already described subtypes.

Parking Distance Control (PDC) Parking Distance Control sensors have found a successful application in supporting automated parking applications as well as adaptive cruise control and blind-spot monitoring. Those systems usually rely on ultrasonic sensors which emit acoustic pulses in frequencies above 20kHz. Angular discrimination is usually obtained by placing independent sensors along the bumper and measurement of the round trip time of the pulse (time of flight) [13]. Those systems are pretty primitive in comparison to previously discussed systems but are good enough for low-speed operations even though they do not provide an abstract interpretation of the environment. The major advantage of those sensors is the low cost of production.

Fusion (Radar + Camera in one sensor) Devices based on merging two sensors gained popularity on the market as they provide a reliable source of information on moving and stationary objects in front of the vehicle, of course, sensor fusion is a popular method of increasing the reliability of measurements and is commonly implemented with all the mentioned above sensors, yet on the system level such fusion is carried out by an external device (like centralized safety unit). There also exist commercially successful implementations in which part of the sensor fusion is carried out in the device directly [12], the rendering of such sensor is presented in figure 2.15. In the case of fusion of radar system (medium and long-range) with mono-vision camera, objects detected and identified by the camera are used to refine object information from radar systems. Due to redundant implementation of sensors, it is possible to implement functional safety mechanisms that allow identifying faults for achieving high goals in functional safety requirements, this includes VRU detection for functions imposing direct activation of breaks. Technically this sensor provides abstract representation instead of measurements however can be included as an independent distributed subsystem. Those types of installations can also provide control and advanced computing capabilities to execute safety features like ACC or LKA[96].

Figure 2.15: Example of Integrated Fusion System - Delphi Integrated Radar and Camera (RACam)

2.2.2. Synthetic Comparison

In this chapter, a comparison of key components will be provided in Table on the next page. The rating in Table relates to the disadvantage marked as "-", neutral as "o", advantage as "+", strong advantage as "++".

		Radar			Camera					Laser		Other	
		Long Range	Medium Range	Short Range	Mono	Stereo	NIR	FIR	Surrounding	Rangefinders	LIDAR	PDCs	Fusion (Camera + Radar)
Performance	Range	++	+	+	+	+	+	+	-	+	+	–	++
	Long. Performance	++	++	++	-	+	-	-	-	+	+	–	++
	Lat. Performance	-	-	-	++	++	++	o	+	o	+	-	++
	Environmental Conditions	+	+	+	-	-	–	-	-	-	-	o	+
	Low Ambient Lighting	++	++	++	-	-	+	++	–	++	++	++	+
	Object Classification	o	o	-	++	++	++	+	+	-	+	–	++
	Stationary Objects Detection	-	-	-	++	++	++	++	+	+	++	++	++
Price	System	++	++	+	+	+	-	–	-	++	–	++	-
	Development	++	++	++	+	+	+	+	-	++	+	++	++
Pros		-longitudinal performance -cost	-longitudinal performance -cost	-versatility -cost	-lateral performance -object classification	-geometrical parameters estimation -object classification	-night vision	-warm object detection	-top view -close range sensing	-cost -low light performance	-precision -low light performance	-cost	-redundancy -performance in geometrical parameters estimation
Cons		-lateral performance	-lateral performance -VRU	-VRU -lateral performance -very short distance sensing	-longitudinal performance -weather influence	-weather influence	-weather influence -cost	-cost -not available at daytime	-range -weather influence	-lack of object identification	-cost -still under development	-limited application	-size

2.3. Occupancy Grid Overview

The methodology presented in this section is based on the occupancy grid principles described by Elfes and Moravec as a tessellation of space into cells, where each cell contains a probabilistic estimate of its occupancy[25]. The probabilistic representation of occupancy can be augmented with additional parameters, such as: drivability classification (certain cells are under-drivable or over-drivable) [44] or information on the particular state of the (e.g. move-ability)[67]. Equation (2.1) describes a 2D probabilistic grid g_{ij} of size N_L by N_W cells. An example is shown in Figure 2.16. This perception method has applications in: robot navigation, trajectory planning, trajectory validation, map building, positioning, and map matching Simultaneous Localization And Mapping (SLAM).

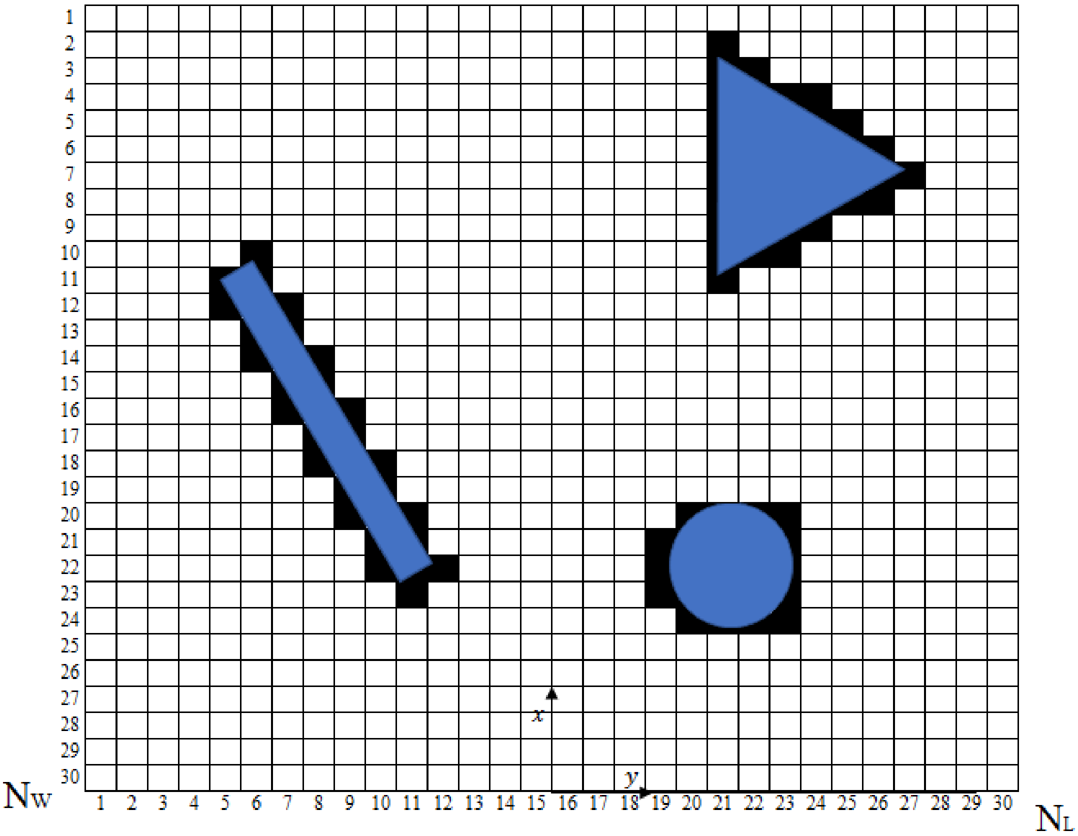


Figure 2.16: A simplified 30 by 30 cell grid with obstacles shown with blue color. Black color indicates occupied and white unoccupied grid cells.

$$\{g_{ij} : 1 <= i <= N_L, 1 <= j <= N_W\} \in \langle 0, 1 \rangle \tag{2.1}$$

The wide range of applications spans beyond automotive; however, the presented assessment focuses on scenarios and tools used in the automotive domain. The process described in this section allowed to enable fast prototyping of perception methods for this research purpose. The proposed approach also allows rapid prototyping in the early stages of commercial projects where real vehicle data is not yet available. The research performed with the described methodology allowed verification of the capability of a virtual validation environment as a source for data-driven performance evaluation.

2.3.1. Inverse Sensor Models

A cornerstone component in grid mapping is the Inverse Sensor Model (ISM), which is used for the calculation of occupancy probability from the measurement information (such as a detection point). The concept of the ISM was introduced by Elfes and Moravec for mapping in mobile robotics applications [59],[58]. ISMs were utilized in this research to update the 3σ region around the detection point. ISMs can be split into two categories - with and without ray casting. An extensive study looking into inverse sensor models developed and assessed in real-world conditions can be found in the work of Jakub Porębski[67].

What Are Inverse Sensor Models

Inverse sensor models are used as a mathematical simplification to transform sensor detections into the probability of the detection surroundings. For Lidar and Radar sensors the detection is a single point reported by the sensor. The detection point is created using the physical sensor by its technological principles. In the case of radar sensors, the detection is a point representing the strongest radar detection of an object in a field of view. The detection is created in a radar look by comparing the emitted and reflected waveform pattern to find out if there were any sources of that reflection[42]. That point is assumed to represent an obstacle. Similarly, in the case of lidar detections, we take into consideration that a single point represents an obstacle on the path of the laser beam emitted from the sensor. Detections provided by vision systems are an abstract representation of image processing allowing to detection of edges or obstacles in the camera's field of view. The detections are characterized by their position with respect to the sensor coordinate system along with an uncertainty of that measurement.

It is assumed that a single reflection defines the presence of an obstacle and can be used to derive the probability of that object existing in space. A 2-dimensional (2D) simplification is proposed to visualize how that detected obstacle is mapped onto an occupancy grid using 2D normal distribution.

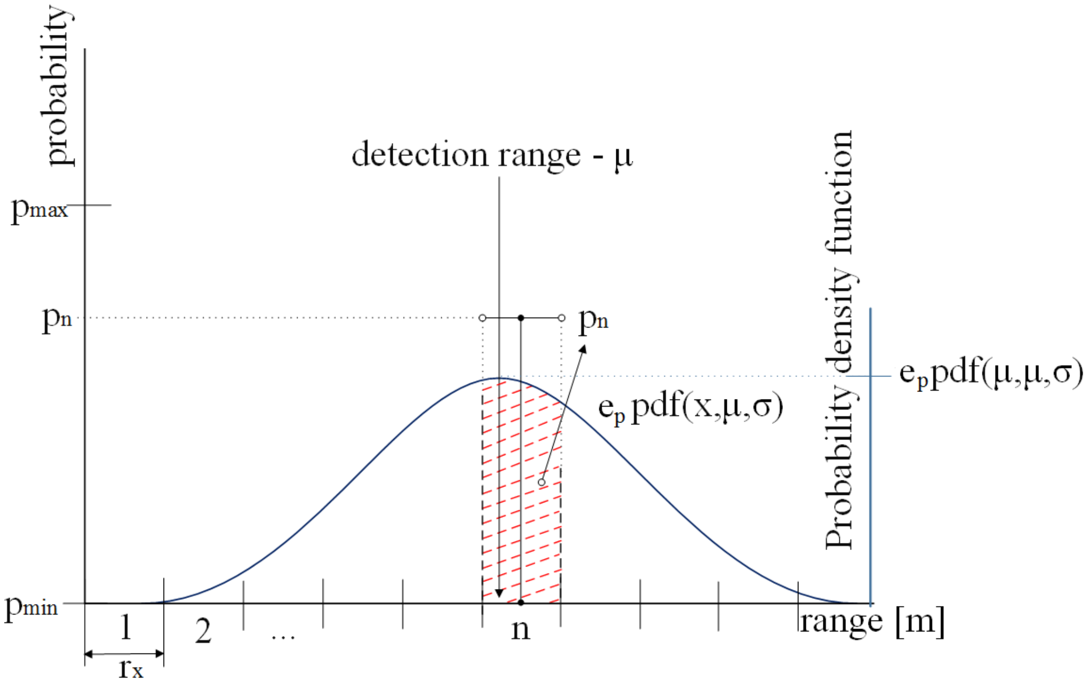


Figure 2.17: 1D simplification of occupancy mapping from a detection response

It is assumed that the detection at a certain range has its position and uncertainty defined which are then used to perform the calculation of a probability density function to estimate the probability in certain grid raster.

The simplest models without ray-casting accumulate information probability for cells surrounding the detection in a 2D space. Sometimes a “hit point" only approach is used by computing probability just for the cell containing the detection. The approach without ray-casting finds uses mainly in binary grids[53].

Models with ray-casting are split into 1-dimensional (1D) and 2D. In the 1D approach, the update is performed on a line of cells connecting the sensor and the detection, whereas in the 2D approach, an area that takes into consideration azimuth uncertainty of a space not being occupied is used. The probability distribution in ray-casting exists in two variants - in the space between the sensor and the detection it can set cell probability to p_{min} along with the cells from the sensor towards the detection area, or apply a probability distribution of the space not being occupied[25][50][51].

Occupancy Probability Calculation

The concept used in this experimental evaluation assumes that the probability of the existence of an obstacle is described by a Gaussian distribution. The explanation of how the uncertainty is related to the detection is visualized in Figure 2.17. The ISMs presented for this application are required to operate for a 2D occupancy grid. This imposes the first requirement to use a Bi-variate Normal Distribution function as the probability density function $\mathrm{pdf}(X;\mu,\Sigma_{ogcs})$ for the calculation of the occupancy probability of the affected cells. The second one is imposed by the sensors that supply the detection points (radar or lidar) with information about the position in 2D with respect to the Sensor Coordinate System (SCS) x_{scs} and y_{scs} and a covariance matrix Σ_{scs} for the detection. Moreover, the detection contains information such as existence probabilities e_p .

The general formula used for the Probability Density Function (PDF) shown in Equation (2.2) requires a covariance matrix Σ_{ogcs} which describes variances and co-variances with respect to Occupancy Grid Coordinate System (OGCS).

$$\mathrm{pdf}(X;\mu,\Sigma_{ogcs}) = \frac{1}{2\pi\sqrt{|\Sigma_{ogcs}|}} \exp\left[-\frac{1}{2}\cdot(X-\mu)\Sigma_{ogcs}^{-1}(X-\mu)^T\right] \tag{2.2}$$

in equation (2.3) where X contains position in x_i and y_i for the PDF values to be calculated.

$$X = \begin{bmatrix} x_i & y_j \end{bmatrix} \tag{2.3}$$

Followed by μ , which in the general formula is the mean value of the distribution. In this case, μ determines the position of the detection in OGCS and consists of position in $x_{det_{ogcs}}$ and $y_{det_{ogcs}}$.

$$\mu = \begin{bmatrix} x_{det_{ogcs}} & y_{det_{ogcs}} \end{bmatrix} \tag{2.4}$$

And the covariance matrix

$$\Sigma_{ogcs} = \begin{bmatrix} \sigma_{xx_{ogcs}} & \sigma_{xy_{ogcs}} \\ \sigma_{yx_{ogcs}} & \sigma_{yy_{ogcs}} \end{bmatrix} \tag{2.5}$$

The sensors report data with respect to SCS. The position of SCS with respect to OGCS is known due to the fact that the sensor is mounted in a known location on the vehicle chassis and the vehicle position is tracked with respect to OGCS. The location in the simulation environment can be obtained directly from a built-in positioning component or in the case of real-world testing using a high fidelity INS (Inertial Navigation Systems (INS). Therefore, a rotation matrix R and a translation vector T can be created to convert the information from SCS to OGCS.

The SCS co-variance matrix Σ_{scs} supplied by the sensor, is presented in Equation (2.6)

$$\Sigma_{scs} = \begin{bmatrix} \sigma^2_{x_{scs}} & \sigma_{xy_{scs}} \\ \sigma_{yx_{scs}} & \sigma^2_{y_{scs}} \end{bmatrix} \tag{2.6}$$

gets rotated to OGCS in Equation (2.7).

$$\Sigma_{ogcs} = R\Sigma_{scs}R^{-1} \tag{2.7}$$

With all the input information, the computation of the probability for cells of the grid lying within 3Σ ellipse (Figure 2.18) can be carried out by the integration of the PDF for each of the cells shown in Equation (2.10). The semi-minor and semi-major axes dimensions can be computed from the Σ_{scs}.

Eigenvalues λ_1 and λ_2 and eigenvectors $\vec{v}_1$ and $\vec{v}_2$ are calculated for the Σ_{scs}. Using this information, we can get the ellipse equation (2.8), where s corresponds to the ellipse size.

$$\left(\frac{x}{\lambda_1}\right) + \left(\frac{y}{\lambda_2}\right) = s \tag{2.8}$$

The value of s can be computed from the desired confidence level for the area we want to update.

To find the orientation of the ellipse, the eigenvector $\vec{v}_{max}$ corresponding to the maximum eigenvalue is used in Equation (2.9).

$$\alpha_{ellipse} = atan\left(\frac{\vec{v}_{max}(x)}{\vec{v}_{max}(y)}\right) \tag{2.9}$$

The ellipse obtained in these steps is used to select the cells of the grid lying inside of the ellipse in order to perform an update either by a lookup table with integration results or by numerically integrating the PDF of selected cell areas.

$$p_{ij} = e_p \int_{x_i-\frac{1}{2}r}^{x_i+\frac{1}{2}r} \int_{y_j-\frac{1}{2}r}^{y_j+\frac{1}{2}r} \mathrm{pdf}(X;\mu,\Sigma_{ogcs})\mathrm{d}y_{ogcs}\mathrm{d}x_{ogcs} \tag{2.10}$$

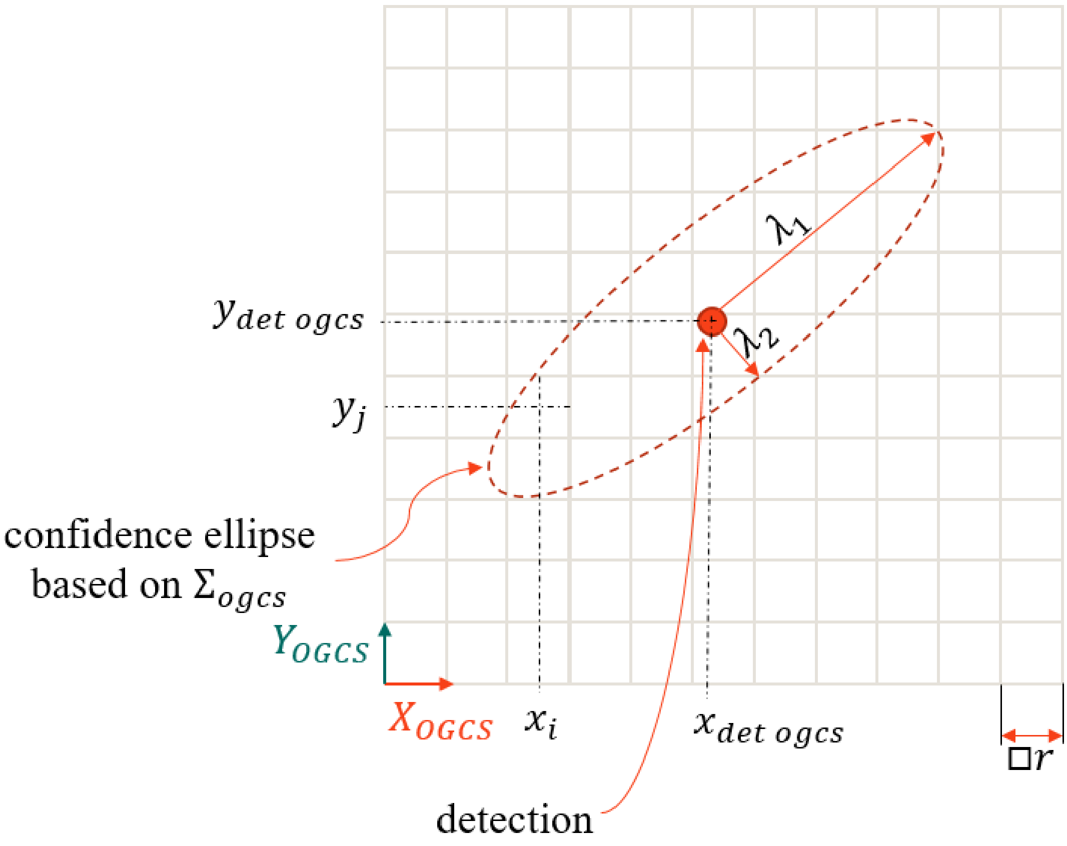


Figure 2.18: Confidence ellipse around the detection with semi-major axis λ_1, semi-minor axis λ_2, and center coordinates x_i, y_j of the cell g_{ij}.

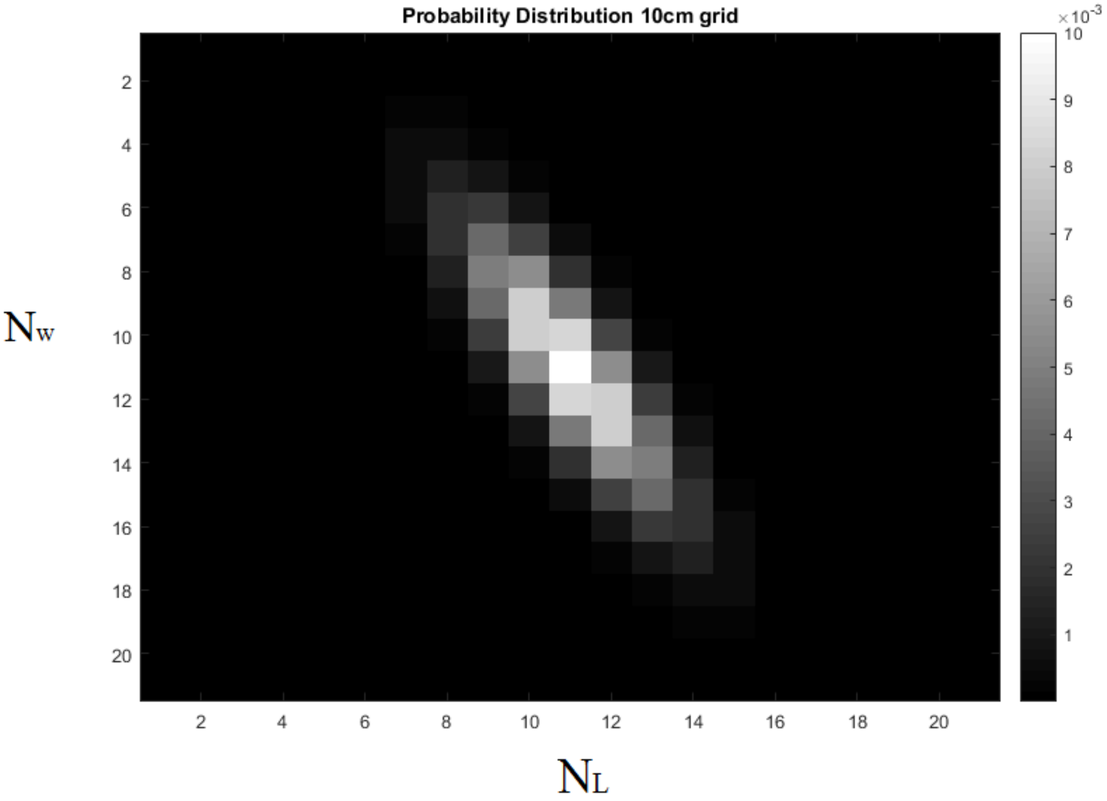


Figure 2.19: Computed result of the probability density function integrated on the grid.

2.3.2. Occupancy Grid Fusion Architectures

Depending on the direct application, with regards to what component is consuming sensor data, different levels of interaction between the data are implemented. Both of the proposed approaches were integrated in the framework presented in Figure 3.1 in Chapter 3.1.

In the measurement-level [69](sometimes referred to as low-level) mode, the detection points and host state from *D2* are used directly for building the grid and storing it in *D3.2*. For the low-level functionality, buffers are implemented to accommodate sensor data, along with prediction methods for the host state and measurements to synchronize buffered data prior to fusion. The main disadvantage of this method is increased difficulty in implementation of scalable architecture, however, the benefits are found in reduced memory usage.

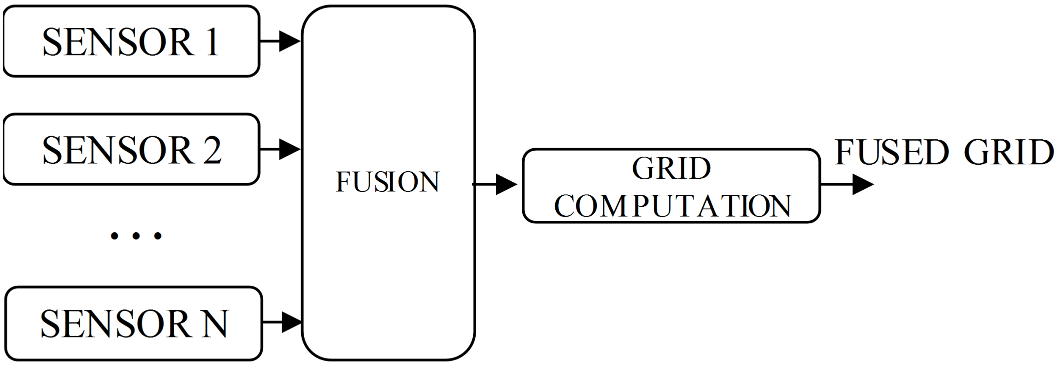


Figure 2.20: Low-level fusion.

In the grid-level fusion, an additional container to store grids for each of the domains is represented by *D3.1*, and then the grids are fused. For all high-level variants, intermediate grids are created for each sensor domain. Table 3.1 in Section 3.4 contains the description of the variants and methods that were used in the virtual validation process. From the automotive perspective, the additional data needed for intermediate grids can be a highly impacting factor as it imposes the requirement for additional memory for sensor grids, which in production application directly impacts the cost of the proposed solution. The advantage of this method is that computation can be distributed through various embedded systems, therefore, distributing the computational load.

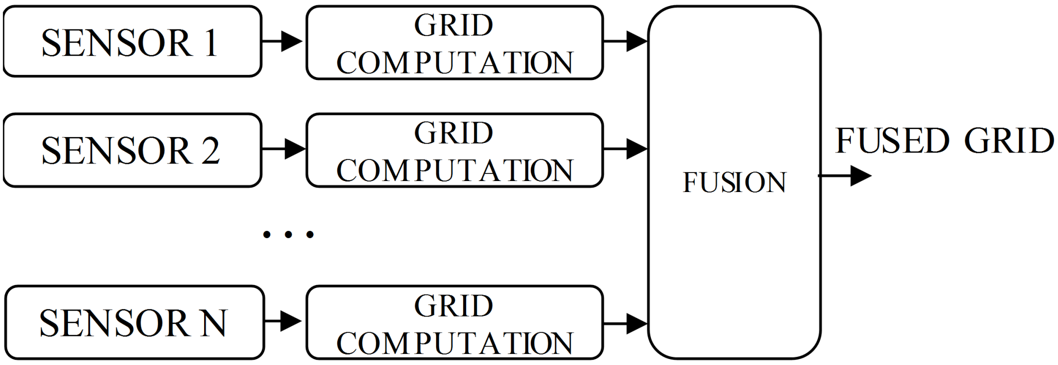


Figure 2.21: Grid-level fusion.

The sensor domain grid can utilize different fusion methods than the grid level fusion algorithms that are executed later. For all high-level algorithms in this evaluation Bayesian method was used except the Dempster-Shafer method.

2.3.3. Probability Fusion Methods

The second component of the mapping occupancy probability is the update and fusion method. The fusion methodology can be diversified in three ways:

1. The first diversification comes from the architecture constraints. The proposed architectures are discussed in depth in further sections. The proposed architectures are presented in Figure 2.21 as High Level and Figure 2.20 as Low Level.
2. The second diversification comes from the mathematical method used for the calculation of the output probability of occupancy. In grid-level fusion, the update method is distinguished from fusion, as it is used to create a domain (sensor) grid. Then, the domain grids are fused together, in which case different methods can be used for building the domain and different methods for the fused grid. This subsection presents the details of the methods identified in the literature and implemented in the framework and used for testing[69].
3. The third diversification originates from the Occupancy Grid Fusion Architecture used in a particular approach. The variants are described in Chapter 2.3.2. The selection of the architecture depends on the use of time filtering in building the input grid or the use of instantaneous (single snapshot of one set of measurements, e.g. a list of radar detections from one scan) grid. This approach applies only to the High-Level approach, where separate grids are created for sensors or domains. A low-Level grid, by nature, is a time-filtered grid.

A Detailed Description of the fusion methods used in this study is given below.

- Bayesian Inference Filter

  The default approach for filtering used in this thesis case is the Bayes filter since the occupancy grid in most prior works is based on Bayesian probability theory. The Bayesian filter is an extension of the Bayes estimator applied in cases where the observed values change over time[7]. Originally, the method found application in the field of radar tracking to estimate beliefs about the recent position of targets[8]. It's general form is presented in Equation (2.11), where the $p(m_t|z_{1:t}, x_{1:t})$ is the posterior probability of occupancy given $z_{1:t}$ measurements and $x_{1:t}$ host poses on the grid.

$$p(m_t|z_{1:t}, x_{1:t}) = \frac{p(z_t|m_t, x_{1:t}, z_{1:t-1}) \cdot p(m_t|z_{1:t-1}, x_{1:t-1})}{p(z_t|z_{1:t-1}, x_{1:t-1})} \tag{2.11}$$

However, with respect to future optimized software implementations, a logarithmic approach presented by Galvez is used where the given probability value $p(m)$ is converted to the log-odds representation $l(m)$ $l(m)$ by means of equation (2.12) [69].

$$l(m) = log\frac{p(m)}{1 - p(m)} \tag{2.12}$$

With the log-odds representation of both $l(m_1)$ inverse sensor model output and$l(m_2)$ the occupancy grid, fusion is performed by summing the log-odds values. The operation is presented in Equation (2.13).

$$l(m) = l(m_2) + l(m_1) \tag{2.13}$$

The Bayes filter is then applied to every updated cell in the occupancy grid in the form of a log-odds ratio, which corresponds to performing a recursive Binary Bayes Filtering using the probability from the ISM. This approach tends to increase the a priori probability when the values exceed a 0.5 threshold and performs oppositely for values below that threshold. The issue is a correct effect from the statistical point of view, but it is causing problems in real applications of the methods, it is “deadlocking" in the case of one input (grid cell or inverse sensor model output) reaching 0 or 1. To prevent this situation, a minimum and maximum probability saturation are set for the complete grid by a tunable parameter. This parameter is used to offset the minimum and maximum values from 0 to for instance 0.1 and from 1 to 0.9.

A graphical representation of resultant probability given the input probabilities from sensors is presented in Figure 2.22. It can be observed that both sensors contribute to the resultant output probability equally with a tendency to amplify the probabilities in case of sensors consensus.

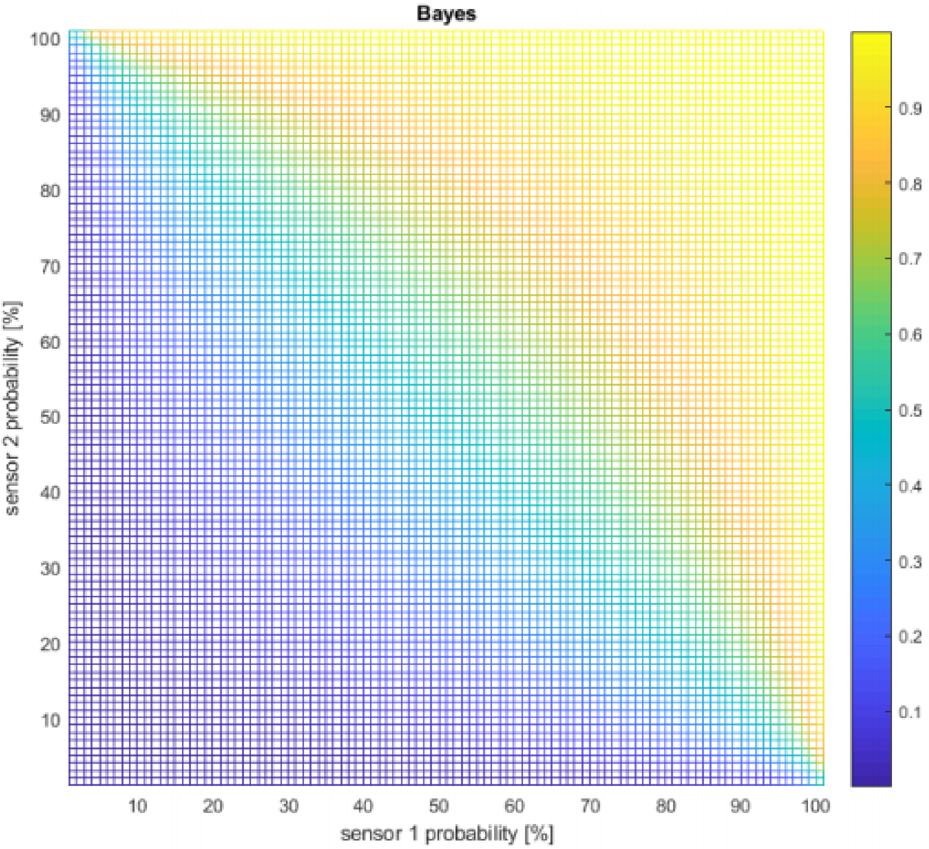


Figure 2.22: Results of Bayesian fusion for two input sensors.

- Dempster-Shafer

  The Dempster-Shafer evidence theory is the second major mathematical apparatus employed in the task of building the occupancy grid. This method finds application in the area of tracking paths of moving targets for the estimation of both beliefs of existence and non-existence of false targets[8]. The primary advantage of this method over the Bayesian approach is the capability for explicit handling of the absence of information, e.g. an unknown state. The second crucial feature is the ability of the method to identify conflicting measurement information[92]. These advantages come with a cost that requires the definition of more than just the occupancy probability grid. Additional layers are needed to represent probability masses. Two mass matrices are needed - occupied $m(O)$ and empty $m(E)$

  The basis for computing the cell probability is the Dempster-Shafer combination equation (2.14) where m_1 and m_2 the two sources of information. The information sources are fused with a belief function $m_{12}(X)$ of X proposition.

$$m_{12}(X) = (m_1 \oplus m_2)(X) = \begin{cases} \frac{(m_1 \cap m_2)(X)}{1-(m_1 \cap m_2)(\emptyset)} & for X \neq \emptyset \\ \\ 0 & for X = \emptyset \end{cases} \tag{2.14}$$

where:

$$(m_1 \cap m_2)(X) = \sum_{A,B \in 2^\Theta | A \cap B = X} w_1 m_1(A) \cdot w_2 m_2(B) \tag{2.15}$$

and the conversion of masses to probability is calculated by:

$$P(A) = \sum_{X \in 2^\Theta} m(X) \frac{|A \cap X|}{|X|} \tag{2.16}$$

Where $m(A)$ and $m(B)$ are Basic Probability Assignment (BPA) functions of mass A and B.

The Dempster-Shafer combination formula applied to the occupancy grid gives the following formulas:

$$m_{12}(O) = \frac{m_1(O)m_2(O) + m_1(O)m_2(\{O,E\}) + m_1(\{O,E\})m_2\{O\}}{1 - m_1(E)m_2(O) - m_1(O)m_2(E)} \tag{2.17}$$

$$m_{12}(E) = \frac{m_1(E)m_2(E) + m_1(E)m_2(\{O,E\}) + m_1(\{O,E\})m_2\{E\}}{1 - m_1(E)m_2(O) - m_1(O)m_2(E)} \tag{2.18}$$

Equations provide a final probability value obtained by the following approach:

$$P(O) = m(O) + \frac{m(\{O,E\})}{2} \tag{2.19}$$

A graphical representation of resultant probability given the input probabilities from sensors is presented in Figure 2.23. It can be observed what happens with the population of sensors probability through masses to the resultant fusion output probability.

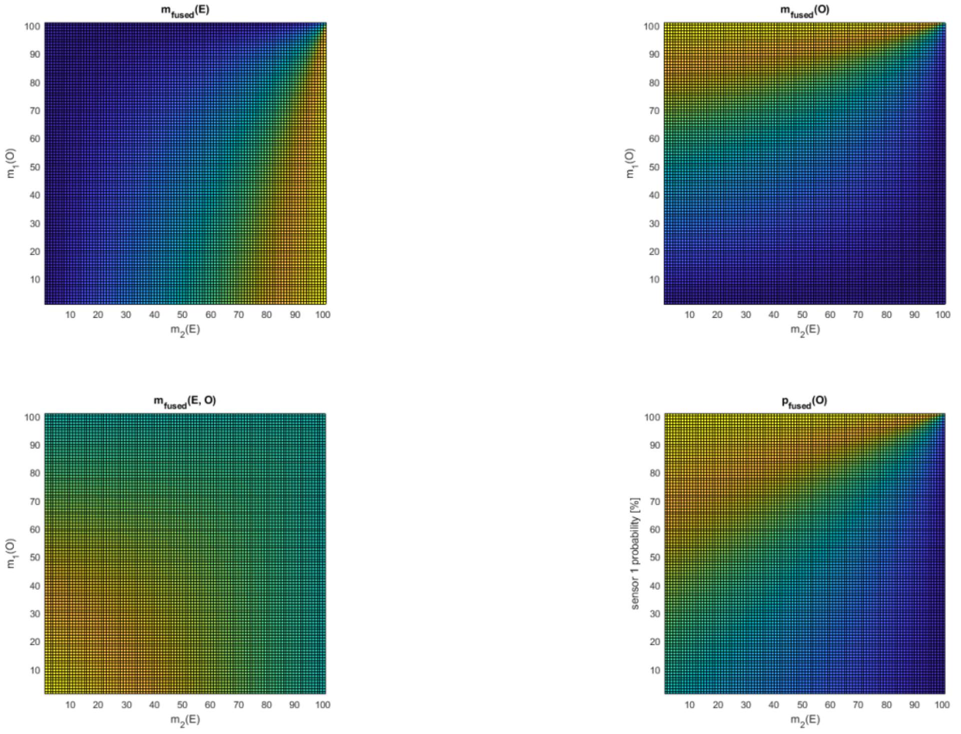


Figure 2.23: Results of Dempster-Shafer mass fusion.

- Linear Opinion Pool

  The method identified in the literature [69] allows for the resolution of conflicting sensor information [4] based on an opinion pooling methodology[21]. An example application of the family of opinion pooling methods is pooling opinions from individuals for drawing conclusions (such as expert opinions)[11].

$$p_c = \alpha \sum_i w_i p_i \tag{2.20}$$

  This method assumes the existence of a weight w_i that can represent the contribution of certain sensors which delivers probability input p_i. [4].

- Independent Opinion Pool (IOP)

  A method that is based on the assumption that the sources are considered to be independent is a modification of the IOP.

$$p_c = a \prod_i p_i \tag{2.21}$$

$$a = (\prod_i p_i + \prod_i (1 - p_i))^{-1} \tag{2.22}$$

The presented modification results in a geometric mean representation of the problem[69] [60].

$$\alpha = [\sum_i w_i]^{-1} \tag{2.23}$$

A graphical representation of resultant probability given the input probabilities from sensors is presented in Figure 2.24. It can be observed that the probabilities are amplified in the case of sensor consensus however not in an as strong manner as in the Bayesian approach.

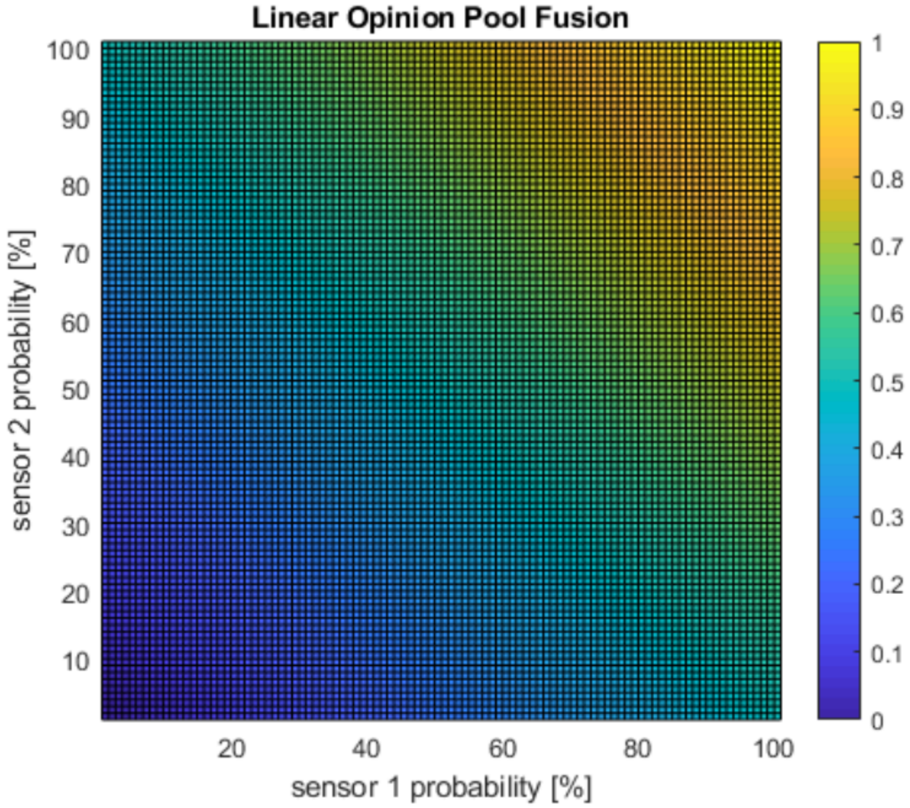


Figure 2.24: Results of Linear Opinion Pool Fusion with equal weights.

- Logarithmic Independent Opinion Pool

  Another generalization of prior methods is a weighting process on top of probability distributions, which comes as a result of splitting the process into two steps: first, a weighted average, and second, a geometrically weighted average[69].

$$p_c = \alpha \prod_i p_i^{w_i} \tag{2.24}$$

$$\alpha = [\prod_i p_i^{w_i} + \prod_i (1 - p_i)^{w_i}]^{-1} \tag{2.25}$$

A graphical representation of resultant probability given the input probabilities from sensors is presented in Figure 2.25. It can be observed that the probabilities are amplified in the case of sensor consensus exactly in the same way as in the Bayesian approach. This method in the case of the same sensor contribution weight behaves in exactly the same manner as the Bayesian approach.

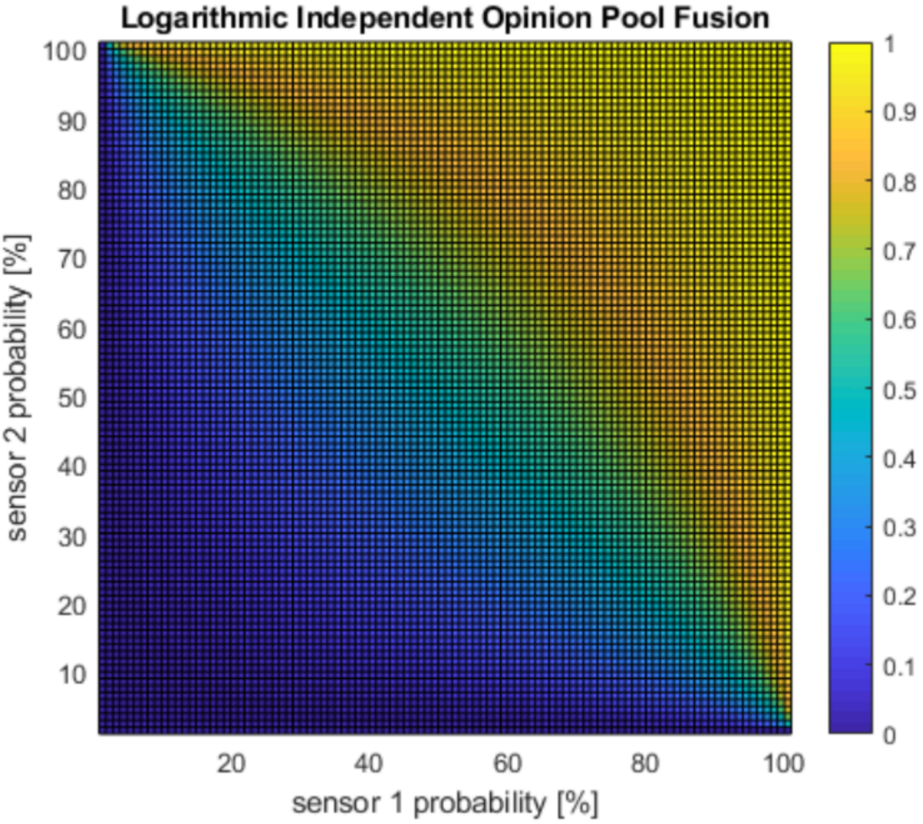


Figure 2.25: Results of Logarithmic Independent Opinion Pool Fusion with equal weights.

- De Morgan′s Law

  Another approach proposed for fusing grids or measurement results inside grids is proposed by Galvez and has a tendency to favor higher probabilities[69]. This method has applications in text searching and it can be applied to simplify Boolean equations to the use of only AND and NOR gates.

$$p_c = 1 - \prod_i (1 - p_i) \tag{2.26}$$

A graphical representation of resultant probability given the input probabilities from sensors is presented in Figure 2.26. It can be observed that the probabilities are added resulting in a much faster reach of sensor consensus, even with contradicting input information. Thus the method does not address the sensor conflict resolution, rather it results in the override of the pessimistic sensor with data from an optimistic one.

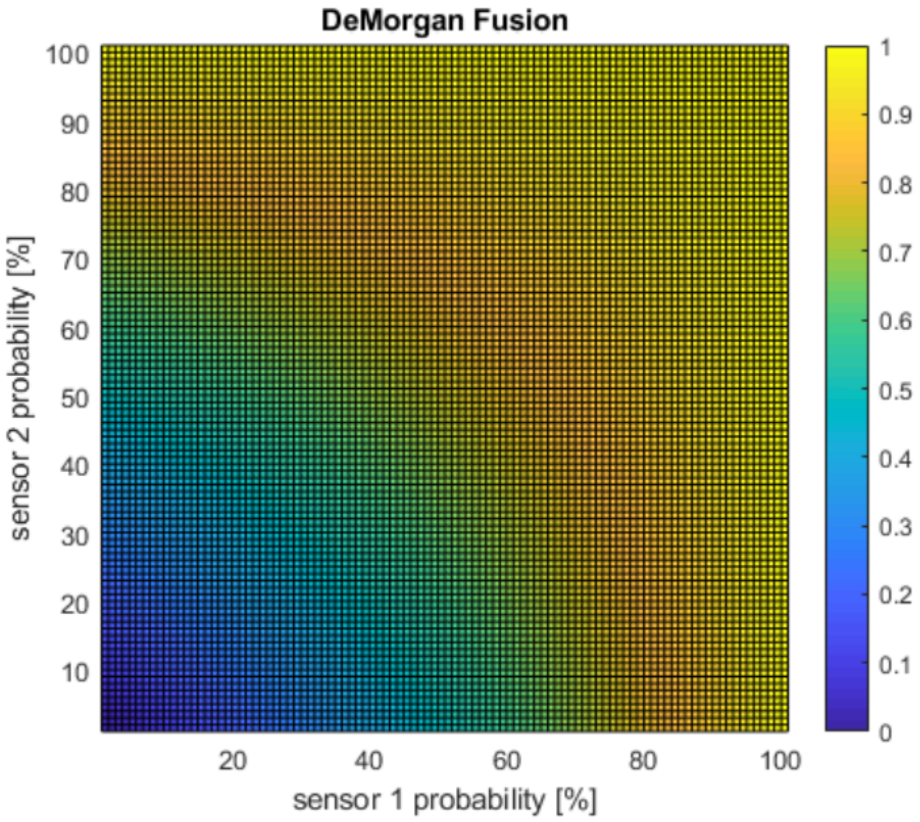


Figure 2.26: Results of DeMorgan based Fusion.

- Maximum Policy

  The simplest of all approaches proposed in the literature [81] assumes a conservative approach for taking the highest values of each cell[69].

$$p_c = max_i(p_i) \tag{2.27}$$

A graphical representation of resultant probability given the input probabilities from sensors is presented in Figure 2.27. It can be observed that this method treats sensors equally even in case of contradicting information therefore conflict resolution capabilities are limited. This is the most conservative method ensuring that in case of any sensor reporting occupied space the resultant grid will be occupied. Those characteristics might lead to an increased number of False Positives (FP) in case of the higher sensitivity of one of the sensors.

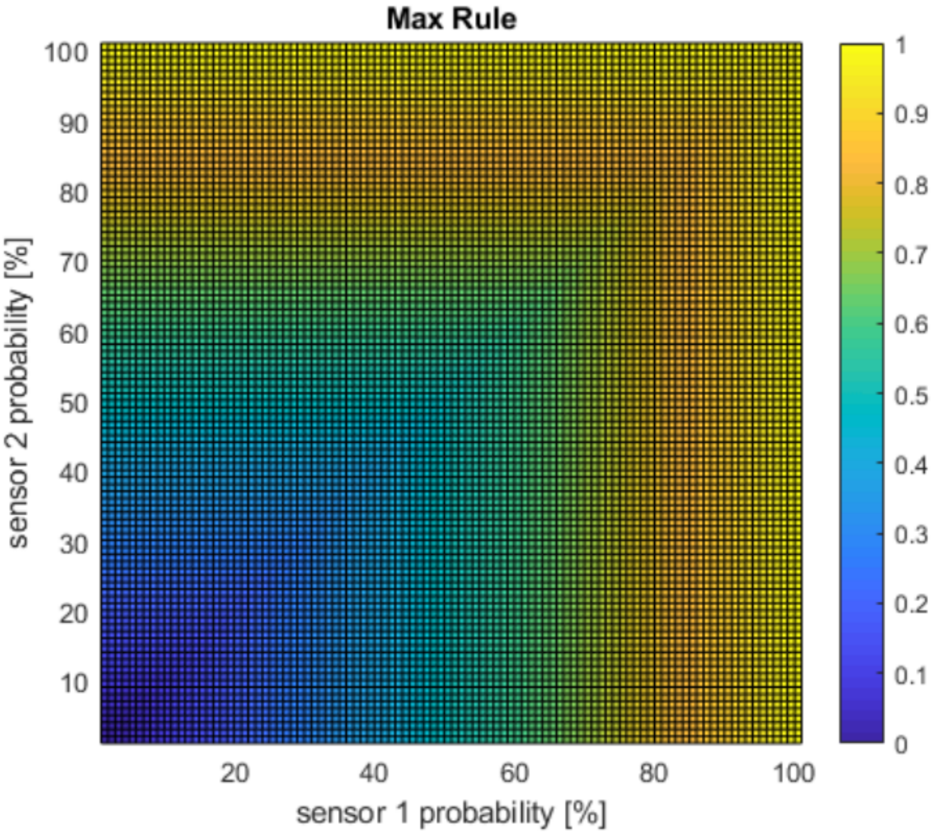


Figure 2.27: Results of Maximum policy based Fusion.

An in-depth look into the lookup table-based comparison of the methods can be found in the work of Galvez[69].

2.4. Virtual Validation Application

The rapid growth of the market for Advanced Safety systems has created a need to enable rapid prototyping and simulation of life-like events involving computer models of traffic participants. Currently, there is a big portfolio of commercial applications allowing the creation end execution of simulated test cases. Those existing tools satisfy the need to perform multiple iterations of test scenarios in perfectly reproducible and controlled virtual conditions[78]. This factor allows the development of the software algorithms at the early stages of prototyping using the software-in-the-loop simulation method[37]. Virtual validation software for the automotive industry supports physics engines for modeling the vehicle dynamics along with complex graphical engines to produce realistic-looking renderings of the environment[90]. The graphical engines are utilized for the generation of the simulated sensor data as well as reference data to be used for performance assessment. This is another major advantage over testing in real-world conditions in the early stage as instrumentation of the test vehicles and assessment platforms are costly and sometimes for early development stages introduces additional overhead making it more difficult to spot potential faults in the concept[77].

Major deficiencies of the virtual validation are primarily related to the low realism of the produced scenarios and results, therefore it cannot be considered as the only way of testing and still is followed up by real-world evaluation[47], however, utilization of the virtual approach provides competitive advantages in prototyping concepts, especially for prototypes of solutions requiring integration of multiple hardware and software subsystems that the integrator does not have full control over (i.e. integration of sensors or actuators from different provides, or generic commercial software components).

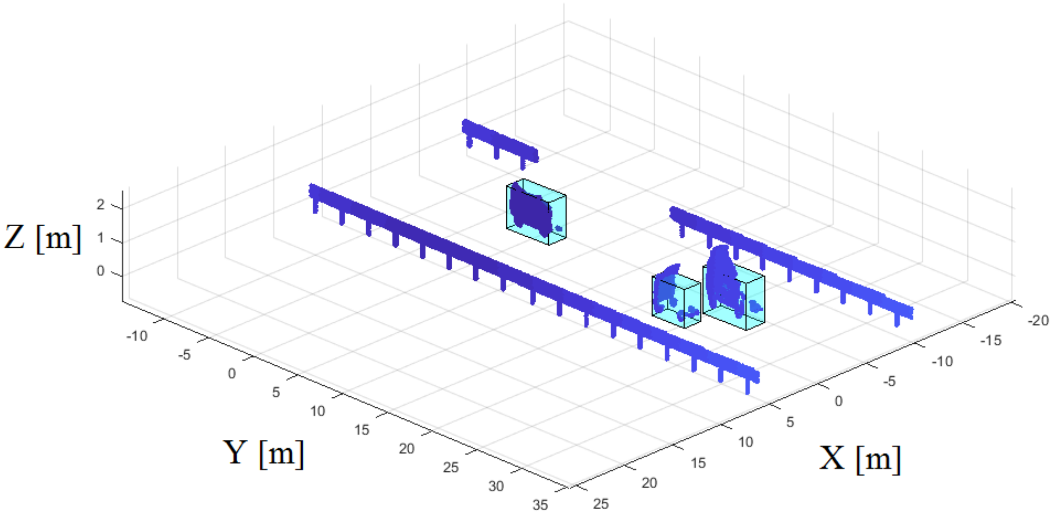
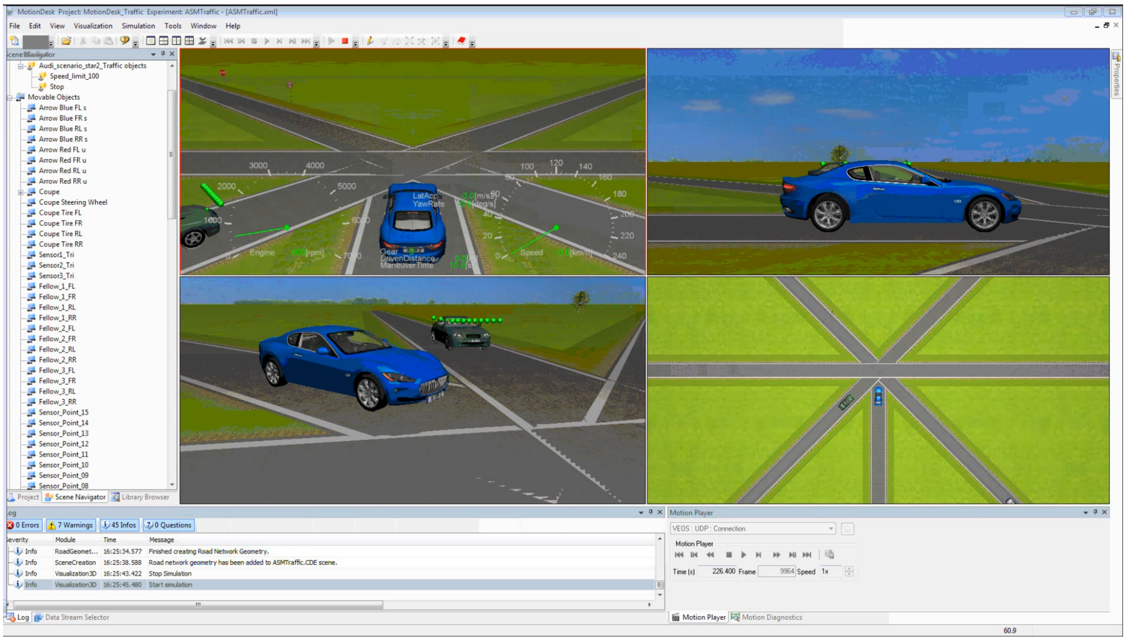


Figure 2.28: Data with points representing barriers and vehicles surrounded by bounding boxes.

Many commercial virtual validation tools allow integration in a co-simulation environment with other prototyping software and simulation platforms (such as Matlab/Simulink)[1] to allow evaluation of complex custom build mechatronic models[37]. The simulation solutions also provide capabilities to simulate simplified perception algorithms extending the utility of such simulation software even further for system integrators.

This research utilized IPG CarMaker software to conduct the virtual experiments. The novel process in which virtual validation was employed for the research purpose is described in Section 3.1. The process utilized dynamic objects state information, a point cloud representing the environment (computed on a GPU using ray tracing), host vehicle motion and position, and parameters describing road curvature. A snapshot of the utilized data contained in *D1* describing the process of simulation from Figure 3.1 is presented in Figure 2.28.



Severity	Module	Time	Message
Info	RoadGeomet...	16:25:34.577	Finished creating Road Network Geometry.
Info	SceneCreation	16:25:38.588	Road network geometry has been added to ASMTraffic.CDE scene.
Info	Visualization3D	16:25:43.422	Stop Simulation
Info	Visualization3D	16:25:45.480	Start simulation

Figure 2.29: Screenshot of an alternative tool for simulation of traffic scenarios available on the market (dSpace Motion Desk).

One of the main things in the course of the product development is to verify whether all the requirements that were posed to it at the beginning of the development process are met. For this purpose, it is necessary to carry out tests during the development of the product. In the case of AS and ADAS products, the most important are tests in the actual road traffic, so-called Real World User Profile (RWUP) tests[77]. Of course, in the initial stages of the project, this is a ride in the so-called open-loop settings, which is the situation when the system is installed in a car, but its outputs are not set on the actuators, and they are only logged[32]. Logging is done using appropriate software, which automatically divides the data into, fixed-length portions of logs. Closed-loop tests are conducted in a controlled environment on a dedicated test field area[77].

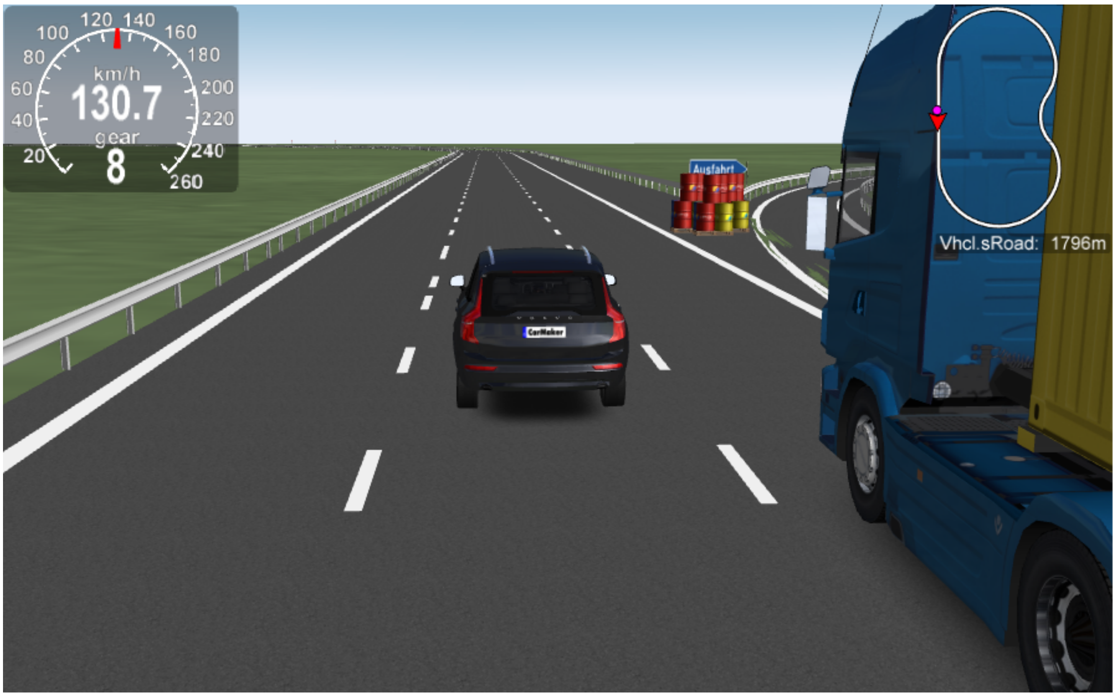


Figure 2.30: Snapshot of the junction, exit sign, and impact attenuating barrels.

2.5. Real World Testing

As stated in the previous section, Virtual Validation is one of the steps in the development process. In the course of the technology getting mature real-world testing is introduced. The industry refers to these types of tests as RWUP[77]. Real-world testing requires the preparation of the prototyping setup installed in the vehicle. The conducted real world tests and vehicle instrumentation are a part of the original contribution performed in the course of this research.

The test setup shown in Figure 2.31 consists of two industrial computers equipped with i7-6700T processors and 64GB of . The Computers operate Robot Operating System (ROS) Melodic and Kinetic instances used for the evaluation. The computers run Linux 18.04 which was used for a complete evaluation. Two PCs were used in the evaluation - one is the ROS Perception computer which runs data parsers to extract information from vehicles and sensors buses. Data is also utilized live in the ROS Perception computer to execute the tested algorithm. The data output from algorithms and raw sensors are then fed over Ethernet to the ROS Logging and Visualization computer. The second computer logs raw data for further reprocessing along with the

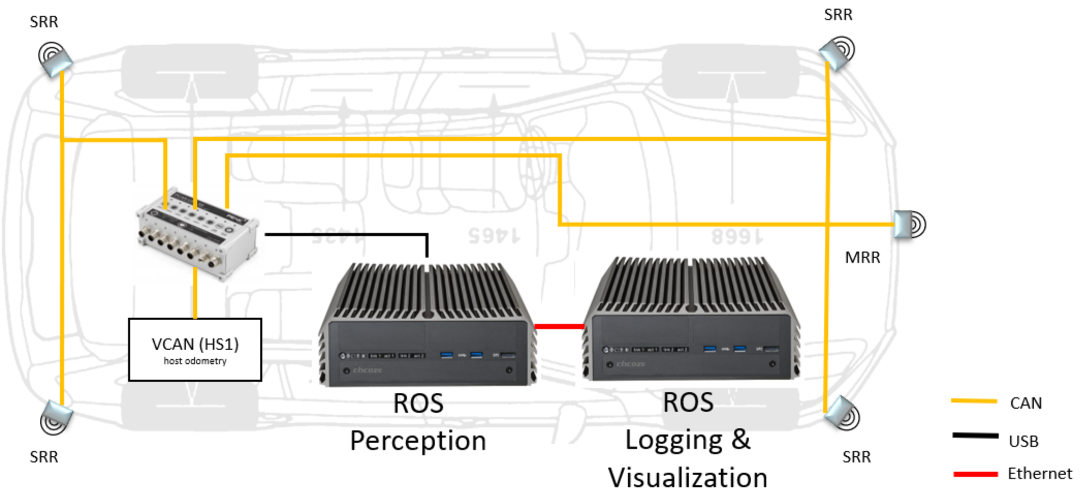


Figure 2.31: Layout of systems installed in the test vehicle used for experimental validation, Medium Range Radar (MRR), Short Range Radar (SRR)

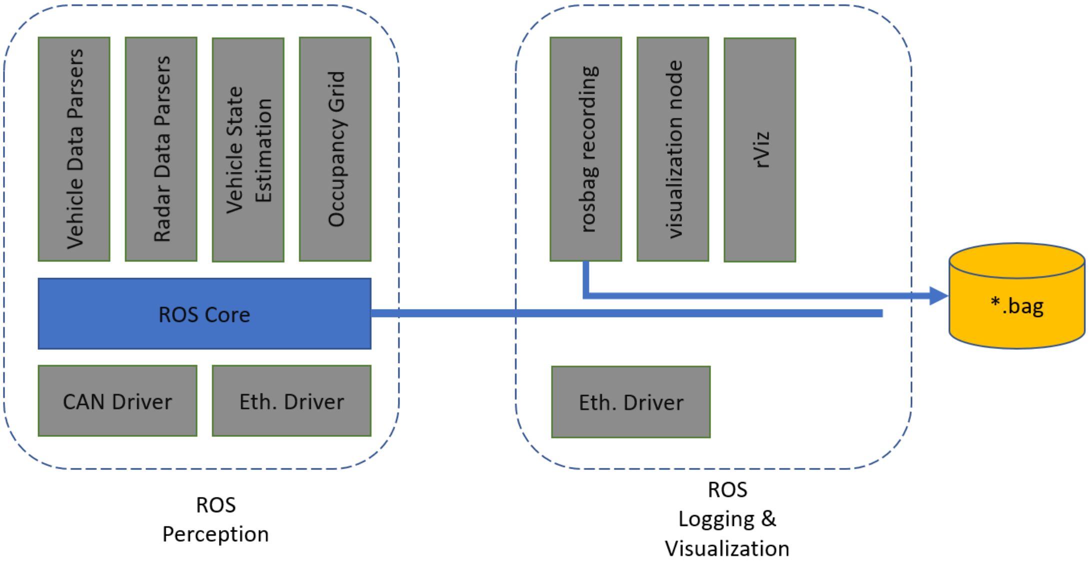


Figure 2.32: Architecture of the software implementation

algorithm results for evaluation.

A crucial component for real-world testing is data visualization. This component allows displaying debug information as well as details on the operation of the algorithm for the logging technician/vehicle operator. Multiple different frameworks exist in the industry (including commercial and open-source). The frameworks allow visualization of all kinds of perception information delivered by in-vehicle systems (such as bounding box representations, grids, raw video streams from cameras, and 3D renderings of the

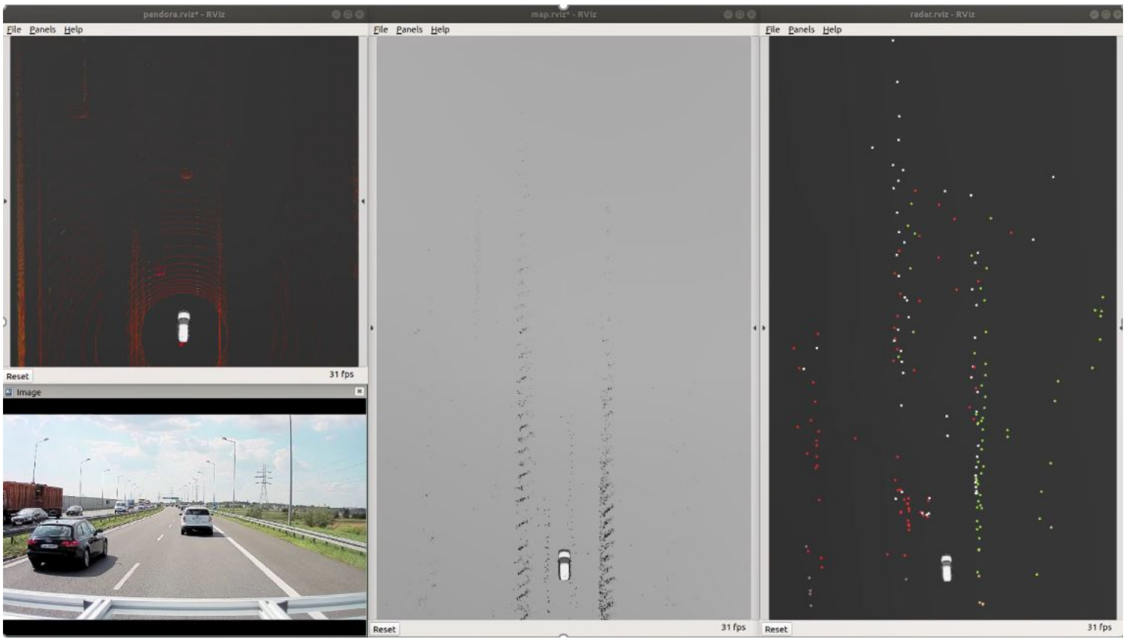


Figure 2.33: Screenshot of the rViz visualization configuration

environmental models)[45]. A snapshot of the visualization user interface implemented in rViz is shown in Figure 2.33. The visualization includes raw radar detection points from all the sensors, the output of the grid algorithm, and reference information. The software architecture of the setup is presented in Figure 2.32. The presented architecture shows a shared ROS core used to channel the information over Ethernet between the two computers.

Sensor data utilized for experimental validation comes from sensors installed on the test vehicle. The details of the sensors are discussed in Section 4.3. The utilized sensors communicate the measured information over Controller Area Network (CAN) interface therefore the vehicle setup is equipped with the interface to acquire CAN data. Peak P-CAN device was used in this case. The interface also acquires the data from the vehicle bus to extract information such as wheel speeds, built-in IMU signals, estimated velocities, and generic vehicle state information. The "PCAN" interface data is connected over USB to the perception computer. The setup allowed real-time sensor data collection and algorithm execution.

The test vehicle used for experiments is a Ford Explorer presented in the 2.35. SUV platform is chosen to accommodate more space for the test setup installation and easy access as the early prototyping setup consumes large amounts of power and requires

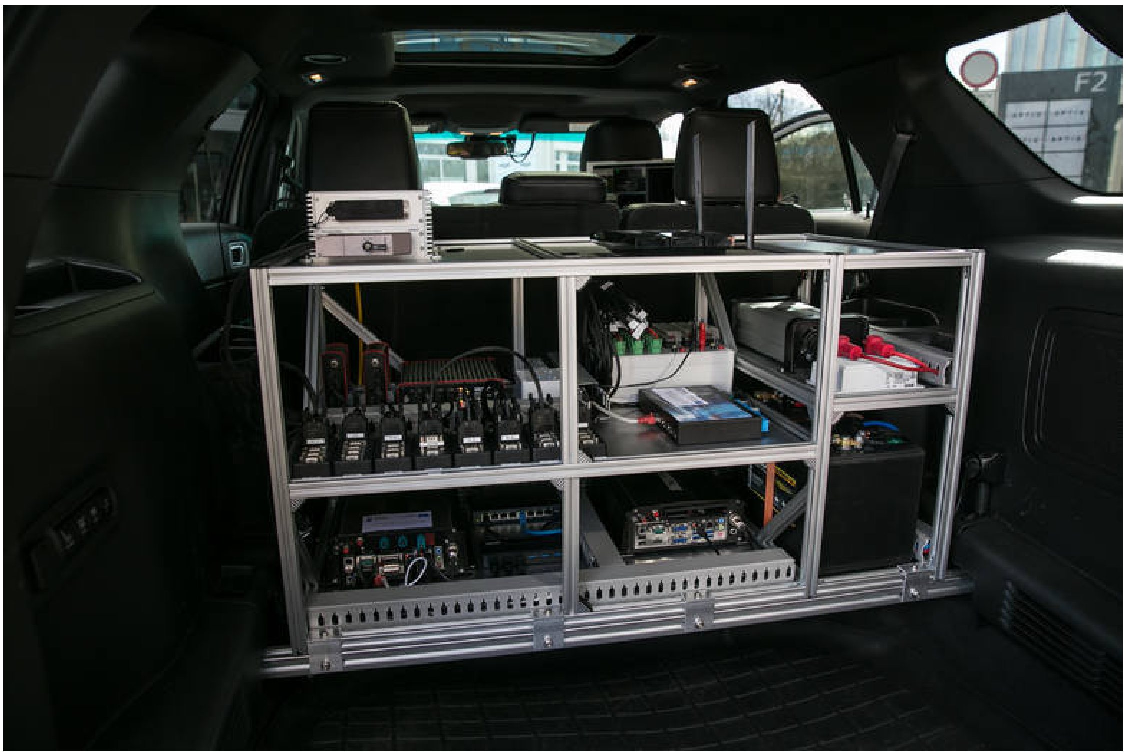


Figure 2.34: Photo of the test equipment installed in the test vehicle

Figure 2.35: Photo of the test vehicle

redundant batteries in the vehicle along with electronic separation between the original vehicle installation. The components of the test system installed in the vehicle's trunk are presented in Figure 2.34.

The application of the test platform to the scenarios will be discussed in Section 4.3 with the description of the procedure used in the real testing.

ROS platform has been chosen as a good option for a quick buildup of multi-module software prototypes due to the fact that it allows quick and easy definition of the interface connecting the building blocks. ROS also provides multiple built-in interfaces and methods for visualization and logging allowing to focus on the algorithms and architecture development[5]. This OS is used in a prototyping stage and for a further move to higher technology readiness level will be substituted by a safe operating system such as Autosar or QNX.

2.6. Hardware in The Loop

Since the beginning of the development of automotive testing methods described as HIL were utilized to cover for needs of unit testing of systems such as Engine Control Module (ECM). The primary reason behind the utilization of this method was to isolate the SUT from the random environment and substitute random environment inputs with controlled signals [39]. This step allowed for repeat-ability conditions and full control over tests. That method also allowed to limit the amount of time spent in system-level tests in real-world conditions thus reducing the development costs[39]. Variations of these tests exist in the industry, where only part of the signals are utilized on SUT resulting in an incomplete testing loop, rather focusing on data replay to the system and evaluation of the responses. The levels on which the SUT is isolated for testing impact the implementation. The system can be isolated only on the level of System On Chip (SOC) resulting in a Processor In The Loop testing where just the compute portion is tested. However, a high level of SUT definition can also be found where the complete vehicle system is defined as a "plant" for end to end testing purposes[9].

On top of the Virtual Validation utilized in the project, there is another efficient approach used to address the topic of checking multiple variants and versions of the implemented software without the need to conduct multiple test drives. The HIL

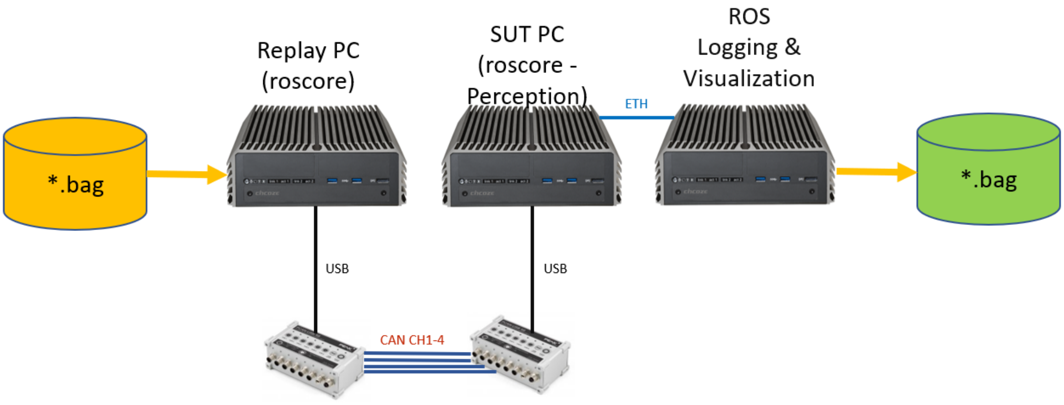


Figure 2.36: HIL setup used for data reprocessing

approach is a commonly employed method of testing and verification in the industry[77]. The HIL setup built for the purpose of this work consists of three computers: Replay PC, SUT (System Under Test), and Logging and Visualization. The SUT and Logging and Visualization setup replicate the setup installed in the vehicle as described in the previous chapter. The Replay PC is set up in a way that allows it to mock up the real vehicle and sensor buses. That type of setup can utilize logs collected during test drives. The collected logs are then replayed to the SUT system. This method allows full repeatability of scenario and input behavior. The system response is then logged in the same manner as in the vehicle - to be used for assessment and evaluation. In further stages of the project along with the migration of the computing platform to embedded systems similar setup can be used for working with those solutions. There are formal requirements related to safety-critical systems that require utilization of Hardware in The Loop for part of the testing process[40]. Here it is utilized to facilitate rapid development in repetitive conditions that would not be possible to achieve in the real world, however, is based on real data. The HIL type systems are commonly used as part of Continuous Development/Continuous Integration/Change Management environments where the SUT is automatically updated with every software release and undergoes automatic testing through a complete test loop. The results of those tests are then utilized to provide feedback information on the tested change-set if it is functional and can be integrated into the official software release thus if it can be integrated into the main branch of development code to the final release. Those systems incorporate automated re-flashing of the tested ECUs and automated Key Performance Indicator (KPI) calculation resulting in a complete performance report for each software change introduced to the baseline. The main disadvantage of this method is that the automation process is time-consuming and can utilize up to 10x more resources than manual testing.

Potential variation of this system to be explored utilizes the connection of the virtual validation and hardware in the loop system in which data from the simulation model is generated in real-time and then streamed to the SUT, and the response of the system is utilized to control the model behavior fully closing the loop. This approach allows testing and development of closed-loop control systems that are utilized for steering the actuators in the plant[77].

2.7. Key Performance Indicators

A set of KPI are selected to complete the data-driven performance evaluation of the researched algorithms. Further sections describe the proposed novel process of ground truth (reference) data generation that is used to compute the presented KPIs. In the process of performance assessment each cell of the reference and tested grid is compared to the fused occupancy grid cells (the 1-D simplification is presented in Figure 3.13). The number of cells resulting with corresponding events is summed up for a scenario [14]. Then the data are used for analysis with the following KPI methods:

- Probability of detection

  This indicator corresponds to the True Positive Rate (TPR) produced by the algorithm, and its increase means that the analyzed method is better at detecting actual obstacles, and therefore is expected to be maximized.

  $$TPR = \frac{TP}{P} = \frac{TP}{TP + FN} \tag{2.28}$$

- Precision

  The factor representing Positive Predictive Value (PPV) indicates how good the method is at judging whether the reported occupied areas are actually occupied.

This factor is therefore expected to be maximized for the best performance.

$$PPV = \frac{TP}{TP + FP} \tag{2.29}$$

- Probability Of False Alarm

  The False Positive Rate (FPR) shows how bad the method is at the estimation of a certain cell as free when it is occupied in the scope of all those events and actually free cells in the experiment. For better performance, this factor has to be minimized, since the severity of false positives are significant in the Advanced Driver Assistance Systems from a functional safety perspective [40].

$$FPR = \frac{FP}{N} = \frac{FP}{FP + TN} \tag{2.30}$$

- False Discovery Rate (FDR)

  The FDR describes a ratio of how many cells identified as occupied were falsely classified. In terms of a better-performing method, this factor must be minimized.

$$FDR = \frac{FP}{FP + TP} \tag{2.31}$$

- F1 Score

  F1 Score is also known as a measure of accuracy, or a harmonic mean of precision and accuracy. The higher the value of this factor, the better the performance of the method.

$$F1 = 2\frac{PPV \cdot TPR}{PPV + TPR} = \frac{2TP}{TP = TN + FP + FN} \tag{2.32}$$

- Normalized Occupied Map Score (NOMS)

  The NOMS method is a variation of the method used for grid performance evaluation proposed by Colleens et al[16]. This method relies on non-binary algorithm outputs (a grid composed of cells with a probability gradient) A and a filtered reference grid B - obtained by subjecting a binary reference grid to spatial filtering that generates gradients of probabilities. The maximization of the score for this method indicates higher performance.

$$NOMS = 2^{\sum_i [\log_2(A_i B_i + \bar{A}_i \bar{B}_i)]} \tag{2.33}$$

2.8. Test Scenarios and Development Process

This section looks into the test scenarios designed for the research that was executed both in simulation and the real world. The flow of the requirements behind the development is presented.

2.8.1. Development Process

The development of any automotive component requires compliance with the rigorous standards for the productization of the solutions. The flow of the information and the development cycle is presented in Figure 2.37 from Tierno et. all [82] article on automotive development and testing. The process of the development starts with the requirements definition. This step in the case of lower Technology Readiness Level (TRL) is an important step where requirements need to be defined rather than derived from direct customer needs. This imposes a need for iterations of this step in research projects such as this to adapt the requirements and extend their definition[43][31] as the project progresses. That process is facilitated by virtual methods as it allows shorter iteration loops between the prototyping stages.

As shown in the model, the requirements are then converted into system design in the process of System Analysis, the result for that step is then carried over to the software design, module design, and then implemented. Along with the development, stages go the validation steps for each of the levels of the design. The full completion of the V-Model along with adherence to the Software Process Improvement and Capability dEtermination (SPICE) allows to track and verify completion of requirements, review, implementation, testing, and traceability. Technology that is defined in the early stages of prototyping in a way compliant with the process allows easy integration into products and fulfilling rigid requirements related to safety[38].

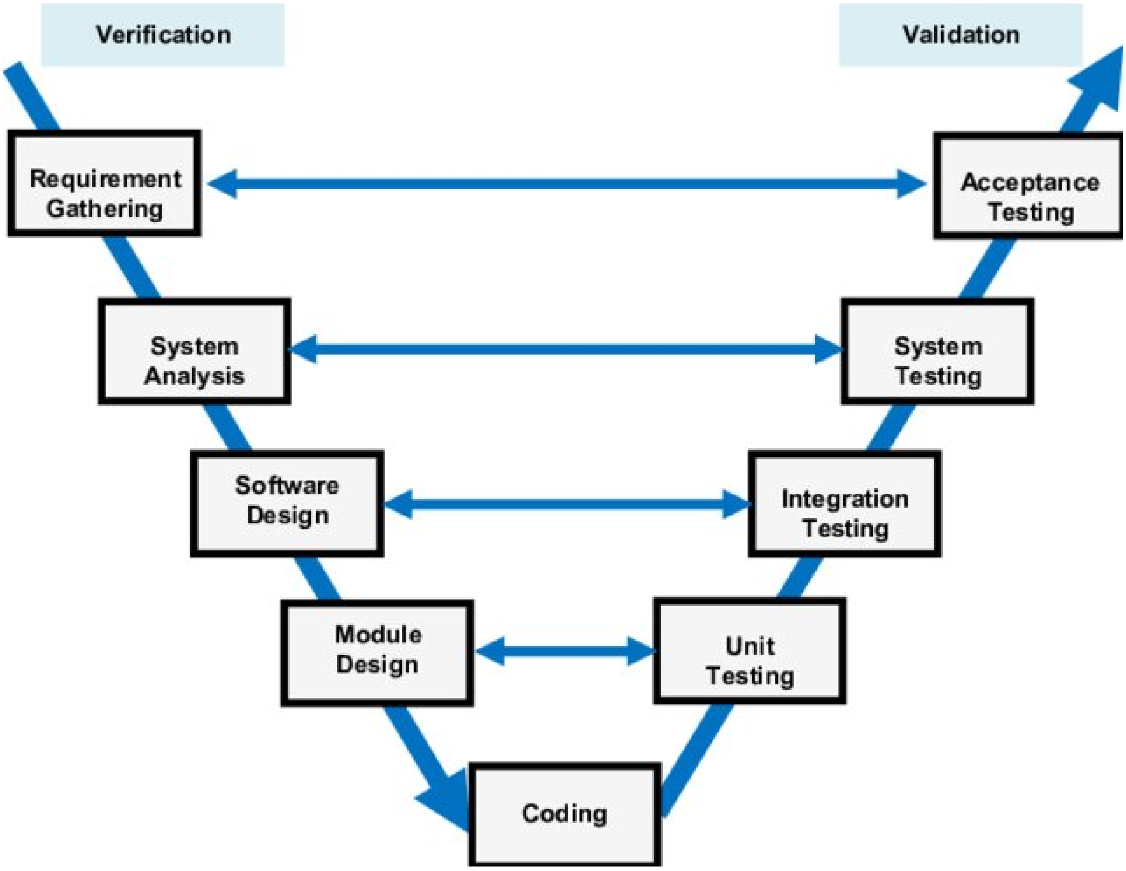


Figure 2.37: V-Model process from Tierno et. all [82] article on automotive development and testing

2.8.2. The Use of Key Performance Indicators for Fused Occupancy Grid Assessment

For basic performance assessment, the fused grid *D3.2* is binarized with a fixed threshold and spatially aligned with the ground truth map *D4*. The results of the comparison are classified as True Positive (TP), True Negative (TN), False Positives (FP), and False Negative (FN). The set of chosen indicators purposely does not include the methods relying heavily on true negatives due to the sparse nature of the grids. These kinds of indicators were identified as having a bias on the results.

2.8.3. Scenario for Virtual Assessment

The virtual experiments performed in this thesis are based on two test scenarios. Both scenarios were executed using the same virtual model (presented in Figure 2.38)

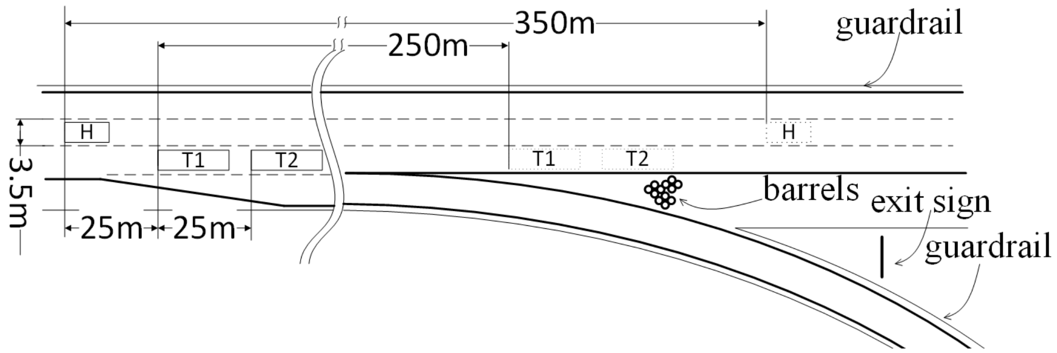


Figure 2.38: Schematics of the scenario modeled in the environment (H – host vehicle, T1 T2 – target vehicles, solid line indicates initial and dotted final positions).

of a straight section of the highway, bounded by guardrails on both sides, followed by an exit junction, with an exit sign and impact attenuating barrels. The test scenario models are shown in Figures 2.41 and 2.30. Both scenarios assumed an approximately constant velocity of 126km/h for the host vehicle along a straight line in the center lane. Scenario number 1 was executed with an empty road that did not contain any traffic participants except the host vehicle. Scenario number 2 contained one light and one heavy commercial vehicle in the right lane (Figure 2.40) with an approximately constant velocity of 90km/h. In both scenarios, the host vehicle traveled 350m in the duration of 10s.

The set of sensors used in the modeling is shown in Figure 2.39. The virtual test vehicle was equipped with six short-range radars - four on the corners of the vehicle, two on the sides. The front of the vehicle was equipped with a forward-looking radar and forward-looking lidar. The primary driver behind the requirements is related to a sensor suite applicable for potential system design satisfying complex highway use-cases with redundant overlapping fields of view of the sensors in the front of the vehicle. Side radars allow additional perception capabilities for adjacent lanes, barriers, and road delimiters.

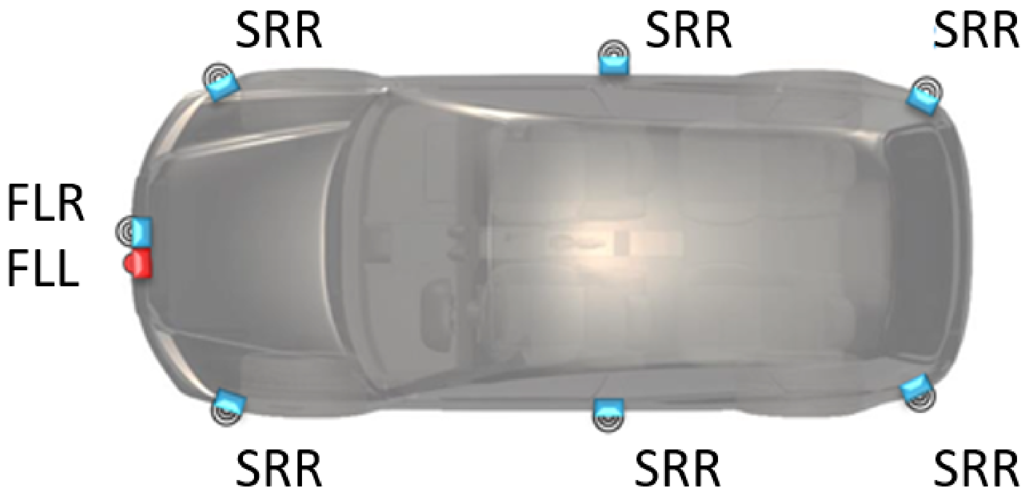


Figure 2.39: Sensor suite on the vehicle model (Short Range Radar (SRR), Forward Looking Radar (FLR) and Forward Looking Lidar (FLL).

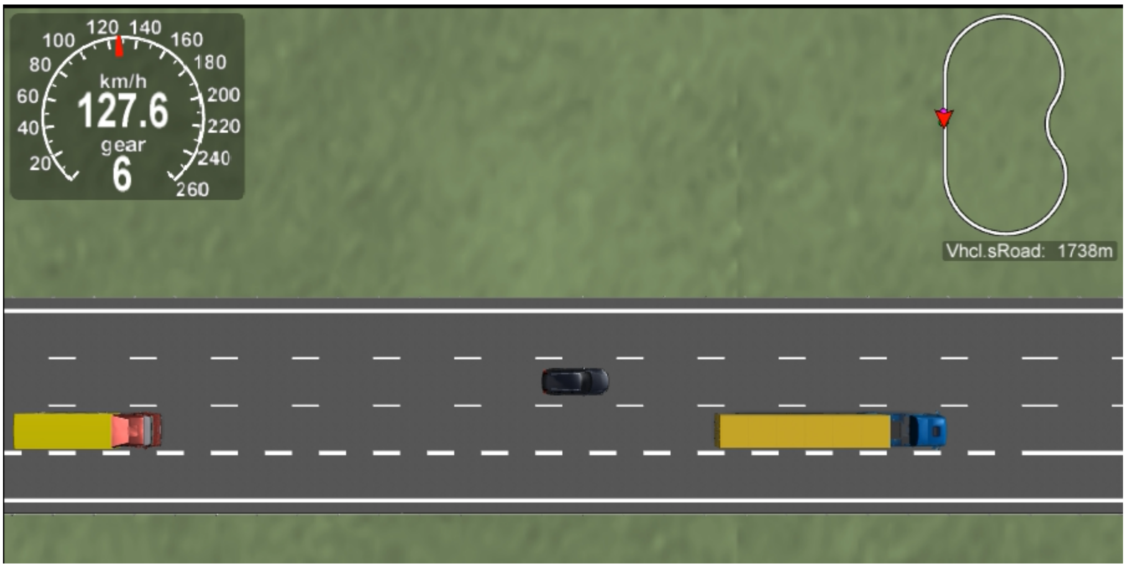


Figure 2.40: Snapshot of the scenario, including two commercial vehicles.

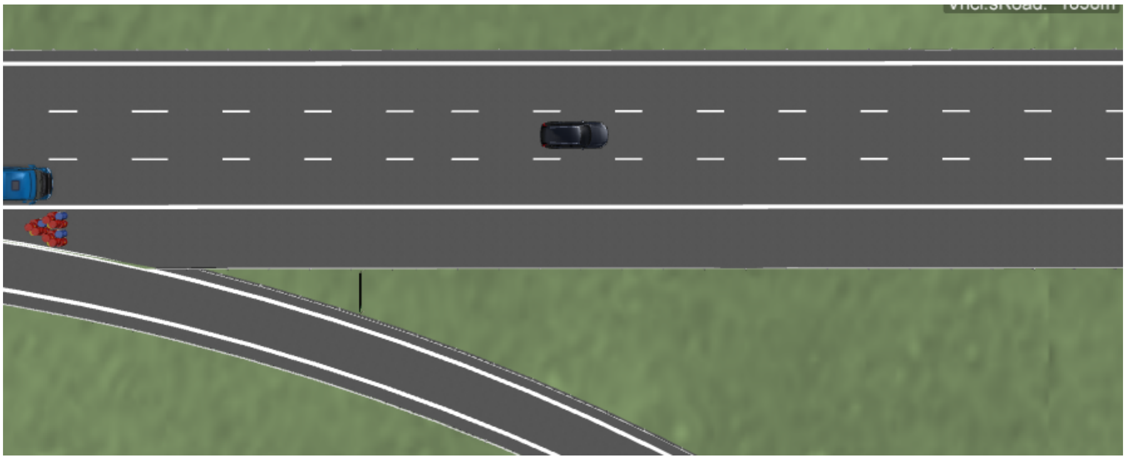


Figure 2.41: Snapshot of the junction in the test environment.

The primary driver behind the selection of two subsets of scenarios (with and without moving vehicles) is driven by the need to investigate the capability of the presented methods on the resolution of moving and stationary obstacles and assessment of performance impact of moving vehicles and occlusion on the perception outcome. An example of the resultant data from the test scenario is shown in Figure 2.42

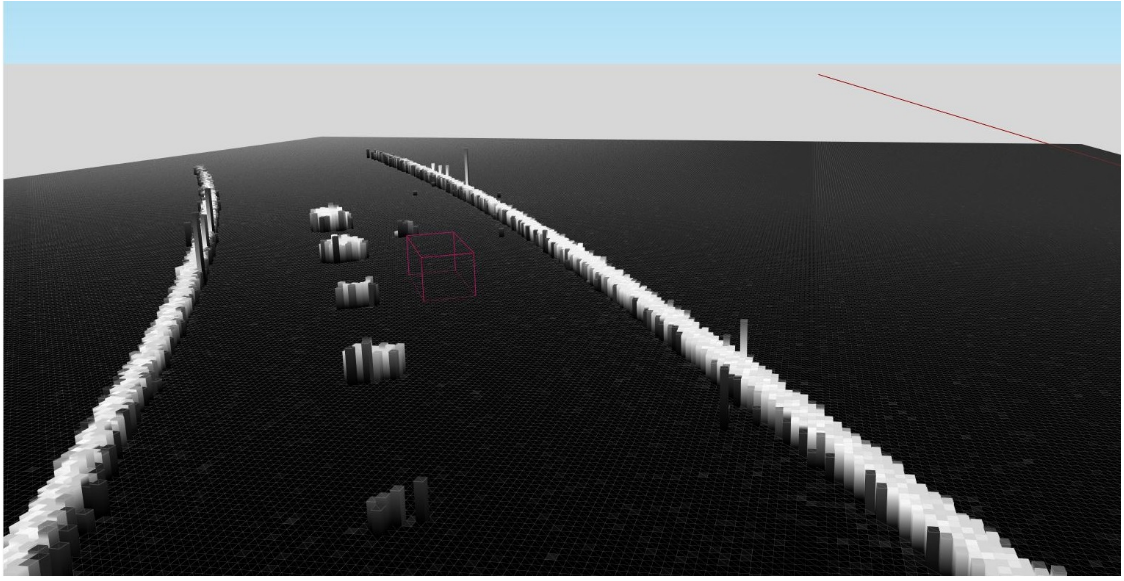


Figure 2.42: Snapshot in perspective of Bayesian occupancy grid - cell color scale represents probability (black - 0 white 1). Bar height represents registered mean height of the cell. Bounding box represent host vehicle.

2.8.4. Scenario for Real testing

As stated in the previous chapter, virtual validation is one of the steps in early prototyping that allows the faster definition of system design and identifying critical issues at an early stage. The need for execution of real-world testing grows however along with the growth of technological maturity of the developed product. The set of sensors used in the modeling is shown in Figure 2.44.

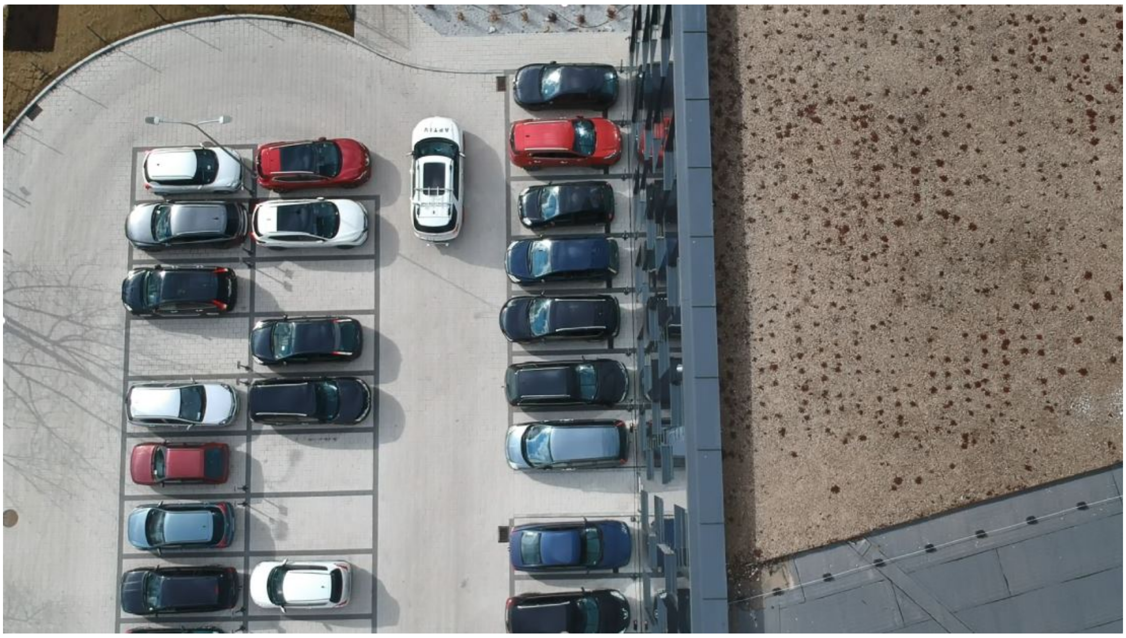


Figure 2.43: Aerial photo of the test vehicle moving through the test scenario area

In the case of this thesis, steps with real-world testing were started at low-speed parking lot scenario with another use-case for the algorithm prototype. The use-case looks into the application of the algorithm for parking space determination. For this purpose, the vehicle described in the previous subsection was equipped with sensors, data acquisition, and a test system. The sensors mounted on the vehicle included four short-range radars mounted on the corners of the vehicle along with a medium-range radar on the front. The setup on the front allowed a partially overlapping field of view of three sensors to exercise sensor fusion capabilities.

The test scenario aerial snapshot is presented in Figure 4.4. The image shows the test vehicle visible in the center - driving through the parking lot. Parking spaces (both occupied and unoccupied) are visible on both sides of the parking. Their landmarks such as curbs and buildings are also visible in the frame - those are static infrastructure features that will be further assessed for potential applications of landmark matching. The vehicle in the scenario passed through the parking with a speed of up to 10km/h, in a straight line to minimize the impact of yaw-rate bias from the in the test.

The components for the algorithms were also tested with a focus on Inverse Sensor Models on open roads in another part of the research[68].

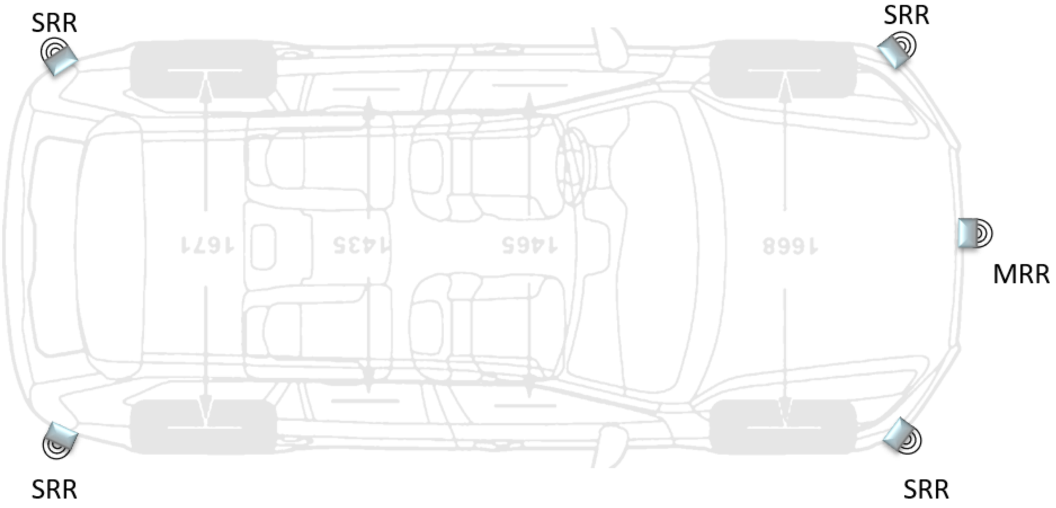


Figure 2.44: Sensor envelope on the test vehicle

3. Architectures and Novel Process Combinations

This chapter walks through the novel implementation of the process used in the research. The methods used for performance assessment in virtual validation are presented with a detailed look into step-by-step data processing. The new proposed method that simplifies the inverse sensor modeling to 1D for optimization purposes. The overview of the process utilized in the research to execute the complete data flow from scenario definition through data generation to algorithm assessment is presented[50]. The explored mixed architecture approaches are explained.

The proposed framework architecture and its sub-modules along with the operation results of the framework were described in my articles titled Developing Occupancy Grid with Automotive Simulation Environment [50] and Occupancy Grid Fusion Prototyping Using Automotive Virtual Validation Environment[51] as well as patented solution for simplification of the inverse sensor modeling to 1D for optimization purposes number:FR3097972B1 [52].

3.1. Cell-Wise Accuracy in Virtual Validation

In this chapter, a look into how the available methods for sensor modeling, scenario building, and algorithm execution were put together to produce experimental evaluation results is presented. The portion of methods described for already existing solutions was given in previous chapters as the existing knowledge along with the extension of description for newly implemented basics is discussed.

An additional approach for validation of occupancy grid methodology focusing on the inverse senors models is discussed in coauthored publication [67]. Other options for performance assessment of the occupancy grid type of perception methods include object-wise accuracy estimation. This approach focuses on how well some particular object is being represented on the occupancy grid. The major advantage of this method is that we focus on grading the performance on a particular region of interest and obstacles which are very relevant for certain use-cases. The disadvantage of this method is that it requires manual effort for labeling certain areas thus making it labor-intensive for vast data-sets however it enables a more specific assessment of performance under certain use-cases.

Another potentially valuable option for performance assessment of occupancy grids is to explore indirectly the performance of features deployed using the grid. For instance, trying to find parking spaces or estimate a derivable lane on the grid. This process similarly to the region of interest approach requires manual effort in the preparation of reference and test data. Moreover, except for the labeling effort, the grid has to be processed through the algorithms estimating the features. Due to that operation the assessment can be biased and is more of an integration test type rather than an isolated test of the grid algorithm. This on the other hand in a better way checks the applicability of the grid in practical solutions in contrary to cell-wise synthetic performance.

3.1.1. Framework Architecture

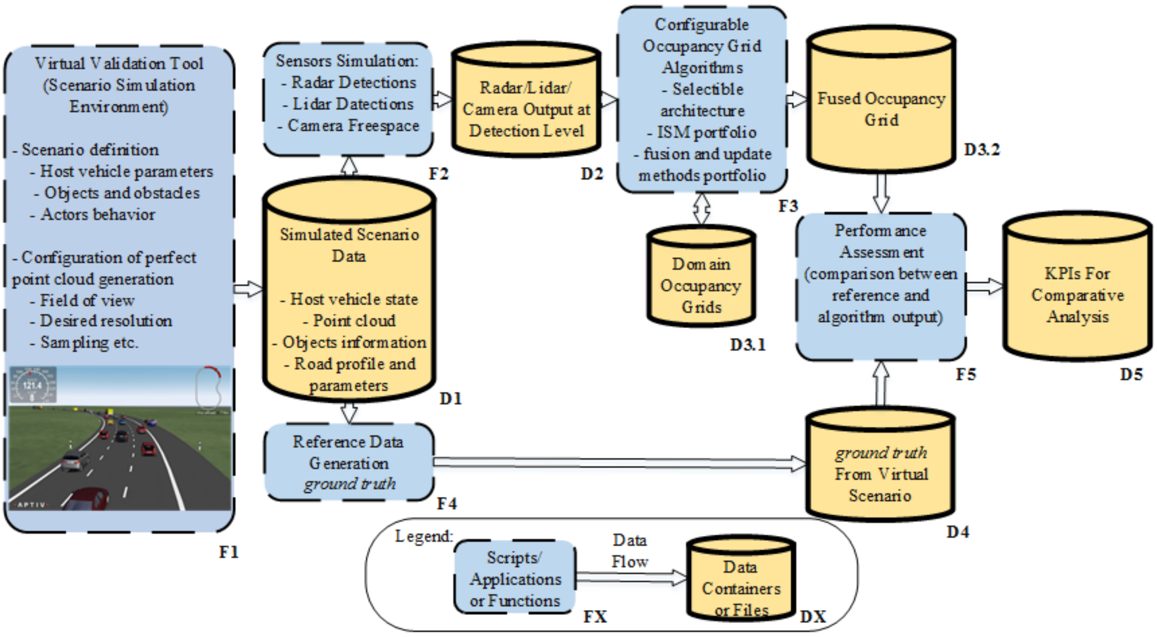

Virtual Validation Tool (Scenario Simulation Environment)
- Scenario definition
- Host vehicle parameters
- Objects and obstacles
- Actors behavior
- Configuration of perfect point cloud generation
- Field of view
- Desired resolution
- Sampling etc.
F1
Sensors Simulation:
- Radar Detections
- Lidar Detections
- Camera Freespace
F2
Radar/Lidar/ Camera Output at Detection Level
D2
Configurable Occupancy Grid Algorithms
- Selectible architecture
- ISM portfolio
- fusion and update methods portfolio
F3
Fused Occupancy Grid
D3.2
Simulated Scenario Data
- Host vehicle state
- Point cloud
- Objects information
- Road profile and parameters
D1
Domain Occupancy Grids
D3.1
Performance Assessment (comparison between reference and algorithm output)
F5
KPIs For Comparative Analysis
D5
Reference Data Generation ground truth
F4
ground truth From Virtual Scenario
D4
Legend:
Scripts/ Applications or Functions
FX
Data Flow
Data Containers or Files
DX

Figure 3.1: The Schematics of Development Framework.

The key innovation of the presented research is the use of a virtual validation environment rather than performing real-world tests, as virtual validation allows quick adaptation, modification, and execution of the test portfolio[77], as well as rapid reference data generation. Virtual validation methodologies have often known deficiencies because sensor models and test scenarios have limited capabilities to replicate all the physical aspects of the real-life sensors and tests that they model. The scope of this research focuses on exploring the capabilities of virtual validation methods and does not seek to determine the absolute performance of these methods[51].

The general flow of data in the framework implemented in this study is presented in Figure 3.1.

The virtual scenarios described before were modeled using the *F1* module. Scenarios are modeled by defining actors – traffic participants, their behavior, the host vehicle behavior, and any stationary infrastructure, i.e. lanes, barriers, buildings, traffic signs, and vegetation. This step was performed using a commercial automotive simulation tool that incorporates modeling of vehicle dynamics, actors' behavior, and simplified sensor models[15].

Figure 2.39 shows the sensors installed on the virtual validation vehicle that was modeled in the simulation tool. Sensor information was used for the parametrization of the sensor models. The sensor models were used to generate the point cloud that was used in further sensor modeling and reference data generation.

The commercial virtual validation tooling described in Chapter 2 is used to generate the vehicles states, reference, and perfect sensor data (utilizing FreeSpace+ and LIDAR models in the software). This data is labeled as *D1* in the process flow. The results of the simulation contain dynamic objects state information, a point cloud representing the environment (computed on a GPU using ray tracing), host vehicle motion and position, and parameters describing road curvature. A snapshot of the data contained in *D1* is presented in Figure 2.28.

3.1.2. Sensor Modeling

The modeling focused on simulating the outputs of sensor domains (radars and lidars). The developed sensor models convert the data from *D1* to virtual detection points. The component *F2*, implemented in Mathworks MATLAB, allows model parametrization by defining the mounting position, orientation, field of view, spatial accuracy, resolution, triggering frequency, latency, and the number of detection points per scan. Additional parameters provide the capability to set up correlations between certain calibrations (i.e. detection range vs radar cross-section for radar model and reflectivity for lidar models). Based on the tuning parameters, the sensor models add noise to the ideal sensor output for the point cloud[88]. Values for the tuning parameters are derived from experimental assessment in which the characteristics of sensors are established (similar to the method described for Aptiv ESR [79]). An example snapshot of generated radar detection points is shown in Figure 3.2.

The second layer of the sensor modeling makes use of object information from the virtual validation tool to replicate the behavior of object-level fusion by adding noise to the input data.

Potential opportunities for the implementation of more realistic models exist and are partially addressed by more sophisticated Finite Element Analysis software. That type of software delivers physical modeling capabilities for wave propagation and reflection

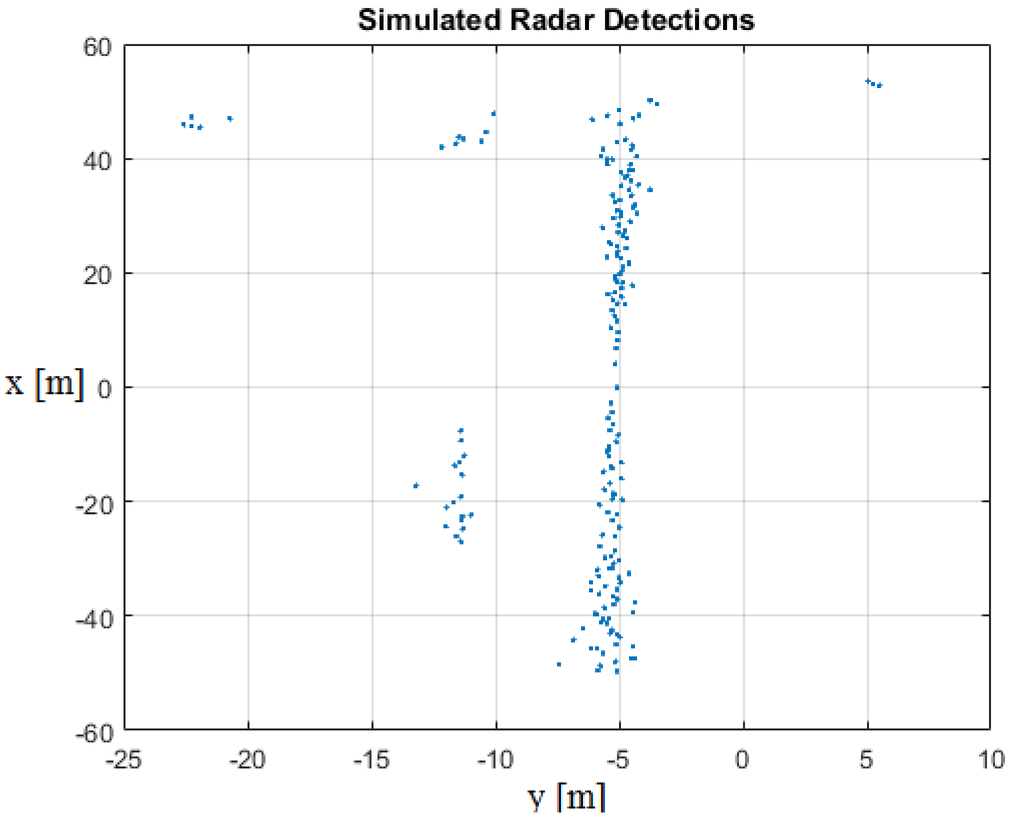


Figure 3.2: Bird′s eye view of the data from the simulated four corner radars. The host vehicle is located at the (0,0) point of the plot. Detection points generated by the barriers are visible on the host vehicle's left side.

taking into consideration the material properties and shapes of objects in the scene. This approach adds severe computational complexity to generation of scenario data. Additional workload needed to implement this approach requires a precise definition of models including their internal construction and materials used. The next step in physical sensor modeling is the low-level sensors model (such as antennas in the case of radar sensors). For additional realism an artificial set of low-level data can be processed through a virtualized embedded software or HIL system including modified sensors, to generate a real response of the actual sensor responsible for the computation of the output signals.

Due to the nature of the research focusing on system-level design, physical modeling would be too complex and a statistical approach was chosen instead. However, the presented process could accommodate such strategy if needed.

3.2. Proposed Novel 1D Inverse Sensor Model Optimisation

The topic of delivery of probability onto occupancy grid has been covered in chapter 2.3.1 mainly by means of 2D Inverse Sensor Models which rely on 2D probability density function which is integrated onto the grid to obtain probability distribution among the cells. Since the grids found in literature and applications originated from ultrasonic sensor the ideas of providing the 2D distribution was a method for coping with measurement uncertainties that are relatively large compared to the assumed grid size.

In case of automotive application the most relevant sensors in this application are lidars and radars which are characterized by a relatively large distribution along only one dimension of measurement. Therefore a sufficient distribution can be obtained only along one axis – either range or cross-range (derived from azimuth) of such measurement. Another part of simplification used in 1D is performing the distribution of probability along a straight line for azimuth uncertainties. The description of this novel approach has been patented as FR3097972B1 patent[52].

The main motivation behind the implementation of such a simplified model is to provide potential for computational optimization in comparison to the traditional methods described in the introduction chapter 2.3.1. The main theoretical advantage of this method is that it is used to compute the probability only along a single line crossing the detection (radial or along the arc). This simplification relies on the assumption that the error of the detection estimation is negligible in the other dimension. The main disadvantage of this method is that in case of higher inaccuracy of the measurements it is rendering false negative results onto the grid. The traditional approach on the other hand is more computationally complex, however not prone to such deficiencies related to false negatives.

A two-dimensional map is defined as in equation 3.1 with N_L and N_W respectively corresponding to number of cells by length and width. The map is composed of N_L by N_W cells. The grid starts in the Occupancy Grid Coordinate System (OGCS). Position in the grid can be described in that coordinate system with physical distance (for instance in meters), therefore each of the cells occupies certain place that is known due to known spatial resolution (res) of the grid. The grid is used further to tessellate (to form of probability) the environment representation derived from sensor measurements.

$$g = \ g_i j : 1 \leq i \leq N_L,\ 1 \leq j \leq N_W \quad \in \ <0,1> \tag{3.1}$$

The sensor measurement is defined as detection (either lidar or radar) with consecutive uncertainties for range and azimuth. Detection is described by the following variables with respect to Sensor Coordinate System (SCS):

- r – detection range in SCS
- α – detection azimuth in SCS
- $\mu_x \mu_y$ - detection x and y coordinates in OGCS (since the location of SCS with respect to OGCS is known using r, α and that location)
- σ_r - standard deviation of range measurement
- σ_α - standard deviation of range measurement

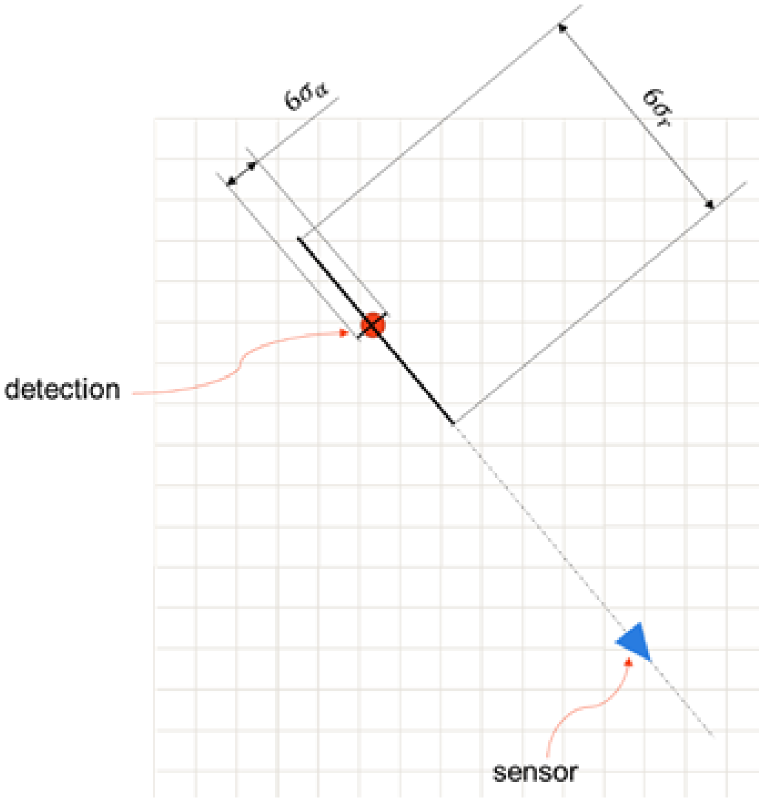


Figure 3.3: Case of detection with negligible azimuth uncertainty

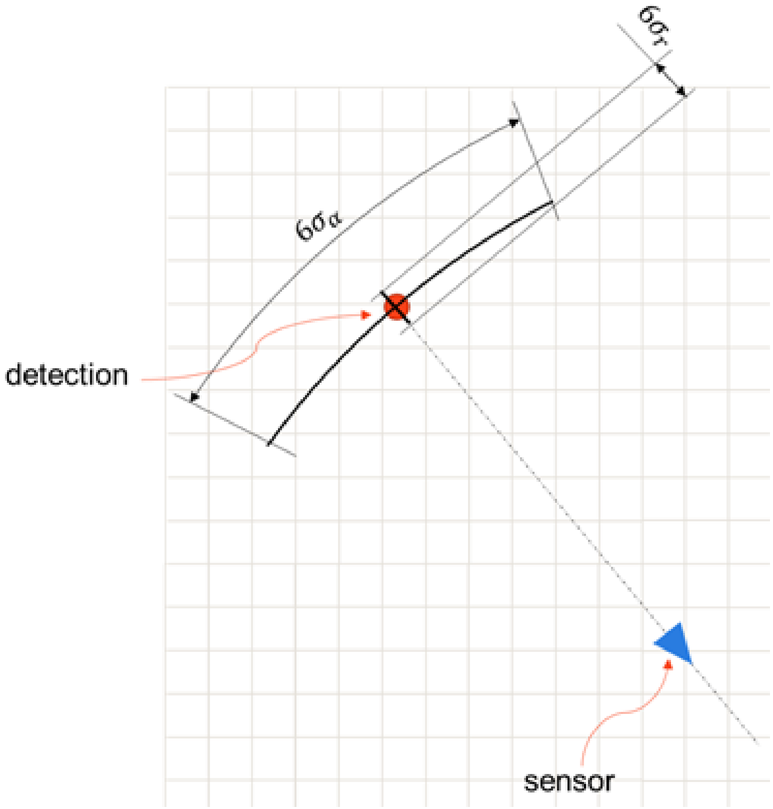


Figure 3.4: Case of detection with negligible range uncertainty

The process for computation of the probability distribution in the proposed 1D Inverse Sensor Model Simplification :

- In case of situation where only range uncertainty is taken into consideration (figure 3.3) the distribution of ISM is applied along the range axis (connecting the sensor origin to the detection) and uncertainty for distribution is taken from range $\sigma = \sigma_r$
- In case where the range uncertainty is negligible ($6\sigma_r < res$). The distribution based on cross range uncertainty is calculated (figure 3.4). Since σ_α is expressed in polar coordinates it is recalculated to σ_{cr}- cross range uncertainty, and the distribution is performed along axis perpendicular to the range. $\sigma = \sigma_{cr}$

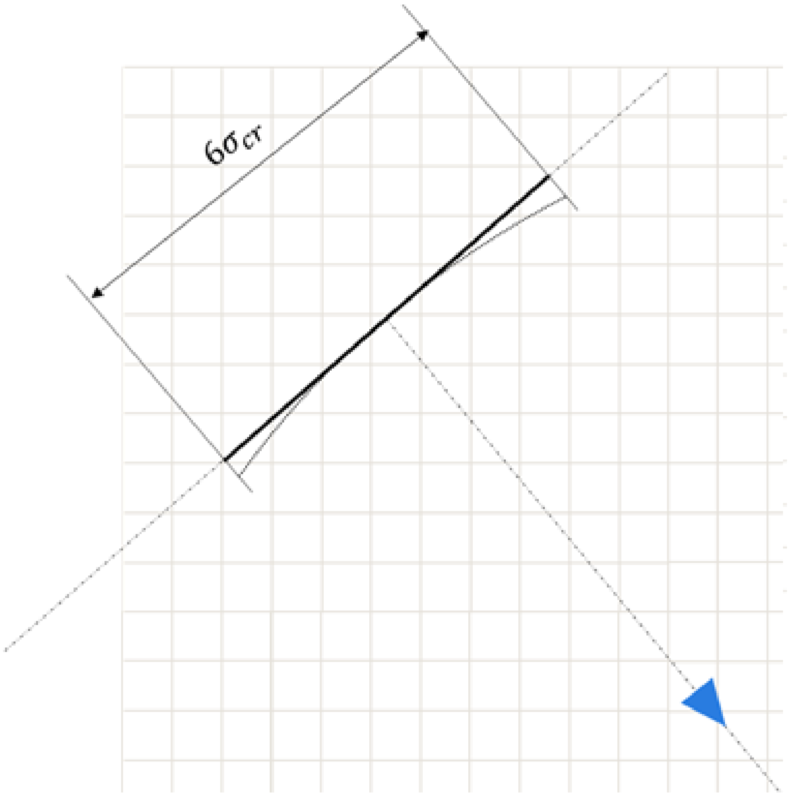


Figure 3.5: Cross-range for distribution

Then the coordinates for $\mu + 3\,\sigma$ and $\mu - 3\,\sigma$ of the beginning and end of application of Bresenahm's algorithm are computed.

$$x_{3\sigma\pm} = \mu_x \pm \cos{(\alpha)} 3\,\sigma \tag{3.2}$$

$$y_{3\sigma\pm} = \mu_y \pm \sin{(\alpha)} 3\,\sigma \tag{3.3}$$

- Computation of line sections (d_i) and distance (r_i) for each cell (either radial or normal -in this steps a radial example is described)
- Coefficients for a line equation are calculated

$$a = \frac{y_{3\sigma+} - \ y_{3\sigma-}}{x_{3\sigma+} - \ x_{3\sigma-}} \tag{3.4}$$

$$b = y_{3\sigma-} - \ x_{3\sigma-}a \tag{3.5}$$

$$y = a \cdot x + b \tag{3.6}$$

- Calculation of intersection points of each grid cell boundary with the line computed in previous step.

$$y_{Li} = a \cdot x_{Li} + b \tag{3.7}$$

$$y_{Ri} = a \cdot x_{Ri} + b \tag{3.8}$$

$$x_{Ti} = (y_{Ti} - b)/a \tag{3.9}$$

$$x_{Bi} = (y_{Bi} - b)/a \tag{3.10}$$

- Validation of which cell area is intersected by the line.

$$y_{Li}\ ,\ y_{Ri}\ \in [y_{Bi},\ y_{Ti}] \tag{3.11}$$

$$x_{Ti}\ ,\ x_{Bi}\ \in [x_{Li},\ x_{Ri}] \tag{3.12}$$

- Section length calculation

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{3.13}$$

- Offset of the cell with respect to the center

$$r_i = \sqrt{\left(\frac{x_1 - x_2}{2} - \mu_x\right)^2 + \left(\frac{y_1 - y_2}{2} - \mu_y\right)^2} \tag{3.14}$$

- Calculation of PDF value for each cell intersecting with radial or corss-radial axis within the 6σ value
- Calculation of probability by simplifying integration of PDF by multiplying the function value by section length shown in Figure 3.6 and for multiple cells in Figure **??**.

$$p\left(r_i\right) = \int \varphi\left(r\right) dr\ \approx \varphi\left(r_i\right) \cdot d_i \tag{3.15}$$

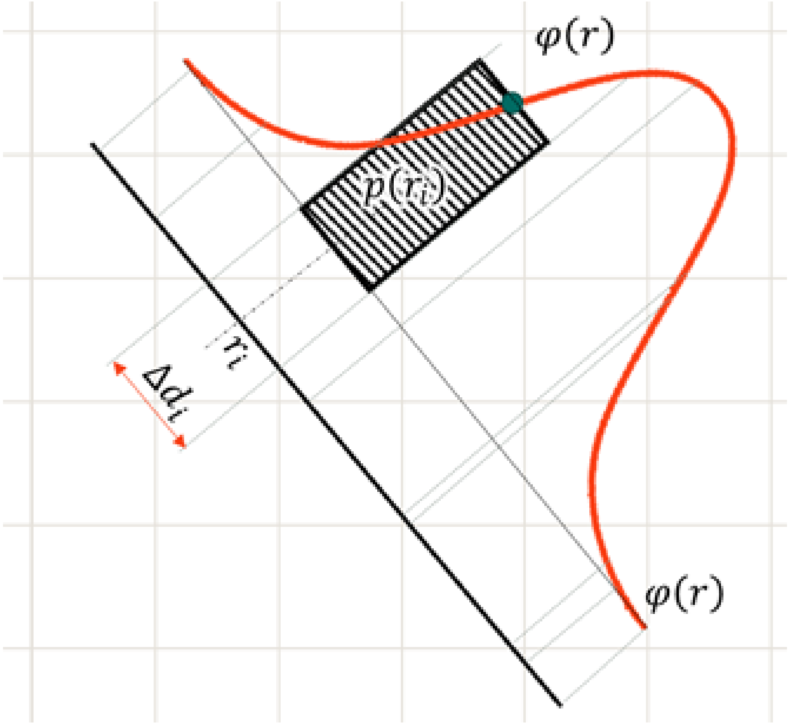


Figure 3.6: Example of probability calculation for a single cell

As a test method for probability calculation, sum of all cells in which update was performed is taken into consideration to be targeted as 1.

$$\sum p\left(r_i\right) = 1 \tag{3.16}$$

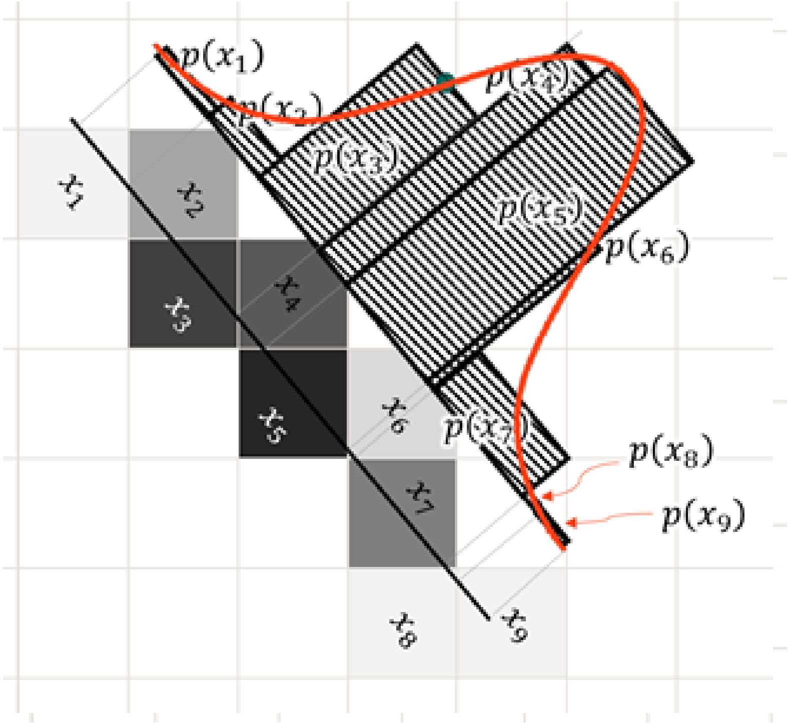


Figure 3.7: Example of areas calculated and remaped in form of intensity to cells

The method has been implemented in form of Matlab scripts working both on virtually simulated and real detections from radars and lidars. The example of the computation results is presented in Figures 3.8 and Figures 3.9.

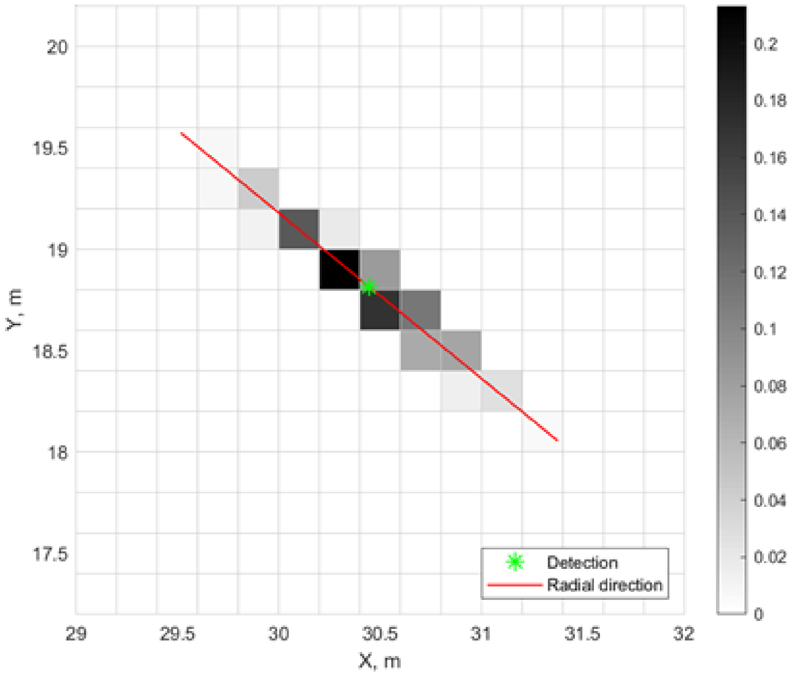


Figure 3.8: Probability distribution from Matlab implementation with sum: 0.99476

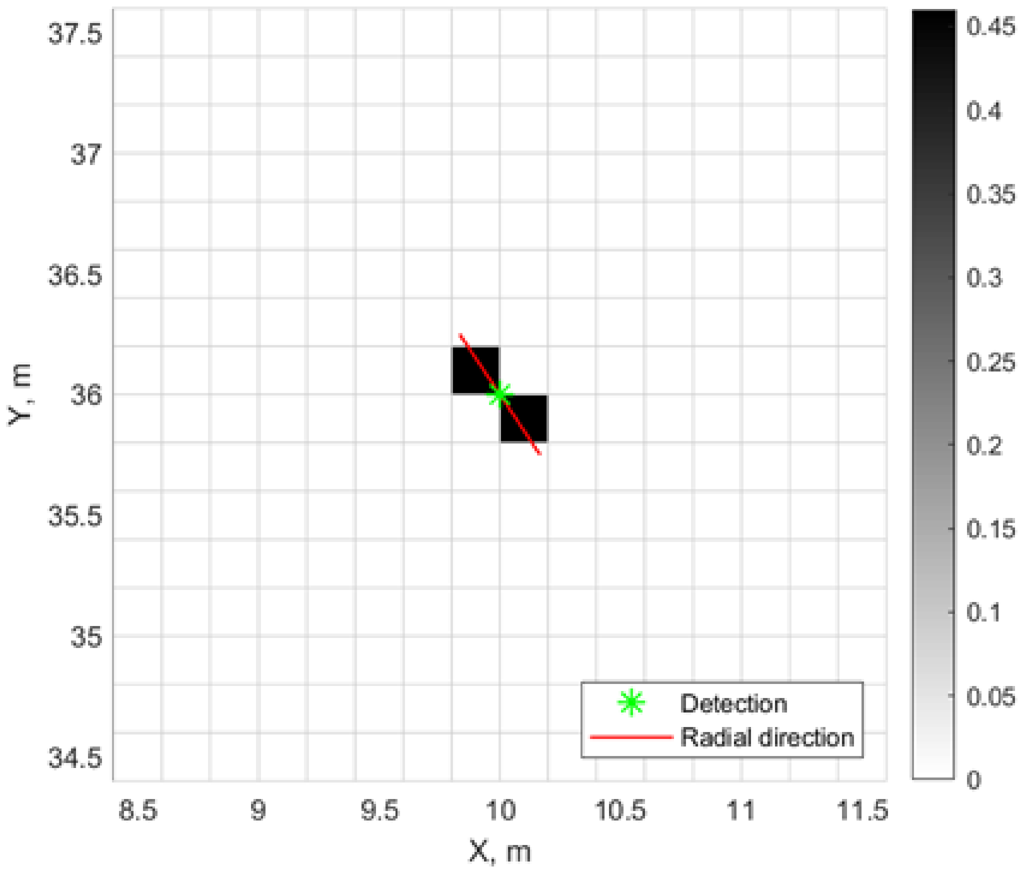


Figure 3.9: Probability distribution from Matlab implementation with sum: 0.92040

Presented method employs three simplifications as compared to the traditional ones found in the literature. The first one is computing the distribution along just one axis, which results in less required computation effort even when using lookup tables for the functions – the main disadvantage however is that in case of uncertainty extended beyond grid resolution some cells are omitted what could result in potential false negative detection. The second one is simplification of the cross range use for addressing the azimuth uncertainty, this one allows relatively accurate identification of cells with use of simple Bresenham's algorithm instead of finding arc intersection with cells. The third simplification relies on substitution of numerical integration of probability density function by approximation of that value to area of a rectangle, which is quite common method but haven't been found in literature as applied to use in inverse sensor models

3.3. Low Level Fusion

Both fusion architecture solutions allow utilizing configurable state of the art inverse sensor model presented in the introduction chapter along with all described mathematical data fusion methods.

The implemented framework accommodates two architectures of grid fusion: centralized (also called low-level or measurement-level - Figure 2.20), and decentralized (also known as grid-level - Figure 2.21)[69]. All of the details discussed in this subsection correspond to the functions and containers *F3 D3.1* and *D3.2*. The configurable architecture is based on a set of parameters defining the size, spatial resolution, methods for handing height information (2.5D approach), inverse sensor model parameters (hit point, 2D probability distribution, use of ray casting), and dynamic objects information removal[73]. To avoid situations in which a Bayesian fusion approach would lock the probability at 0, a minimum probability in the grid is assumed $p_{min} > 0$ and $p_{max} < 1$.

Proposal of Sensor Optimization for Fusion Purposes

Different methods for performing data fusion have been presented, described, and implemented through recent years[24]. Discussed methods are visualized in Figure 3.10. From fully centralized methods in which raw data from sensors is fed to the central processing unit for deriving object properties - Example A, to approaches in which already processed information in a form of object information is supplied to the central unit for sensor fusion - Example B through hybrid approaches in which object data from one type of sensor are used to facilitate object detection in a different type of sensor - Example C.

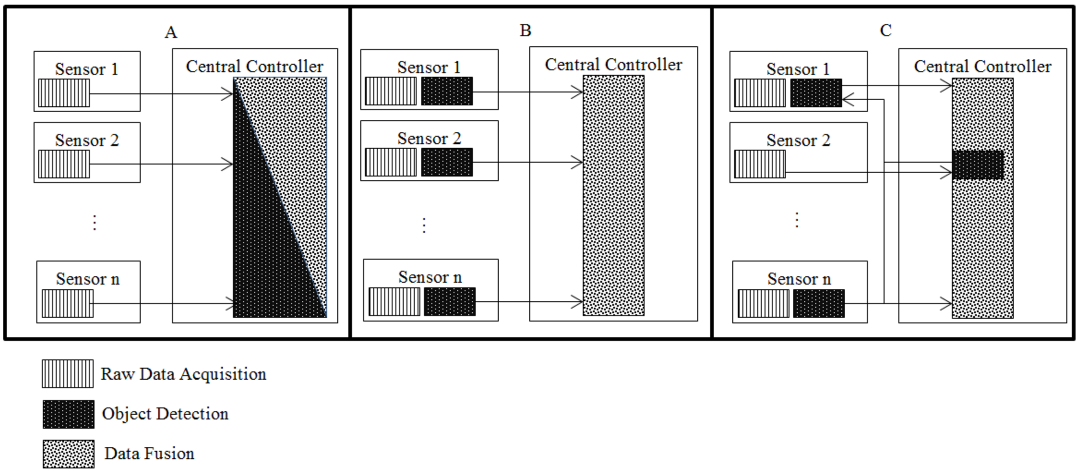


Figure 3.10: Examples of Fusion Solutions

As most of the sensors operate on quite complicated raw data which usually would require large amounts of scarce memory to store (such as raw video data or raw radar reflection data), an approach is proposed in which chunks of raw data are being populated to the centralized unit. This could be easily confused with the original method presented in Example C and B - as objects generated by sensors when the detection part is performed inside the sensor - are a derivative of raw data. But in this case, the idea behind this approach is to populate windowed-out areas of low-level data, what is shown in Figure 3.11.

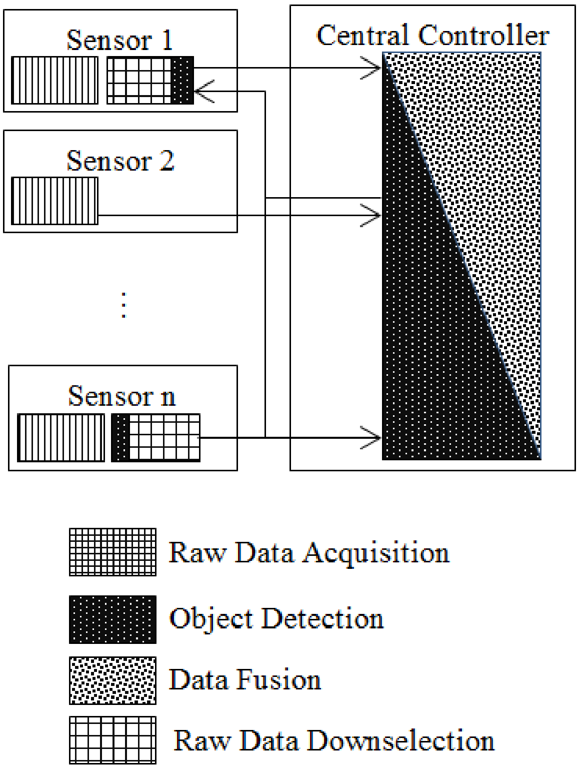


Figure 3.11: Proposed Down-Selection Method

Populated chunks of low-level data then would be processed for object detection which could include time-domain tracking of low-level information - as the central unit would allow for bigger computational resources and complex fusion algorithms.

As we are focusing on sensor design in this subject–this would allow reduction of sensor computational needs–to the point in which sensor is capable of determining regions of interest (without tracking) - from which low-level data would be populated to the central computational unit at a cost of higher need for bus throughput. The busload on the other hand would be lower than in Example A from Fig 5. On top of that, reduce the computational needs of the sensor, area of interest selection will be auxiliary supported by information from the central unit on already identified objects similarly to Example C. Proposed approach would be beneficial mainly for Radar and LIDAR systems which in most cases are limited to object building on their own. Reduction of tasks performed by an externally mounted sensor would transfer cost distribution towards the Centralized Controller, which is beneficial because sensors mounted on the boundaries of the vehicle are prone to damage.

The proposed approach has been published in my work on the review of tracking and object detection methods[49].

3.4. Combination of Methods

This section presents the list of methods implemented and tested during the scope of the project. The methods used in this research are presented in Table 3.1. The first column named "Variant" presents an abbreviated name used further to uniquely distinguish the implemented configuration. The second column named "Architecture Variant" distinguishes between High-Level and Low-Level sensor fusion modes. The third column named "Filtering Mode" distinguishes the use of Accumulated grids (filtered in time) versus Instantaneous - ones that did not accumulate time history information. The fourth column "Method" indicates what fusion method was used for High-Level and Low-Level fusion when merging information from multiple sensors. The proposed combinations in the case of high-level fusion relied on the Bayesian approach for individual sensors grids and then were fused by the high-level fusion using listed algorithms. Some of the algorithms (FHL-LI, FHL-L, FHL-I) taken into consideration for high-level fusion in this research were skipped from analysis in the prior art of Galvez[69], however, were included in the experimental analysis using a virtual environment. The new approach proposed in this research was assessed using all the instantaneous combinations of the high-level methods with domain grids created with the Bayesian approach (HL-M, HL-LI, HL-L, HL-I, HL-De and HL-S). An overall new approach to using the sum of probabilities was also proposed in this research (HL-S).

Performance Assessment Methods

To assess the performance of presented fusion rules, a container *D4* with the reference data has to be created by reference data generation tooling *F4*. The generated reference data is then compared against fused grid results *D3.2* using performance assessment tooling *F5*, which employs a set of Key Performance Indicators (KPIs) for use in qualitative analysis. This analysis results in a report *D5*, which contains KPIs for a set of input data.

Variant	Architecture Variant	Filtering Mode	Method
LL-B	Low Level	Accumulated	Bayes
LL-DS	Low Level	Accumulated	D-S
HL-B	High Level	Instantaneous	Bayes
HL-M	High Level	Instantaneous	Max
HL-LI	High Level	Instantaneous	LIOP
HL-L	High Level	Instantaneous	LOP
HL-I	High Level	Instantaneous	IOP
HL-De	High Level	Instantaneous	DeMorgan
HL-S	High Level	Instantaneous	Sum
FHL-B	High Level	Accumulated	Bayes
FHL-LI	High Level	Accumulated	LIOP
FHL-L	High Level	Accumulated	LOP
FHL-I	High Level	Accumulated	IOP

Table 3.1: Tested configurations and their naming

Reference Data Generation

In this research, a proprietary novel approach was proposed to generate reference data based on simulated data.

The objective of generating reference data is to obtain an idealistic representation of the test scenario. Therefore, a relatively high tessellation resolution of 0.05m compared to 0.2m in the fused grid is chosen. Cells in the high-resolution reference grid that are physically occupied by stationary obstacles (barriers, vegetation, curbs, etc.) are flagged. As the analysis focuses on the static grid, footprints from dynamic objects are removed.

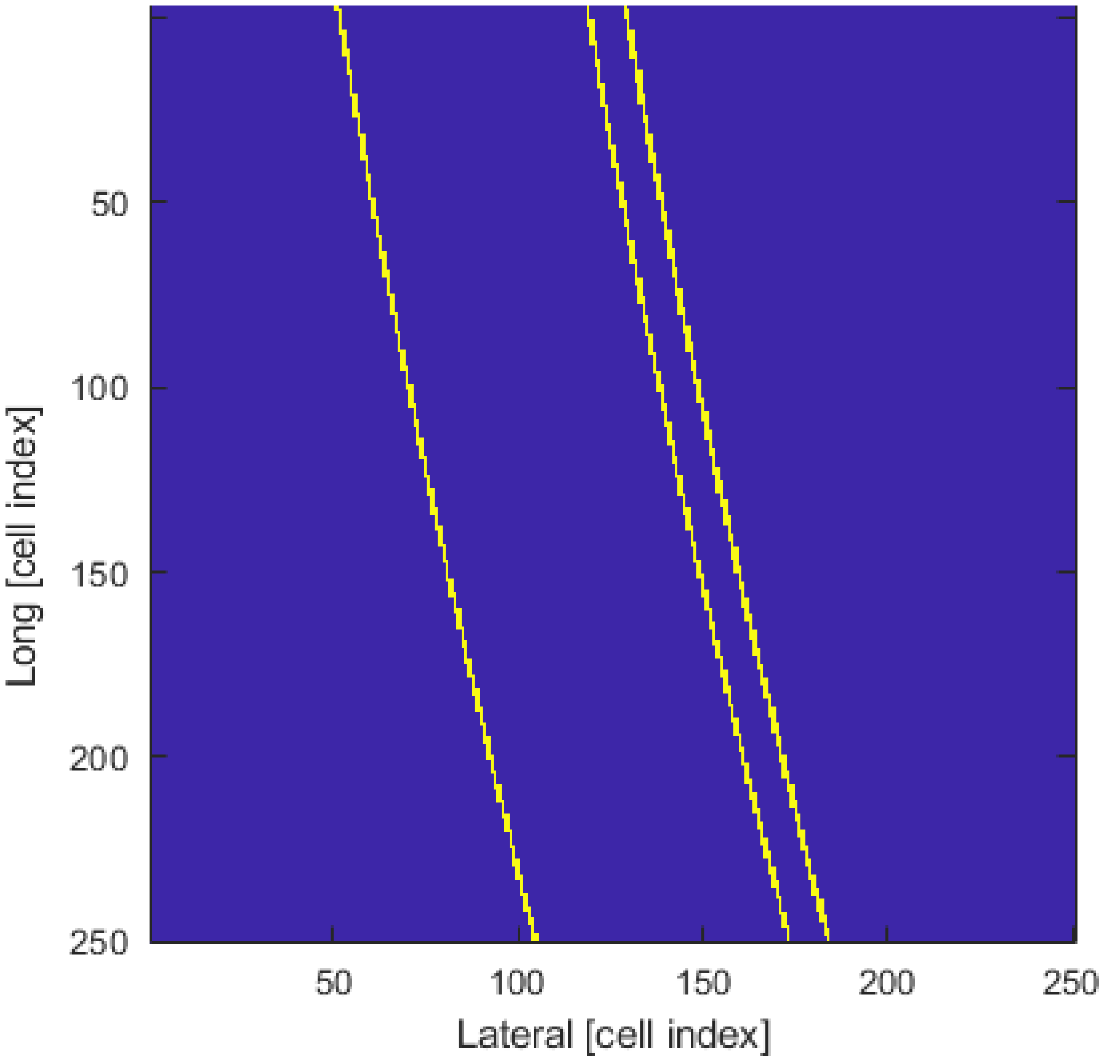


Figure 3.12: Top view of the reference data generated for a curved section of the road (yellow color represents occupied and blue unoccupied cells).

In the end, the reference data grid is defined with a binary state (Equation 3.17). Where N_{Lr} is the length of the reference gird and N_{Wr} and width of the reference gird. An example of the reference data is shown in Figure3.12.

$$\{r_{ij} : 1 <= i <= N_{Lr}, 1 <= j <= N_{Wr}\} \in \{0, 1\} \tag{3.17}$$

The process of performance assessment is explained by the 1D simplification where both the reference and the computed grids, which normally consist of rectangular cells, are represented by line segments with given values. In Figure 3.13, the 1D simplification of a process of ground-truth (reference) grid creation based on the perfect sensor is

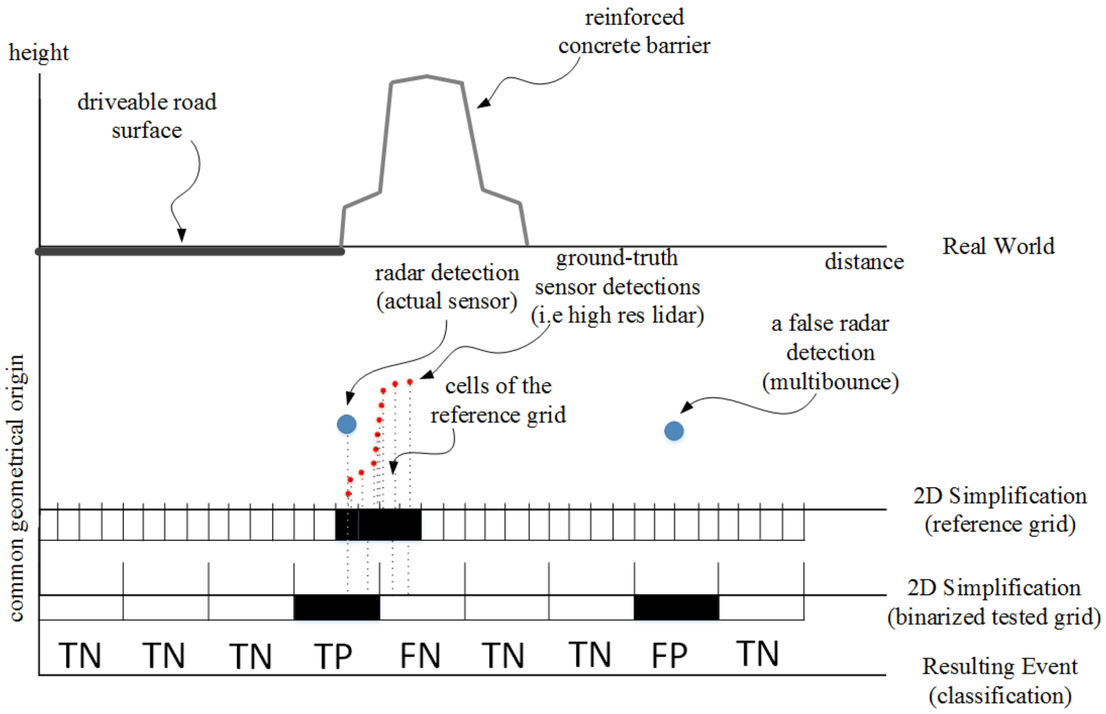


Figure 3.13: Drawing of the 1D simplification of the reference data generation and comparison with the high resolution reference data along with the resulting event classification for the performance assessment.

presented. The assumption for the reference data is that anywhere on the reference grid where there was a registered detection (from the reference sensor) above the road surface level, the binary occupancy state of that cell is set to true. This process is performed for the complete length of the tested scenario to accumulate the information using reference positioning information of the perfect sensor.

This process can be scaled to be utilized in real-world testing by substituting the point cloud generated with the simulation environment, with a data source from a real lidar sensor. This approach however requires a very high accuracy lidar for reference generation and high accuracy positioning data source to accumulate the performed scans into a 3d representation of the environment. Therefore for the early stages of prototyping, it makes the most sense to utilize the simulation method.

Proposed KPIs

After ground truth data generation, each cell was compared to the fused occupancy grid cells (the 1-D simplification is presented in Figure 3.13). For basic performance assessment, the fused grid *D3.2* is binarized with a fixed threshold and spatially aligned with the ground truth map *D4*. The results of the comparison are classified as: True Positive (TP), True Negative (TN), False Positives (FP), and False Negative (FN). The number of cells resulting with corresponding events is summed up for a scenario [14]. Then the data are used for analysis with the KPI methods described in Chapter 2. The basic KPI computation for the scenario (presented in the time domain) is shown Figure 3.14.

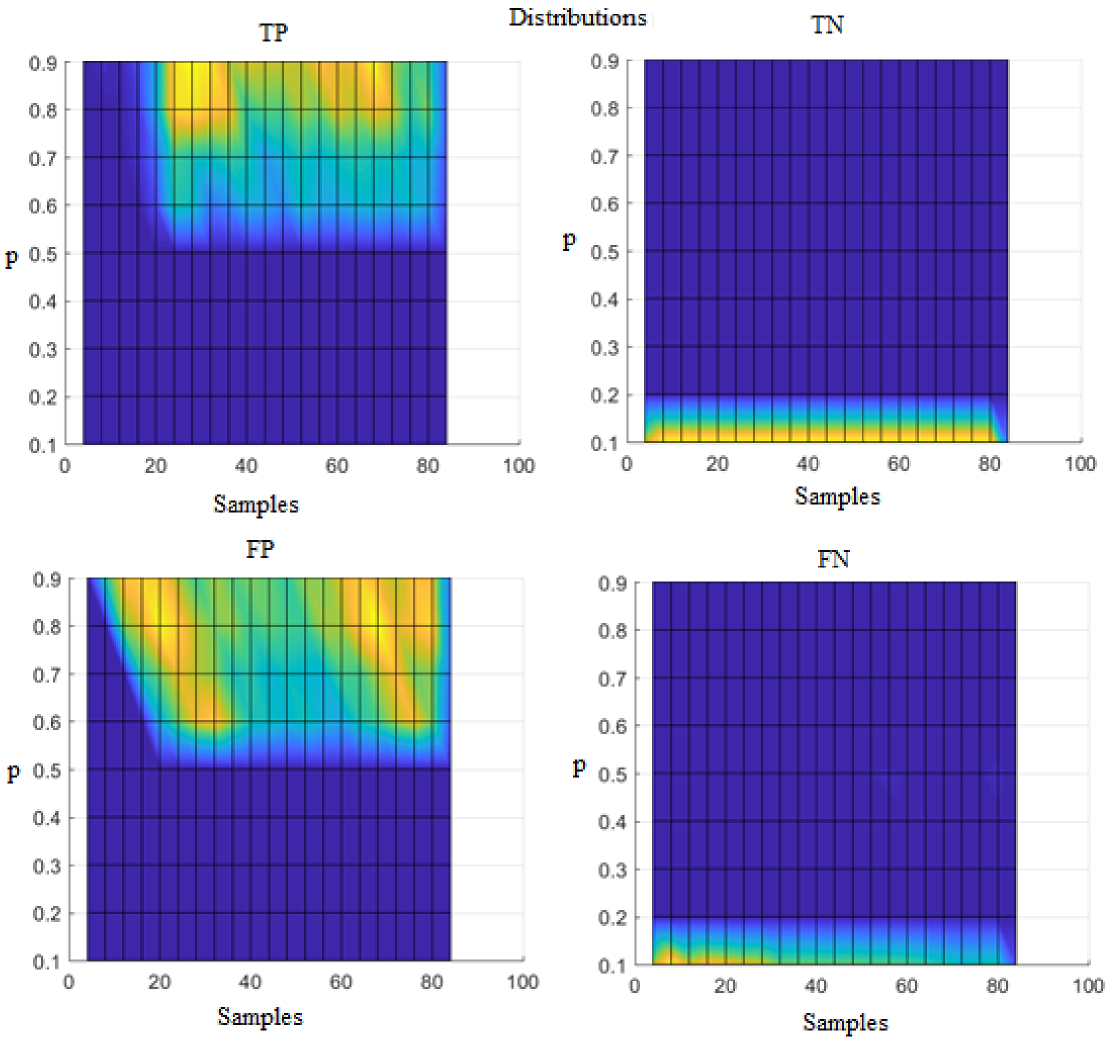


Figure 3.14: Accumulation of basic TP in the scenario in the synthetic data sample domain versus the tested grid probability prior to binarization

3.5. Parking Space Rule Based Assessment

For the last experiment conducted in the experimental evaluation real-world testing has been carried out. The test scenario as described previously was conducted on an open-air parking lot with visual image reference data from a drone.

The evaluation was based on the data collection in the real world (the algorithms were run along in the vehicle in real-time as well for indication and framing purposes). The primary goal of the test drives was to accumulate raw signals from the vehicle and sensors buses to provide a timestamped synchronized data recording.

The collected data recording then was used in the HIL system from the previous chapter on the replay PC. The data was streamed over the physical interface to the SUT with different algorithm configurations to obtain the resultant algorithm data to be used for further assessment.

During the analysis, the resultant algorithm data was accumulated for the length of the scenario for all algorithms variants and then labeled manually to assess the capability to distinguish parking spaces based on the resulting grids. The manual labeling process consisted of finding parking space rectangles and marking them based on the reference aerial image onto the occupancy grid results. This type of performance assessment corresponds to the feature-based performance assessment described in the previous chapter.

Execution using the HIL setup was also used to measure the execution timing of the processes performing fusion computation using the code profiler.

4. Experimental Results

This chapter shows the results of the experiments performed in the virtual validation and real environments. The snapshots of visualization of resultant occupancy grids are presented along with resultant KPIs for the tested events. Results of computation time are shown.

4.1. Virtual Validation

Experimental evaluation was performed using the set of configurations of the methods and architectures. The results of this evaluation are presented in Table 3.1. The presented methods were run in both of the test scenarios (Scenario 1 - without moving obstacles and Scenario 2 - including two moving targets). Reprocessing was carried out through the complete processing chain and resulted in the list of performance indicators presented in Figure 4.3.

Figure 4.3 presents the results for the Scenario 1 and the Scenario 2. The relation between a higher sensitivity of a method (having higher TPR and PPV) and its susceptibility to false classification when subjected to clutter (having higher FDR and TPR) was observed. This is reflected in small differences in the F1 score. This relation can also be observed in the raw probability of occupancy presented in a form of snapshots in Figures 4.1 and 4.2.

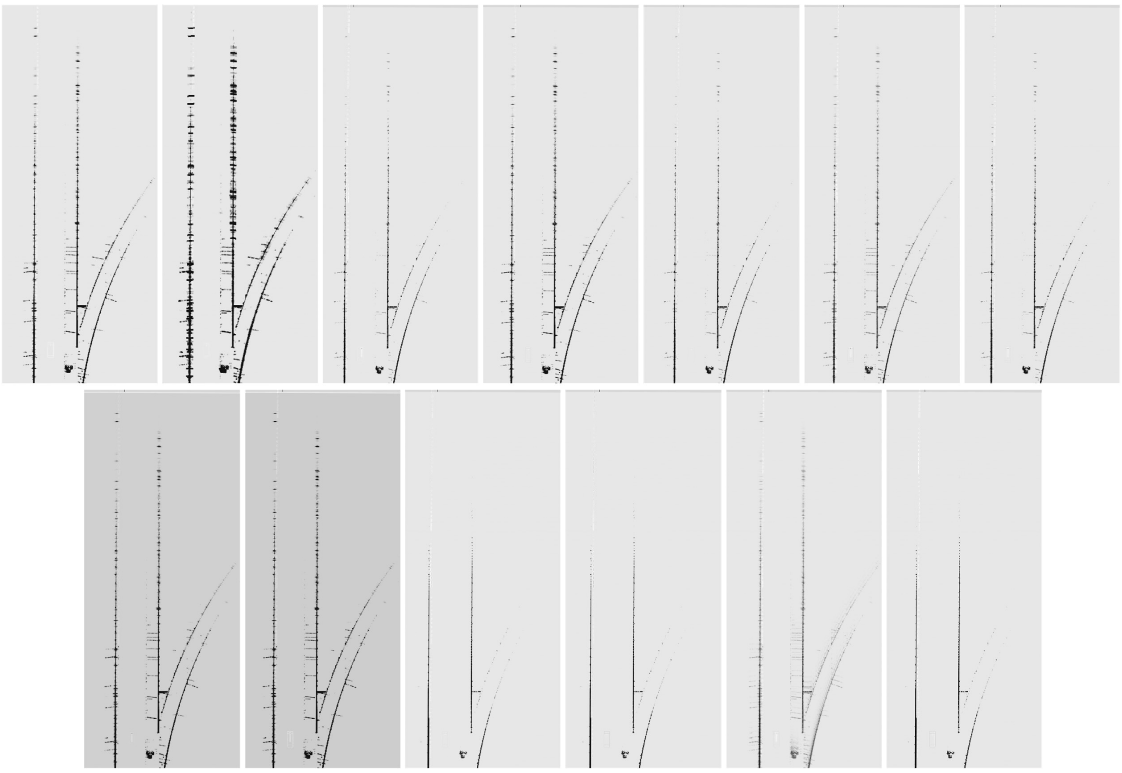


Figure 4.1: Snapshots of resultant occupancy grid for fusion variants in the first row: LL-B, LL-DS, HL-B, HL-M, HL-LI, HL-L, HL-I, in the second row: HL-De, HL-S, FHL-B, FHL-LI, FHL-L, FHL-I for Scenario 1.

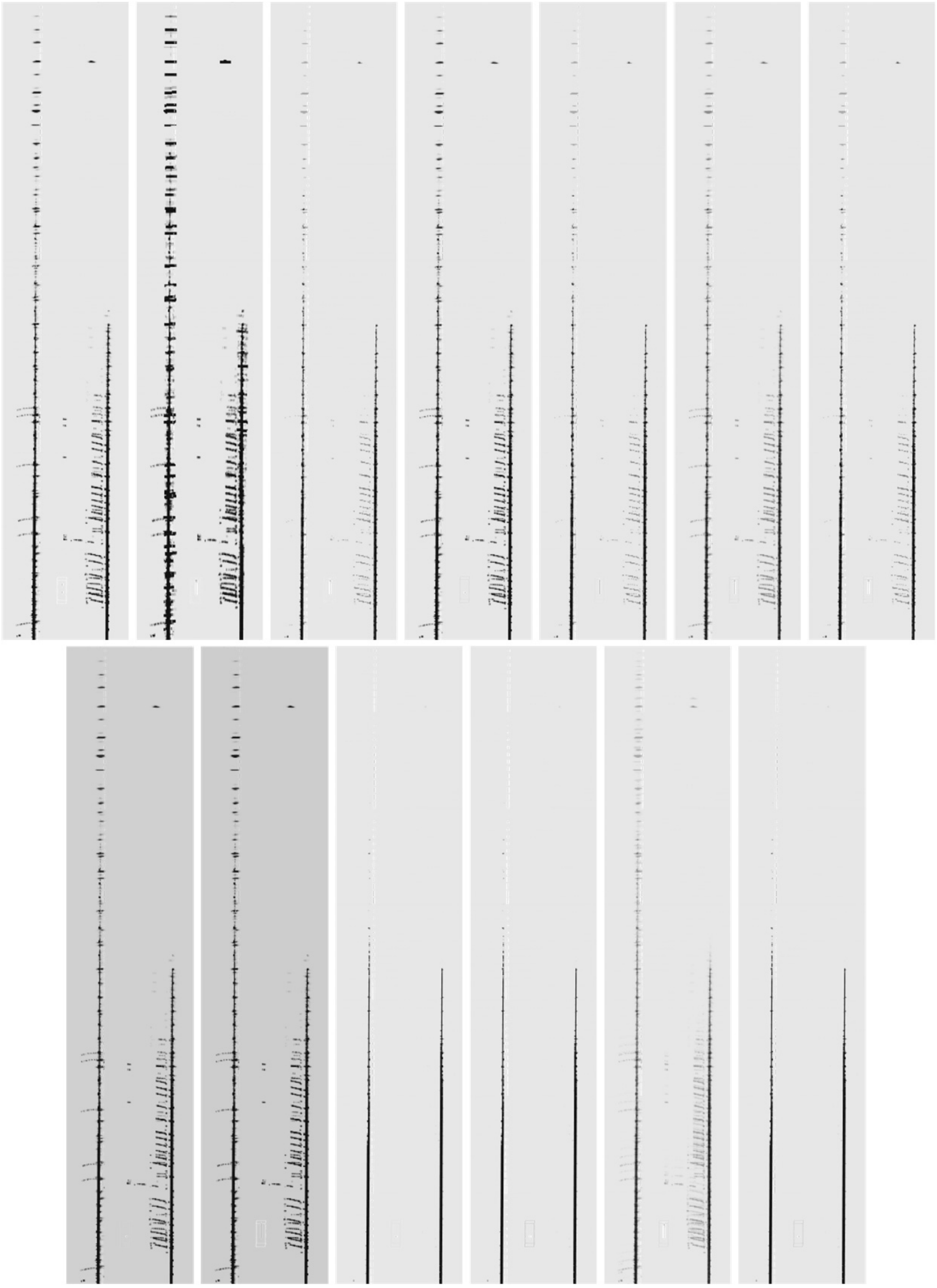


Figure 4.2: Snapshots of resultant occupancy grid for fusion variants in the first row: LL-B, LL-DS, HL-B, HL-M, HL-LI, HL-L, HL-I, in the second row: HL-De, HL-S, FHL-B, FHL-LI, FHL-L, FHL-I for Scenario 2.

Overall, most of the methods have lower performance in Scenario 1 (visible in blue in Figure 4.3) compared to Scenario 2 (visible in orange in Figure 4.3). This observation relates to the inability of the performance assessment method to explicitly handle occlusions and dynamic objects. The numerical results from the Scenarios are presented in Tables 4.1 and 4.2

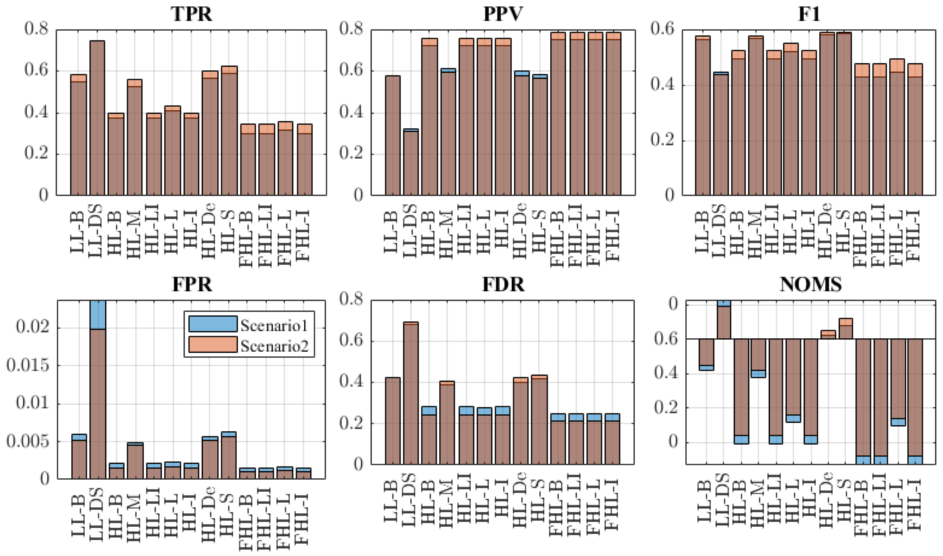


Figure 4.3: Colored bar plots of KPI results per scenario.

As seen in Figure 4.1. both of the Low-Level methods present high replication of the ground reflections occupancy, similar tendencies are observed for the High-Level De-Morgan and Instantaneous Sum, moreover, the tendency is visible also for obstacles such as barriers at higher ranges which represent higher angular inaccuracy of detections, reflected in the wider spread of occupancy probability. This effect is less pronounced in the High-Level Maximum, High-Level LOP and Accumulated High-Level LOP. The least susceptible methods are showing the lowest residue of the occupancy probability and in case of Accumulated approach include the Bayesian and LIOP along with IOP, the remaining instantaneous high-level approaches based on Bayesian, LIOP and IOP. The observation of the response of the algorithm indicates that in general the same methods but used in high level and especially filtered mode are less prone to those areas of increased occupancy. Similarly, the tendencies can be observed in Figure 4.2. The filtering methods in the Scenario number 2 seem to also suppress the false positives better, additional reason behind this is the fact that the test includes moving vehicles that are suppressed by the filtering approach.

Variant	**TPR**	**PPV**	**FPR**	**FDR**	**F1**	**NOMS**
LL-B	0.549	0.576	0.006	0.424	0.563	-0.18
LL-DS	0.745	0.319	0.024	0.681	0.447	0.23
HL-B	0.374	0.720	0.002	0.281	0.492	-0.61
HL-M	0.525	0.613	0.005	0.387	0.566	-0.22
HL-LI	0.374	0.720	0.002	0.281	0.492	-0.61
HL-L	0.408	0.723	0.002	0.277	0.522	-0.48
HL-I	0.374	0.720	0.002	0.281	0.492	-0.61
HL-De	0.564	0.598	0.006	0.403	0.580	0.02
HL-S	0.589	0.583	0.006	0.417	0.586	0.08
FHL-B	0.298	0.753	0.002	0.247	0.427	-0.73
FHL-LI	0.298	0.753	0.002	0.247	0.427	-0.73
FHL-L	0.319	0.750	0.002	0.250	0.448	-0.50
FHL-I	0.298	0.753	0.002	0.247	0.427	-0.73

Table 4.1: Scenario 1 KPIs for the configurations.

Variant	TPR	PPV	FPR	FDR	F1	NOMS
LL-B	0.581	0.575	0.005	0.425	0.578	-0.15
LL-DS	0.747	0.309	0.020	0.691	0.438	0.19
HL-B	0.398	0.759	0.002	0.241	0.522	-0.56
HL-M	0.559	0.594	0.005	0.406	0.576	-0.18
HL-LI	0.398	0.759	0.002	0.241	0.522	-0.56
HL-L	0.430	0.759	0.002	0.241	0.549	-0.44
HL-I	0.398	0.759	0.002	0.241	0.522	-0.56
HL-De	0.598	0.579	0.005	0.421	0.588	0.05
HL-S	0.621	0.564	0.006	0.436	0.591	0.12
FHL-B	0.343	0.786	0.001	0.215	0.477	-0.68
FHL-LI	0.343	0.786	0.001	0.215	0.477	-0.68
FHL-L	0.358	0.785	0.001	0.215	0.492	-0.46
FHL-I	0.343	0.786	0.001	0.215	0.477	-0.68

Table 4.2: Scenario 2 KPIs for the configurations.

During the course of the analysis of the probability representation presented in Figure 4.1, an inconsistency in occupied areas was identified in the right lane. Further investigation of the observed behavior led to the conclusion that deficiencies in the sensor simulation process resulted in false ground detections. Those deficiencies result in a reduction of reliability of the KPIs and create an opportunity for further development.

4.2. Real World Testing

As part of the research scope, an experimental evaluation in real-world conditions was performed. The primary goal behind the real-world testing was to check the feasibility of the proposed solution to operate as an embedded system on-board of the test vehicle and crosscheck the fact of the usefulness of virtual prototyping for the development of production intended software. The real-world testing was carried out on closed premises instead of actual traffic due to the prototype nature of the implementation. Due to the tests on closed premises, it was required to stick to speed limitations in the infrastructure used for testing. Due to that fact it was decided to focus on parking scenario applications as those tests are safer and require lower speeds than highway scenarios. The test was conducted in a parking lot near to Aptiv engineering offices. Aptiv's test vehicle has been driven through the parking lot with a constant speed of 10km/h in a straight line. The linear motion without turning has been chosen to minimize the impact of host position estimation on the results.

The reference data collected during the experiment consisted of videos and images recorded with a DJI Spark hobby drone. The area in the Field Of View (FOV) consisted of the parking view with the host vehicle and the parked vehicles along with the parking infrastructure. An example snapshot of the aerial data is presented in Figure 4.4.

Data collected during the experimental activities consisted of raw bus traffic in the test vehicle (including sensor data traffic and vehicle bus). The tests were repeated three times for consistency purposes. The data was recorded with a test system described in the Section 2.5. The recorded data was obtained from the ROS environment in a form of rosbag files. The files consisted of ROS topics with bus communication and internal information timestamps. The system run on-board of the vehicle was using the prototyped occupancy gird results with Bayesian fusion configuration in real-time. The results of the algorithms were also recorded for inspection purposes.

The main goal of those test activities was to register data to be used in further reprocessing in the HIL system described in Section 2.6. For the organization purpose of the thesis, the HIL reprocessing is also considered as part of the real-world testing.

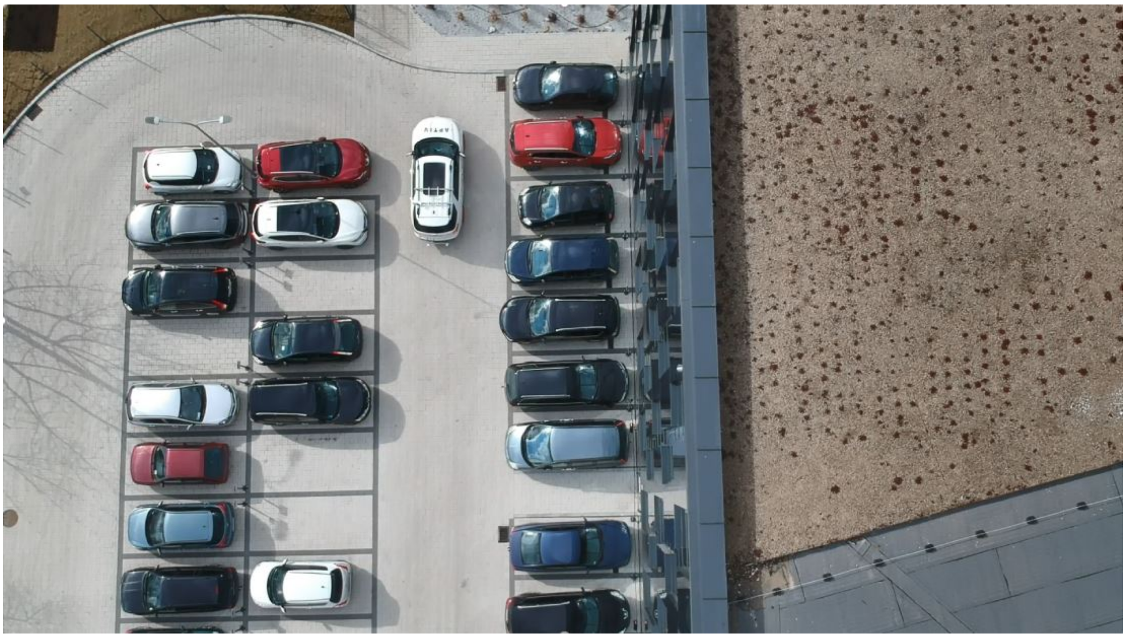
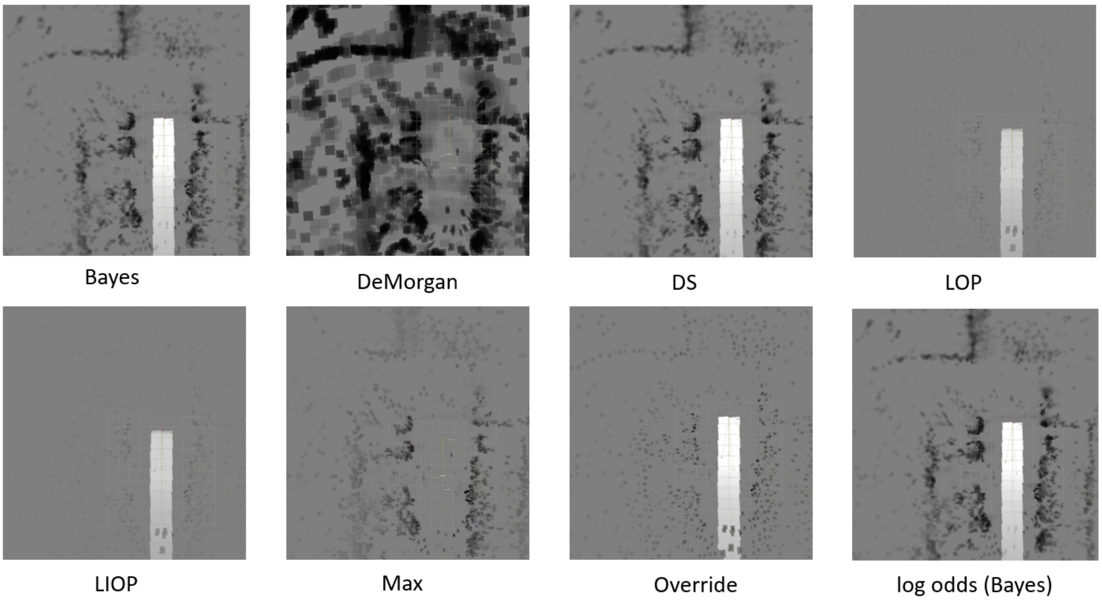


Figure 4.4: Aerial photo of the test vehicle moving through the test scenario area

Registered sensor data were reprocessed in the offline HIL environment to produce results from various selected configurations of the tested algorithms. The results produced by the tested variants of the algorithms are presented in Figure 4.5. The tested set of configurations includes:

- Bayes
- DeMorgan configuration
- DS - Dempster-Shafer approach
- LOP - Logarithmic Opinion Pool
- Max - Maximum probability configuration
- Override - an approach where the results from inverse sensor model probability was directly used in the grid by overriding the original values in the existing grid from the previous iteration.
- log-odds (Bayes) - an optimized implementation of the Bayesian approach where the logarithms of odds were used to represent both ISM output and the grid values, The operation carried out in this test was added to replicate the Bayesian equation in the log-odds domain. The results were converted to the probability domain to produce the presented plots.



Bayes DeMorgan DS LOP

LIOP Max Override log odds (Bayes)

Figure 4.5: The resultant grids for various methods in the real world test scenario

The resultant probability of the grids subjected to visual inspection indicates differences between the used methods. The three similarly performing variants are the Bayes, Dempster-Schafer, and log odds implementation of the Bayesian methods. The listed methods seem to be acceptable for further processing to determine parking space allocations. Results from the DeMorgan approach are heavily biased by the inverse sensor models resulting in false occupancy presented in the unoccupied spaces. The opinion pool-based methods seem to present a tendency to not properly replicate the occupied areas. A similar lack of precision is demonstrated by the override method. The maximum policy-based approach presents a good trade-off between the representation of the proper representation of occupied spaces to the false occupancy in the remaining area.

4.3. Execution Timing

The third part of the research scoped evaluation of the computation timing of the selected methods on PC hardware. In the previous steps, a subset of the best-suited methods was selected in terms of subjective assessment of performance by inspection of computational results of the methods on real-world experimental data. Two Bayesian approaches were selected (both the regular implementation and the log-odds variant)

Method:	Measured Time [ns]
Bayesian (probability)	115
Bayesian (log odds)	68
Dempster-Shafer	81

along with Dempster-Shafer. The selected agnostic libraries behind the implementation were executed on the test data to measure the execution timing of a single algorithm iteration. The time of execution was measured by registering the time in which the function working on data fusion starts until it is done with processing. The registered times com form a system clock using the ros::Time::now()[91]. The start and end values were subtracted to get the time it takes for one iteration of the algorithm to execute. The measured times were averaged for the complete test scenario to reduce the influence of time jitter related to underlying characteristics of the computation. The assessed time represents how long it takes to fuse two instances of input data using a given method. The hardware used for test execution is the same setup that was used to reprocess data in the HIL section. It is observed that the most time-demanding variant is the Bayesian using probability implementation. The second in terms of time needed for computation was the Dempster-Shafer method, this on the other hand most memory consuming due to the need of multiple matrices used for computations. The third and the fastest approach was the Bayesian method using log-odds implementation as the operations needed to execute this process is only the addition of two variables. The execution timing for the complete scan can be obtained as it scales linearly in a function of updated cells. The number of cells that require updating is dependent on the number of detections and the area of the probability distribution needed to cover detections uncertainty. This study found that all three methods allow for the system to be operated in real-time given the hardware platform chosen for the assessment. The results however can undergo a reduction in a result of optimization for the algorithm execution on Graphical Processing Unit (GPU), Digital Signal Processor (DSP), or dedicated Application Specific Integrated Circuits (ASIC).

5. Summary and Conclusions

The research presented in this dissertation investigated the use of an automotive virtual validation tool-chain for prototyping perception algorithms.

The work performed on the topic was done in several stages. The first stage incorporated literature and state of the art review consisting of exploring the existing fusion and occupancy grid architectures. The list of most common methods for static environment perception was compiled along with a review of prototyping environments.

In the second stage, a prototype implementation of a new occupancy grid suite was performed (including implementation of Matlab based framework for ISMs and Occupancy Grid Fusion algorithms). This stage utilized synthetic sensor data generated using Matlab random data generation tools. The steps for virtual validation were then joined to the system by utilization of the automotive simulation tooling. The virtual simulation scenarios were defined as the system use case scenarios. Then the scenarios were modeled in the Virtual Simulation tooling by placing actors and infrastructure, as well as the definition of the sensor suite. The sensor suite definition was performed in iterations to define the optimal mounting positions, FOVs and anticipated sensor types. The step of the definition of test cases consisted of adding the reference sensor to the suite. Then the implementation of sensor models based on real sensors chosen for the project was done in Matlab to handle outputs of the virtual validation tooling. The sensor modeling part was also implemented in Matlab based on the built-in randomization to add noise and uncertainty to generated sensors results. The next step consisted of the reference data tooling implementation that utilized the virtual validation reference sensor data to generate the reference high resolution grids. The tooling allowing execution of new fusion combination method and data handling pipelines were autocoded to mex functions and integrated into the Matlab framework[66]. The design choice to use this method was motivated by the large amount of data produced by all the components. Then the proposed KPIs were defined as a result of further literature review. The KPI tooling was implemented to be used on the data produced by the reference data generation tooling and the algorithms suite. The next step consisted of the integration of all the components together to allow batch reprocessing of the simulation data to obtain the first set of test results. After a small-scale experimental evaluation, a portfolio of scenarios was implemented and reprocessed by the complete suite. This stage confirms the thesis of this dissertation which is that it is possible to prototype, develop and comparatively assess state-of-the-art occupancy grid algorithms using virtual validation methods. The assessment performed in this stage indicated tendencies in the methods. The methods that are more prone to higher rates of false detections are also better in representing obstacles at further ranges, however, this results in a degraded performance presented in Figures in Chapter 4.1.

In the third stage, a migration of the algorithm part was performed. The developed algorithms were migrated to the platform-agnostic core library written in C++. The primary novel goal achieved in the implementation was focused on merging and evaluating variants of the fusion algorithms. In the next step, the agnostic library was wrapped to be used as executable in the ROS environment in a form of a configurable node. The HIL solution has been built to allow offline reprocessing of data. The project included real-world assessment in the test car. The car was reused from an existing project to allow data collection in real-world conditions along with the live execution of the tested algorithms. The system was ported to the test vehicle and real-world testing was conducted. After completion of the real-world tests, the data acquisition was performed and the collected samples were used in offline reprocessing to produce the real-world experimental results. After completion of real-world tests, the data was analyzed to assess the performance of the selected methods. The results of real-world tests included the assessment of computation time for the three selected methods. Similar to the assessment of the virtual validation the results showed differences between the tested fusion methods with similar tendencies of more susceptible methods to represent better performance at further distances and worse false detections in unoccupied spaces.

The main original contribution to the field being a direct result of the research work consists of proposed and implemented novel methods for combining existing methods, along with the novel performance assessment approach for use with virtual validation process, which were published by the author of this thesis in articles[50][51]. Original contribution consisting of the sensor modeling implemented in the research was based on a sensor suite review published in article[49]. The new proposed method that simplifies the inverse sensor modeling to 1D for optimization purposes has been covered by a patent number FR3097972B1[52]. Original engineering contribution performed in this work consists of vehicle instrumentation and data acquisition system build as well as execution of both virtual and real world experiments.

The key takeaways from the presented research indicate that the proposed virtual tool-chain employed in the experiment allows early identification, testing, and definition of architectures and system-level approaches. Previously mentioned takeaways led to the conclusion that such an approach enables quick development of the prototype software and verification of validation concepts. It has been also observed that the approach allows for early identification of architecture limitations. This factor is crucial for designing architectures as early identification of constraints is crucial for saving costs in the long term scope of the project.

The proposed tooling allows for integration and definition of the functional requirements when the hardware is not yet available which is a common case in the industry. The major advantages of the evaluated method are enabling complete repeatability in comparison to real-world testing and the generation of high precision reference data that would not be possible to obtain using a reference lidar mounted on a moving platform. These features are crucial for complex system providers for the definition of products to better correspond to the customer's technical needs.

The employed tool-chain will aid in the generation of dangerous scenarios, e.g. dense traffic, that would require vast effort and cost to be produced in the real world with controlled conditions. The tool-chain can also help in the definition of KPIs and the validation of their reliability. The limitations of this approach are related to its ability to realistically represent the critical properties of test scenarios and sensors. Those limitations, however, underline that the final validation and performance assessment of more technically mature products requires the extensive use of real data. Virtual validation frameworks will benefit from increasing the realism of sensor models, which will lead to more widespread adoption of virtual validation in perception algorithm development. The reduced ability to represent a physical phenomenon in the evaluated simulation process decreased the reliability of the proposed performance assessment scheme. This was reflected in the fact that a set of numerical KPIs has been identified to be not consistent with expert assessment. Numerical results obtained from the comparison indicate that the higher the sensitivity of the method, the more prone it is to clutter and noise. A valid testing methodology requires tailoring depending on the application of the perception module to yield useful results.

The research also showed that the solution used as a module *F3* described in Section 3.1 can be transferred to the ROS and executed in real-time on vehicle platforms. Real-world tests showed the tendency for fusion methods also observed in virtual validation. The PC-based implementation is still undergoing testing in a real-world environment. Further plans include integration of the perception chain to be used with actual data consumers. This method can be used for variety of potential applications. One prospective usage might be parking place detection for parking lot application. Another possible implementation can be detection of road edges and shoulder areas for driver assistance take-off action. The third prospective application is the usage of an estimated grid map to augment the vehicle positioning utilizing map matching algorithms.

Bibliography

[1] Kareem Abdelgawad et al. “Advanced Traffic Simulation Framework for Networked Driving Simulators”. In: *8th IFAC Symposium on Advances in Automotive Control (AAC 2016)*. June 2016.

[2] Hillary Abraham et al. “What’s in a Name: Vehicle Technology Branding Consumer Expectations for Automation”. In: *AutomotiveUI ’17: Proceedings of the 9th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. Sept. 2017. DOI: 10.1145/3122986.3123018.

[3] Evan Ackerman. “Lidar that will make self-driving cars affordable [News]”. In: *IEEE Spectrum* 53 (Oct. 2016), pp. 14–14. DOI: 10.1109/MSPEC.2016.7572525.

[4] Juan David Adarve et al. “Computing Occupancy Grids from Multiple Sensors using Linear Opinion Pools”. In: *IEEE International Conference on Robotics and Automation*. St Paul, Minnesota, United States, May 2012. URL: https://hal.inria.fr/hal-00671211.

[5] M. Aeberhard, Thomas Kuhbeck, and B. Seidl. “Automated Driving with ROS at BMW”. In: *ROSCon2015*. 2015.

[6] A. Aijazi et al. “Systematic Evaluation and Characterization of 3D Solid State Lidar Sensors For Autonomous Ground Vehicles”. In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLIII-B1-2020 (Aug. 2020), pp. 199–203. DOI: 10.5194/isprs-archives-XLIII-B1-2020-199-2020.

[7] M. S. Arulampalam et al. “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking”. In: *IEEE Transactions on Signal Processing* 50.2 (2002), pp. 174–188. DOI: 10.1109/78.978374.

[8] Y. Bar-Shalom, P.K. Willett, and X. Tian. *Tracking and Data Fusion: A Handbook of Algorithms*. YBS Publishing, 2011. ISBN: 9780964831278. URL: https://books.google.pl/books?id=2aOiuAAACAAJ.

[9] Guy Berg, Verena Nitsch, and Berthold Faerber. “Vehicle in the Loop”. In: *Handbook of Driver Assistance Systems*. Jan. 2016, pp. 199–210. DOI: 10.1007/978-3-319-12352-3_10.

[10] Niels Berger. “Lane Change Path Planning With state-dependent safety constraints using Nonlinear Model Predictive Control”. PhD thesis. Feb. 2018. DOI: 10.13140/RG.2.2.33869.33768.

[11] Richard Bradley. “Learning from others: conditioning versus averaging”. In: *Theory and Decision* 85.1 (2018), pp. 5–20. DOI: 10.1007/s11238-017-9615-y. URL: https://ideas.repec.org/a/kap/theord/v85y2018i1d10.1007_s11238-017-9615-y.html.

[12] Lindsay Brooke. “Delphi’s RACam integrated radar-vision system enables active-safety suites”. In: *SAE* (2015).

[13] Alessio Carullo and Marco Parvis. “An ultrasonic sensor for distance measurement in automotive applications”. In: *Sensors Journal, IEEE* 1 (Sept. 2001), pp. 143 –147. DOI: 10.1109/JSEN.2001.936931.

[14] João Carvalho and Rodrigo Ventura. “Comparative Evaluation of Occupancy Grid Mapping Methods Using Sonar Sensors”. In: *Pattern Recognition and Image Analysis*. Ed. by João M. Sanches, Luisa Micó, and Jaime S. Cardoso. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 889–896. ISBN: 978-3-642-38628-2.

[15] Alex Chen and Michael Sarazen. *Simulations Pave the Road for Self-Driving Technologies*. 2017. URL: https://medium.com/syncedreview/simulations-pave-the-road-for-self-driving-technologies-78b696227383 (visited on 08/06/2018).

[16] T. Colleens, J. J. Colleens, and D. Ryan. “Occupancy grid mapping: An empirical evaluation”. In: *2007 Mediterranean Conference on Control Automation*. 2007, pp. 1–6. DOI: 10.1109/MED.2007.4433772.

[17] On-Road Automated Driving (ORAD) committee. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. 2021. DOI: https://doi.org/10.4271/J3016_202104. URL: https://doi.org/10.4271/J3016_202104.

[18] Soumyo Das et al. "Path Tracking and Control for Parallel Parking". In: *2020 International Conference on Image Processing and Robotics (ICIP)*. 2020, pp. 1–6. DOI: 10.1109/ICIP48927.2020.9367343.

[19] Data.gov.uk. *UK Road Accidents Safety Data. Retrieved from Road Safety Data - Datasets*. 2015. URL: https://data.gov.uk/dataset/road-accidents-safety-data.

[20] Pär Degerman, Jochen Pohl, and Magnus Sethson. "Ultrasonic Sensor Modeling for Automatic Parallel Parking Systems in Passenger Cars". In: *Infotainment Systems-PT-135, IVI Technology and Intelligent Transportation Systems-SP-2099*. Apr. 2007. DOI: 10.4271/2007-01-1103.

[21] Morris H. Degroot. "Reaching a Consensus". In: *Journal of the American Statistical Association* 69.345 (1974), pp. 118–121. DOI: 10.1080/01621459.1974.10480137.

[22] Xuejun Ding and Yuping He. "Design of an Active Safety System for Multi-Trailer Articulated Heavy Vehicles Using Real-Time Simulations". In: *The 3rd International Conference on Digital Manufacturing Automation (ICDMA2012)*. Aug. 2012.

[23] Jianmin Duan et al. "Moving Objects Detection in Evidential Occupancy Grids Using Laser Radar". In: *2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*. Vol. 02. 2016, pp. 73–76. DOI: 10.1109/IHMSC.2016.211.

[24] eetimes. *Hannes Estl - Sensor fusion: A critical step on the road to autonomous vehicles (2016) EE Times Europe*. 2016. URL: http://www.electronics-eetimes.com/news/sensor-fusion-critical-step-road-autonomous-vehicles/page/0/4.

[25] A. Elfes and L. Matthies. "Sensor integration for robot navigation: Combining sonar and stereo range data in a grid-based representataion". In: *26th IEEE*

Conference on Decision and Control. Vol. 26. 1987, pp. 1802–1807. DOI: 10.1109/CDC.1987.272800.

[26] Pierre Exertier et al. “Contribution of laser ranging to Earth’s sciences”. In: *Comptes Rendus Geoscience* 338.14 (2006). La Terre observée depuis l’espace, pp. 958–967. ISSN: 1631-0713. DOI: https://doi.org/10.1016/j.crte.2006.09.019. URL: https://www.sciencedirect.com/science/article/pii/S1631071306002756.

[27] Uwe Franke et al. “Making Bertha See”. In: *2013 IEEE International Conference on Computer Vision Workshops*. 2013, pp. 214–221. DOI: 10.1109/ICCVW.2013.36.

[28] Florian Fölster and Hermann Rohling. “Signal processing structure for automotive radar”. In: *Frequenz* 60.1-2 (2006), pp. 20–24. DOI: doi:10.1515/FREQ.2006.60.1-2.20. URL: https://doi.org/10.1515/FREQ.2006.60.1-2.20.

[29] Jorge Godoy et al. “A Grid-Based Framework for Collective Perception in Autonomous Vehicles”. In: *Sensors* 21.3 (2021). ISSN: 1424-8220. DOI: 10.3390/s21030744. URL: https://www.mdpi.com/1424-8220/21/3/744.

[30] Yinglong He et al. “Adaptive Cruise Control Strategies Implemented on Experimental Vehicles: A Review”. In: *IFAC-PapersOnLine* 52.5 (2019). 9th IFAC Symposium on Advances in Automotive Control AAC 2019, pp. 21–27. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2019.09.004. URL: https://www.sciencedirect.com/science/article/pii/S2405896319306238.

[31] Karl Heckemann et al. “Safe Automotive Software”. In: *Knowledge-Based and Intelligent Information and Engineering Systems - 15th International Conference*. Vol. 6884. Sept. 2011, pp. 167–176. ISBN: 978-3-642-23865-9. DOI: 10.1007/978-3-642-23866-6_18.

[32] Daniel Heß, Matthias Althoff, and Thomas Sattel. “Should collision avoidance systems use yaw stabilization?” In: *Knowledge-Based and Intelligent Information and Engineering Systems - 15th International Conference, KES 2011, Kaiserslautern, Germany*. Oct. 2013, pp. 2058–2062. ISBN: 978-1-4799-2914-6. DOI: 10.1109/ITSC.2013.6728532.

[33] Ping-Min Hsu and Zhen-Wei Zhu. “Car trajectory prediction in image processing and control manners”. In: *2016 IEEE International Conference on Intelligent Transportation Engineering (ICITE)*. 2016, pp. 45–49. DOI: 10.1109/ICITE.2016.7581305.

[34] Yaofu Huang, Zengshan Tian, and Qing Jiang. “A Radar and Monocular Camera-Based Fusion Approach for Pedestrian Detection”. In: *The 2nd International Conference on Computing and Data Science*. New York, NY, USA: Association for Computing Machinery, 2021. ISBN: 9781450389570. URL: https://doi.org/10.1145/3448734.3450461.

[35] Norma Hubele and Kathryn Kennedy. “Forward collision warning system impact”. In: *Traffic Injury Prevention* 19.sup2 (2018). PMID: 30001148, S78–S83. DOI: 10.1080/15389588.2018.1490020. eprint: https://doi.org/10.1080/15389588.2018.1490020. URL: https://doi.org/10.1080/15389588.2018.1490020.

[36] Constantin Ilas. “Electronic sensing technologies for autonomous ground vehicles: A review”. In: *2013 - 8th International Symposium on Advanced Topics in Electrical Engineering, ATEE 2013*. May 2013, pp. 1–6. ISBN: 978-1-4673-5979-5. DOI: 10.1109/ATEE.2013.6563528.

[37] V. Ilic et al. “Development of Sensor Fusion Based ADAS Modules in Virtual Environments”. In: *2018 Zooming Innovation in Consumer Technologies Conference (ZINC)*. 2018, pp. 88–91. DOI: 10.1109/ZINC.2018.8448849.

[38] International Organization for Standardization. *ISO/IEC 15504:2003 Information technology – Process assessment – Part 2: Performing an assessment*. Tech. rep. International Organization for Standardization, 2003.

[39] R. Isermann, J. Schaffnit, and S. Sinsel. “Hardware-in-the-loop simulation for the design and testing of engine-control systems”. In: *Control Engineering Practice* 7.5 (1999), pp. 643–653. ISSN: 0967-0661. DOI: https://doi.org/10.1016/S0967-0661(98)00205-6. URL: https://www.sciencedirect.com/science/article/pii/S0967066198002056.

[40] ISO. *Road vehicles – Functional safety*. Norm. 2011.

[41] Prem Jain. “Trends in Next Generation Intelligent Transportation Systems”. In: *Self-Driving Vehicles and Enabling Technologies*. June 2021. DOI: 10.5772/intechopen.97690.

[42] Youngseok Jin et al. “Design and Implementation of FMCW Surveillance Radar Based on Dual Chirps”. In: *Elektronika ir Elektrotechnika* 24 (Dec. 2018). DOI: 10.5755/j01.eie.24.6.22292.

[43] Katharina Juhnke, Matthias Tichy, and Frank Houdek. “Challenges Concerning Test Case Specifications in Automotive Software Testing”. In: *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. 2018, pp. 33–40. DOI: 10.1109/SEAA.2018.00015.

[44] R. Jungnickel, M. Kohler, and F. Korf. “Efficient automotive grid maps using a sensor ray based refinement process”. In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. 2016, pp. 668–675. DOI: 10.1109/IVS.2016.7535459.

[45] HyeongRyeol Kam et al. “RViz: a toolkit for real domain data visualization”. In: *Telecommunication Systems* 60 (Oct. 2015), pp. 1–9. DOI: 10.1007/s11235-015-0034-5.

[46] Dae Jung Kim and Chung Choo Chung. “Automated Perpendicular Parking System With Approximated Clothoid-Based Local Path Planning”. In: *IEEE Control Systems Letters* 5.6 (2021), pp. 1940–1945. DOI: 10.1109/LCSYS.2020.3044254.

[47] Glyn Lawson, Davide Salanitri, and Brian Waterfield. “VR Processes in the Automotive Industry”. In: *17th International Conference, HCI International*. Sept. 2015. ISBN: 978-3-319-21005-6. DOI: 10.1007/978-3-319-21006-3_21.

[48] Mark Mages, Felix Klanner, and Alexander Stoff. “Intersection Assistance”. In: *Handbook of Driver Assistance Systems: Basic Information, Components and Systems for Active Safety and Comfort*. Ed. by Hermann Winner et al. Cham: Springer International Publishing, 2016, pp. 1259–1286. ISBN: 978-3-319-12352-3. DOI: 10.1007/978-3-319-12352-3_51. URL: https://doi.org/10.1007/978-3-319-12352-3_51.

[49] Paweł Markiewicz, Marek Długosz, and Paweł Skruch. “Review of tracking and object detection systems for advanced driver assistance and autonomous driving applications with focus on vulnerable road users sensing”. In: *Trends in Advanced Intelligent Control, Optimization and Automation*. Ed. by Wojciech Mitkowski et al. Cham: Springer International Publishing, 2017, pp. 224–237. ISBN: 978-3-319-60699-6.

[50] Paweł Markiewicz and Jakub Porębski. “Developing Occupancy Grid with Automotive Simulation Environment”. In: *Applied Sciences* 10.21 (2020). ISSN: 2076-3417. DOI: 10.3390/app10217629. URL: https://www.mdpi.com/2076-3417/10/21/7629.

[51] Paweł Markiewicz et al. “Occupancy Grid Fusion Prototyping Using Automotive Virtual Validation Environment”. In: ICCMA 2018 (2018), pp. 81–85. DOI: 10.1145/3284516.3284540. URL: http://doi.acm.org/10.1145/3284516.3284540.

[52] Paweł Markiewicz, Jakub Porębski, and Dominik Sasin. *Procédé et système pour cartographier un environnement physique au moyen d’une grille d’occupation*. 2021. URL: https://patents.google.com/patent/FR3097972B1/.

[53] Mathworks. *Documentation - Robotics System Toolbox - Occupancy Grids*. 2018. URL: https://www.mathworks.com/help/robotics/ug/occupancy-grids.html (visited on 08/06/2018).

[54] Sanket Mehta, Arpita Patel, and Jagrat Mehta. “CCD or CMOS Image sensor for photography”. In: *2015 International Conference on Communications and Signal Processing (ICCSP)* (2015), pp. 0291–0294.

[55] Andrew Miller et al. *Euro NCAP 2025 Roadmap - IN PURSUIT OF VISION ZERO*. 2017. URL: https://cdn.euroncap.com/media/30700/euroncap-roadmap-2025-v4.pdf.

[56] David Miller and Wendy Ju. “Does The First Officer Concur? Shared Control with Smart Vehicle Systems”. In: *AutomotiveUI ’14: Adjunct Proceedings of the 6th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. Sept. 2014, pp. 1–6. DOI: 10.1145/2667239.2667286.

[57] Abdul Sajeed Mohammed et al. “The Perception System of Intelligent Ground Vehicles in All Weather Conditions: A Systematic Literature Review”. In: *Sensors* 20.22 (2020). ISSN: 1424-8220. DOI: 10.3390/s20226532. URL: https://www.mdpi.com/1424-8220/20/22/6532.

[58] H. Moravec and A. Elfes. “High resolution maps from wide angle sonar”. In: *Proceedings. 1985 IEEE International Conference on Robotics and Automation*. Vol. 2. 1985, pp. 116–121. DOI: 10.1109/ROBOT.1985.1087316.

[59] Hans Moravec. “Sensor Fusion in Certainty Grids for Mobile Robots”. In: *AI Mag.* 9.2 (July 1988), pp. 61–74. ISSN: 0738-4602. URL: http://dl.acm.org/citation.cfm?id=46184.46187.

[60] Mongi A. Abidi Muhamad Abdulghafour. “Data fusion through nondeterministic approaches: a comparison”. In: *Proceedings Volume 2059, Sensor Fusion VI*. Vol. 2059. 1993. DOI: 10.1117/12.150253. URL: https://doi.org/10.1117/12.150253.

[61] Jordan Navarro, Franck Mars, and Mark Young. “Lateral control assistance in car driving: Classification, review and future prospects”. In: *Intelligent Transport Systems, IET* 5 (Oct. 2011), pp. 207 –220. DOI: 10.1049/iet-its.2010.0087.

[62] Sandra Nowok et al. “Millimeter wave radar for high resolution 3D near field imaging for robotics and security scans”. In: *International Radar Symposium IRS 2017*. June 2017. DOI: 10.23919/IRS.2017.8008132.

[63] Ronan O’Malley, Martin Glavin, and Edward Jones. “A review of automotive infrared pedestrian detection techniques”. In: *Signals and Systems Conference, 208*. July 2008, pp. 168 –173. DOI: 10.1049/cp:20080657.

[64] Ü. Özgüner, T. Acarman, and K. Redmill. *Autonomous Ground Vehicles*. Artech House ITS series. Artech House, 2011. ISBN: 9781608071920. URL: https://books.google.pl/books?id=OTF-uQAACAAJ.

[65] Min Woo Park, Kyung Jang, and Soon Jung. “Panoramic Vision System to Eliminate Driver’s Blind Spots using a Laser Sensor and Cameras”. In: *International Journal of Intelligent Transportation Systems Research* 10 (Aug. 2012), pp. 101–114. DOI: 10.1007/s13177-012-0046-4.

[66] Nikolaos Ploskas and Nikolaos Samaras. "MATLAB MEX functions containing CUDA code". In: *GPU Programming in MATLAB*. Dec. 2016, pp. 219–240. ISBN: 978-0-12-805132-0. DOI: 10.1016/B978-0-12-805132-0.00008-4.

[67] Jakub Porebski et al. "Occupancy grid for static environment perception in series automotive applications". In: *IFAC-PapersOnLine* 52.8 (2019). 10th IFAC Symposium on Intelligent Autonomous Vehicles IAV 2019, pp. 148 –153. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2019.08.063. URL: http://www.sciencedirect.com/science/article/pii/S2405896319303945.

[68] Jakub Porębski and Krzysztof Kogut. "Performance Evaluation of the Highway Radar Occupancy Grid". In: *Sensors* 21 (Mar. 2021), p. 2177. DOI: 10.3390/s21062177.

[69] Carlos Galvez del Postigo Fernandez. "Grid-Based Multi-Sensor Fusion for On-Road Obstacle Detection: Application to Autonomous Driving". MA thesis. 2015. URL: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-173316.

[70] Ralph H. Rasshofer. "Functional requirements of future automotive radar systems". In: *2007 European Radar Conference*. 2007, pp. 259–262. DOI: 10.1109/EURAD.2007.4404986.

[71] Gerhard Rollmann et al. "Short Range Radar System for Automotive Applications". In: *IEEE International Conference on Ultra-Wideband ICUWB*. Jan. 2003, pp. 215–221. ISBN: 978-3-540-00597-1. DOI: 10.1007/978-3-540-76988-0_18.

[72] Mario Sabbatelli et al. "Real-Time Obstacle Detection Using Stereo Vision for Autonomous Ground Vehicles: A Survey". In: *IEEE International Conference on Intelligent Transportation Systems*. Oct. 2014. DOI: 10.1109/ITSC.2014.6957799.

[73] J. Schauer and A. Nüchter. "The Peopleremover - Removing Dynamic Objects From 3-D Point Cloud Data by Traversing a Voxel Occupancy Grid". In: *IEEE Robotics and Automation Letters* 3.3 (2018), pp. 1679–1686. ISSN: 2377-3766. DOI: 10.1109/LRA.2018.2801797.

[74] Bobbie Seppelt et al. "Considering the Human Across Levels of Automation: Implications for Reliance". In: *Driving Assessment Conference*. Nov. 2017, pp. 228–234.

[75] Abdullah Omer Sevil, Mehmet Canevi, and Mehmet Turan Soylemez. "Development of an Adaptive Autonomous Emergency Braking System Based on Road Friction". In: *2019 11th International Conference on Electrical and Electronics Engineering (ELECO)*. 2019, pp. 815–819. DOI: 10.23919/ELECO47770.2019.8990620.

[76] Markus Sieber et al. "Automatic Emergency Steering with Distracted Drivers: Effects of Intervention Design". In: *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. 2015, pp. 2040–2045. DOI: 10.1109/ITSC.2015.330.

[77] Pawel Skruch et al. "The Simulation Strategy and Its Realization in the Development Process of Active Safety and Advanced Driver Assistance Systems". In: *SAE Technical Paper*. SAE International, 2015. DOI: 10.4271/2015-01-1401. URL: https://DOI.org/10.4271/2015-01-1401.

[78] Sandeep Sovani. "Simulation Accelerates Development of Autonomous Driving". In: *ATZ worldwide* 119 (Sept. 2017), pp. 24–29. DOI: 10.1007/s38311-017-0088-y.

[79] Leo Stanislas and Thierry Peynot. "Characterisation of the Delphi Electronically Scanning Radar for robotics applications". In: *Australasian Conference on Robotics and Automation (ACRA 2015)*. Canberra, A.C.T, 2015. URL: https://eprints.qut.edu.au/92567/.

[80] Emese Szádeczky-Kardoss and Bálint Kiss. "Path Planning and Tracking Control for an Automatic Parking Assist System". In: *European Robotics Symposium 2008*. Vol. 44. Dec. 2008, pp. 175–184. ISBN: 978-3-540-78315-2. DOI: 10.1007/978-3-540-78317-6_18.

[81] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. ISBN: 0262201623.

[82] Antonio Tierno et al. "Open issues for the automotive software testing". In: *2016 12th IEEE International Conference on Industry Applications (INDUSCON)*. Nov. 2016, pp. 1–8. DOI: 10.1109/INDUSCON.2016.7874609.

[83] Sandra Trösterer et al. "Using a Parking Assist System Over Time: Insights on Acceptance and Experiences". In: *Proceedings of the 6th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. AutomotiveUI '14. Seattle, WA, USA: Association for Computing Machinery, 2014, 1–8. ISBN: 9781450332125. DOI: 10.1145/2667317.2667327. URL: https://doi.org/10.1145/2667317.2667327.

[84] D.-C Tseng, Yu-Chi Lin, and Tat-Wa Chao. "Wide-scoped Surrounding Top-view Monitor for Advanced Driver Assistance Systems". In: *2015 4th International Conference on Mechatronics, Materials, Chemistry and Computer Engineering*. Jan. 2015. DOI: 10.2991/icmmcce-15.2015.90.

[85] Omer Tsimhoni, Jonas Bärgman, and Michael Flannagan. "Pedestrian Detection with near and far Infrared Night Vision Enhancement". In: *LEUKOS: The Journal of the Illuminating Engineering Society of North America* 4 (Sept. 2013), pp. 113–128. DOI: 10.1582/LEUKOS.2007.04.02.003.

[86] UNECE. *Agreement concerning the Adoption of Harmonized Technical United Nations Regulations for Wheeled Vehicles, Equipment and Parts which can be Fitted and/or be Used on Wheeled Vehicles and the Conditions for Reciprocal Recognition of Approvals Granted on the Basis of these United Nations Regulations*. 2017. URL: https://unece.org/fileadmin/DAM/trans/main/wp29/wp29regs/2020/R139am2e.pdf.

[87] Gorka Velez et al. "On creating vision-based advanced driver assistance systems". In: *IET Intelligent Transport Systems* 9 (Feb. 2015), pp. 59–66. DOI: 10.1049/iet-its.2013.0167.

[88] Z. Wang et al. *Measurement Data Modeling and Parameter Estimation*. Boca Raton: CRC Press, 2012, p. 290.

[89] Erik Ward and John Folkesson. "Vehicle localization with low cost radar sensors". In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. 2016, pp. 864–870. DOI: 10.1109/IVS.2016.7535489.

[90] Yannik Weber and Stratis Kanarachos. "The Correlation between Vehicle Vertical Dynamics and Deep Learning-Based Visual Target State Estimation: A Sensitivity Study". In: *Sensors* 19 (Nov. 2019), p. 4870. DOI: 10.3390/s19224870.

[91] Bryce Willey. *ROS Documentation*. 2017. URL: http://wiki.ros.org/roscpp/Overview/Time.

[92] Huadong Wu et al. "Fusion using Dempster-Shafer theory [for context-aware HCI]". In: *Conference Record - IEEE Instrumentation and Measurement Technology Conference*. Vol. 1. Feb. 2002, 7 –12 vol.1. ISBN: 0-7803-7218-2. DOI: 10.1109/IMTC.2002.1006807.

[93] Tzu-Sung Wu. "Pedestrian and Vehicle Recognition Based on Radar for Autonomous Emergency Braking". In: *WCX™ 17: SAE World Congress Experience*. Mar. 2017. DOI: 10.4271/2017-01-1405.

[94] Chunlei Yi, Kunfan Zhang, and Nengling Peng. "A multi-sensor fusion and object tracking algorithm for self-driving vehicles". In: *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 233 (Aug. 2019), pp. 2293–2300. DOI: 10.1177/0954407019867492.

[95] Buyue Zhang et al. "A Surround View Camera Solution for Embedded Systems". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. June 2014, pp. 676–681. DOI: 10.1109/CVPRW.2014.103.

[96] Adam Ziębiński et al. "Review of advanced driver assistance systems (ADAS)". In: *AIP Conference Proceedings 1906, 120002*. Vol. 1906. Nov. 2017, p. 120002. DOI: 10.1063/1.5012394.

Acronyms

1D 1-dimensional. 7, 38, 71, 76–84, 107

2D 2-dimensional. 37, 38, 76, 85

ACC Adaptive/Active Cruise Control. 15, 16, 21, 25, 33, 128

ADAS Advanced Driver Assistance Systems. 8

AEB Autonomous Emergency Braking. 12–14, 16, 128

AES Automatic Emergency Steering. 14

ASIC Application Specific Integrated Circuits. 104

CAN Controller Area Network. 58

D-S Dempster-Shafer. 89

DMS Driver Monitoring Systems. 21

DSP Digital Signal Processor. 104

EBA Emergency Brake Assist. 14

ECM Engine Control Module. 60

ESR Electronically Scanning Radar. 74

F1 harmonic mean of precision and accuracy. 63, 99, 100

FCW Forward Collision Warning. 14, 16

FDR False Discovery Rate. 63, 95, 99, 100

FFT Fast Fourier Transform. 25

FHL-B High Level Accumulated Bayes. 89, 96, 97, 99, 131

FHL-I High Level Accumulated Independent Opinion Pool. 88, 89, 96, 97, 99, 131

FHL-L High Level Accumulated Linear Opinion Pool. 88, 89, 96, 97, 99, 131

FHL-LI High Level Accumulated Logarithmic Independent Opinion Pool. 88, 89, 96, 97, 99, 131

FIR Far Infrared. 29

FLL Forward Looking Lidar. 67, 130

FLR Forward Looking Radar. 67, 130

FN False Negative. 65, 92

FOV Field Of View. 101, 105

FP False Positives. 52, 65, 92

FPR False Positive Rate. 63, 99, 100

GPU Graphical Processing Unit. 30, 104

HCV Heavy Commercial Vehicle. 23

HIL Hardware in The Loop. 4–6, 10, 60, 61, 75, 93, 101, 102, 104, 106

HL-B High Level Instantaneous Bayes. 89, 96, 97, 99, 131

HL-De High Level Instantaneous DeMorgan. 88, 89, 96, 97, 99, 131

HL-I High Level Instantaneous Independent Opinion Pool. 88, 89, 96, 97, 99, 131

HL-L High Level Instantaneous Linear Opinion Pool. 88, 89, 96, 97, 99, 131

HL-LI High Level Instantaneous Logarithmic Independent Opinion Pool. 88, 89, 96, 97, 99, 131

HL-M High Level Instantaneous Max. 88, 89, 96, 97, 99, 131

HL-S High Level Instantaneous Sum. 88, 89, 96, 97, 99, 131

INS INS (Inertial Navigation Systems. 40

IOP Independent Opinion Pool. 48, 89, 98

IR infrared. 28, 29

ISM Inverse Sensor Model. 10, 37, 105

KPI Key Performance Indicator. 7, 62, 65, 92, 98, 100, 106–108, 131

LIOP Logarithmic Independent Opinion Pool. 89, 98

LKA Lane Keeping Aid. 15, 16, 21, 33, 128

LL-B Low Level Accumulated Bayes. 89, 96, 97, 99, 131

LL-DS Low Level Accumulated Dempster-Shafer. 89, 96, 97, 99, 131

LOP Linear Opinion Pool. 89, 98, 102

MRR Medium Range Radar. 57, 130

NCAP New Car Assessment Program. 6, 11, 12, 128

NIR Near Infrared. 29

NOMS Normalized Occupied Map Score. 63, 99, 100

ODD Operational Driving Domain. 21

OGCS Occupancy Grid Coordinate System. 39, 40, 76, 77

PCB Printed Circuit Boards. 20

PDC Parking Distance Control. 20, 25, 32

PDF Probability Density Function. 40, 80

PPV Positive Predictive Value. 62, 95, 99, 100

RACam Integrated Radar and Camera. 33, 129

RGB Red Green Blue. 28

RGB-IR Red Green Blue and IR. 29, 30, 129

ROS Robot Operating System. 56, 58, 60, 101, 108

RWUP Real World User Profile. 55, 56

SCS Sensor Coordinate System. 39, 40, 77

SLAM Simultaneous Localization And Mapping. 35

SOC System On Chip. 60

SPICE Software Process Improvement and Capability dEtermination. 64

SRR Short Range Radar. 57, 67, 130

SUT System Under Test. 60–62, 94

SUV Sport Utility Vehicle. 58

TN True Negative. 65, 92

TP True Positive. 65, 92, 93, 131

TPR True Positive Rate. 62, 95, 99, 100

TRL Technology Readiness Level. 64

VGA Video Graphics Array. 27

VRU Vulnerable Road Users. 13, 14, 33

Symbols

A	grid composed of cells with a probability gradient	63
α	detection azimuth in SCS	77
B	filtered reference grid	63
e_p	detection existence probability	39
g_{ij}	occupancy grid	35
$l(m)$	logarithm of odds of occupancy given the measurement m	45
$l(m_1)$	logarithm of odds of occupancy from sensor number 1	45
$l(m_2)$	logarithm of odds of occupancy from sensor number 2	45
λ_1	first eigenvalue	40
λ_2	second eigenvalue	40
$m(A)$	Basic Probability Assignment function of the mass A	47
$m(B)$	Basic Probability Assignment function of the mass B	47
m_1	information source number one	46

$m_{12}(X)$	belief function	47
m_2	information source number two	46
$m(E)$	mass of cell being empty	46
$m(O)$	mass of cell being occupied	46
μ	mean value of the distribution	39
$\mu_x\mu_y$	detection x and y coordinates in OGCS (since the location of SCS with respect to OGCS is known using r, $\alpha andthatlocation)$	77
N_L	grid length expressed in number of cells	35
N_{Lr}	length of the reference gird	90
N_W	grid width expressed in number of cells	35
N_{Wr}	width of the reference gird	90
$p(m)$	probability of occupancy given the measurement m	45
p_i	probability input from sensor	48
$p(m_t\|z_{1:t}, x_{1:t})$	posterior probability of occupancy	44
p_{max}	maximum probability	85
p_{min}	minimum probability	85
pdf$(X; \mu, \Sigma_{ogcs})$	probability density function for the calculation of the occupancy	39
r	detection range in SCS	77
σ_α	standard deviation of range measurement	77

Σ_{ogcs}	covariance matrix in occupancy grid coordinates system	39
σ_r	standard deviation of range measurement	77
Σ_{scs}	covariance matrix in sensor coordinates system	39
$\vec{v}_1$	first eigenvector	40
$\vec{v}_2$	second eigenvector	40
$\vec{v}_{max}$	eigenvector corresponding to maximum eigenvalue	41
w_i	weight of sensors contribution	48
X	position vector	39
$x_{det_{ogcs}}$	mean position in x	39
X	proposition	47
x_i	i-th x position	39
x_{scs}	position in x axis in sensor coordinates system	39
$x_{1:t}$	host poses until time t	45
$y_{det_{ogcs}}$	mean position in y	39
y_i	i-th y position	39
y_{scs}	position in y axis in sensor coordinate system	39
$z_{1:t}$	sensor measurements until time t	44

List of Figures

List of Tables

Notes