



AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY

FIELD OF SCIENCE: ENGINEERING AND TECHNOLOGY

SCIENTIFIC DISCIPLINE: AUTOMATION, ELECTRONIC AND ELECTRICAL
ENGINEERING

DOCTORAL DISSERTATION

Occupancy grid environmental modeling for automotive applications

Author: *Jakub Porębski*

Supervisor: *dr hab. inż. Paweł Skruch, prof.AGH*

Assisting supervisor: *dr inż. Krzysztof Kogut*

Completed in: *AGH University of Science and Technology
Faculty of Electrical Engineering, Automatics, Computer Science
and Biomedical Engineering
Department of Automatic Control and Robotics*

Kraków, 2022



AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

DZIEDZINA NAUK INŻYNIERYJNO-TECHNICZNYCH

DYSCYPLINA AUTOMATYKA, ELEKTRONIKA I ELEKTROTECHNIKA

ROZPRAWA DOKTORSKA

*Modelowanie otoczenia pojazdu za pomocą siatek zajętości
w zastosowaniach automatyki samochodowej*

Autor: *Jakub Porębski*

Promotor pracy: *dr hab. inż. Paweł Skruch, prof.AGH*

Promotor pomocniczy: *dr inż. Krzysztof Kogut*

Praca wykonana: *Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie
Wydział Elektrotechniki, Automatyki, Informatyki
i Inżynierii Biomedycznej
Katedra Automatyki i Robotyki*

Kraków, 2022

The acknowledgments were removed in the online version for privacy reasons.

Preamble

This research is a result of an industrial PhD program, funded by the Polish Ministry of Science and Higher Education (MNiSW), project number 0014/DW/2018/02, and carried out in cooperation with Aptiv Services Poland S.A., Technical Center Kraków and AGH University of Science and Technology, Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering.

Abstract

Reliable vehicle environmental modeling is a fundamental prerequisite for advanced driver assistance and autonomous driving systems. A commonly used representation of environment is the occupancy grid map. It divides the vehicle surroundings into a grid of cells and estimates the occupancy state for each cell, assuming the cells are independent of each other. This perception system builds a model of the driving environment by fusing measurements from multiple perceptual sensors including LIDARs, radars, vision sensors, etc.

In state-of-the-art research, many methods have been developed to build a perception system using the occupancy grid, but these methods are only tailored to specific sensor configurations, and their performance is not verified on the existing automotive hardware. Nowadays, car manufacturers, within a single model, offer a wide range of vehicle options that may differ in mounted sensors or driving comfort features. The occupancy grid algorithm offers a single point of reference for environment perception and can work with almost any set of sensors. This in turn can enable cost reduction of the overall system.

This thesis proposes a modular and scalable architecture for the occupancy grid algorithm that can be customized to a variety of sensor configurations, allowing the proposed solution to be easily adapted to different vehicle variants. Presented architecture allows for an effective reduction of measurement uncertainties, which is confirmed by experiments in this work. Guidelines for tuning the occupancy mesh algorithm are also described. Furthermore, the first application of the Dezert-Smarandache fusion method in a three-state automotive occupancy grid is shown. The results of the dissertation show a successful integration of the occupancy grid algorithm in the automated vehicle that can be easily implemented in production.

Keywords: occupancy grid; automotive; perception; environmental modeling; sensor fusion; Dezert-Smarandache rule of combination

Streszczenie

Wiarygodne modelowanie środowiska pojazdu jest podstawowym warunkiem wstępnym dla zaawansowanych systemów wspomagania kierowcy i jazdy autonomicznej. Powszechnie stosowaną formą reprezentacji środowiska jest mapa siatki zajętości (*ang. occupancy grid*). Dzieli ona środowisko na siatkę komórek i szacuje stan zajętości dla każdej komórki, zakładając, że komórki siatki są od siebie niezależne. Ten system percepcji buduje model środowiska jazdy poprzez połączenie pomiarów z wielu czujników percepcyjnych, w tym LIDAR-ów, radarów, czujników wizyjnych itp.

W najnowocześniejszych badaniach opracowano wiele metod budowy systemu percepcji wykorzystujących siatki zajętości, jednak metody te są dostosowane tylko do określonych konfiguracji czujników, a ich wydajność nie jest weryfikowana na istniejącym sprzęcie samochodowym. Obecnie producenci samochodów, w ramach jednego modelu, oferują szeroki zakres opcji, które mogą różnić się zamontowanymi czujnikami lub funkcjami zwiększającymi komfort jazdy. Algorytm siatki zajętości oferuje pojedynczy punkt odniesienia dla percepcji otoczenia i może współpracować z niemal dowolnym zestawem czujników, co z kolei może umożliwić redukcję kosztów całego systemu.

W niniejszej pracy zaproponowano modułarną i skalowalną architekturę dla algorytmu siatki zajętości, która może być dostosowana do różnych konfiguracji czujników, co pozwala na łatwe dostosowanie proponowanego rozwiązania do różnych wariantów pojazdów. Prezentowana architektura pozwala na efektywną redukcję niepewności pomiarowych, co zostało potwierdzone eksperymentalnie w niniejszej pracy. Opisano również wskazówki dotyczące dostrajania opisanego algorytmu. Ponadto przedstawiono pierwsze zastosowanie metody fuzji Dezerta-Smarandache'a w trójstanowej samochodowej siatce zajętości. Wyniki rozprawy przedstawiają udaną integrację algorytmu siatki zajętości w zautomatyzowanym pojeździe, który może być łatwo wdrożony do produkcji.

Słowa kluczowe: mapa zajętości; przemysł samochodowy; systemy percepcji; modelowanie środowiska; fuzja sensoryczna; metoda kombinacji Dezerta-Smarandache'a

Contents

List of abbreviations	17
Mathematical notation	19
1 Introduction	21
1.1 Objectives	21
1.2 Contributions	22
1.3 Outline	22
2 Automated driving systems	25
2.1 Historical background	25
2.2 Automation levels.....	26
2.2.1 Human driver in control.....	27
2.2.2 Automated driving systems in control	28
2.3 Automotive systems development process	29
2.3.1 The V-model	29
2.3.2 Standardization of automotive systems.....	31
2.4 Tasks of the automated driving systems	31
2.4.1 Positioning	32
2.4.2 Environment sensing.....	32
2.4.3 Perception systems.....	33
2.4.4 Motion planning systems	33
2.5 Occupancy grid application in automotive	34
3 Occupancy grid framework	37
3.1 Foundation of the occupancy grid	37
3.2 State-of-the-art of automotive occupancy grids	40
3.2.1 Dynamic occupancy grid	41
3.2.2 Multidimensional representations.....	41
3.2.3 Multi resolution occupancy maps	41
3.3 Occupancy grid architectures	42
3.4 Occupancy grid data flow	44
3.4.1 Sensor information processing.....	44
3.4.2 Sensor modeling.....	46
3.4.3 Evidence fusion and accumulation	49
3.4.4 Decay	49
3.4.5 Grid shift	50
3.5 Sensor conflict resolution	52
3.5.1 Self-sensor conflicts	52
3.5.2 Multi sensor conflicts.....	52

4	Heterogeneous information fusion	53
4.1	Probability and uncertainty representation formalisms	53
4.1.1	Bayesian probability framework	54
4.1.2	Dempster-Shafer Theory (DST)	54
4.1.3	Dezert-Smarandache Theory (DSmT)	57
4.2	Heterogeneous fusion rules	58
4.2.1	Measurement evidence representation	60
4.2.2	Bayesian framework fusion rules	60
4.2.3	Combination rules for the Dempster Shafer evidence framework	61
4.2.4	Combination rules for the Dezert-Smarandache evidence framework	62
4.2.5	Visualization of the occupancy grids	63
4.2.6	Quantitative evaluation of contradictory evidences	65
4.3	Occupancy grid fusion	67
4.3.1	Evaluated fusion architectures	67
4.3.2	Fusion rules comparison	71
5	Occupancy grid noise filtering	77
5.1	Types of uncertainty	77
5.1.1	Data synchronization delays	77
5.1.2	Sensor detection characteristics	78
5.1.3	Sensor mounting vibrations	79
5.1.4	Host positioning uncertainties	79
5.2	Assessing occupancy grid mapping quality	79
5.2.1	Extraction of interesting occupancy grid features	80
5.2.2	Definition of grid quality metrics	83
5.3	Occupancy grid filtering capabilities	85
5.3.1	Sensor detection modeling	85
5.3.2	Free space estimation filtering	86
5.3.3	Dealing with modeling overconfidence	87
5.3.4	Sensor synchronization	87
6	Simulation evaluation	89
6.1	Simulation environment	89
6.1.1	Driving Scenario Designer	89
6.1.2	Sensor uncertainty simulation	90
6.1.3	Occupancy grid configuration used for the simulation evaluation	91
6.2	Comparison of different uncertainty types on the occupancy grid	92
6.2.1	Qualitative evaluation of noise types	92
6.2.2	Quantitative evaluation of noise types	94
6.3	Uncertainty filtering assessment	95
6.3.1	Inverse sensor model types	96
6.3.2	Decay filtering	97
6.3.3	Free space modeling	98
6.3.4	Fusion rules comparison	100
6.3.5	Combination of filtering methods	102

6.3.6	Grid uncertainty filtering conclusions.....	106
7	Experimental evaluation	107
7.1	Experimental setup	107
7.2	Example occupancy grid tuning procedure	108
7.2.1	Selection of proper grid dimensions	108
7.2.2	Selection of inverse sensor model parameters	109
7.2.3	Tuning of decay and free space modeling filters	110
7.2.4	Occupancy grid before and after tuning.....	112
7.3	Sensor conflict on different occupancy grids.....	113
7.3.1	Bayesian occupancy grid	113
7.3.2	Dempster-Shafer occupancy grid.....	114
7.4	Experimental evaluation of different fusion rules	115
7.5	Performance evaluation of the algorithm	119
7.6	Execution on different combinations of automotive sensors.....	120
8	Conclusions and future work	123
8.1	Summary.....	123
8.2	Conclusions	123
8.3	Suggestions for future work	124
A	Fusion matrices and equations	125
A.1	Basic definitions	125
A.1.1	Dempster-Shafer model definition	125
A.1.2	Dezert-Smarandache model definition.....	126
A.2	Dempster-Shafer conjunctive rule	126
A.3	DSm common rule of combination for free-DSm models	128
A.4	DSm rule of combination for hybrid-DSm models	129
A.4.1	Component S_1	130
A.4.2	Component S_2	131
A.4.3	Component S_3	131
A.4.4	Compression to non-empty propositions	134
B	Additional simulation evaluation results	137
B.1	Threshold dependency.....	137
B.2	Additional results for the combination of filtering methods	139
B.2.1	Bayesian grid fusion with hit point inverse sensor model	139
B.2.2	Bayesian grid fusion with two-dimensional inverse sensor model.....	140
B.2.3	Dempster combination rule with hit point inverse sensor model.....	141
B.2.4	Dempster combination rule with two-dimensional inverse sensor model	142
B.2.5	Hybrid Dezert-Smarandache combination rule with hit point inverse sensor model	143
B.2.6	Hybrid Dezert-Smarandache combination rule with two-dimensional inverse sensor model	144
	Bibliography	145
	List of Figures	153
	List of Tables	157

List of abbreviations

The following table describes the significance of various abbreviations and acronyms used throughout the thesis, in the alphabetical order. The page on which each one is defined or first used is also given. Common abbreviations or acronyms used once are not on this list.

Abbreviations	Meaning	Page
ADAS	Advanced Driver Assistance System	25
ASPICE	Automotive Software Process Improvement and Capability determination	29
BBA	Basic Belief Assignment	55
DAS	Driver Assistance System	25
DOG	Dynamic Occupancy Grid	41
DSmC	Classic Dezert-Smarandache fusion rule	62
DSmH	Hybrid Dezert-Smarandache fusion rule	62
DSmT	Dempster Shafer modified or Dezert-Smarandache Theory	53
DST	Dempster-Shafer Theory	53
ECU	Electronic Control Unit	25
FOV	Field of View	89
FSM	Forward Sensor Model	46
ISM	Inverse Sensor Model	47
KPI	Key Performance Indicator	79
LIDAR	Laser Imaging, Detection And Ranging	21
MDC	Multi Domain Controller	26
NTHSA	National Highway Traffic Safety Administration	26
OG	Occupancy Grid	38
RADAR	Radio Detection And Ranging	21
SAE	Society of Automotive Engineers	27
TBM	Transferable Belief Model	56
VBA	Vacuous Belief Assignment	55

Mathematical notation

The following table lists the mathematical symbols used throughout the thesis in the order of occurrence in thesis. The page on which each one is defined or first used is also given.

Symbol	Meaning	Page
G	occupancy grid	38
N	number of occupancy grid cells	38
c_i	i-th cell of the occupancy grid	38
X	probabilistic state of the cell	38
t	time	38
z	sensor measurement	38
Θ	frame of discernment	55
θ	elemental hypothesis	55
\mathcal{P}^Θ	proposition set over frame of discernment Θ	55
$m(A)$	Basic Belief Assignment (BBA) of A	55
α	element of proposition set \mathcal{P}^Θ – subset of elemental hypotheses θ	55
2^Θ	power set	55
\mathcal{D}^Θ	Dedekind lattice, hyper-power set	57

1 Introduction

One of the most challenging problems in the autonomous driving field is a reliable perception of the vehicle environment. This ability is vital to safely navigate through dense traffic and avoid collisions with other traffic participants. Almost every modern driver assistance system relies on an environment perception to trigger an emergency braking maneuver, an evasion maneuver, a lane change, or simply to plan the future trajectory of the vehicle.

An automotive perception system operates in a wide variety of driving conditions. It has to be resistant to different weather conditions and cannot be constrained by complex, cluttered or unseen environments. To match these requirements, modern vehicles are equipped with a variety of sensors like LIDARs, radars, cameras, and ultrasonics. The perception system accumulates, fuses, classifies, and filters uncertain sensor measurements in order to create an accurate representation of the actual scene around the host vehicle.

Besides that, car manufacturers offer multiple options with different prices of the same vehicle model. The car options may differ by an equipment, electronics, or convenience features installed onboard. To limit the software development costs, the automotive perception system should be compatible with multiple vehicle options without the need of redesigning the solution.

There are multiple perception approaches in the automotive industry. This thesis focuses on the occupancy grid mapping algorithm, which depicts an environment as a multidimensional spatial lattice, where each cell stands for an independent portion of a space [Elfes and Matthies, 1987; Moravec, 1988]. Since several measurements occur over time, the grid map combines these evidences with a recursive probability filter. One of the simplest filters used for this application is the Binary Bayes Filter (BBF), which estimates the probability that the grid cell is either occupied or free [Dietmayer et al., 2014]. The recursive update process has capabilities to model and filter out measurements' uncertainties. More complex frameworks such as Dempster-Shafer or Dezert-Smarandache evidence theories are additionally able to solve the sensor information conflicts [Yager, 1987]. All grid map solutions assume that individual grid cells are independent stochastic processes, which facilitates a fast implementation at the cost of approximation errors.

The key advantage of the occupancy grid perception is its flexibility. Grid based environment representation can detect arbitrarily shaped obstacles, can be configured to distinguish between free space and occluded areas, and ultimately it can be used to identify and track dynamic road users.

1.1 Objectives

The occupancy grid has become an increasingly interesting area of research in both industry and academia nowadays. The algorithm is considered a well-known solution in the field of robotics, where it is used to solve the simultaneous localization and mapping problem. In the research reports, the occupancy grid algorithm is presented as applicable to a variety of experimental vehicles, but the series automotive applications are still limited.

The long-term objective of the occupancy grid development is to embed the perception module on cars commercialized on the automotive market. This intends to enable cars to monitor their driving

environment by using multiple sensors mounted on board. As aforementioned, the perception system has to be robust and provide high quality output regardless of neither environmental conditions nor the sensor uncertainties.

The purpose of this thesis is to investigate if the occupancy grid can be used as an automated vehicle perception system and how well it performs. Nevertheless, due to the fast development in this research area, the presentation of a single algorithm for occupancy grid mapping will be already outdated at the time of the thesis publication. Instead, the author focuses on the development of the general occupancy grid architecture, which is modular and can be adapted to various sensor types and automotive applications.

Moreover, this thesis aims to address some of the problems that almost every occupancy grid algorithm has to manage, such as the estimation of the map quality and the filtering capabilities of the algorithm with different types of system uncertainties.

1.2 Contributions

This thesis work incorporates the following contributions to the field.

- **A novel multilevel information fusion architecture**, which uses intermediate grids to solve sensor conflicts and speed up the algorithm computation (Section 4.2).
- **First application of the Dezert-Smarandache rules of combination** in the three-state automotive occupancy grid environment (Section 4.3).
- Theoretical and experimental **comparison of Bayesian, Dempster-Shafer and Dezert-Smarandache approaches** for the grid fusion (Sections 4.3.2, 6.3.4 and 7.4).
- **A new method of an occupancy grid quality assessment**, which has been tested in simulated and environmental conditions. The method is able to evaluate the algorithm using only a small amount of ground truth information by analyzing only the map representation of specific road objects (Section 5.2).
- **Analysis of the impact of multiple sensor and system uncertainties on the quality of occupancy grid** using a simulation environment (Section 6.2).
- A grid filtering capabilities investigation allowed to prepare **a guideline for tuning** the algorithm on a specific vehicle setup (Sections 6.3 and 7.2).

1.3 Outline

Chapter 2 introduces a brief overview of the automated driving systems available in the automotive. It highlights a historical overview of the driving systems, their development process, and the role of perception in the automated driving industry.

In Chapter 3, the occupancy grid framework is introduced. The algorithm foundations and state-of-the-art solutions are presented. The basic probabilistic theories are described. A detailed look into the algorithm architecture and its individual components is described.

Chapter 4 expands the topic of heterogeneous information fusion in the occupancy grid framework. The multilevel fusion approach as well as the Bayesian, Dempster-Shafer and Dezert-Smarandache probabilistic theories are introduced. Additionally, the combination rules, equations, and a simple comparison are presented.

Chapter 5 focuses on the uncertainty filtering in the occupancy grid, presenting different types of uncertainties and ways to filter them out in the occupancy grid. Moreover, the method for assessing the occupancy grid mapping quality is defined.

In Chapter 6, simulation experiments setup and results are described. First, the simulations focus on assessment how the occupancy grid is affected by different noise types. Secondly, the filtering capabilities of the occupancy grid are evaluated qualitatively and quantitatively in order to present their performance.

In Chapter 7, an experimental evaluation is defined. The algorithm's real-time capabilities are assessed. Furthermore, the proposed tuning procedure is described. In addition, the effects of various sensor conflicts are characterized with an evaluation of different fusion rules.

In Chapter 8, conclusions of this thesis and future work opportunities are provided.

Appendixes A and B contain supplementary information which can be useful to fully describe results presented in this thesis. Appendix A presents a detailed derivation of all fusion rules equations and Appendix B shows some additional simulation evaluation results.

2 Automated driving systems

Modern vehicles are equipped with a growing number of Electronic Control Units (ECUs) which serves the purpose of vehicle's automation. This chapter describes how these systems evolved in time (Section 2.1) and what automation levels they could provide (Section 2.2).

Automated driving systems can be divided based on the task they streamline. The primary objective is to improve road safety, which can be realized using passive or active safety systems. A simplified distinction between these systems is that passive safety is designed to protect vehicle occupants in the event of crash, while active safety systems operate during precrash traffic scenarios and their goal is to prevent the accident occurrence. Additional objective of automated systems is to improve the driving quality via convenience features [Cieslar et al., 2016; Galvani, 2019]. Regardless of the application, the automated driving system has to meet the automotive standards and procedures as described briefly in Section 2.3.

The Driver Assistance Systems (DAS) or their extended version Advanced Driver Assistance Systems (ADAS) has to perform a set of driving tasks in order to maintain its operation. The first task is to sense and self-position in the driving environment. Secondly, the collected information has to be perceived and formatted into an easy-to-understand image of the road situation. Thirdly, the vehicle might plan future actions based on the collected data and finally act based on the derived plans [Eskandarian, 2012]. Section 2.4 focuses on the aforementioned tasks describing their responsibilities in detail.

The occupancy grid algorithm is a part of the perception system of the vehicle. It can easily merge multiple sensors into a single representation of the environment. Occupancy maps created as an output can be easily utilized by multiple ADAS systems in both safety and convenience areas of the application as presented in Section 2.5.

2.1 Historical background

The chronological order of the Driver Assistance Systems development is a very delicate topic. Sometimes well-known solutions wait years to be integrated in the automotive industry due to various factors such as reliability, cost of production, or consumer satisfaction. The rapid advancement in this area was disrupted by the Second World War, and the development process was not gradual, therefore.

Nevertheless, the first Driver Assistance System in the automotive market is considered to be the anti-lock braking system (ABS), introduced in the 1950s. ABS detects if one or more wheels are about to lock up under braking and prevents the wheels from locking up to ensure that the vehicle remains steerable [Galvani, 2019].

On the other hand, government regulations might force car manufacturers to develop specific controllers, or to install some systems as obligatory. This might be the factor that popularized the usage of electronics in vehicles, where the government environmental regulations regarding exhaust emissions and fuel economy introduced in the late 1960s required computer-based techniques to replace the methods applied at the time to increase engine control performance [Happian-Smith, 2006]. Nowadays, this practice is still common, the TPMS (Tire Pressure Monitoring System) is obligatory for the new vehicles

since 2014 and the Driver Monitoring Systems will have to be installed in all new European vehicles since 2026, based on current Euro NCAP regulations [Bieńkowska, 2019].

Starting from the 1960s, some types of electronic systems are usually present in the vehicle. Commonly, every specific functionality is controlled by a separate device called an electronic control unit (ECU). Since 1990s, the number of ECUs mounted in vehicles is growing rapidly. Today's vehicles may contain 100 ECUs or more controlling functions that range from essential (such as engine and power steering control) through comfort (such as power windows, heated seats, and air conditioning), to security and access (such as door locks and keyless entry). ECUs also control passive safety features, such as airbags, and basic active safety features, i.e., automatic emergency braking. Therefore, the modern vehicle architecture is a complicated network of multiple electronic control units, each dedicated to a specific purpose. In order to limit the number of independent computing units, several ECUs can be evaluated on a single multidomain controller (MDC). In addition to the advantages of signal processing and vehicle control, the utilization of MDCs facilitates cost savings in design and manufacturing by decreasing the number of hardware components and the amount of materials [Korta et al., 2021]. A detailed overview of this architecture is available in Figure 2.1.

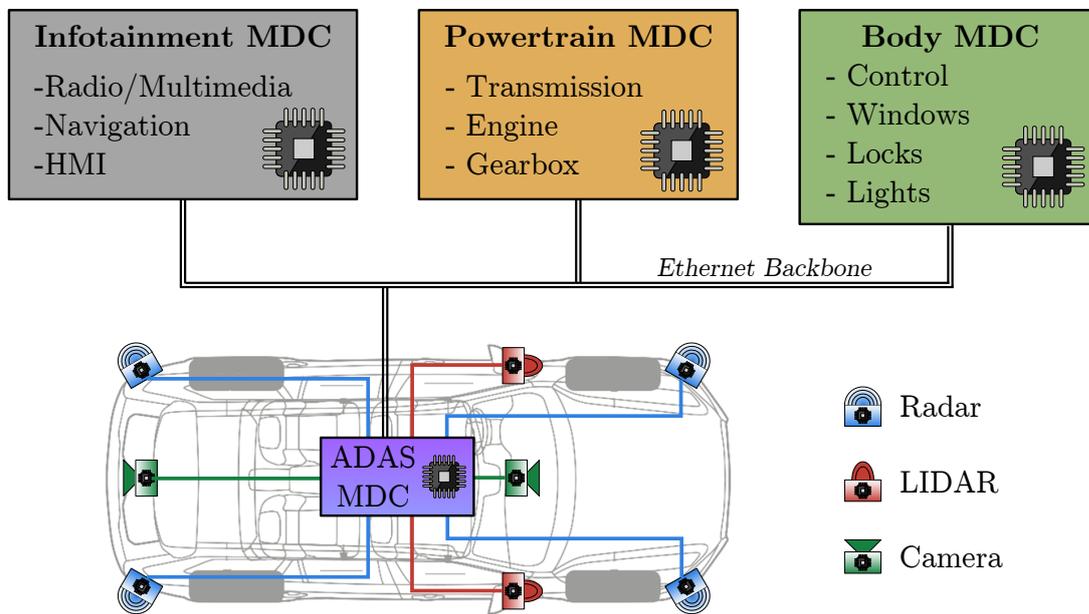


Figure 2.1. A modern vehicle control system architecture based on MDCs (inspired by [Korta et al., 2021]).

Presently, electronics make up a significant percentage of a vehicle's price and define many of its functionalities. The world's biggest Consumer Electronics Show (CES), which over time presented many technological gadgets such as CD players, high-definition television, or internet services, is now presenting also new developments in the automotive industry. Since 2010, almost all automotive companies are presenting their products at CES [Anktiengesellschaft Volkswagen, 2021].

2.2 Automation levels

Levels of automation designate the degree of driver and computer control over a dynamic task. This distinction of automation has been applied to the automotive industry to provide a framework to describe system responsibilities. There are different automation level definitions from National Highway Traffic Safety Administration's (NHTSA) and Federal Highway Research Institute (BAST), but the most com-

monly applied is the definition provided by the Society of Automotive Engineers (SAE) [Payre et al., 2021].

The Society of Automotive Engineers defined six levels for motor vehicle automation as shown in Figure 2.2. In order to achieve different levels of automation, different sensor redundancy and system architectures have to be applied. This in turn differentiates ADAS features on the automation level, where it is viable to apply them. This section lists some automated driving features grouped by the automation levels where they could be applied. There are some exceptions from the provided classification, as not all features correspond to a single automation level.

As mentioned before, the automation levels specify the degree of driver and computer control over the vehicle. Starting from level 0 to level 2, the human driver has to monitor the driving environment. While there is no automation at level 0, systems at level 1 assist the driver in the driving task (driving assistance). At level 2, partial automation is implemented, and the system can steer, brake, or accelerate the car in predefined situations. From level 3 to level 5, the system is capable of monitoring the driving environment and even replacing the human driver. Conditional automation at level 3 means that the automated vehicle can operate by itself in some conditions, but the driver should always be ready to intervene if the system reaches its limits. At level 4, human intervention is no more required (high automation), but the system is still limited by some environmental conditions. Fully autonomous level 5 defines a vehicle that can be self-driving in every road situation.

Currently, most of the vehicles on the market belong to the level 0-2 category, where the human driver is in control all the time and the assistance systems provide guidance for the driver. Higher levels of automation are currently present only in small geo-fenced areas where strictly monitored vehicles could take responsibility for driving.

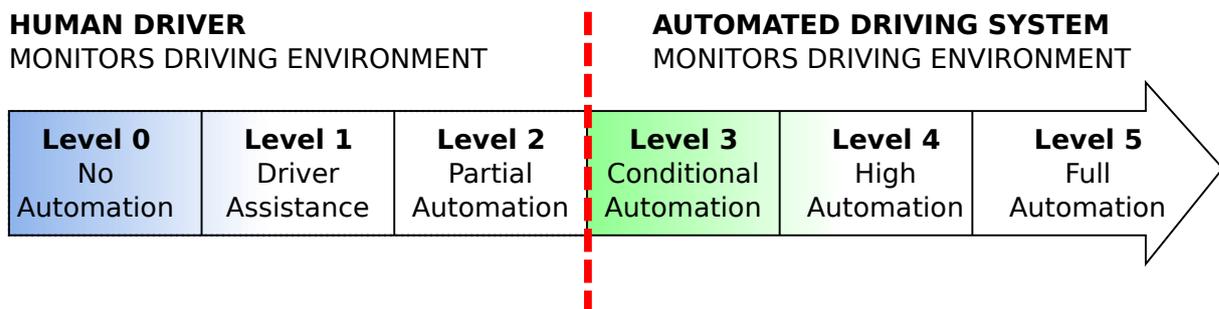


Figure 2.2. Levels of driving automation (inspired by [Andriamahefa, 2017]).

2.2.1 Human driver in control

Starting from level 0 to level 2, the human driver has to monitor the driving environment. Even as on level 0 there is no automation, there are systems that can inform the driver of specific conditions or passively improve the safety and comfort of driving. Driver assistance systems available in level 0 are, for example:

- Parking sensors – introduced to cars in the 1980s, provide an acoustic warning depending on the distance from surrounding obstacles while parking the car, to avoid damaging the vehicle.
- Lane Departure Warning (LDW) – this feature aids in the case when the vehicle begins to move out of its lane and no turn indicators are activated. The LDW warns the driver by acoustic and/or visual and/or haptic feedback if he is accidentally leaving the current lane.
- Blind Spot Information System (BLIS) – a feature responsible for monitoring the blind spots and warning the driver if lane change intention is detected.
- Forward Collision Warning (FCW) – warns the driver in the event of an imminent collision.

- Tire Pressure Monitoring System (TPMS) – real-time monitoring and low-pressure warning for the driver.

Level 1 systems take care of a single functionality in specific cases. Driver assistance systems utilized in this level of automation can control vehicle dynamics in lateral or longitudinal directions and are usually based on conventional sensors which are already widely present on the market. Example level 1 ADAS systems are:

- Anti-Lock Braking System (ABS)– supports the driver while braking, avoiding wheel blockage.
- Traction Control System (TCS) – introduced in the 1970s, prevents the wheels from spinning by reducing the drive torque at each driven wheel.
- Electronic Stability Program (ESP) – combines and improves both AEB and TCS capabilities to prevent the vehicle’s tendency to become unstable (over-steered or under-steered) – introduced in the 1980s.
- Hill Hold Control (HHC) and Hill Descent Control (HDC) – assist the driver in driving upwards or downwards, respectively.
- Cruise Control (CC) – maintains the car at the speed selected by the driver, by controlling both the engine and the automatic gearbox. This system was available as an option in vehicles since 1970s.
- Adaptive Cruise Control (ACC) – like the CC, ACC maintains the vehicle at a desired speed. However, if there is another vehicle traveling ahead, the system is able to detect it and reduce the speed in order to keep a safe distance from it.
- Autonomous Emergency Braking (AEB) and Emergency Brake Assist (EBA) – these features responsible for warning and/or providing brake support if an imminent collision is detected.
- Lane Keep Assist/Aid (LKA) and Lane Centering (LC)– these features aid in the case when the vehicle begins to move out of its lane and no turn indicators are activated. The LKA helps the vehicle stay in the lane by applying counter-steering force to the steering wheel. The LC continuously controls the steering wheel to keep the vehicle in the center of the lane.
- Driver Monitoring Systems – assess the driver’s alertness and warn the driver if needed, and eventually apply the brakes.

The level 2 systems share the same level of authority with the driver as level 1 but are able to perform more complex maneuvers, combining longitudinal and lateral dynamics, which mainly lead to perform a desired trajectory with a desired speed. Examples of these systems are:

- Highway Assist (HA) – combination of ACC, LKA, and BLIS which continuously control longitudinally and laterally the vehicle.
- Autonomous Parking – assists the driver with both finding the parking spot and parking the car in it. The authority is left to the driver, who remains in charge of monitoring the procedure or controlling the pedals according to guidance.

2.2.2 Automated driving systems in control

In the automation levels 3 and above, the driver can pass full vehicle control with the specialized ADAS systems. The systems might operate only in some conditions, but they have to be "fail-safe", i.e., when they assess that they are not capable of handling the current situation, or they detect a fault by themselves, they warn the driver and ask to take back vehicle control.

According to the SAE [SAE International, 2014], recommended practice to safely operate these systems need redundancy both in sensors and in decision electronic control units. Several companies provided guidelines for the development of automation level 3 systems, nevertheless there are still some legal controversies according to vehicles satisfying this level of automation, e.g., who is responsible in the case of an accident, how to safely transfer control to the driver [Aptiv et al., 2019].

Systems in a vehicle which satisfies the 4th level of automation shall be not only fail-safe but fail-operational, i.e., able to safely work in case of failure without the intervention of the driver. An example

of L4 ADAS system is an Automatic Valet Parking, where the vehicle might automatically park itself on a specified parking lot and return to the driver on request.

The final automation step is level 5, which is a fully autonomous vehicle. Level 5 vehicles do not require the intervention of the driver and have such redundancy, sensing coverage, and decision intelligence that they can even lack interface with the driver to control the car, like the steering wheel and pedals. The driver becomes a passenger, who just sets a destination and sleeps while the vehicle is transporting him to his decided location [Galvani, 2019].

2.3 Automotive systems development process

In each advanced engineering industry such as automotive, the introduction of a new product requires considerable effort and throughout planning to limit the costs and ensure final product reliability. Engineers on all levels must cooperate and coordinate their work during the entire product development lifecycle.

For instance, if the manufacturing company would like to productize a novel electronic control unit (ECU), then except for the concept of a unique software controller, they must also design the hardware where this unit is executed. This chain of work must be compliant with different standardization bodies and follow a detailed procedure to ensure that the resulting device will behave and perform in the way that its concept is assumed.

One of the main standards defining automotive software process improvement and capability determination is the ISO/IEC 15504 standard, referred also as Automotive SPICE [International Organization for Standardization, 2012]. This standard defines best practices for embedded software in automotive development. Among them, the V-model system engineering process for software development is recommended.

2.3.1 The V-model

The V-Model, also known as the Verification and Validation model, is a disciplined model that requires rigorous evaluation to ensure continuous assessment and development. It requires a testing phase corresponding to each stage of development, which benefits both providers who can eliminate potential problems in the initial stages, and clients who can assume a meticulous approach to both innovation and development.

The V-model diagram is depicted in Figure 2.3. This process defines and characterizes several development steps that summarize the development of the embedded automotive control system. These steps include software design, implementation, and testing.

First, the system requirements analysis phase is performed. Concrete requirements from stakeholders are gathered and combined into a single document. It is essential that such requirements are achievable, verifiable, consistent, and complete due to the fact that the outcome will be utilized as input for the subsequent work of software engineers. Simultaneously with the system requirements definition, the system tests are defined, and the evaluation procedure starts. It is crucial for the V-model and ASPICE standard to test the solution in every system definition and design step.

Based on the prepared requirement document, the software architects prepare the logical and technical system architecture. In this design process, the engineers distribute functions among specific software components. At the end of this stage, the high-level system architecture documentation is prepared, and the system integration tests are ready to verify the architecture cohesion.

Concurrently, the software architects/engineers design the interfaces of subsystems that resulted from splitting the overall software into modules, interfaces, and specific component alignment. The detailed software architecture documents will later serve as a specification for the developers and engineers.

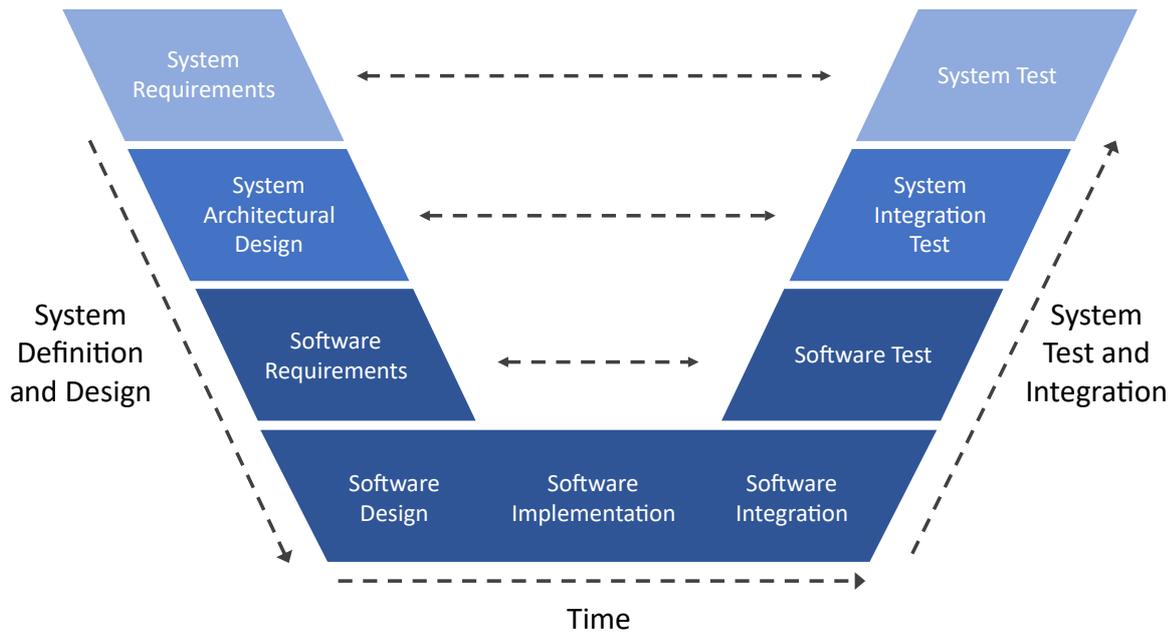


Figure 2.3. V-model development process diagram.

The most bottom part of the V-model is the component's implementation. During this phase, software of the product is prepared based on the documentation from previous steps. Later, the system integration and the testing phase begin. The tests used to verify the requirements and architecture are used to assess the actual software correctness and performance. If the tests show some errors, the software requirements are updated, and the product has to be corrected.

After the component unit test finishes, the system integration tests, and validation are performed. In the context of automotive software development process, system integration tests constitute a verification of whether the unit works as expected when it is a part of the entire distributed vehicle system. To effectively test the component, other parts of the vehicle have to be simulated.

According to the V-model, multiple development and testing tasks can be paralleled and pipelined to ensure cost-effective management of time and engineers' efforts. During all testing phases, if some irregularities are detected, the process returns to the corresponding design and documentation phase, which is denoted as a dashed line on the Figure 2.3. The final test of the software component is the full system test in the real environment, which can be often performed by the carmaker client instead of the manufacturing company.

The V-model provides guidance for equipment manufacturers for communication and coordination of multiple teams of engineers. From this perspective, the V-Model reflects a project management view of software development and fits the needs of project managers, accountants, and lawyers rather than software developers or users. At the same time, the V-model itself is largely generic. It does not dictate specific tools or techniques, but rather an approach to the internally selected development methods.

Design of the occupancy grid software component requires a specification related to different levels of V-model. The first part of this thesis (chapters 3 and 4) focuses on the system architectural design. Software requirements, integration testing guidelines (tuning) and results of specific implementation are presented in chapters 5, 6 and 7.

2.3.2 Standardization of automotive systems

To ensure that the automotive software and hardware is safe, it should be prepared according to certain standards. The most important software standards and frameworks are:

- ISO/IEC 15504 (ASPICE) – Automotive Software Process Improvement and Capability Determination [International Organization for Standardization, 2012],
- IEC 61508 – Functional Safety [International Electrotechnical Commission, 2021],
- ISO 26262, ISO 21448 – Safety of the Intended Functionality (SOTIF) [International Organization for Standardization, 2019],
- Automotive Open System Architecture (AUTOSAR) [AUTOSAR, 2021],
- The Motor Industry Software Reliability Association (MISRA) [MISRA, 2021].

The first standard provides guidelines for the automotive software development process. The next two standards refer to functional safety, which was developed to address the new safety challenges that autonomous (and semi-autonomous) vehicle software developers are facing. The last two standards define guidelines for the code style implementation in both C and C++ programming languages.

Moreover, the offices and employees of the manufacturers should ensure that the newly developed products are not prone to cyberattacks, which may lead to leakage of protected intellectual property to expose the product and the customers' data. To ensure the required security level, the Trusted Information Security Assessment Exchange (TISAX) procedures are followed, which match with the ISO/IEC 27002 standard [International Organization for Standardization, 2013].

2.4 Tasks of the automated driving systems

Driving tasks performed by Advanced Driver Assistance Systems (ADAS) require four steps similar to the human driver cognitive process [Eskandarian, 2012] (Figure 2.4). At first, the system senses the environment and determine vehicle position. The host has to be equipped with divergent and redundant sensors listed in Section 2.4.2. In order to mitigate the weaknesses of individual sensors, the non-trivial task of data fusion has to be developed in perception ADAS features [Dickmann et al., 2015; Dietmayer et al., 2014]. For that purpose, the occupancy grid framework together with the object tracking module are utilized. The system understands the road situation to plan and decide which action should be performed. The vehicle's planning modules are described in Section 2.4.4. The final step is to act accordingly to the circumstances.

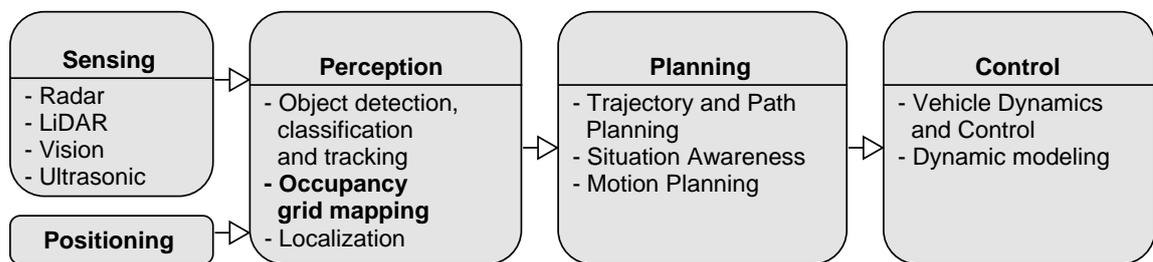


Figure 2.4. Cognitive information flow of the ADAS features with elements used to perform each step.

2.4.1 Positioning

Vehicle state estimation is a complex task of determining the car relative pose (position and orientation), velocities, and acceleration.

To determine the car pose (position and orientation) in a world fixed frame, several positioning sensors are needed. The satellite-based navigation systems such as global positioning system (GPS), differential GPS (DGPS), or real-time kinematics (RTK) may provide only the estimated vehicle state and work only in open sky conditions.

To improve vehicle positioning and state estimation, inertial measurement units (IMU) are utilized. The positioning module filters the data to compute the full vehicle state even if the satellite location is not available.

2.4.2 Environment sensing

The ADAS features need to perceive the environment with high precision and accuracy in all weather conditions. Automotive environment imaging sensors include cameras, radars, LIDARs, and ultrasonic sensors. Each of the sensor types has an integral role in the perception module. To ensure a safe driving experience, a combination of sensors is usually developed to mitigate possible sensor uncertain measurements (see table 2.1).

The ultrasonic sensors utilize the echo times from sound waves that bounce off nearby objects. They are suitable especially for low speed, short range applications, such as lateral moving, blind spot detection and parking [Yi, 2001].

Radars, LIDARs and vision sensors are the masters of medium to long range imaging systems. Radar (Radio Detection And Ranging) uses radio waves to determine the range, azimuth, and elevation of objects. As the only sensor, the radar could measure motion as a range rate and operates in every weather condition. LIDAR, short for light detection and ranging, is a device that measures distance to objects using laser light. It can provide the most accurate 3D map of the car's surroundings; however, it is the most expensive sensor in the set. Cameras could perform classification and scene interpretation. The stereo vision provides additionally the depth map of the environment which could be further fused with other sensors' data [Steinbaeck et al., 2017; Zheng et al., 2019].

Comparison of the capabilities of different sensors is presented in Table 2.1.

Table 2.1. Comparison of environment imaging sensor capabilities (extended from [Steinbaeck et al., 2017]). Symbol '✓✓' means very good performance, '✓' acceptable execution, 'X' and 'XX' refers to low and very low measurement capability. Fields marked with '—' describe not available capabilities for the sensor.

Sensor capability	Radar	LIDAR	Vision	Ultrasonic
Short range	✓✓	✓	✓✓	✓✓
Long range	✓✓	✓	✓	X
Range resolution	✓	✓✓	—	✓✓
Angular resolution	X	✓✓	✓	XX
Works in bad weather	✓✓	X	X	X
Works in dark	✓✓	✓✓	XX	✓✓
Works in bright	✓✓	✓	✓	✓✓
Color/contrast	—	X	✓✓	—
Velocity estimation	✓✓	—	✓	—

2.4.3 Perception systems

Each of the imaging sensors has its own advantages and disadvantages as presented in Table 2.1. To mitigate the ubiquitous noises and weaknesses of individual sensors, the data from all sensors has to be fused in the perception system.

The utmost objective of the perception system is to provide the most accurate representation of the vehicle surroundings, which could be easily used by the downstream motion planning systems. Different ADAS features require varying perception capabilities. For example, the Blind Spot Detection module (BLIS) would require monitoring of specific areas on the sides of the vehicle against any dynamic objects. The Adaptive Cruise Control (ACC) would use the front target position to compute the desired drive speed. The Lane Keep Assist (LKA) would require monitoring of lane markers as well as the side target positions.

Additional objective of the perception system is to precisely locate the vehicle on the road and in the world. The positioning module delivers the filtered vehicle state data, but the position has to be located in a world or road frame of reference. It can be performed using a combination of the sensed environment representation and the existing maps of the area [Filliat and Meyer, 2003; Meyer and Filliat, 2003; Valente et al., 2018].

Historically, the most commonly used solution for the vehicle's perception is object tracking. Tracker collects all sensor detections and delivers a list of classified objects to multiple ADAS modules. It utilizes multiple objects tracking algorithms which use a variety of models to identify and estimate the shape and velocities of various road users such as vehicles, trucks, bicyclists, and pedestrians. Object tracking is widely applied in automotive as a component in all automation levels. Nevertheless, the list of rigid objects is not enough for all modern ADAS features. For example, the object list is not able to determine the free area around the vehicle without some other assumptions about the tracked object quality.

The second approach for perception systems is the occupancy grid mapping. Initially, the occupancy grid mapping focused on stationary objects, but recent improvements in this area allowed mapping also dynamic occupancy [Danescu et al., 2011; Tanzmeister et al., 2014]. The grid-based representation of the environment is more general than the list of objects, therefore it can be used by a wider set of ADAS features. Additionally, vehicles with automation level 3 and above require redundancy on all levels of processing, including perception systems. In these vehicles, the occupancy grid combined with the object tracking provides full redundancy, as both approaches are complementary and able to detect both stationary and dynamic objects in the environment.

The type of environment representation depends on the requirements of downstream elements and the communication bandwidth capabilities of the whole ADAS system. Output of the perception system can be a list of classified stationary and moving objects or the occupancy map, where each cell value stands for a portion of space. To compress the grid-based map representation, it can be analyzed, for example, to determine the road type [Seeger et al., 2016] or to detect the contours of static objects [Foroughi et al., 2015]. The object track might be augmented with camera sensor to detect traffic objects like signs or lights [Dickmann et al., 2015]. Detailed features which could be extracted from the occupancy map representation are explained later in Section 2.5.

2.4.4 Motion planning systems

The objective of motion planning systems is to determine the path which a vehicle should follow based on the current road situation and automated driving goals.

The path planning module combines the outputs of perception components to generate multiple trajectories as shown in Figure 2.5. The situation assessment creates a comprehensive understanding of the road scene. The module analyses the scene drivability and computes different factors determining the safety of all actions [Guo et al., 2018]. Further on, a motion planning component classifies perceived environment surroundings into one of the modeled traffic configurations and selects the most feasible

trajectory. In further processing the behavior generation is performed and the vehicle controller executes selected path [Urmson et al., 2008; Ziegler et al., 2014].

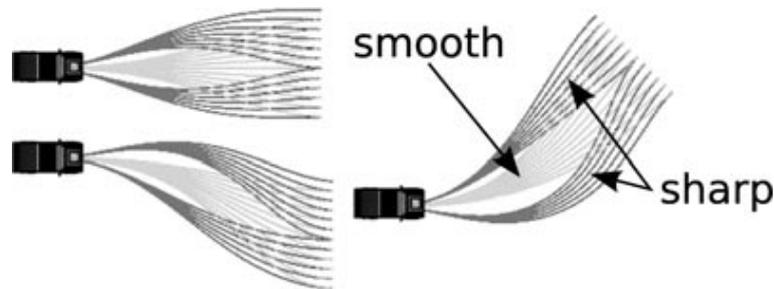


Figure 2.5. Smooth and sharp trajectories. The most viable trajectory according to road scene can be directly executed by the vehicle controller [Urmson et al., 2008].

2.5 Occupancy grid application in automotive

The occupancy grid algorithm is present in the automotive market since late 2000s [Urmson et al., 2008]. The occupancy mapping can simplify the perception system in almost all autonomous systems, nonetheless its benefits are clearly present in the automation level 2 and above. The algorithm can easily work with multiple different sensors and provide redundant tracking solution, guardrail detection, road situation assessment, and localization of the vehicle on the road as presented in Figure 2.6.

The long-term objective of the occupancy grid development is to embed the perception module on cars commercialized on the automotive market. This intends to enable cars to monitor their driving environment by using multiple sensors mounted on board. Targeting the automotive market, and the automotive domain in general, implies that the perception module is subjected to multiple constrains:

- First, the computations have to be performed in real-time. The rate of production of occupancy grids must be high enough to spot any hazards on the road in both high and low velocities.
- Second, a mass production cannot afford expensive hardware. Thus, the integration must be realized on low-cost computing platforms.
- Third, the computing platform must have a low electrical power-consumption to fit within the limited source of electrical energy in cars.
- Finally, the module has to be safe, which includes handling sensor uncertainties, knowing numerical errors during computation, and guarantying determinism.

Two-dimensional static occupancy grid managed to meet these constrains and has been successfully integrated into the autonomous vehicle software stack [Kunz et al., 2015; Urmson et al., 2008]. Grid-based representation is suited to perform sensor fusion and solve conflicting measures from various vehicle environmental sensors such as radars, LIDARs, cameras and ultrasonic. Different sensor models can be specified to adapt to the distinct characteristics of the various sensors, facilitating efficient fusion in the grid.

Other approaches for automotive perception systems are multi-object tracking algorithms, which maintain a list of currently known objects. However, the main limitation of these methods is that the tracked object has to follow a set of motion and shape models. The grid-based model of the environment allows for the representation of arbitrarily shaped objects without the need to track each object separately. The absence of object-based representation allows easier fusion of low-level descriptive sensory information onto the grid without requiring data association [Saval-Calvo et al., 2017].

The occupancy grid module enables vehicle self-localization [Milstein, 2008; Stachniss, 2006], path planning [Stepan et al., 2005] and trajectory validation, which are essential in the automated driv-

ing. There are also proposed methods of extending the occupancy grid framework into tracking solution by adding the dynamic and velocity layer to the algorithm and creating dynamic occupancy grid [Godoy et al., 2021; Steyer et al., 2017].

In the autonomous vehicle, we cannot think about the occupancy grid as an independent component. The map provides an accurate approximation of the surroundings and acts as a tool to represent the environment used by other components. Therefore, the occupancy grid can be easily adapted for specific applications and requirements of the downstream elements in the software stack. The probability map can be clustered to extract object bounding boxes or contours [Godoy et al., 2021; Pae et al., 2021]. For other applications, especially the path planning, drivable area around the host can be extracted as an envelope and the proposed trajectories might be invalidated on the occupancy grid [Ding et al., 2019; Laconte et al., 2019; Szlachetka et al., 2020]. Examples of the environment representations which might be extracted from the occupancy grid are presented on Figure 2.6.

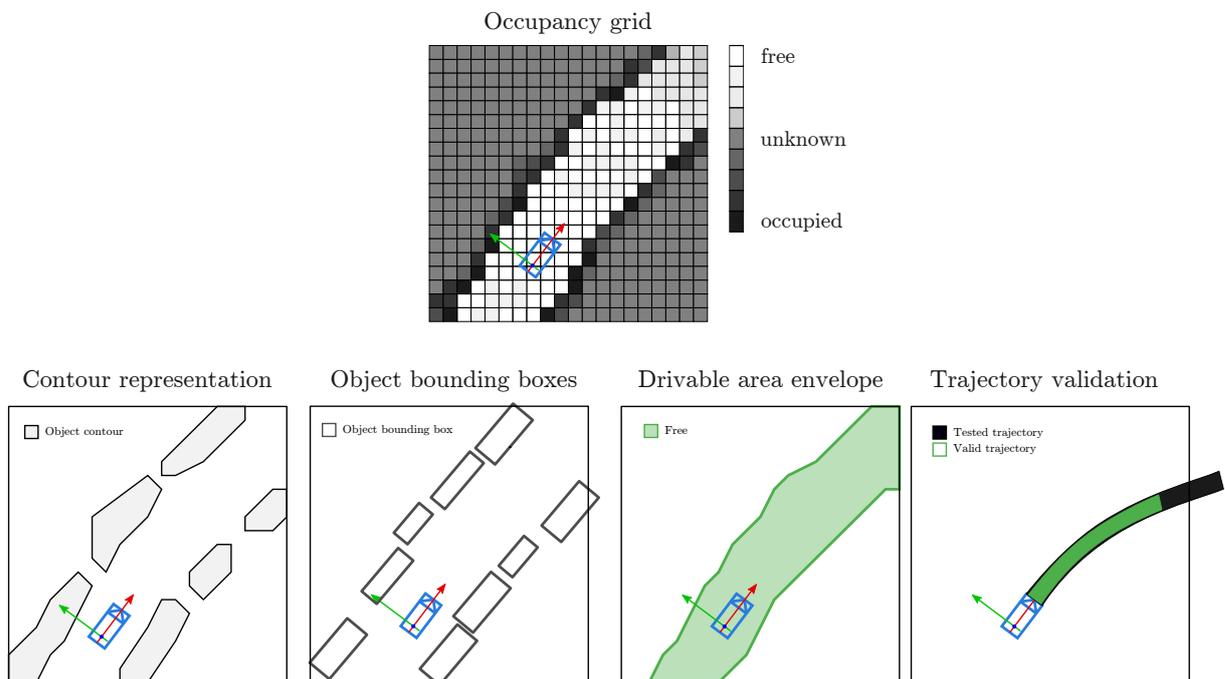


Figure 2.6. Examples of information extracted from the occupancy grid. The grid representation (top) could be transformed depending on application into, for example (from left): contours, objects, free space envelope or a trajectory validation tool.

3 Occupancy grid framework

Among multiple environment mapping methods, the occupancy grid is an excellent framework to perform sensor fusion. The occupancy grid representation employs a multidimensional tessellation of space into cells, where each cell stores a probabilistic estimate of its state. The discretization of space representation reduces the algorithm complexity and enables processing of multiple inhomogeneous sensors measurements.

The following chapter presents a comprehensive description of the occupancy grid framework designed for automotive applications. Processing of the data is a computational intense and algorithmically complex job. The assumptions needed to make the algorithm computationally feasible are described in Section 3.1. The occupancy grid method is receiving great attention in recent years. The general review of the current state-of-the-art automotive occupancy grids is presented in Section 3.2. An overall description of the occupancy grid data flow and possible architectures is presented in Sections 3.3 and 3.4. As the occupancy grid fuses together different sensor domains, some conflicts might occur. Differentiation and examples of contradictions resolved in the algorithm are presented in Section 3.5.

3.1 Foundation of the occupancy grid

The occupancy grid mapping was introduced in the late 1980s by Elfes [Elfes, 1989; Elfes and Matthies, 1987] and Moravec [Moravec, 1988]. It is a well-established method solving the problem of integration of noisy data from multiple sensors and various robot positions into a common description of the environment.

To define the occupancy grid from the mathematical point of view, it is needed to introduce some basic definitions.

Random variable *A random variable X is a measurable function defined on a probability space that maps from the sample space to the real numbers. A random variable can designate the result of an experiment, the state of a system, the value of a measurement, etc. A quantitative random variable can be continuous or discrete, depending on if it is obtained by measuring (i.e., continuous) or counting (i.e., discrete). For instance, the result of the toss of a coin is a discrete random variable X that can take on two values: head or tail [Rakotovao et al., 2016].*

The occupancy grid makes use only of discrete random variables; therefore, the following definitions implicitly assume only the usage of discrete random variables.

Probability of a discrete event *Let X be a discrete random variable, S the set of all possible values of X , and x an element of S . The probability distribution of X is a function P that assigns a non-negative real number to each value x of X such that*

$$P(x) \geq 0 \qquad \sum_{x \in S} P(x) = 1 \qquad (3.1.1)$$

The quantity $P(x)$ denotes the probability that X takes a value x . From Equation (3.1.1) follows that $P(x)$ is a real number between 0 and 1.

Conditional probability *It is a measure of the probability of an event occurring, given that another event has already occurred.*

Let X and Y be two random variables, and S_X and S_Y the sets of all possible values of X and Y with the non-zero probability of event y ($P(y) > 0$). The conditional probability of the joint of events (x, y) where $x \in S_X$ and $y \in S_Y$ denotes the quotient:

$$P(x|y) = \frac{P(x \wedge y)}{P(y)} \quad (3.1.2)$$

where $P(x \wedge y)$ is the probability that X takes value x and Y takes value y .

Bayesian theorem *This theorem describes the probability of an event, based on prior knowledge of the conditions that might be related to the event. It allows to redefine the conditional probability equation (3.1.2) as:*

$$P(x|y) = \frac{P(y|x) P(x)}{P(y)} \quad (3.1.3)$$

where $P(y|x)$ is also a conditional probability: the probability of event Y to take value y occurring given that event X produced a value x .

Occupancy grid *The occupancy grid G models a physical world with a grid subdivided into cells c , which can be defined as:*

$$G = \{c_i\} \quad i = 1, \dots, N \quad \bigvee_{i \neq j} : c_i \cap c_j = \emptyset \quad S = \bigcup_{i=1}^N c_i \quad (3.1.4)$$

where:

- G – the grid,
- N – number of cells,
- c_i – the i -th grid cell,
- S – area of the grid.

Occupancy grid cell *Each cell of the grid describes a portion of the environment as an independent stochastic process with the Markov property. The probabilistic state of the cell evolves over time by the accumulation of sensor measurements.*

Depending on the fusion framework, the cell state might be a binary continuous random variable describing if the cell is occupied or in general a collection of random variables defined on a common probability space. Every cell of the grid is an independent stochastic process defined as:

$$\forall_{i=1}^N c_i = \{X_i(t, z) : t \in T, z \in Z\} \quad (3.1.5)$$

where:

- $t \in T$ – time,
- $z \in Z$ – sensor measurements,
- $X_i(t, z)$ – random variable describing state of the i -th cell over time and measurements.

The occupancy grid cell estimation of the environment is periodically updated by the new sensor measurement to compute the best prediction of the real environment. New sensor information data denoted as z_{t_n} are fused into **prior grid cell** $c_i(t_{n-1})$ state in order to produce the **posterior cell state** $c_i(t_n)$. 'Prior' refers to the state estimation up to the time of the update (from t_0 to t_{n-1}), while 'posterior' describe the latest estimate of the state (from t_0 up to t_n).

Markov property *The stochastic process of the i -th occupancy grid cell c_i depends only on time and prior sensor measurements. In the occupancy grid inference framework, this process is the Markovian one (holds the Markov property).*

A stochastic process $\{X(t) : t \in T\}$ is Markovian if, for any n , the distribution of $X(t_n)$ conditional on the values of $X(t_0), X(t_1), \dots, X(t_{n-1})$ with $t_0 < t_1 < \dots < t_{n-1} < t_n$ and $t_i \in T$, depends only on $X(t_{n-1})$:

$$P(X_{t_n} = x_n | X_{t_{n-1}} = x_{n-1}, \dots, X_{t_0} = x_0) = P(X_{t_n} = x_n | X_{t_{n-1}} = x_{n-1}) = F(x_n, x_{n-1}; t_n, t_{n-1}) \quad (3.1.6)$$

In other words, the Markov property of the cell process ensures that the posterior state of the process $X(t_n)$ depends only on the most recent known value of the process $X(t_{n-1})$ and on the current measurements z_{t_n} :

$$X_{t_n} = F(X(t_{n-1}), z_{t_n}) \quad (3.1.7)$$

The occupancy grid accumulation is performed in a discrete time, only when the sensor measurement is ready for processing. Thus, the inference step might be described as a discrete time process as in equation (3.1.6). Other processing steps might require a continuous time approach, but the Markov property will be still preserved [Gómez-Corral et al., 2015].

Cell independence *As the grid is a collection of cells, representing the real environment, the cells might be coupled together by multiple dependencies. Nevertheless, modeling cell dependency leads to a combination explosion of possible grid states. To avoid that, it is commonly assumed that the cells on the grid are independent processes. Therefore, this assumption allows the computational unit to process the sensor fusion for each portion of the space separately, making it computationally feasible.*

Robotic and automotive grids *The occupancy grid mapping method was originally suited to solve the simultaneous localization and mapping (SLAM) problem for autonomous mobile robots. However, this method has been recently adapted for automotive industry purposes [Saval-Calvo et al., 2017].*

The key difference between automotive and robot occupancy grids is the area which the map represents. In robotics, the occupancy grid builds the map for the whole operational area (one room, parking, one-floor of building). In automotive, however, a vehicle operates on a much bigger area and keeping track of the whole grid-based map is ineffective. Thus, in the automotive, the occupancy grid represents only the local environment as presented in Figure 3.1.

Host positioning *Representation of the local environment around the vehicle requires more maintenance steps than in the robotic approach. Firstly, the coordinate system of the host has to be defined. In this case, a derivative of the ISO 8855:2011 standard on vehicle dynamics is applied. Following this norm, a right-hand-sided occupancy grid coordinate system can be defined. An example of the occupancy grid is shown in Figure 3.2.*

The environment's representation on the grid is discrete, although the vehicle is moving and turning in the real, continuous world. To eliminate discretization errors which would occur if the map were to be rotated, the host performs its motion relative to the grid with a fixed orientation [Weiss et al., 2007].

Moreover, since the host may turn on the map, the occupancy grid has to be a square to ensure a constant representation of the area in front of the vehicle. This condition implicates that the grid resolution in both dimensions is equal and the cells are squares.

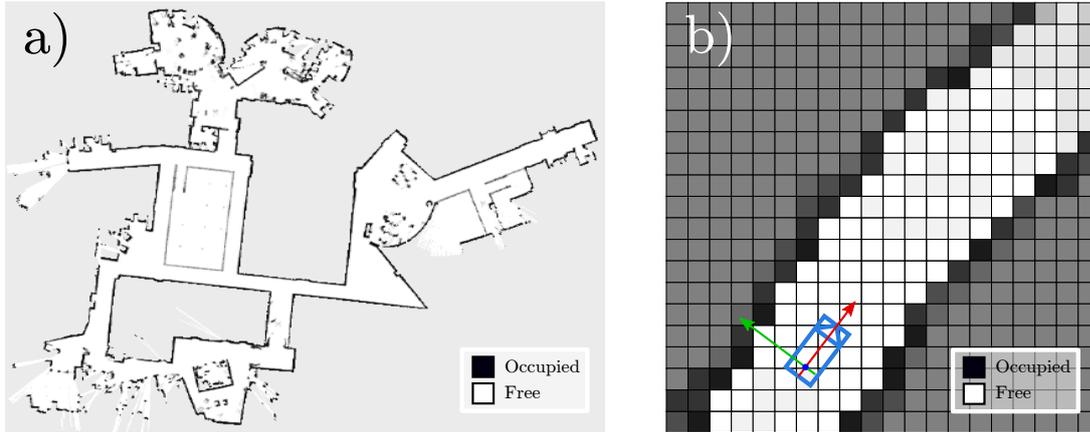


Figure 3.1. Robot and automotive occupancy grids. (a) Robot grids depict whole operational area of robot (image adapted from [Stachniss, 2006]). (b) Automotive occupancy grid map only local environment and is shifted as the vehicle moves to the new area.

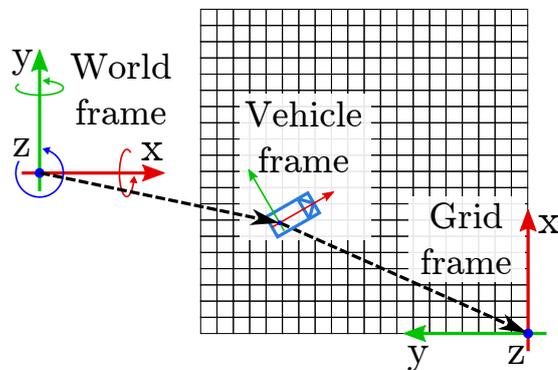


Figure 3.2. Coordinate systems used to position the occupancy grid in the world frame.

Automotive grid assumptions outline All assumptions of the automotive occupancy grid can be assembled into six points:

- Each cell of the grid is an independent Markovian stochastic process representing the state of the portion of the space,
- In automotive, the occupancy grid represents only the local area around the vehicle,
- Host position on the map is known variable,
- Occupancy map is only shifted relative to the fixed world frame, it cannot be rotated,
- Host is moving and turning in continuous coordinate system relative to the map origin,
- Automotive grid map is square with the same cell resolution in both directions.

3.2 State-of-the-art of automotive occupancy grids

The automotive occupancy grid was first proposed by [Weiss et al., 2007] in 2007. This publication was followed by numerous others and until today the topic of occupancy grid mapping in automotive is continually active. The industry is also interested in this solution, with more than 200 patents related to the applications and different types of occupancy maps.

Current trends in occupancy grid development can be differentiated into three groups: maps which estimate the velocities for each cell, called dynamic occupancy grids, maps which operate on higher than two dimensions and grids with variable resolution.

Each of the seminal approaches to the occupancy grid presented in this section offers new capabilities of the algorithm at the cost of the higher computational load. In contrary, most of the modern automotive vehicles in series production has limited computational resources, because they contribute to total vehicle manufacturing cost. Therefore, regardless of numerous occupancy grid techniques, in the discussed application, the grid is usually used to represent the environment in a 2D map with a constant cell size [Andriamahefa, 2017; Pietzsch et al., 2009].

3.2.1 Dynamic occupancy grid

Dynamic occupancy grid (DOG) is an extension of the occupancy grid allowing estimation of the dynamic environment and its velocities [Danescu et al., 2011; Nègre et al., 2014]. The dynamic occupancy grid approach utilizes a particle filter together with a Dempster-Shafer evidence theory. The particle filter empowers the estimation of a velocity for each occupied cell [Gies et al., 2018], thus reinforcing the fusion method to distinguish static and dynamic occupancy. In this type of occupancy grid, the Dempster-Shafer evidence theory is utilized, supporting more than only two occupancy states of each cell. An example of the dynamic occupancy grid is presented in Figure 3.3(a).

Seminal research on the dynamic occupancy grid topic is the DOGMA algorithm [Nuß, 2017]. Multiple implementations of the algorithm are available as open-source frameworks. Nevertheless, to support the DOGMA real-time operation, a parallel computing unit such as GPU is preferred. From the automotive perspective a graphical processing unit is an expensive module, which can be installed only in luxury or highly automated vehicles nowadays (level 4 or 5). Due to that, modern DOG implementations are too computationally complex to be utilized within existing cheap automotive electronic control units.

3.2.2 Multidimensional representations

Most of the indoor occupancy grid applications represent the environment as a 2D spacial lattice. However, in external applications, due to problems with an elevated or slanted road, the height handling for the grid is needed. The proposed extensions of the occupancy grid are two-and-half-dimensional (2.5D) stixel or multi-level maps [Oniga and Nedevschi, 2010; Pfeiffer and Franke, 2011; Saleem et al., 2018; Triebel et al., 2006] and a complete three-dimensional (3D) voxel representation [Fankhauser and Hutter, 2016; Wurm et al., 2010].

The 2.5D stixel world representation (Figure 3.3(b)) model the environment with adjacent rectangular sticks of a given width and height. Elevation maps (Figure 3.3(c)) store at a cell level the height of the object occupying the cell. Established grid surface may be utilized to classify non-drivable space.

The full 3D environment representation can be modeled by 3D occupancy grids. The environment is subdivided into adjacent cubic cells called voxels. The probability that each voxel is occupied by an obstacle is computed from the sensor measurements. An example of robotic 3D occupancy grid application is shown in Figure 3.3(d).

Currently, the computational complexity of the 3D occupancy grid makes them unfeasible for automotive applications, however, with the increasing computational power, this approach might be transferred from robotics to automotive.

3.2.3 Multi resolution occupancy maps

Depending on the granularity of the mapped environment, the occupancy grid map could be compressed to save the memory. It could be done by dynamically changing the cell size. One of the approaches for lossless occupancy map compression are the 2^d -trees which spacial subdivision is presented in Figure 3.4. For 2D occupancy grid maps, this approach is called quadtree [Andriamahefa, 2017] and for 3D – octree [Wurm et al., 2010].

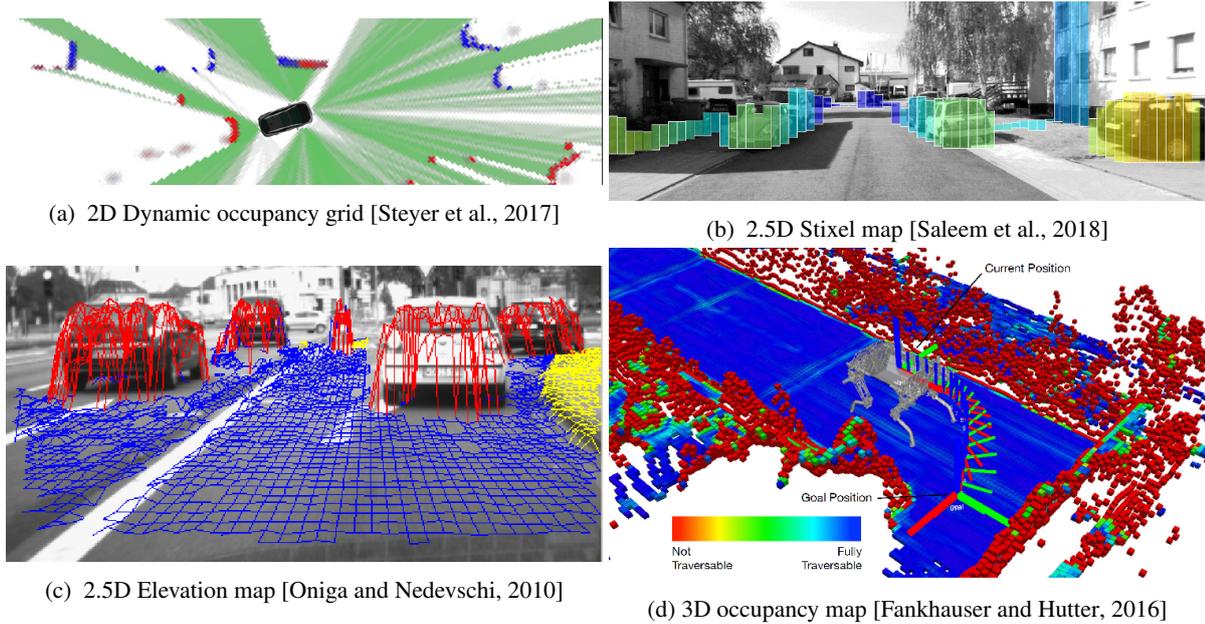


Figure 3.3. Examples of occupancy grid extensions for handling the dynamic objects and the multidimensional environment structure.

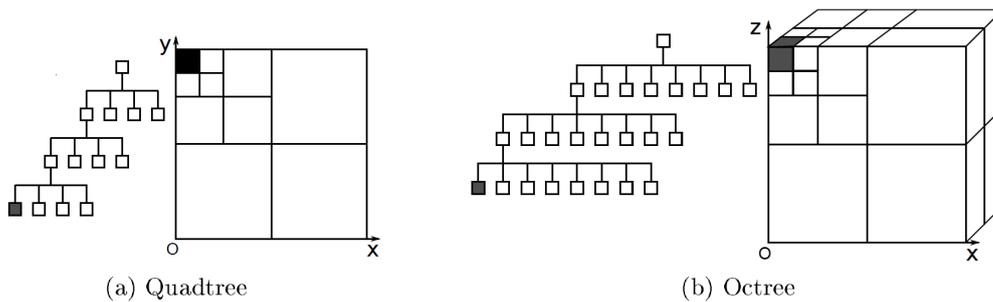


Figure 3.4. The tree structure of 2^d -trees with the corresponding spatial subdivision [Andriamahefa, 2017].

3.3 Occupancy grid architectures

The occupancy grid framework is usually used for the fusion of multiple sensors' data. Depending on the accumulation method, the occupancy grid algorithm may be divided into two different methodologies: low- and high-level [Gálvez del Postigo Fernández, 2015; Porębski et al., 2019]. The comparison of those two methods is presented in Figure 3.5.

The low-level or centralized occupancy grid architecture contains only one instance of the grid filter feedback loop. The main principle of this architecture is a direct fusion of the sensor data. The advantage of this approach is the low latency of the fused grids, which translates to possible shorter accumulation times of the algorithm.

The high-level occupancy grid is characterized by multiple accumulation loops in the algorithm flow. The occupancy filtering is usually performed separately for each sensor or for a selected domain of sensors. Intrinsic occupancy grids contain the history of the measurements, and the final fusion is performed on all intrinsic maps. The high-level framework is more modular than the low-level approach, it could fuse occupancy grid maps from different sources or other modules. This approach is more memory and computation consuming, nevertheless it allows better sensor data validation.

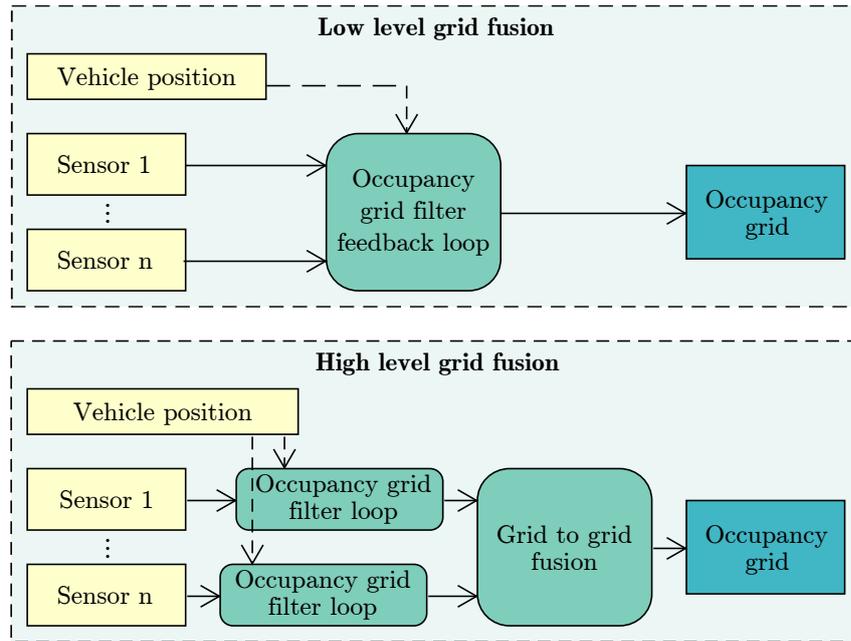


Figure 3.5. Comparison of the low- and high-level grid fusion architectures.

The main distinction between low and high architecture levels is that in the low-level grid the accumulation loop is performed only once per algorithm step, while in high-level it is carried multiple times.

As the accumulation loop is the most time-consuming part of the algorithm, therefore, the low-level grid fusion approach will almost always execute faster than the high-level one. Moreover, the ability to connect all sensor scans in a single node of the algorithm offers the developer more flexibility for sensor conflict resolution or different optimization techniques.

A lot of research does not provide information about the detailed architecture of their occupancy grid implementation; however, the low-level fusion could be assumed to be the most common one due to its simplicity and flexibility. All occupancy grid algorithms presented in this thesis would be based on this architecture, however, elements of the high-level option can be useful for the separation of independent structures if only the synchronization between different grid maps is provided.

3.4 Occupancy grid data flow

At the core of the occupancy grid algorithm is the idea of environment state estimation from the sensor data. This computation addresses the problem of estimating quantities from sensor data that are not directly observable, but that can be inferred. Notwithstanding, sensors carry only partial information about those quantities, and their measurements are corrupted by noise. As the vehicle has to rely on its sensors to gather occupancy data, the sensor information processing is a crucial step which might assess all the uncertainties of the system. Later, the probabilistic state estimation of the occupancy grid algorithm computes belief distributions over possible world states.

The occupancy grid data flow can be divided into four steps, which forms a filtering loop as presented in Figure 3.6. To properly fuse evidences, adequate information from sensor measurements has to be

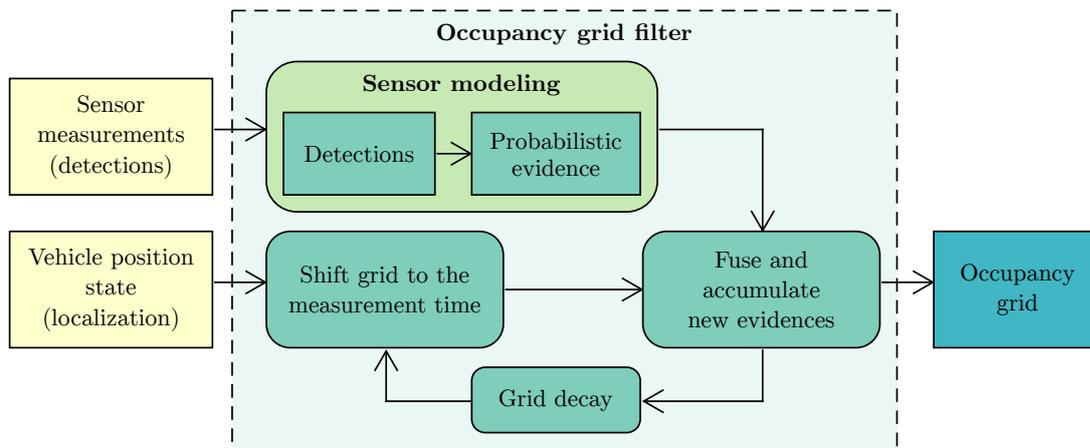


Figure 3.6. Occupancy grid data flow.

extracted and converted in the grid information domain in the sensor modeling step. The key step of producing the occupancy grid is the fusion and accumulation of sensor evidences into the persistent occupancy grid. The fusion step considers the prior map information and measurement probabilistic evidence and produces the predicted occupancy grid. The algorithm processes the new data recursively and to mitigate the accumulation overconfidence, the decay step is needed. The decay acts as a forgetting mechanism, slowly declining cells' evidences. Finally, as the vehicle moves, the grid has to be shifted to ensure the correct positioning relative to the vehicle.

3.4.1 Sensor information processing

In automotive applications, different types of sensors are utilized to ensure the redundancy and robustness of the perception system. Each sensor has different characteristics, which drive the way in which the measurement is represented. Majority of the devices used for environment perception are so-called range sensors [Elfes and Matthies, 1987; Eskandarian, 2012; Thrun et al., 2005]. A range sensor is a device that senses the world by exploiting the properties of a physical support such as light, radio waves, or acoustic waves. The process of sensing outputs a measurement that reflects the world as sensed by the sensor [Andriamahefa, 2017].

3.4.1.1 Vehicle position state interface

In order to properly accumulate environment representation around a moving vehicle the occupancy grid algorithm has to receive precise information about current vehicle state. Such data are usually obtained from a separate component which combines information from the vehicle's inertial measurement

units (IMUs) and from global positioning systems such as GPS. The minimal information required for the occupancy grid to work is the local position and rotation of the vehicle, which can be computed based on vehicle linear and angular velocity or taken from global positioning unit.

3.4.1.2 Sensor data interface and format

Each measurement z from range sensors is able to provide two not complementary types of information. Firstly, range measurements sense the presence of an object (o). Secondly, the reflected rays carry the information about the possible traversal space to the target (f)

$$z \in \{o, f\} \quad (3.4.1)$$

where:

- z – sensor measurement,
- o – range detection describing occupancy from the reflection,
- f – possible traversal space to the target describing free space.

Moreover, the radar sensor used in automotive utilizes the Doppler shift effect to determine the relative velocity of the detection called range rate. Using this information, the Static-Unknown-Dynamic classification can be determined from the single detection

$$o \in \{s, d, \{s \cup d\}\} \quad (3.4.2)$$

- s – static detection,
- d – dynamic object,
- $\{s \cup d\}$ – ambiguous or unknown detection motion status.

The ISO/DIS 23150 standard about logical interface in automotive sensors specifies different sensor measurement representation [International Organization for Standardization, 2021]. For the occupancy grid purposes, the types can be divided into two main groups: point and contour shape measurements (as presented in Figure 3.7). Point detections are implemented in radar and LIDAR sensors, while the contour shape is utilized in vision free space measurements.

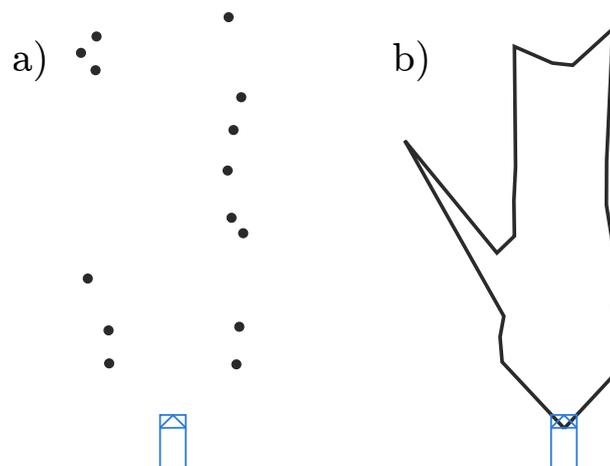


Figure 3.7. Sensor scan measurement types: (a) point (radar and LIDAR detections), (b) contour shape (vision free space envelope).

3.4.2 Sensor modeling

Each occupancy grid input sensor has unique properties, which can be utilized to extract more information about the environment. The sensor modeling part of the occupancy grid algorithm aims to improve the accumulation procedure by transferring as much information as possible from the detection model to the occupancy grid. The sensor model has to consider all types of uncertainties present in the occupancy grid. Usually in the experimental setup, many uncertainty characteristics are not directly measurable, and the sensor model has to approximate the overall grid detection uncertainty, not only the sensor parameters itself.

In the literature, there are two families of sensor models for the occupancy grid: forward and inverse sensor models, further referenced as FSM and ISM accordingly. Both of these sensor models derive the *posteriori* grid probability $p(G|z)$, which can be further used in the fusion step.

Physical sensor model aims to model the behavior of the sensor given some simulated environment to imitate the sensor behavior in the virtual domain. Classical sensor model (e.g., radar model) takes the virtual environment representation and computes the measurements of it. As a result, the model delivers the probability of measurement given map – $p(z|G)$.

In the occupancy grid, the problem of modeling, however, requires the opposite kind of information. The aim is to estimate the probability of a cell being occupied $p(G)$ given the sensor measurement z – $p(G|z)$.

3.4.2.1 Forward sensor model (FSM)

The forward sensor model gives the probability of observing a reading $p(z|G)$ based on map knowledge $p(G)$ [Thrun, 2003]. Forward models have the advantage that they can be determined experimentally and can characterize a sensor in a more straightforward manner.

A prerequisite for using FSM is to have an accurate physical sensor model, which generates $p(z|G)$ from given map probabilities $p(G)$.

The physical sensor model for the sensor is known, forward sensor models promise accurate occupancy modeling. They might consider all sensor specifications, such as multi-bounces or reflections (if the sensor model supports them).

To convert the physical sensor model probability into the forward sensor model output, the Bayes theorem might be applied:

$$p(G|z) = \frac{p(z|G) \cdot p(G)}{p(z|G = \text{Occupied}) \cdot p(z|G = \overline{\text{Occupied}})} \quad (3.4.3)$$

where:

- $p(G|z)$ – probability of a grid cell being occupied given measurements,
- $p(z|G)$ – probability of measurement given map – known, computed by the physical sensor model,
- $p(G)$ – occupancy probability of the given map – unknown,
- $p(z|G = \text{Occupied})$ – true positive rate of sensor – given by the sensor characteristics,
- $p(z|G = \overline{\text{Occupied}})$ – false positive rate of sensor – given by the sensor characteristics.

Nonetheless, the physical sensor model has to be evaluated on already known grid maps (normally the virtual environment provides that, but here only historical information is known). Therefore, the forward sensor model became a single high-dimensional optimization problem aiming to estimate the map $p(G)$ which would generate measured sensor detection z .

In the literature, there are a multitude of FSM techniques tackling this optimization algorithm [Carvalho and Ventura, 2013; Thrun, 2003]. In practice, however, the forward sensor models are still too computationally complex to be applicable in the automotive software stack, which has to process hundreds of detections from multiple sensors at the same time. The alternative for that is an inverse sensor model (ISM).

3.4.2.2 Inverse sensor model (ISM)

Inverse sensor model (ISM) is a tool used to determine the occupancy probability based on sensor detection and their inaccuracies. The aim of the ISM is to detect the real environmental obstacle that causes sensor detection and update the grid with proper probabilities. In opposite of FSM, the ISM is a method to deduce the occupancy (cause) from sensor readings (effect).

The first Bayesian inverse sensor model (ISM) designed for the ultrasonic sensors was presented in the late-80s by Elfes and Matthies [Elfes and Matthies, 1987]. The Bayesian approach computed the ISM utilizing the sensor model, therefore expressing the measurement uncertainties given the physical location of the sensed obstacle. This approach requires the enumeration of all possible grid configurations, which causes the exponential computation complexity.

In order to create the ISM with a linear complexity, the analytical approach has been proposed [Andriamahefa, 2017]. It approximates the sensor model using a continuous function defined over the distance from the sensor. It is based on the Gaussian distribution [Konolige, 1997] or on the power function [Yguel et al., 2008].

The further simplification of the analytical ISM has been presented in [Homm et al., 2010; Weiss et al., 2007; Wurm et al., 2010], which approximates the occupancy depending on the distance from detection. This approach uses simple analytical functions like linear approximation or 3-valued spline [Weiss et al., 2007; Wurm et al., 2010].

Inverse sensor model specifies a probability distribution over map cells m given sensor measurements $z \Rightarrow p(G|z)$. It decomposes the high-dimensional mapping problem into many binary estimation problems, which are then solved independently of each other. Using arbitrary distribution profiles, the ISM might be divided into many binary estimation problems, which can be easily optimized and paralleled.

3.4.2.3 Dual inverse sensor model

Range sensors such as laser-based sensors, stereo cameras, radars, and ultrasonic sensors are commonly used in robotics and for autonomous vehicles. Each detection from these sensors is able to provide two not complementary types of information. Firstly, range measurements sense the presence of an object. Secondly, the reflected rays carry the information about the possible traversal space to the target.

In common implementations of the inverse sensor models, those two information are packed in one distribution [Elfes and Matthies, 1987; Joubert, 2012; Konolige, 1997; Yguel et al., 2008]. Nevertheless, various grid applications require separation of free and occupancy data to solve sensor conflicts [Foroughi et al., 2015] or to estimate additional environment parameters [Valente et al., 2018].

In order to address this problem, the proposed framework unravels the ISM distribution by splitting the process into two separate paths, which internally accumulate two independent information. This implementation of the ISM could be paralleled since the reflection and traversal space paths are independent, as presented in Figure 3.8.

The processing flow presented in Figure 3.8 consists of cell selection probability calculation for a single range sensor detection.

The first step of the processing is the selection of significant grid cells relative to the applied probability distribution. The cell selection policy significantly reduces the computational complexity of the modeling step, because the probability has to be computed only for a subset of cells. The second step is probability calculation, where the actual evidence is extracted from the cell area. For the probability distribution, either Gaussian or power functions are utilized [Porębski, 2020].

The ISO/DIS 23150:2021 standard states that each sensor detection should have an existence probability measurement. This quantity depicts the uncertainty and can be used in the discounting operation in the ISM probability calculation step. Therefore, the discounting step is already embedded in the ISM processing and will not be covered in the sensor fusion section [International Organization for Standardization, 2021].

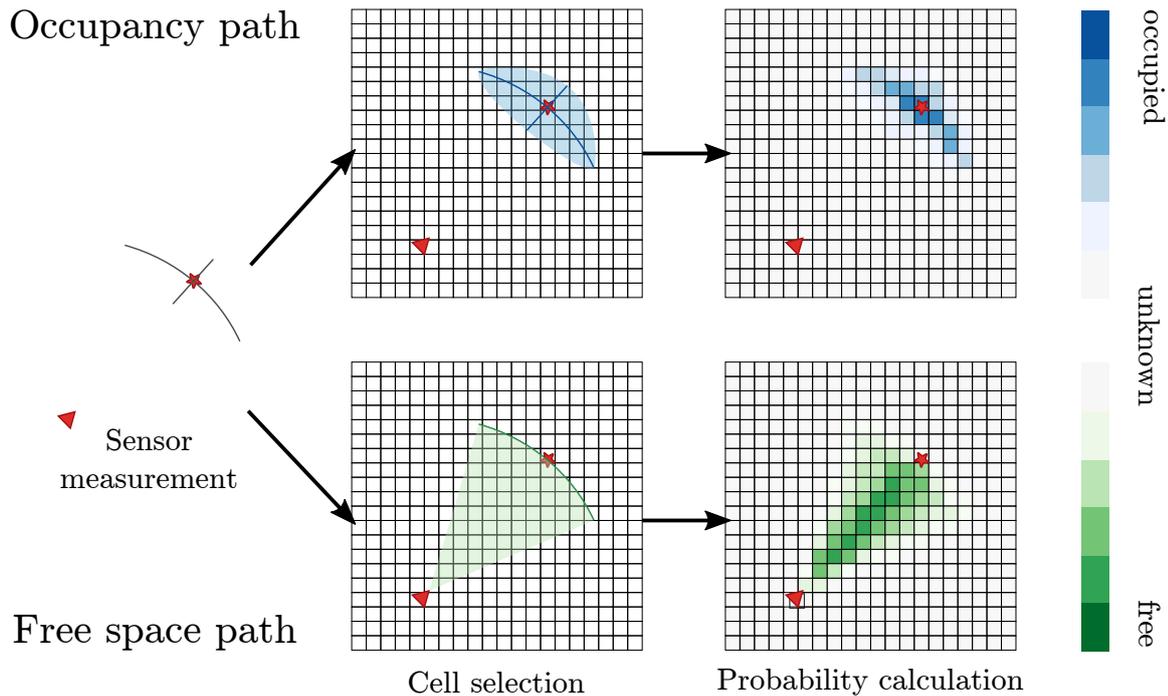


Figure 3.8. Graphical illustration of the sensor model data flow.

3.4.2.4 Intermediate grid evidence separation

In common implementations of occupancy grid algorithms, all the aforementioned information types are packed into a single processing data flow [Elfes and Matthies, 1987; Joubert, 2012; Konolige, 1997; Yguel et al., 2008]. This approach is not commutative, so the output grid depends on the order of data processing even within single sensor scan [Thrun, 2003].

As a solution to that problem, Foroughi et al. in [Foroughi et al., 2015] introduced two independent grid maps: one for occupancy and one for free space accumulation. This approach requires separation of cell states, but can be used to solve sensor conflicts [Foroughi et al., 2015] or to estimate additional environment parameters [Valente et al., 2018].

Accordingly, the proposed occupancy grid framework unravels the probability representation by splitting the process into separate paths, which internally accumulate independent information. The types of evidences accumulated within the occupancy grid framework mimics all the information provided from sensors. Automotive sensors, however, might provide four types of evidences and their hierarchical illustration is presented in Figure 3.9.

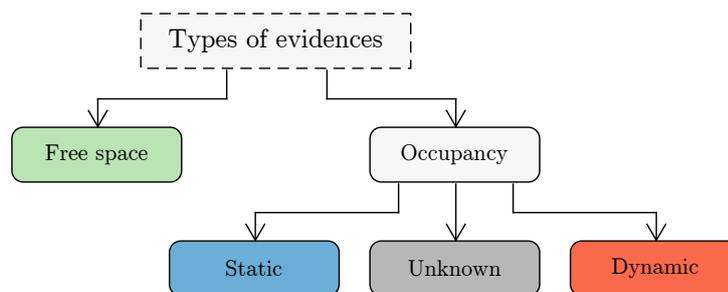


Figure 3.9. Types of independent evidences maintained in the proposed occupancy grid framework.

Each processing path accumulates a different probability type on a separate intermediate grid. The intermediate grids (IG), similarly to the main occupancy grid, are matrices, where each element describes a portion of a space.

The occupancy information can be internally divided into three groups as specified in (3.4.2), thus they are managed on three intermediate grids representing different evidences: Static IG_S , Dynamic IG_D and Unknown occupancy IG_{SUD} . Depending on the used sensors and fusion framework, some of these intermediate grids might be disabled to reduce the memory requirement of the algorithm. Values of the occupancy intermediate grid express the probability of an i -th cell c_i being an origin for any of processed sensor occupancy $\{s, d, \{s \cup d\}\}$

$$\begin{aligned} IG_S(c_i) &= p(c_i|s) \\ IG_D(c_i) &= p(c_i|d) \\ IG_{SUD}(c_i) &= p(c_i|\{s \cup d\}) \end{aligned} \quad (3.4.4)$$

Respectively, the free space data IG_F describe probability of the cell being free from obstacles f

$$IG_F(c_i) = p(c_i|f) \quad (3.4.5)$$

Intermediate grids store information only between each fusion iteration. In that period, the incoming measurements z^t are fused into intermediate grids. The fusion rule for each grid could be different and should depict the kind of information provided by sensors. Each consecutive reflection from some obstacle is a new evidence, so it should increase the occupancy intermediate grid probability. For that grid, the De Morgan's fusion method is used:

$$\begin{aligned} IG_S^{t+1}(c_i) &= 1 - (1 - IG_S^t(c_i)) (1 - p(c_i|s^t)) \\ IG_D^{t+1}(c_i) &= 1 - (1 - IG_D^t(c_i)) (1 - p(c_i|d^t)) \\ IG_{SUD}^{t+1}(c_i) &= 1 - (1 - IG_{SUD}^t(c_i)) (1 - p(c_i|u^t)) \end{aligned} \quad (3.4.6)$$

On the other hand, the free-space information IG_F^t depend mostly on different sensor capabilities instead as the number of traversed rays. To meet that assumption, a maximum policy could be applied:

$$IG_F^{t+1}(c_i) = \max(IG_F^t(c_i), p(c_i|f^t)) \quad (3.4.7)$$

3.4.3 Evidence fusion and accumulation

At the core of the occupancy grid algorithm is the idea of estimating the state from sensor data. Probabilistic state estimation algorithms compute belief distributions over possible world states. The fusion and accumulation might be refereed as a measurement update step in common filtering algorithms such as Bayes of Kalman Filter [Thrun et al., 2005].

In the occupancy grid, evidence accumulation is performed by recursive fusion of new sensor data into the persistent occupancy grid. The topic of different fusion frameworks is discussed in detail in the next Chapter 4.

3.4.4 Decay

Occupancy and free space modeling simplify many physical dependencies of the sensor measurement. For example, it assumes data completion and cell independence. These hidden dependencies are omitted during the fusion process of integrating multiple pieces of evidence. Consequently, the results quickly become overconfident [Thrun et al., 2005].

In order to deal with the problem, the information decaying procedure is applied. In the occupancy grid algorithm, the decay corresponds to the prediction update step in other filters. To model the forgetting of the older evidences, it artificially diminishes the evidence on the entire grid over time. The

overconfidence of occupancy modeling may be handled using different methods, applying the grid prediction [Thrun et al., 2005].

Nevertheless, the common forgetting procedure is an exponential decay. It increases the uncertainty of grid cells while preserving the overall variance of the map. The exponential decay for the occupancy grid is described as:

$$p(c_i)^{t+\Delta t} = (p(c_i)^t - p(c_i)^{t_0}) \cdot e^{-\frac{\Delta t}{\tau}} + p(c_i)^{t_0} \quad (3.4.8)$$

where:

- $p(c_i)^t$ – probability of the i -th cell at time t ,
- $p(c_i)^{t_0}$ – initial probability of the cell, for Bayesian occupancy grid $p(c_i)^{t_0} = 0.5$, for evidence-based grid types $p(c_i)^{t_0} = 0.0$
- Δt – time elapsed between two decay operations,
- τ – mean lifetime.

Mean lifetime represents the time period after which only $\frac{1}{e} \approx 36.7\%$ of the evidence remains in the cell.

The behavior of the decay depends on the update rate of the occupancy grid, which influences the decay rate value. The example processes of decaying for both occupied and free cells for various τ values is depicted in Figure 3.10. The decay always tends towards the most uncertain probability value, $p(c_i)^{t_0}$ which is 0.5 for the Bayesian inference method and 0.0 for the evidence-based grids.

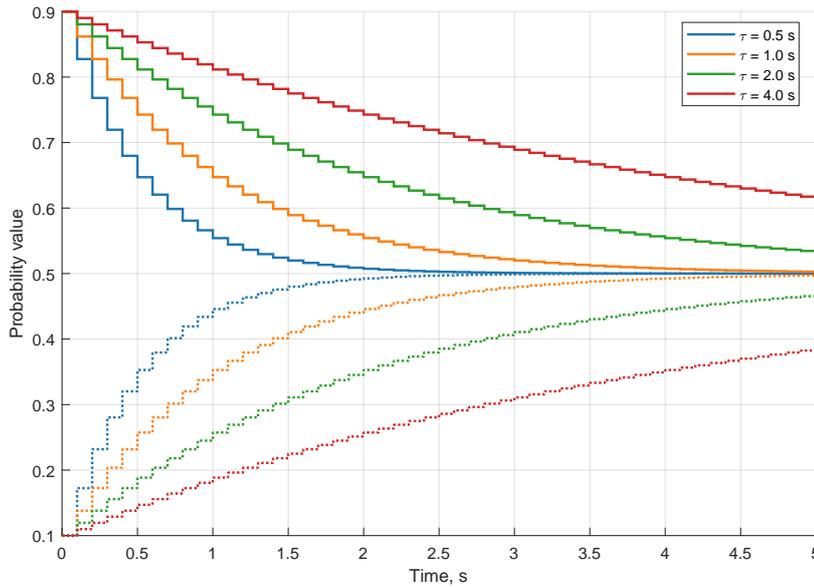


Figure 3.10. Decaying process of the occupied cell with $p = 0.9$ (solid lines) and free cell $p = 0.1$ (dotted lines) for different mean lifetimes τ . The decay is applied with 10 Hz frequency.

3.4.5 Grid shift

The automotive occupancy grid algorithm is highly limited by the memory consumption requirements. Therefore, the automotive occupancy grid represents only the closest vehicle surroundings. Each cell of the grid, however, always represents the same area of the environment, thus as the host moves, the grid must be shifted. The shift might be performed perpendicular to the cell side by the distance equal to the integer multiplication of the cell size. Illustration of the grid shift procedure is presented in Figure 3.11.

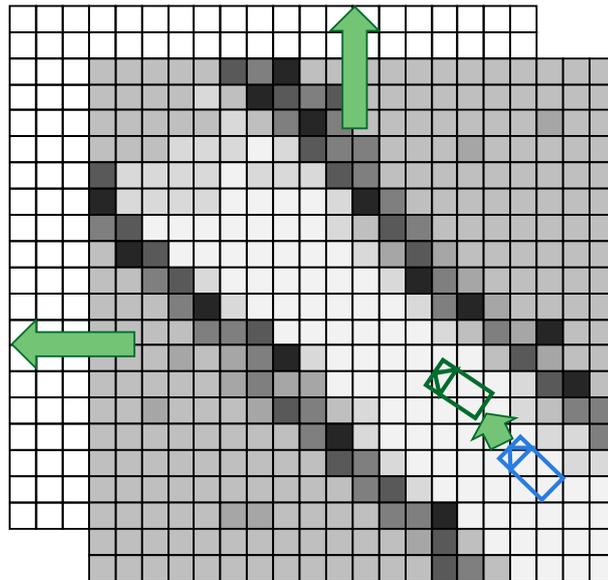


Figure 3.11. Graphical illustration of the grid shift procedure.

The grid representation is never rotated to prevent cell approximation and Moiré patterns [Weiss et al., 2007]. To enable vehicle turning, the host position on the grid is continuous and can rotate freely.

Vehicle might be positioned in the center of the occupancy map as presented in Figure 3.12(a). This positioning is useful in low velocity or urban scenarios, where the vehicle has to monitor the surroundings on all sides. For other applications, where the vehicle is moving relatively fast in a forward direction, the host might be moved closer to the grid border. In this application, the ego vehicle is positioned on a constant radius circle from the grid center, allowing better usage of the available grid space (Figure 3.12b) [Porębski et al., 2019]. Shift distance is computed before each sensor fusion step ensuring correct grid position for every sensor measurement.

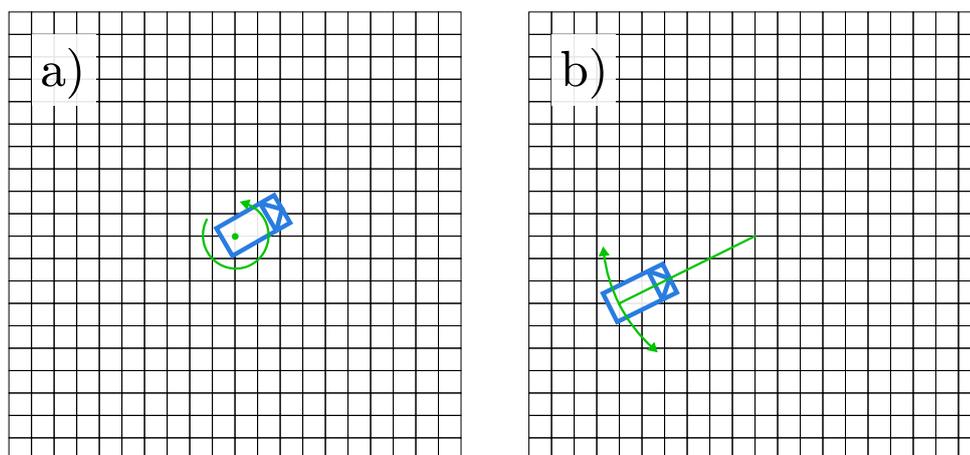


Figure 3.12. Types of host positioning (a) center, (b) constant radius [Porębski et al., 2019].

3.5 Sensor conflict resolution

The occupancy grid algorithm has to deal with heterogeneous data sources. As mentioned in the Section 2.4.2, each sensor has its own advantages and disadvantages. Therefore, a conflict in the perceived environment state might happen and has to be resolved by the occupancy grid fusion.

Conflicts in the occupancy grid processing can be divided depending on the period in which they happen. If conflicting sensor information comes from the same sensor and time stamp, they can cause self-sensor conflicts. If the disagreement is between two separate sensor scans coming from different timestamps, it results in multi-sensor conflicts.

Causes of those conflicts can be different, hence they should be handled accordingly in the occupancy grid processing. This section describes the types of sensor conflicts, while detailed examples of these conflicts' resolution are presented in Section 4.3.

3.5.1 Self-sensor conflicts

Commonly, automotive perception sensors produce sensor detection in batches, called sensor scans. Sensor scan is an instant measure of the environment state. In the occupancy grid processing presented in this thesis, four different evidences are extracted from the scan and represented on four intermediate grids (IG_S , IG_D , IG_F , IG_{SUD}). As the intermediate grids are independent evidences, fusion of those sources of information may cause self-sensor conflict.

Wrongly classified radar detection can be an example of conflict measurement. If a dynamic object is moving perpendicular to the radar center line, the detection coming from this object might be wrongly classified as stationary. This may result in, i.e., two detections close to each other with conflicting stationary and dynamic evidences.

This family of sensor conflicts also consists all false positive detections (detection of not existing objects) and sensor modeling inaccuracies. Single timestamp conflicts are instantly resolved by the intermediate grid fusion step of the occupancy grid algorithm. An example of this conflict resolution is presented in Section 4.3.

3.5.2 Multi sensor conflicts

Automated vehicle has to operate in a dynamic environment. Nevertheless, due to sensor measurement latency or rapid changes in the surroundings, consecutive sensor readings may vary significantly. As every grid cell stands for the same portion of space regardless of time, that causes the fusion of conflicting measurements with the prior cell state.

As an example of that, multi-sensor conflict can be caused by dynamic objects present in the occupancy grid. As the object moves, its dynamic occupancy traverses to new grid cells. The area behind the object has to be made free with new sensor measurements, creating a dynamic occupancy residual.

Multi sensor conflicts also cover contradictions between different sensors, which might happen in case of changing weather conditions or due to differences in sensor accuracy. The contradicting information in this type of conflicts are subjected to the stochastic filtering of occupancy grid fusion. An example of this conflict resolution is presented in Section 4.3.

4 Heterogeneous information fusion

The occupancy grid is a comprehensive perception module, interconnecting the sensors with more abstract downstream components. One of the major advantages of the grid-based environment representation is that it can unify heterogeneous sensor domains into a single map of occupancy. In automotive applications, different types of sensors are utilized to ensure the redundancy and robustness of the perception system.

A probabilistic model of the environment, the occupancy grid, can be defined using different probability frameworks. Focusing on their applicability for the automotive perception module, three main probability frameworks are defined in Section 4.1.

The output occupancy map representation is an amalgamation of the input data. Section 4.2 investigates different fusion architectures and rules of combination which can be applied in the automotive occupancy grid. In order to validate the correctness of the occupancy grid, the model of the environment has to be properly visualized. Proposed approach for grid visualization techniques is presented in Section 4.2.5.

Finally, Section 4.3 defines the combination rule equations for the three evaluated fusion frameworks and provides a simple example to present and compare different combination rules.

4.1 Probability and uncertainty representation formalisms

The management and combination of uncertain, imprecise, or highly conflicting sources of information is a crucial part of the occupancy grid algorithm. Historically, the first probabilistic inference method was Bayes theorem, founded in the XVIII century. More recently, the emergence of artificial intelligence and machine learning algorithms resulted in a growth of interest relating to the management of uncertainty and evidential reasoning. This resulted in the generalization of the classical Bayesian probability by the Dempster-Shafer theory (DST) of belief functions [Shafer, 1976]. Nevertheless, rapid progress continues and in 2004, F. Smarandache and J. Dezert proposed further generalization of the DST framework into Dezert-Smarandache Theory (DSmT) of plausible and paradoxical reasoning [Smarandache and Dezert, 2004]. All these three methods found application as occupancy grid information fusion frameworks.

The Bayesian probability framework has been used since the inception of occupancy grids [Elfes and Matthies, 1987; Rakotovo et al., 2016; Weiss et al., 2007]. The Bayesian framework assumes complete knowledge about the cell state which can be either "occupied" or "free" and these states are mutually exclusive. Each grid cell maintains a binary random variable, which values depict the probability of a cell state. When the probability is high, the cell is more likely occupied, while when it is low, the cell is believably free. Due to the limited hypothesis set, no measurement uncertainty modeling is possible, and the binary cell states are interconnected. If any sensor decreases the occupancy probability, it implies increasing free space estimation.

As an extension to the possible cell state, the Dempster-Shafer approach introduces unions of states, along with the idea of a "mass", or a measure of confidence in each of the alternatives [Challa and Koks, 2004]. Therefore, Dempster-Shafer grid cell will be described not only by the "occupied" and "free"

states, but also as their union "unknown" hypothesis ("occupied" or "free" states). This breaks the connection between "occupied" and "free" and these beliefs are now independent values. In the occupancy grid area, the Dempster-Shafer evidence has gained advantages over Bayesian for it being able to clearly distinguish the ignorance and the contradiction [Yi et al., 2000]. Specifically for the automotive application, DST is used mostly for the dynamic occupancy grid application, because it enables multiple cell state representation [Nuß, 2017; Tanzmeister et al., 2014].

The third and latest probabilistic reasoning framework is the Dezert-Smarandache Theory (DSmT). It is a generalization of the DST framework, which extends the possible beliefs with also intersections of states. As an example, for the "occupied"–"free" grid cell state, the DSmT cell would estimate both "unknown" ("occupied or free") and "occupied and free" beliefs. In the actual environment, the second one: "occupied and free" is never a real cell state, however, this belief can be especially useful when managing the resolution of sensor conflicts. The framework is relatively new, yet it has already proposed multiple new fusion rules, and it can be easily reduced to the classical Dempster-Shafer frame of discernment [Smarandache and Dezert, 2005]. Despite its novelty, the DSmT framework has been successfully applied in some occupancy grid applications for example in [Dezert et al., 2015; Moras et al., 2015; Wang et al., 2018].

This thesis presents a comparison of different occupancy grid frameworks, focusing on their applicability in automated vehicle perception. The following paragraphs describe the aforementioned frameworks in the details needed for implementation of the functional occupancy grid algorithm.

4.1.1 Bayesian probability framework

The binary Bayesian framework is the simplest one considered in this chapter. Assumptions and applications of the Bayesian framework have been thoroughly described in multiple prior arts such as [Challa and Koks, 2004; Thrun et al., 2005]. Therefore, in this thesis, the author describes only the minimal level of detail required for understanding and implementation of the presented variants of the occupancy grid algorithms.

In the Bayesian occupancy grid, each cell state X is described as a single binary random variable defining if the cell is occupied or not.

$$X \in \{\text{occupied}, \text{free}\} \quad (4.1.1)$$

These two cell states are mutually exclusive; therefore, the sum of their probabilities is equal to 1.

$$p(\{\text{occupied}\}) + p(\{\text{free}\}) = 1 \quad (4.1.2)$$

The probability $p(\{\text{occupied}\})$ denotes the probability that the cell is occupied.

In this framework, there is no place for any uncertainty management or reallocation of sensor conflicts. All measurements are taken as true without any confidence approximation.

4.1.2 Dempster-Shafer Theory (DST)

Dempster-Shafer Theory (DST), also called mathematical theory of evidence, was introduced in 1976. This framework is attractive for the occupancy grid application because it gives a nice mathematical model for the representation of uncertainty, and it includes Bayesian theory as a special case [Shafer, 1976; Yi et al., 2000].

This section presents a short introduction to the Dempster-Shafer theory required for the occupancy grid implementation and development. A complete presentation of the mathematical theory of evidence proposed by Glenn Shafer can be found in his book in [Shafer, 1976].

The classic Dempster-Shafer Theory assumes an exhaustive finite frame of discernment Θ :

$$\Theta = \{\theta_1, \theta_2, \dots, \theta_n\} \quad (4.1.3)$$

whose elements are mutually exclusive. That means the hypotheses θ_i (also called elemental hypotheses) are well precisely defined (identified) in such a way that they do not overlap and fill the whole possibility space. The frame of discernment is a key element building the proposition set \mathcal{P}^Θ , which describe all permitted combination of hypotheses. The proposition set \mathcal{P}^Θ is also called body or corpus of evidence and sometimes noted as \mathfrak{B} .

A Basic Belief Assignment (BBA) is a mapping $m(\cdot) : \mathcal{P}^\Theta \rightarrow [0, 1]$, such that:

$$\sum_{A \in \mathcal{P}^\Theta} m(A) = 1 \quad (4.1.4)$$

Despite that the DST is a matured mathematical theory, it is in permanent dynamic evolution, therefore, there is no established nomenclature for some of its elements. The quantity $m(A)$ can be called: Basic Belief Assignment (BBA), Basic Probability Assignment (BPA) or Basic Belief Mass (BBM) of A . For the consistency of naming in this work, the value $m(A)$ will be referred to as Basic Belief Assignment (BBA) or mass of A .

The belief (credibility) and plausibility functions of $A \subseteq \mathcal{P}^\Theta$ are defined as:

$$\text{Bel}(A) = \sum_{X \subseteq A} m(X) \quad (4.1.5)$$

$$\text{Pl}(A) = \sum_{X \in \mathcal{P}^\Theta, X \cap A \neq \emptyset} m(X) = 1 - \text{Bel}(\bar{A}) \quad (4.1.6)$$

where \bar{A} denotes the complement of the proposition A in \mathcal{P}^Θ .

The belief function Bel uniquely corresponds to BBA of m and vice versa. Belief and plausibility can be interpreted as an upper and lower probability assigned to the BBA.

Another method of conversion from the BBA into probability is the transformation defined as:

$$\text{betP}(A) = \sum_{X \subseteq \Theta} \frac{|X \cap A|}{|X|} \frac{m(A)}{1 - m(\emptyset)} \quad (4.1.7)$$

where $|X|$ denotes the number of states in the set X (cardinality of X).

Glenn Shafer in [Shafer, 1976, p.35–37] considers the proposition set called power set 2^Θ , which consists of all elementary hypotheses θ_i and their unions ($2^\Theta := (\Theta, \cup)$).

As an example of the power set elements, consider a three-element frame of discernment:

$$\Theta = \{\theta_1, \theta_2, \theta_3\} \quad (4.1.8)$$

The power set based on that frame has a cardinality of $|2^\Theta| = 8$ with elements denoted as:

$$2^\Theta = \{\alpha_0, \alpha_1, \dots, \alpha_7\} \quad (4.1.9)$$

where each α_i corresponds to a set of elemental hypotheses θ_i . In the example power set, the α_i sets are defined as:

$$\begin{aligned} \alpha_0 &:= \emptyset & \alpha_4 &:= \theta_1 \cup \theta_2 \\ \alpha_1 &:= \theta_1 & \alpha_5 &:= \theta_1 \cup \theta_3 \\ \alpha_2 &:= \theta_2 & \alpha_6 &:= \theta_2 \cup \theta_3 \\ \alpha_3 &:= \theta_3 & \alpha_7 &:= \theta_1 \cup \theta_2 \cup \theta_3 \end{aligned} \quad (4.1.10)$$

The set $\alpha_7 = \theta_1 \cup \theta_2 \cup \theta_3 = \{\Theta\}$ is called vacuous set and the $m(\Theta)$ is described as the Vacuous Belief. If $m(\Theta) = 1$ it defines a fully ignorant state, where no hypotheses are available, assignment. This belief assignment is called Vacuous Belief Assignment (VBA) and will be denoted as $m_v(\cdot)$

$$m_v(\cdot) : m_v(\Theta) = 1 \quad \wedge \quad m_v(A \neq \Theta) = 0 \quad (4.1.11)$$

The role of the set $\alpha_0 = \emptyset$ is also important as it defines the conflict state and might indicate that the frame of discernment set Θ is not exhaustive. In the classic Dempster-Shafer Theory, $m(\emptyset) := 0$, which is also called as "closed world assumption". This restriction was revoked by Smets in the Transferable Belief Model (TBM), which allows $m(\emptyset) \geq 0$. If this mass is greater than zero, it means that the frame of discernment does not define the environment well enough. That generalizes DST by enabling the accumulation of conflict belief and can be useful in the fusion methods definition. More details about the Transferable Belief Model can be found in [Smets, 1990].

In the occupancy grid area, the DST has been applied due to its property to distinguish the ignorance and contradiction from sensor reading. Using Dempster-Shafer theory, the measurements can have some uncertainty, which is applied in a discarding procedure of sensor fusion. The most common frames of discernment used in the occupancy grids are:

$$\Theta = \{O, F\} \quad \text{or} \quad \Theta = \{S, D, F\} \quad (4.1.12)$$

- O – occupancy state – cell is occupied,
- F – free state – area is drivable,
- S – cell is occupied by stationary obstacle,
- D – there is a dynamic/moving object within the cell.

This thesis will cover only the second frame of discernment $\Theta = \{S, D, F\}$, because it also includes the smaller frame $\Theta = \{O, F\}$ and provides more flexibility in fusion and dynamic object handling in the occupancy grid.

For the frame of discernment $\Theta = \{S, D, F\}$, the power set 2^Θ consist of eight elements as defined in (4.1.10). For this frame elemental hypotheses θ_i are defined as:

$$\theta_1 = \{S\} \quad \theta_2 = \{D\} \quad \theta_3 = \{F\} \quad (4.1.13)$$

and the meaning of all possible Basic Belief Assignments is:

- $m(\alpha_0) = m(\emptyset)$ – conflict evidence about states not covered by other BBAs,
- $m(\alpha_1) = m(\{S\})$ – static evidence,
- $m(\alpha_2) = m(\{D\})$ – dynamic evidence,
- $m(\alpha_3) = m(\{F\})$ – free space evidence,
- $m(\alpha_4) = m(\{S \cup D\})$ – static or dynamic evidence – ambiguous evidence of undefined occupancy also referred as $m(O)$,
- $m(\alpha_5) = m(\{S \cup F\})$ – static or free space evidence – cell state is uncertain but not dynamic,
- $m(\alpha_6) = m(\{D \cup F\})$ – dynamic or free space evidence – cell state is uncertain but not stationary,
- $m(\alpha_7) = m(\{S \cup D \cup F\})$ – static or dynamic or free space evidence – no information, cell state is fully uncertain.

Belief $m(\alpha_0)$ represent conflicting information or evidence not covered by the other states. By the definition of the DST $m(\alpha_0) := 0$.

Beliefs $\{m(\alpha_1), m(\alpha_2), m(\alpha_3)\}$ provide evidences for the raw elemental hypotheses of the frame of discernment, i.e., stationary, dynamic, or free. Masses $\{m(\alpha_4), m(\alpha_5), m(\alpha_6)\}$ can be explained as the negative evidence for the corresponding elemental hypothesis, i.e., not-free, non-dynamic, non-stationary. As will be presented in the Section 4.3.1.2 some fusion rules might result in locking evidences $m(\alpha_5)$ and $m(\alpha_6)$ on the value of 0 therefore, they are discarded by some works such as [Steyer et al., 2018]. Nevertheless, these cell states might be correct, especially if some fusion rule might provide evidences for negative evidence.

Belief $m(\alpha_7)$ is also called unknown state or VBA (Vacuous Belief Assignment) and sometimes denoted as $m(\Theta)$. This mass is equal to 1 in the initial state of the system, when no evidences are available.

4.1.3 Dezert-Smarandache Theory (DSmT)

Dempster-Shafer modified theory or Dezert-Smarandache theory (DSmT) by Dezert and Smarandache [Dezert, 2002; Smarandache and Dezert, 2004] allows mutually overlapping elements of a frame of discernment. Thus, a frame of discernment is a finite exhaustive set of elements Θ , but not necessarily exclusive in DSmT [Daniel, 2014]. DSmT uses Basic Belief Assignments and belief functions defined analogically to the classic Dempster-Shafer theory (DST), but they are defined on a so-called hyper-power set or Dedekind lattice D^Θ instead of the classic power set 2^Θ of the frame of discernment.

The hyper-power set D^Θ is defined as the set of all composite propositions built from elements of Θ with \cup and \cap operators ($D^\Theta := (\Theta, \cup, \cap)$) [Smarandache and Dezert, 2004].

As an example of the hyper-power set definition, let us consider the same three-elements' frame of discernment as in the previous paragraph:

$$\Theta = \{\theta_1, \theta_2, \theta_3\} \quad (4.1.14)$$

The hyper-power set based on that frame has a cardinality of $|D^\Theta| = 19$ with elements denoted as:

$$D^\Theta = \{\alpha_0, \alpha_1, \dots, \alpha_{18}\} \quad (4.1.15)$$

where each $\alpha_i : i \in \{0, \dots, 18\}$ corresponds to a set of elemental hypotheses $\theta_j : j \in \{1, 2, 3\}$. In the example hyper-power set, the α_i sets consist of all possible intersection sets:

$$\begin{array}{ll}
\alpha_0 := \emptyset & \alpha_{10} := (\theta_2 \cap \theta_3) \cup \theta_1 \\
\alpha_1 := \theta_1 & \alpha_{11} := \theta_1 \cap \theta_2 \cap \theta_3 \\
\alpha_2 := \theta_2 & \alpha_{12} := (\theta_1 \cup \theta_2) \cap \theta_3 \\
\alpha_3 := \theta_3 & \alpha_{13} := (\theta_1 \cup \theta_3) \cap \theta_2 \\
\alpha_4 := \theta_1 \cup \theta_2 & \alpha_{14} := (\theta_2 \cup \theta_3) \cap \theta_1 \\
\alpha_5 := \theta_1 \cup \theta_3 & \alpha_{15} := \theta_1 \cap \theta_2 \\
\alpha_6 := \theta_2 \cup \theta_3 & \alpha_{16} := \theta_1 \cap \theta_3 \\
\alpha_7 := \theta_1 \cup \theta_2 \cup \theta_3 & \alpha_{17} := \theta_2 \cap \theta_3 \\
\alpha_8 := (\theta_1 \cap \theta_2) \cup \theta_3 & \alpha_{18} := (\theta_1 \cap \theta_2) \cup (\theta_1 \cap \theta_3) \cup (\theta_2 \cap \theta_3) \\
\alpha_9 := (\theta_1 \cap \theta_3) \cup \theta_2 &
\end{array} \quad (4.1.16)$$

The basis of the DSmT is the refutation of the principle of exclusivity of the Shafer's propositions. The Dezert Smarandache model (DSm) can be applied to describe fuzzy sets of states, where the definition of elements θ_i cannot be properly identified and precisely separated. The full propositions hyper-power set with all possible intersections and unions is called free, because no other assumption is done on the hypotheses. Nevertheless, depending on the intrinsic nature of the elements, the free model might not fit the reality. Some elements of D^Θ can be truly exhaustive or truly non-existing at all at the given time. For example, in the occupancy grid state $\{O \cap F\}$ ("occupied" and "free") is never possible, as the area cannot be occupied and free at the same time. Therefore, these integrity constraints can be explicitly and formally introduced to form a hybrid Dezert Smarandache model [Smarandache and Dezert, 2004].

The classical Dempster-Shafer model is an example of hybrid DSm, where all intersections are constrained to be empty sets. For the three-element example from (4.1.16) the hyper-power set (4.1.15) is

defined as:

$$\begin{aligned}
\alpha_0 &:= \emptyset & \alpha_{10} &:= (\theta_2 \cap \theta_3) \cup \theta_1 = \alpha_1 \neq \emptyset \\
\alpha_1 &:= \theta_1 \neq \emptyset & \alpha_{11} &:= \theta_1 \cap \theta_2 \cap \theta_3 = \emptyset \\
\alpha_2 &:= \theta_2 \neq \emptyset & \alpha_{12} &:= (\theta_1 \cup \theta_2) \cap \theta_3 = \emptyset \\
\alpha_3 &:= \theta_3 \neq \emptyset & \alpha_{13} &:= (\theta_1 \cup \theta_3) \cap \theta_2 = \emptyset \\
\alpha_4 &:= \theta_1 \cup \theta_2 \neq \emptyset & \alpha_{14} &:= (\theta_2 \cup \theta_3) \cap \theta_1 = \emptyset \\
\alpha_5 &:= \theta_1 \cup \theta_3 \neq \emptyset & \alpha_{15} &:= \theta_1 \cap \theta_2 = \emptyset \\
\alpha_6 &:= \theta_2 \cup \theta_3 \neq \emptyset & \alpha_{16} &:= \theta_1 \cap \theta_3 = \emptyset \\
\alpha_7 &:= \theta_1 \cup \theta_2 \cup \theta_3 \neq \emptyset & \alpha_{17} &:= \theta_2 \cap \theta_3 = \emptyset \\
\alpha_8 &:= (\theta_1 \cap \theta_2) \cup \theta_3 = \alpha_3 \neq \emptyset & \alpha_{18} &:= (\theta_1 \cap \theta_2) \cup (\theta_1 \cap \theta_3) \cup (\theta_2 \cap \theta_3) = \emptyset \\
\alpha_9 &:= (\theta_1 \cap \theta_3) \cup \theta_2 = \alpha_4 \neq \emptyset & &
\end{aligned} \tag{4.1.17}$$

The frame D^Θ defined with constrains (4.1.17) has now $|D^\Theta| = 8 = |2^\Theta|$ has different elements and corresponds to Shafer's model. This proposition set definition will be used in the application of the hybrid DSm fusion rules in occupancy grids.

In the hybrid Dezert-Smarandache model definition for the stationary-dynamic-free example, all elemental hypotheses are exclusive, i.e., there should not be any cells which are at the same time for example stationary and free. Meaning of the states $\{\alpha_0, \dots, \alpha_7\}$ is the same as in the Dempster frame as described in Section 4.1.2. Propositions $\{\alpha_8, \dots, \alpha_{18}\}$ in the hybrid DSmT model describe conflict states. These sets of elemental hypotheses cannot be observed in the real world, but they might be used for deduction of the origin of contradictory information. Meaning of the states $\{\alpha_{15}, \alpha_{16}, \alpha_{17}, \alpha_{11}\}$ could be described as:

$$\begin{aligned}
\alpha_{15} &= \{S \cap D\} & - & \text{stationary and dynamic state,} \\
\alpha_{16} &= \{S \cap F\} & - & \text{stationary and free state,} \\
\alpha_{17} &= \{D \cap F\} & - & \text{dynamic and free state,} \\
\alpha_{11} &= \{S \cap D \cap F\} & - & \text{stationary and dynamic and free state.}
\end{aligned}$$

The state $\alpha_{18} = \{(S \cap D) \cup (S \cap F) \cup (D \cap F)\}$ describes that the grid cell is in either $\{S \cap D\}$ or $\{S \cap F\}$ or $\{D \cap F\}$ states. Propositions $\{\alpha_8, \alpha_9, \alpha_{10}\}$ narrows down the hypothesis to a single intersection and a union of states. And finally, the states $\{\alpha_{12}, \alpha_{13}, \alpha_{14}\}$ define which intersection can be excluded, as the states $\{\alpha_4, \alpha_5, \alpha_6\}$ exclude one of the elemental hypotheses.

In the occupancy grid area, the DSmT was applied to utilize the fusion rules specially defined for this framework on the two elemental frame of discernment $\Theta = \{O, F\}$ [Dezert et al., 2015; Moras et al., 2015; Wang et al., 2018]. This thesis presents application of the hybrid DSm with Shafer constrains on the three-element frame of discernment $\Theta = \{S, D, F\}$.

4.2 Heterogeneous fusion rules

Probabilistic information fusion is the core step of the occupancy grid algorithm. In this step, the processed sensor scan in the form of intermediate grid evidences is fused into a prior occupancy grid state. This produces a new grid posteriori estimate as in the measurement update step in common filtering algorithms such as Bayes of Kalman Filter [Thrun et al., 2005]. The sensor information fusion can be performed using different fusion rules, tailored for specific applications and requirements.

Each sensor scan consists of a number of detections or a list of contour points (see Section 3.4.1.2). The sensor scan corresponds to a single time measurement, meaning all detections have the same time stamp.

As presented in the Section 3.4.2.4, automotive sensor information can be divided into four different evidences represented on four intermediate grids. All intermediate grids can be produced from the same sensor scan; therefore, they should contribute equally to the a posteriori grid output. Moreover, the fusion of the intermediate grids should be associative and commutative, because all these evidences come from the same time and contribute the same environment state.

On the other hand, new cell state is inferred only from prior information and current measurements. In a real system we might safely assume that the time cannot flow backward, and the new measurements are fused in natural chronological order (from the oldest to newest). Based on that there is no need to maintain the full associativity of different timestamps data fusion and this operation may not be reversible (reverting time flow can give different results).

Therefore, the fusion of the intermediate grids has to be associative, nevertheless the combination with the prior cell state does not have to hold that condition. For this specific occupancy grid application, three common fusion architectures can be specified as presented in Figure 4.1.

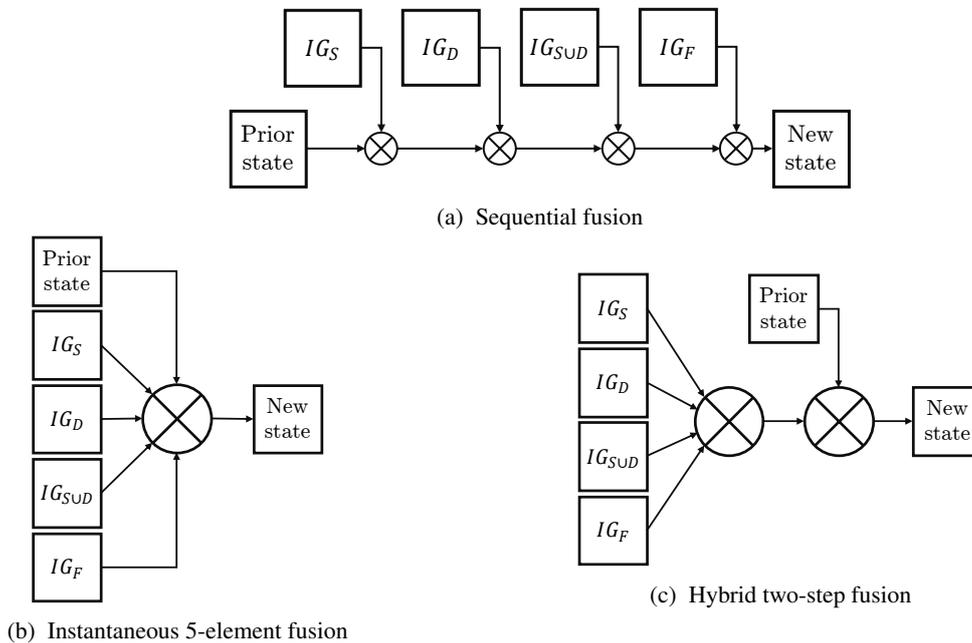


Figure 4.1. Fusion architectures designed for combination of intermediate grids into prior occupancy grid cell states. The symbol \otimes represents fusion operation.

The first option is to fuse the intermediate grids sequentially directly into the prior state (Figure 4.1(a)). As an opposite, the second type of fusion is instantaneous or centralized fusion (Figure 4.1(b)). Third option separates the fusion of measurement intermediate grids and the prior state in a hybrid two-step fusion (Figure 4.1(c)).

While there are various fusion rules which can be applied in the occupancy grid algorithm, a few of them hold the associativity property: Bayesian, Dempster and conjunctive rule. Only these rules can be applied if the sequential or centralized fusion architecture is utilized. In the hybrid two-step fusion, the associative rule has to be applied in the first step, while the other step can be selected from a bigger set of non- or semi-associative combination rules.

The hybrid two-step fusion is already recommended for application in some complex fusion rules such as hybrid Dezert-Smarandache (DSmH) or Proportional Conflict Redistribution v.1-6 (PCR1-6) rules of combination [Smarandache and Dezert, 2004, 2005]. Based on these premises, the hybrid two-step fusion is applied in the presented occupancy grid algorithm.

The following section will describe the fusion rules applicable for the different probability representation formalisms.

4.2.1 Measurement evidence representation

The intermediate grid evidences are stored as a single number in the range $[0, 1]$, which describe the hypothesis evidence. The value describes the measurement belief that the hypothesis is true (static/dynamic/occupied or free).

To convert the evidence into occupancy probability, one can apply the linear scaling [Porębski, 2020]:

$$\begin{aligned} p_S(\text{occupied}) &= 0.5 \cdot (1 + \text{IG}_S) & p_{SUD}(\text{occupied}) &= 0.5 \cdot (1 + \text{IG}_{SUD}) \\ p_D(\text{occupied}) &= 0.5 \cdot (1 + \text{IG}_D) & p_F(\text{occupied}) &= 0.5 \cdot (1 - \text{IG}_F) \end{aligned} \quad (4.2.1)$$

Additionally, evidences might be used to directly construct BBAs for the Dempster-Shafer framework. Using the notation defined in (4.1.13) the BBAs definition follows:

$$\begin{aligned} m_S(\alpha_1) &= \text{IG}_S & m_F(\alpha_7) &= 1 - \text{IG}_S & \forall_{i=\{0,2,3,4,5,6\}} m_S(\alpha_i) &= 0 \\ m_D(\alpha_2) &= \text{IG}_D & m_D(\alpha_7) &= 1 - \text{IG}_D & \forall_{i=\{0,1,3,4,5,6\}} m_D(\alpha_i) &= 0 \\ m_F(\alpha_3) &= \text{IG}_F & m_F(\alpha_7) &= 1 - \text{IG}_F & \forall_{i=\{0,1,2,4,5,6\}} m_F(\alpha_i) &= 0 \\ m_{SUD}(\alpha_4) &= \text{IG}_{SUD} & m_{SUD}(\alpha_7) &= 1 - \text{IG}_{SUD} & \forall_{i=\{0,1,2,3,5,6\}} m_{SUD}(\alpha_i) &= 0 \end{aligned} \quad (4.2.2)$$

4.2.2 Bayesian framework fusion rules

The Bayesian probability still dominates the area of occupancy grid probabilistic frameworks. As the name follows, the most commonly used fusion rule in this approach is the Bayesian rule defined as:

$$p(G^{t+1}) = \frac{p(G^t) \cdot p(G|z^t)}{p(G^t)p(G|z^t) + (1 - p(G^t))(1 - p(G|z^t))} \quad (4.2.3)$$

where:

- $p(G^{t+1})$ – new, updated occupancy map probabilities (*a posteriori* grid),
- $p(G^t)$ – prior map of occupancy probabilities (*a priori* grid),
- $p(G|z^t)$ – set of probabilities of the grid being occupied given sensor measurements. This value can be derived from the intermediate grid evidences using equation (4.2.1).

This fusion rule is fully associative and commutative, and it can be used in any fusion architecture. Nonetheless, Bayesian fusion is unable to model sensor uncertainty and lack of sensor conflict resolution. However, this fusion rule has a major advantage for embedded automotive applications: it can be easily optimized to significantly reduce its computational complexity. To do that, the fusion rule must undergo a logit transformation:

$$l = \text{logit}(p) = \log\left(\frac{p}{1-p}\right) \quad p = \frac{e^l}{1+e^l} \quad (4.2.4)$$

Using this transformation, the Bayesian fusion equation takes form:

$$l(G^{t+1}) = l(G^t) + l(G|z^t) \quad (4.2.5)$$

This log-odds form of the Bayesian fusion rule requires only a single addition operation in order to fuse new evidence, therefore it does not require much computational resources to be developed in the automotive environment [Markiewicz and Porębski, 2020; Markiewicz et al., 2018].

There are plenty of different fusion rules applicable in the Bayesian framework, such as the aforementioned De Morgan and Maximum Policy (Equations (3.4.6) and (3.4.7)). Definition of other fusion rules can be found in [Markiewicz and Porębski, 2020].

4.2.3 Combination rules for the Dempster Shafer evidence framework

The Bayes and Dempster–Shafer approaches are both based on the concept of attaching weights to the postulated states of the system being measured. While Bayes applies a more "classical" meaning to these in terms of well-known ideas about probability, Dempster–Shafer theory assigns its mass to all the unions of the entities that comprise a system. This approach requires the definition of new rules of combination that operates on the Basic Belief Assignments.

The Shafer's model requires for all elementary hypotheses $\theta_1, \theta_2, \dots, \theta_n$ to be truly exhaustive and exclusive. Therefore, for the Shafer's model, the BBA of the null set is always zero ($m(\emptyset) = 0$). The exhaustive constraint is revoked in the Smets' model, and there the mass of the null set can be greater than zero ($m(\emptyset) \geq 0$).

This section covers two of the most common rules of combination applicable to the DST framework: Conjunctive rule, which operates on Smets' model and the Dempster rule of combination working in classical Shafer's model.

4.2.3.1 Conjunctive rule of combination

The conjunctive rule (known also as conjunctive consensus) for two BBAs m_1 and m_2 in the DST framework is defined $\forall A \in 2^\Theta$ by:

$$m_{\cap}(A) = (m_1 \odot m_2)(A) = \sum_{X \cap Y = A} m_1(X) m_2(Y) \quad \forall X, Y \in 2^\Theta \quad (4.2.6)$$

$m_{\cap}(\cdot)$ is not a proper belief assignment satisfying the Shafer's definition, since in most cases the sources do not totally agree (there exists partial and/or total conflicts between sources of evidence), so that $m_{\cap}(\emptyset) > 0$. Therefore, this fusion rule can be applied only to the Smets' model.

This fusion rule is commutative, associative, preserves the neutral impact of VBA, and can be defined for $k \geq 2$ sources of information:

$$m_{\cap}(A) = (m_1 \odot m_2 \odot \dots \odot m_k)(A) = \sum_{X_1 \cap X_2 \dots \cap X_k = A} \prod_{i=1}^k m_i(X_i) \quad \forall X_1, X_2, \dots, X_k \in 2^\Theta \quad (4.2.7)$$

The conjunctive rule for $k \geq 2$ sources can be also fused sequentially, meaning it can be used as a first fusion step for the hybrid fusion [Smarandache and Dezert, 2004]. When fusing four intermediate grids together with the equation of the intermediate grid fusion case, equation (4.2.7) takes form:

$$m_{\cap}(A) = (m_S \odot m_D \odot m_F \odot m_{SUD})(A) = (((m_S \odot m_D) \odot m_F) \odot m_{SUD})(A) \quad (4.2.8)$$

Therefore, each intermediate grid BBA can be fused sequentially using the same equation set.

4.2.3.2 Dempster rule of combination

The Dempster's rule of combination is the most widely used rule of combination so far in many expert systems based on belief functions since historically it was proposed in the seminal book of Shafer in [Shafer, 1976]. This rule, although presenting interesting advantages (mainly the commutativity, associativity and neutral impact of VBA) fails however to provide coherent results due to the normalization procedure it involves.

Dempster's rule is based on the conjunctive consensus, however, in order to meet with the Shafer's definition, it transfers the conflicting mass to non-empty sets proportionally to their resulting masses

$$m_{DS}(A) = \begin{cases} m_{\cap}(X) \cdot \frac{1}{1 - m_{\cap}(\emptyset)} & , \forall X \subseteq 2^\Theta \setminus \emptyset \\ 0 & , X \in \{\emptyset\} \end{cases} \quad (4.2.9)$$

As mentioned in [Smarandache and Dezert, 2005] Dempster's rule consists of two steps: first applies the conjunctive consensus rule and the second normalizes the masses using conflict measurement

$$k_{12} = \frac{1}{1 - m_{\cap}(\emptyset)} \quad (4.2.10)$$

Therefore, in the hybrid two-step fusion, the Dempster's rule can be applied as a second step coupled with conjunctive consensus.

4.2.4 Combination rules for the Dezert-Smarandache evidence framework

The DSMT evidence framework extends the DST by the introduction of additional possible propositions in the power set 2^{Θ} by creating hyper-power set D^{Θ} as presented in Section 4.1.3.

Smarandache and Dezert in [Smarandache and Dezert, 2004] defines two types of hyper power sets: free and hybrid. The free hyper-power set allows the existence of all possible unions and intersections of elemental hypotheses θ_i in the hyper-power set D^{Θ} . The hybrid power set enables the definition of constrains on the permitted subsets. It can be used to define DST power set as a variant of hybrid hyper-power set and use DSMT rules in combination with the Dempster's model.

4.2.4.1 Classic DSMT rule of combination

When the free DSMT model holds for the fusion problem under consideration, the classic DSMT rule of combination (DSMT in short) of two independent sources of evidences over the same frame Θ with belief functions corresponds to the conjunctive consensus of the sources. It is given by:

$$m_{DSm}(A) = \sum_{X \cap Y = A} m_1(X)m_2(Y) \quad \forall X, Y \in D^{\Theta} \quad (4.2.11)$$

This rule of combination is commutative and associative and can always be used for the fusion of sources involving fuzzy concepts. This rule can be directly and easily extended for the combination of $k \geq 2$ independent sources of evidence similarly as the conjunctive rule presented in (4.2.7):

$$m_{DSm}(A) = \sum_{X_1 \cap X_2 \dots \cap X_k = A} \prod_{i=1}^k m_i(X_i) \quad \forall X_1, X_2, \dots, X_k \in D^{\Theta} \quad (4.2.12)$$

4.2.4.2 Hybrid DSMT rule of combination

As aforementioned, the DSMT is a new theory that is in permanent dynamic evolution, which may lead to different kinds of inconsistencies in nomenclature. Daniel noticed that there are two different definitions of the hybrid DSMT rule of combination [Daniel, 2014]. In this thesis the first definition of the hybrid DSMT rule (DSMH in short) is assumed based on [Dezert, 2002; Smarandache and Dezert, 2004].

The DSMH is defined as:

$$m_{DSmH}(X) = \phi(X) [S_1(X) + S_2(X) + S_3(X)] \quad (4.2.13)$$

where $\phi(X)$ is a characteristic non-emptiness function of a set X , i.e., $\phi(X) = 1$ if $X \notin \emptyset$ and $\phi(X) = 0$ otherwise.

$S_1(A)$, $S_2(A)$ and $S_3(A)$ are defined for two sources (for n-ary versions see [Smarandache and Dezert, 2004]) as follows:

$$S_1(A) = m_{DSmC}(X) = \sum_{X \cap Y = A} m_1(X)m_2(Y) \quad \forall X, Y \in D^\Theta \quad (4.2.14)$$

$$S_2(A) = \sum_{\substack{[(u(X) \cup u(Y)) = A] \vee \\ [(u(X) \cup u(Y)) \in \emptyset \wedge A = \Theta]}} m_1(X)m_2(Y) \quad \forall X, Y \in \emptyset \quad (4.2.15)$$

$$S_3(A) = \sum_{X \cup Y = A, X \cap Y = \emptyset} m_1(X)m_2(Y) \quad \forall X, Y \in D^\Theta \quad (4.2.16)$$

where $u(X)$ is the union of all singletons θ_i that compose X and Y and the set Θ describes a fully ignorant/vacuous belief.

The formulas $S_1(A)$, $S_2(A)$ and $S_3(A)$ of the DS_mH combination can be explained as:

$S_1(A)$ – corresponds to the classic DS_m rule on the free DS_m model,

$S_2(A)$ – represents the mass of all relatively and absolutely empty sets in both the input BBA's, which arises due to non-existential constraints and which is transferred to the total or relative ignorance,

$S_3(A)$ – transfers the sum of masses of relatively and absolutely empty sets, which arise as conflicts of the input BBA's, to the non-empty union of input sets [Daniel, 2014].

Hybrid DS_m rule of combination is commutative but not associative. Therefore, it cannot be applied in a sequential fusion step in the occupancy grid algorithm. Nevertheless, it can be coupled with the DS_mC rule in the two-step fusion architecture as presented in [Smarandache and Dezert, 2004].

4.2.5 Visualization of the occupancy grids

Presentation of the occupancy grid is often an underestimated problem of data visualization. In automotive applications, grids are relatively small – usually a grid has around half a million of cells [Porębski et al., 2019]. A natural choice to present such data is to convert the grid into an image. On this scene, each pixel corresponds to the cell state. Conveniently, the time sequence of grids may be showcased as a video presenting how the occupancy evolves with changing environment [Andriamahefa, 2017; Thrun et al., 2005].

For a binary state or single-valued occupancy grid, the grayscale image is usually chosen for the occupancy probability or other single value representation.

Nevertheless, the human eye can be easily fooled by the gray scale images. Our perception of contrast depends on the overall image lightness, and some gray scale levels might appear different depending on their surroundings. An example of this so-called brightness contrast illusion is presented in Figure 4.2. The middle gray square appears darker in the bright environment and brighter in the dusky surrounding [Takahashi, 2017].

As a solution for that problem, the author proposes the usage of grid-structured pie charts, where each pie represents the cell state. This approach is not feasible for visualizing high areas of the environment, but might be used for displaying small grid areas, omitting brightness problems.

Moreover, as the grid quality is usually determined based on a visual assessment, a secondary presentation method can be useful to spot any grid representation issues even during the algorithm development.

In this thesis, the mapping of probabilities p into the color value is defined as an inverted gray scale:

$$\text{RGB} = \{1 - p, 1 - p, 1 - p\} \quad (4.2.17)$$

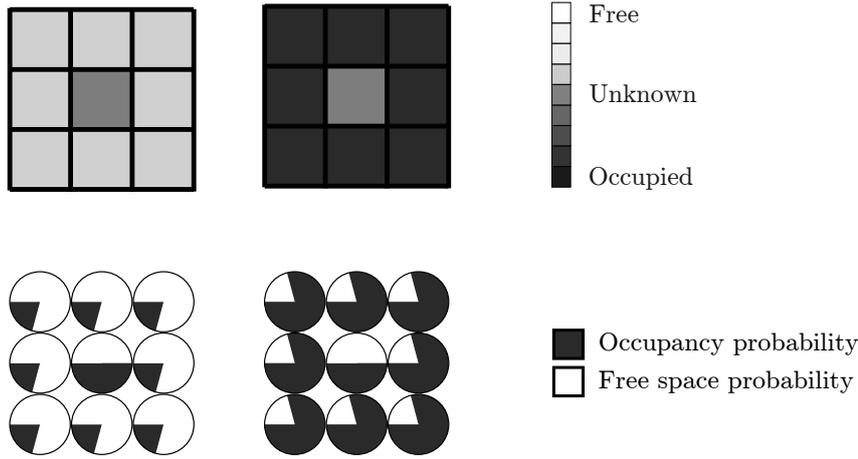


Figure 4.2. Brightness contrast illusion in the gray-scale occupancy grid and proposed grid structured pie charts as a grid visualization.

where RGB is a Red-Green-Blue triplet, where each value represents the intensity of the corresponding color, i.e., $RGB = \{1, 1, 1\}$ – white, $RGB = \{0, 0, 0\}$ – black, $RGB = \{0, 1, 1\}$ – cyan, etc.

An example of a grid mapped using (4.2.17) is presented in the Section 7.3 in Figure 7.8.

Visualization of the occupancy grids which are based on the three-element frame of discernment $\Theta = \{S, D, F\}$ can be done similarly by using colored map of cells. For the grid visualization Steyer et al. proposed linear colormap based on plausibility [Steyer et al., 2018]. In this thesis, the color mapping definition is modified such that the red color defines the dynamic belief, blue color represents the static and the green coefficient mean free space.

$$RGB = \left\{ 1 - \sum_{\substack{X \in 2^\Theta, \\ X \cap \{D\} = \emptyset}} m(X), 1 - \sum_{\substack{X \in 2^\Theta, \\ X \cap \{F\} = \emptyset}} m(X), 1 - \sum_{\substack{X \in 2^\Theta, \\ X \cap \{S\} = \emptyset}} m(X) \right\} \quad (4.2.18)$$

$$RGB = \{pl(\{D\}), pl(\{F\}), pl(\{S\})\} \quad (4.2.19)$$

An example of a grid mapped using (4.2.19) is presented in the Section 4.2.6 in Figure 4.4 and later in the Chapter 7.

The Dempster-Shafer BBA cell state can be also visualized as a triangular graph plot as presented in [Jøsang et al., 2005]. Nevertheless, in this work, the pie chart grid is used also in the Bayesian cell visualization. The color of each belief mass is determined using (4.2.19) to keep consistent colormaps between different pictures. Legend for pie charts is presented in Figure 4.3.

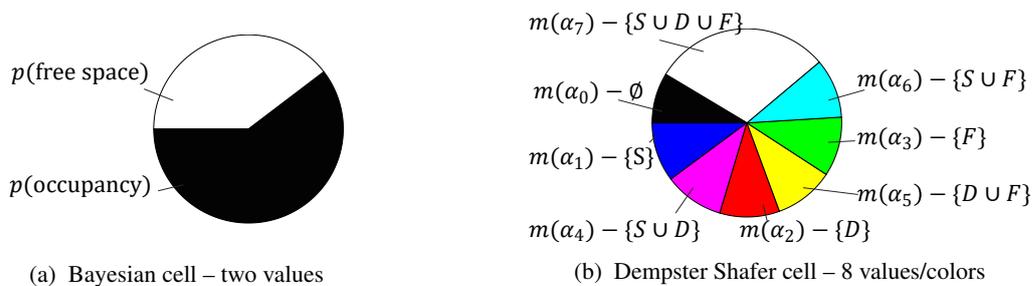


Figure 4.3. Legend for the pie chart cell state visualization.

4.2.6 Quantitative evaluation of contradictory evidences

Occupancy grid visualizations presented in Section 4.2.5 are able to display the occupancy grid state and qualitatively determine the presence of sensor conflict. Nevertheless, for an automated occupancy grid assessment, the sensor conflict should be measured quantitatively. Depending on the application and researcher preferences, different metrics are used to measure evidence disagreement.

Commonly, sensor conflict metrics are used to measure the distance between two Basic Belief Assignments to measure the disagreement between them. An example of such metric can be Dempster normalization factor k_{12} [Shafer, 1976], *diffBetP* metric [Jousselme et al., 2001] or hidden sensor conflict [Daniel and Kratochvíl, 2020].

In the occupancy grid conflict assessment, however, self-conflict measurement is needed. For simple quantitative grid conflict measurement, three different metrics can be used: entropy, specificity and auto-conflict. The values of these metrics for the example grid from the previous section are visualized in Figure 4.4.

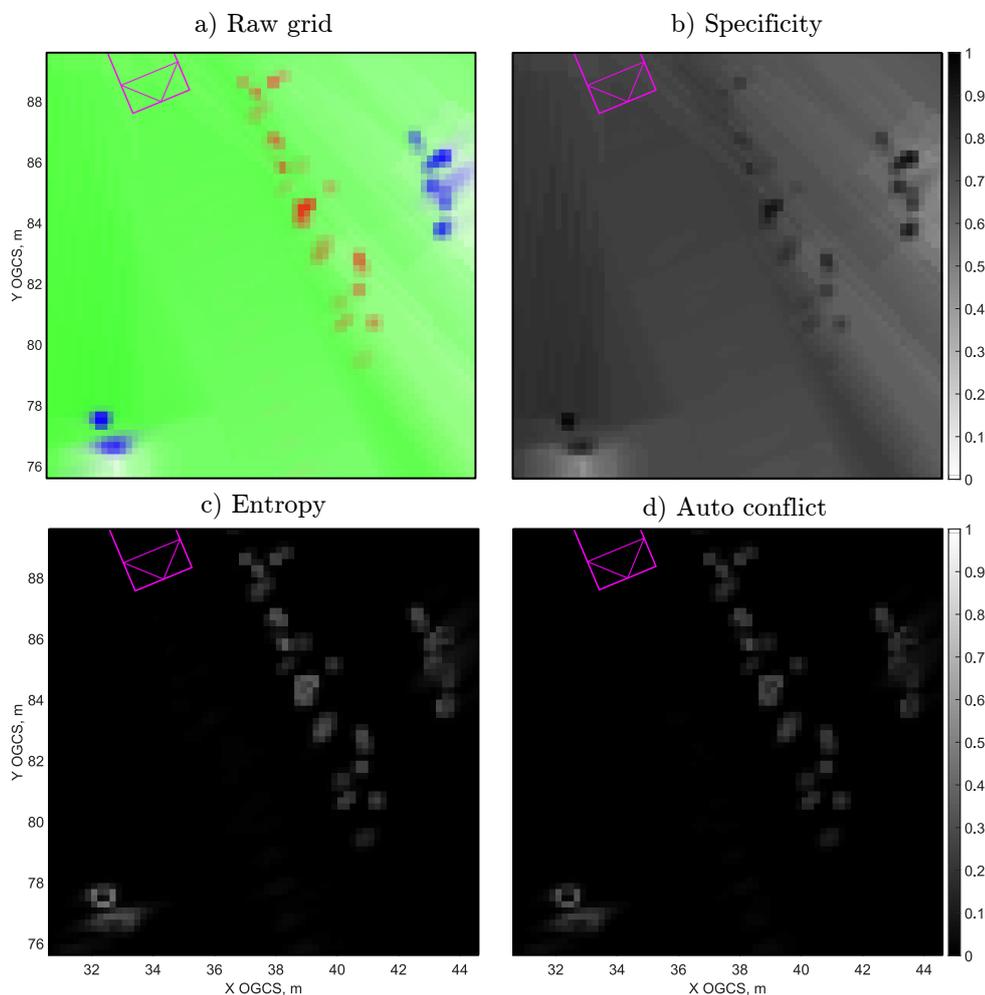


Figure 4.4. Presentation of quantitative conflict metrics for the example Dempster-Shafer’s grid snapshot. The gray-scale for entropy and auto conflict is changed in a way that white areas represent more inconsistency in the cell evidence. Most of the conflicts are visible in the dynamic residual area (red area in the middle of the top left grid image and as an envelope of stationary object (blue area in the bottom of the top left grid image).

4.2.6.1 Entropy and specificity

In [Yager, 2008] Yager introduced the concepts of entropy and specificity in the framework of Dempster-Shafer's theory. These parameters are complementary and can be used to indicate the quality of evidence.

A high value of entropy can indicate inconsistency in the distribution of mass beliefs. The entropy of a cell can be calculated as follows:

$$E_c = - \sum_{X \subseteq \mathcal{P}^\Theta} m(X) \cdot \ln(pl(X)) \quad (4.2.20)$$

where $pl(X)$ is the plausibility of X .

The specificity parameter provides an indication of how the belief mass is dispersed. Therefore, it has a higher value if the mass distribution is less doubtful. The specificity value of a cell can be calculated as follows:

$$S_c = \sum_{X \subseteq \mathcal{P}^\Theta \setminus \emptyset} \frac{m(X)}{|X|} \quad (4.2.21)$$

Considering the two parameters, we can conclude that: the lower the entropy, the more consistent is the evidence; and the higher the specificity, the less diverse it is. Therefore, for better certainty we need low entropy and high specificity [Valente et al., 2018].

The values of these evidence quality measurements are visualized using gray-scale images in Figure 4.4. The specificity is presented using the same inverted gray scale as the Bayesian occupancy grid (4.2.17). For the entropy, however, the normal gray scale was used to amplify the contrast in the output image.

4.2.6.2 Auto conflict

Another method of assessing sensor conflict is the auto-conflict measurement defined in [Martin et al., 2008; Osswald and Martin, 2006]. It is computed by applying the conjunctive rule of combination to the cell state itself:

$$a(m) = m_{\cap}(\emptyset) = (m \circledast m)(\emptyset) \quad (4.2.22)$$

This definition can be extended to higher degrees of auto conflict, but in most cases the first-order auto conflict is enough to detect and measure sensor conflicts.

The auto-conflict behaves similarly to the entropy measure, the area with higher value is characterized with more inconsistencies in evidences as visualized in Figure 4.4.c.

4.3 Occupancy grid fusion

Information fusion is the core of automotive occupancy grid processing. Every valid sensor detection is firstly a subject for the inverse sensor modeling, then it is fused into the occupancy grid environment representation. The probabilistic fusion has to retain reliability robustness even during the processing of uncertain or even false sensor measurements.

Probability representation frameworks, as well as fusion architectures and rules of combination, are already presented previously in the current chapter. This section focuses on the application and testing of three fusion architectures, which put to use five different rules of combination.

4.3.1 Evaluated fusion architectures

This thesis investigates the topic of multi-sensor automotive occupancy grid, which should be able to operate on heterogeneous sensor sources. As explained in section 4.2, the best fusion setup for that set of sensors is the two-step hybrid architecture. Based on the capabilities of the presented sensor fusion rules, three fusion architectures models were selected to highlight the character of every probabilistic representation framework. Developed and implemented hybrid fusion architectures are presented visually in Figure 4.5.

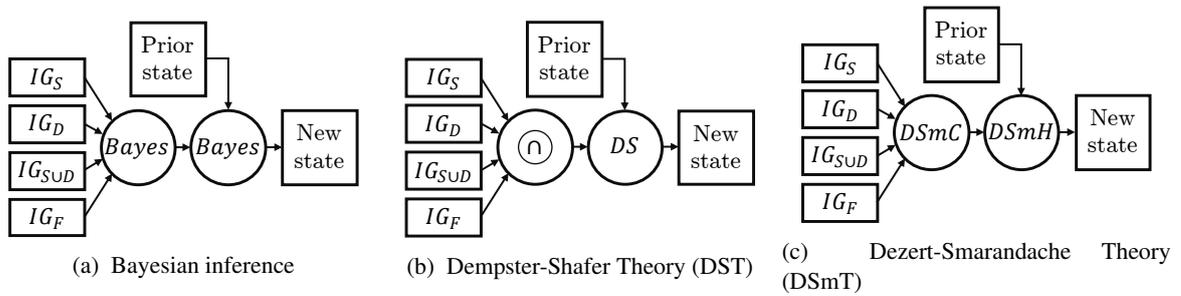


Figure 4.5. Developed hybrid fusion architectures tailored for usage within different fusion frameworks. Symbol \odot represents conjunctive fusion rule, DS – Dempster rule of combination, $DSmC$ and $DSmH$ – classical and hybrid Dezert-Smarandache rules accordingly.

Recent directives for automated driving such as [Aptiv et al., 2019; International Organization for Standardization, 2021] suggest that some value of uncertainty should be measured within the perception system. Bayesian inference framework is not able to model sensor uncertainty, but the DST and DSmT frameworks can. For these two cases, the hybrid fusion architecture accumulates sensor conflicts in the first stage. Redistribution of the internal sensor conflict is modeled in the second step, together with the application of the Dempster’s model constraints.

4.3.1.1 Bayesian inference

Figure 4.5(a) depict developed Bayesian inference architecture. In this setup, both fusion steps are managed using the same Bayesian fusion rule:

$$p_{posteriori} = \frac{p_{prior} \cdot p_{measurement}}{p_{prior} \cdot p_{measurement} + (1 - p_{prior})(1 - p_{measurement})} \quad (4.3.1)$$

where:

- $p_{posteriori}$ – *a posteriori* cell occupancy probability (updated cell state),
- p_{prior} – *prior* occupancy probability of a cell,
- $p_{measurement}$ – measurement occupancy probability computed from intermediate grid evidences using equation (4.2.1).

Equation (4.3.1) is used for sequential fusion of the intermediate grid evidences in the first step and then may be used to fuse the prior cell probability into the temporal result. In front of application of the Bayesian fusion rule, the conversion of evidences into Bayesian probabilities has to be computed using equation (4.2.1).

For an embedded algorithm's implementation, the logit transformation can be used, but it requires that all inverse sensor models deliver occupancy probabilities in the same domain of log odds. This thesis considers only the probability representation, because it ensures that the same inverse sensor model can be reused for all fusion frameworks.

As aforementioned, the Bayesian inference method is a simple and efficient way to represent the occupancy grid. This solution can be applied for systems which require low computational complexity without uncertainty modeling.

4.3.1.2 Dempster Shafer Theory

Representation of probabilities as beliefs and masses offers new possibilities for uncertainty modeling and sensor conflict redistribution. This architecture, visually presented in Figure 4.5(b), makes use of the two most commonly used fusion rules: conjunctive and Dempster combination. Nevertheless, the proposed setup is designed in a way that the Dempster combination rule can be swapped with other fusion rules if only they are able to properly redistribute the sensor conflict.

For the DST and DSMT occupancy grid in this thesis static-dynamic-free frame of discernment is analyzed ($\Theta = \{S, D, F\}$), which beliefs are described in (4.1.13). In order to present the fusion examples clearly the following notation is assumed:

$$m_1^i = m_1(\alpha_i) \qquad m_2^i = m_2(\alpha_i) \qquad (4.3.2)$$

where the definition of each subset α_i is specified also in 4.1.2.

Conjunctive rule of combination

The conjunctive rule of combination is described in Section 4.2.3.1. This rule holds associative and commutative properties; therefore, the intermediate grid fusion can be done sequentially without loss of generality.

Assuming that the initial state ($m_{1(\text{init})}$) is a VBA:

$$\forall_{i \in \{0, \dots, 6\}} m_{1(\text{init})}^i = 0 \qquad m_{1(\text{init})}^7 = 1 \qquad (4.3.3)$$

And all IG BBAs defined as (4.2.2) are fused as m_2 with constrains:

$$\forall_{i \in \{1, 2, 3, 4, 7\}} m_2^i \geq 0 \qquad \forall_{i \in \{0, 5, 6\}} m_2^i = 0 \qquad (4.3.4)$$

The simplified conjunctive rule equation takes form:

$$\begin{aligned} m_{\cap}(\alpha_0) &= m_1^0 + m_1^1(m_2^2 + m_2^3) + m_1^2(m_2^1 + m_2^3) + m_1^3(m_2^1 + m_2^2 + m_2^4) + m_1^4 m_2^3 \\ m_{\cap}(\alpha_1) &= m_1^1(m_2^1 + m_2^4 + m_2^7) + m_2^1(m_1^4 + m_1^7) \\ m_{\cap}(\alpha_2) &= m_1^2(m_2^2 + m_2^4 + m_2^7) + m_2^2(m_1^4 + m_1^7) \\ m_{\cap}(\alpha_3) &= m_1^3(m_2^3 + m_2^7) + m_1^7 m_2^3 \\ m_{\cap}(\alpha_4) &= m_1^4(m_2^4 + m_2^7) + m_1^7 m_2^4 \\ m_{\cap}(\alpha_5) &= 0 \\ m_{\cap}(\alpha_6) &= 0 \\ m_{\cap}(\alpha_7) &= m_1^7 m_2^7 \end{aligned} \qquad (4.3.5)$$

This fusion rule operates on the Smets' model, and it accumulates sensor conflict measurements in the form of mass $m_{\cap}(\alpha_0)$. Therefore, the first step of the Dempster fusion architecture has to be normalized to valid Dempster's model of evidence.

For the direct occupancy grid implementation, the equation (4.3.5) can be simplified for the sequential fusion of the intermediate grid due to the specific structure of their BBAs (4.2.2). Detailed derivation of the equation (4.3.5) is presented in the Appendix A.

Dempster rule of combination

Second stage of the DST fusion architecture is realized using Dempster rule of combination defined in Section 4.2.3.2. Similarly, to the conjunctive rule, it also holds associate and commutative properties.

For the three-element frame of discernment, the fusion rule takes form:

$$k_{12} = \frac{1}{1 - m_{\cap}(\alpha_0)} \quad (4.3.6)$$

$$m_{DS}(A) = \begin{cases} m_{\cap}(\alpha_i) \cdot k_{12} & , A \in \{\alpha_1, \dots, \alpha_7\} \\ 0 & , A = \alpha_0 \end{cases} \quad (4.3.7)$$

which redistributes the sensor conflict $m_{\cap}(\alpha_0)$ uniformly to all other non-zero masses.

Uniform conflict redistribution assumes that the sensor disagreement originates equally from different evidences. This approach is a controversial topic for some researchers [Dezert et al., 2012; Khan and Anwar, 2019; Odgerel and Lee, 2016; Zadeh, 1996] who argue its correctness on various examples. In order to resolve controversies related to the Dempster rule, in presented fusion setup, it can be replaced with other weighted operator (WO) or proportional conflict redistribution (PCR) rules as described in [Smarandache and Dezert, 2005].

The Dempster combination rule equation (4.3.7) cannot be computed for completely conflicting sources, where $m_{\cap}(\alpha_0) = 1$. Nevertheless, in the real automotive application, this condition has to be handled in the fusion algorithm. For the implementations presented in this thesis, in case of full conflict, the output cell is reverted to the initial vacuous hypothesis state (VBA):

$$\text{if } m_{\cap}(\alpha_0) = 1 \quad \text{then } m_{DS}(A) = m_v(A) = \begin{cases} 0 & , A \in \{\alpha_0, \dots, \alpha_6\} \\ 1 & , A = \alpha_7 \end{cases} \quad (4.3.8)$$

Based on the characteristics of the occupancy grid sensor inputs, the masses corresponding to propositions $\alpha_5 = \{S \cup F\}$ and $\alpha_6 = \{D \cup F\}$ will be always equal to zero if the Dempster rule of combination would be applied in the second fusion step. This means that some sensor conflicts, e.g., between stationary and free evidences cannot be effectively solved using the DST framework with the default fusion rules.

4.3.1.3 Dezert-Smarandache Theory

Desert-Smarandache model of evidences provides a more complete description of the frame of discernment. Elements of the hyper-power set define not only unions, but also intersections of elemental hypotheses. Moreover, DSMT offers methods for the reduction of the full hyper-power proposition set of the free DSMT model into the Dempster's frame of discernment, defined as a specific hybrid DSMT model.

The hybrid DSMT two-step fusion is an analog to the Dempster architecture, utilizing the classic DSMT rule (DSMT) as a first step and the hybrid DSMT (DSMTH) rule as a second as presented in Figure 4.5(c).

Classic DSm rule of combination (DSmC)

The classic DSm rule of combination (DSmC) is an equivalent of the conjunctive Dempster rule of combination. It is associative and commutative but can be applied only to the free DSm model. As described in Section 4.1.3 the intersections of elemental hypotheses do not have any physical explanation for the occupancy grid states. Nevertheless, using the DSmC rule, these masses can be exploited as a detailed sensor conflict representation. For example, the belief of subset $\alpha_{16} = \{S \cap F\}$ can be explained as a conflict between static and free space measurements. Therefore, the free DSm model of evidences is better suited for managing sensor conflict, as it can represent conflicts with 13 beliefs instead of a single mass of empty set in the DST.

The detailed conflict representation can be also a downside of the DSmC rule, as it increases the complexity and number of computations required for this fusion. Nevertheless, sensor measurement evidences presented in the form of intermediate grids have specific and simple BBA definitions (equation (4.2.2)), which can simplify the DSmC fusion. For clarity, this section provides only the final, reduced equations of the rules of combination. Derivation procedure of this equation can be found in Appendix A.

First step fusion using DSmC rule starts with the assumption that the initial state $m_{1(\text{init})}$ is a fully ignorant state – a Vacuous Belief (VBA):

$$\forall_{i \in \{0, \dots, 18\} \setminus \{7\}} m_{1(\text{init})}^i = 0 \quad m_{1(\text{init})}^7 = 1 \quad (4.3.9)$$

Furthermore, the intermediate grid BBAs (4.2.2) are constrained by the DSmT framework as:

$$\forall_{i \in \{1, 2, 3, 4, 7\}} m_2^i \geq 0 \quad \forall_{i \in \{0, 5, 6, 8, 9, \dots, 18\}} m_2^i = 0 \quad (4.3.10)$$

Based on these two assumptions, the DSmC fusion rule takes form:

$$\begin{aligned} m_{DSmC}(\alpha_0) &= 0 \\ m_{DSmC}(\alpha_1) &= m_1^1 m_2^1 + m_1^1 m_2^4 + m_1^4 m_2^1 + m_1^1 m_2^7 + m_1^7 m_2^1 \\ m_{DSmC}(\alpha_2) &= m_1^2 m_2^2 + m_1^2 m_2^4 + m_1^4 m_2^2 + m_1^2 m_2^7 + m_1^7 m_2^2 \\ m_{DSmC}(\alpha_3) &= m_1^3 m_2^3 + m_1^3 m_2^7 + m_1^7 m_2^3 \\ m_{DSmC}(\alpha_4) &= m_1^4 m_2^4 + m_1^4 m_2^7 + m_1^7 m_2^4 \\ m_{DSmC}(\alpha_5) &= 0 \\ m_{DSmC}(\alpha_6) &= 0 \\ m_{DSmC}(\alpha_7) &= m_1^7 m_2^7 \\ m_{DSmC}(\alpha_8) &= 0 \\ m_{DSmC}(\alpha_9) &= 0 \\ m_{DSmC}(\alpha_{10}) &= 0 \\ m_{DSmC}(\alpha_{11}) &= m_1^{11} \cdot 1 + m_1^{15} m_2^3 + m_1^{16} m_2^2 + m_1^{17} m_2^1 \\ m_{DSmC}(\alpha_{12}) &= m_1^{12} m_2^3 + m_1^{12} m_2^4 + m_1^{12} m_2^7 + m_1^3 m_2^4 + m_1^4 m_2^3 \\ m_{DSmC}(\alpha_{13}) &= 0 \\ m_{DSmC}(\alpha_{14}) &= 0 \\ m_{DSmC}(\alpha_{15}) &= m_1^{15} (m_2^1 + m_2^2 + m_2^4 + m_2^7) + m_1^1 m_2^2 + m_1^2 m_2^1 \\ m_{DSmC}(\alpha_{16}) &= m_1^{16} (m_2^1 + m_2^3 + m_2^4 + m_2^7) + m_1^1 m_2^3 + m_1^3 m_2^1 + m_1^{12} m_2^1 \\ m_{DSmC}(\alpha_{17}) &= m_1^{17} (m_2^1 + m_2^3 + m_2^4 + m_2^7) + m_1^2 m_2^3 + m_1^3 m_2^2 + m_1^{12} m_2^2 \\ m_{DSmC}(\alpha_{18}) &= 0 \end{aligned} \quad (4.3.11)$$

Same as the conjunctive rule, equation (4.3.11) can be further simplified for each particular IG BBA. Nevertheless, using evidences in the form of four intermediate grids, the DS_m rule assigns non-zero beliefs to only 10 propositions out of 19 possible subsets of hypotheses.

Hybrid DS_m rule of combination (DS_mH)

The hybrid DS_m rule (DS_mH), similarly to the Dempster fusion, aims to distribute sensor conflict into the propositions supported by the hybrid DS_m model. The hybrid DS_m rule can differentiate between various types of conflicts and transfer the conflicting mass more accurately than the Dempster rule. The DS_mH rule does that by transferring conflict to the partial total ignorance upon the principle that between two conflicting hypotheses, one is right [Smarandache and Dezert, 2005]. Nevertheless, if this assumption is not enough, this particular fusion rule might be as well replaced by other combination rules i.e., proportional conflict redistribution or minC rule.

The DS_mH fusion step is performed only once to fuse together the output of DS_mC combination (m_1) with the prior cell state (m_2). Both input BBAs have some constrains, defined as:

$$\forall_{i \in \{1,2,3,4,7,11,12,15,16,17\}} m_1^i \geq 0 \quad \forall_{i \in \{0,5,6,8,9,10,13,14,18\}} m_1^i = 0 \quad (4.3.12)$$

$$\forall_{i \in \{1, \dots, 7\}} m_2^i \geq 0 \quad \forall_{i \in \{0,8, \dots, 18\}} m_2^i = 0 \quad (4.3.13)$$

The hybrid DS_m rule of combination for the aforementioned BBAs can be simplified into the following set of equations:

$$\begin{aligned} m_1^z &= m_1^7 + m_1^{11} + m_1^{12} + m_1^{15} + m_1^{16} + m_1^{17} \\ m_{DSmH}(\alpha_1) &= m_1^z m_2^1 + m_1^1 (m_2^1 + m_2^4 + m_2^5 + m_2^7) + m_1^4 m_2^1 + m_1^4 m_2^5 \\ m_{DSmH}(\alpha_2) &= m_1^z m_2^2 + m_1^2 (m_2^2 + m_2^4 + m_2^6 + m_2^7) + m_1^4 m_2^2 + m_1^4 m_2^6 \\ m_{DSmH}(\alpha_3) &= m_1^z m_2^3 + m_1^3 (m_2^3 + m_2^5 + m_2^6 + m_2^7) \\ m_{DSmH}(\alpha_4) &= m_1^z m_2^4 + m_1^4 m_2^4 + m_1^1 m_2^2 + m_1^2 m_2^1 + m_1^4 m_2^7 \\ m_{DSmH}(\alpha_5) &= m_1^z m_2^5 + m_1^1 m_2^3 + m_1^3 m_2^1 \\ m_{DSmH}(\alpha_6) &= m_1^z m_2^6 + m_1^2 m_2^3 + m_1^3 m_2^2 \\ m_{DSmH}(\alpha_7) &= m_1^z m_2^7 + m_1^1 m_2^6 + m_1^6 m_2^1 + m_1^2 m_2^5 + m_1^5 m_2^2 + m_1^3 m_2^4 + m_1^4 m_2^3 \end{aligned} \quad (4.3.14)$$

Detailed derivation of the equation (4.3.14) is presented in the Appendix A.

4.3.2 Fusion rules comparison

To compare the three aforementioned two-step fusion architectures (Figure 4.1) an example of a single cell evidences fusion is presented in this section.

The example consists of three consecutive fusion iterations. First iteration describes the initialization phase of the occupancy grid. The second takes the output of the first iteration and presents a common situation when fusion improves the state estimate in the presence of small internal sensor conflict. The third iteration of this example takes the output of the second iteration and fuses it with highly conflicting input data to depict multi-sensor conflict redistribution differences between fusion rules.

Cell values for both steps of each fusion iteration are presented in the form of pie charts in Figures 4.6, 4.7 and 4.8 and as numerical values in Tables 4.2, 4.3 and 4.4.

Input evidences

The intermediate grid input values for each fusion iteration are presented in Table 4.1. Input values were selected arbitrarily in order to present clear differences between different fusion methods.

In the first step, only the static or dynamic evidence is fused into the initial cell state. Second input data contains a partially free and static cell state, which corresponds to a meager instant sensor conflict. The third measurement delivers strong evidence about the dynamic cell state with some chance of free area. Third fusion iteration is an example of multi-sensor conflict, which is described in Section 3.5.

Table 4.1. Intermediate grid input values for each fusion iteration.

Intermediate grid evidence	First iteration	Second iteration	Third iteration
IG_S	0	0.1	0
IG_D	0	0	0.2
IG_F	0	0.2	0.4
IG_{SUD}	0.3	0	0

Visualization of fusion iterations

Three consecutive fusion iterations present how each grid framework handles different sensor conflicts. Each iteration can be described as:

First Iteration – Initialization of the cell state – Figure 4.6

Second Iteration – Instant sensor conflict – Figure 4.7

Third Iteration – Multi sensor conflict – Figure 4.8

As presented in Figure 4.6 the initial state for the Bayesian occupancy grid is $p = 0.5$, while for the DS and DS_m frameworks it is the full ignorance state (VBA). Presented fusion architectures hold the property of neutral impact of $p = 0.5$ and VBA, therefore, the results of the first input measurements are an exact copy of the input data. Result of the Bayesian fusion is obtained using normalization (4.2.1), which converts intermediate grid evidences into probability.

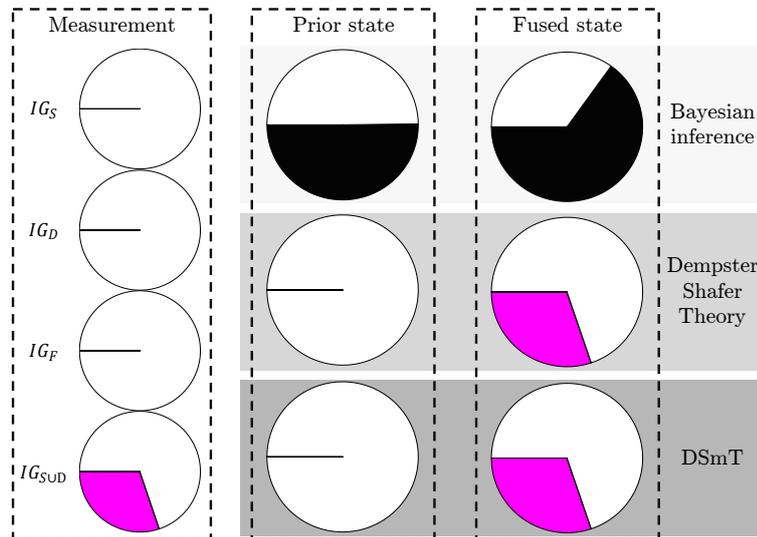


Figure 4.6. Graphical visualization of fusion example – Iteration 1: Fusion into initial cell state.

Second fusion iteration presented in Figure 4.7 fuses 2^{nd} intermediate grid evidences into the first fusion output. In this example, the outputs from DST and DS_mT architectures are visually similar. Small differences in values are still present and are discussed in the next section.

The result of the third fusion iteration (Figure 4.8) varies significantly depending on the fusion architecture. Dempster framework is able to redistribute conflict only to already non-zero masses, while

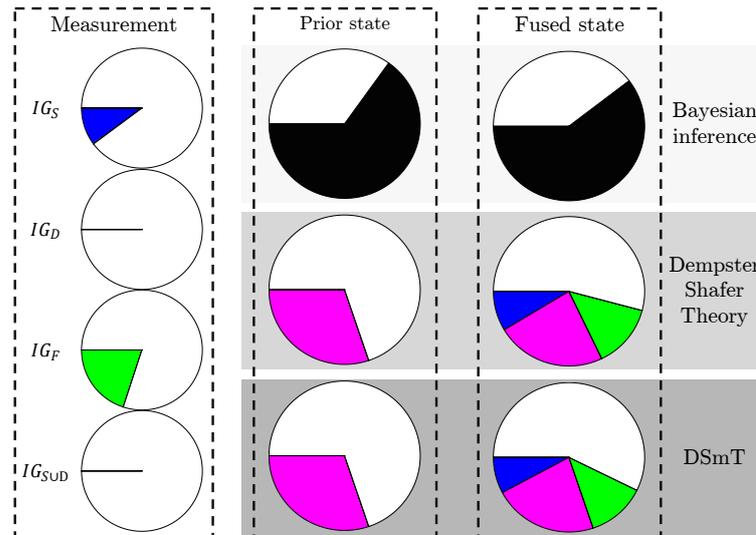


Figure 4.7. Graphical visualization of fusion example – Iteration 2: Instant sensor conflict.

the DSMT framework can affect all valid subsets of propositions. In this example, a mass for subset $\alpha_6 = \{D \cup F\}$ is generated and is clearly visible as a yellow slice in the DSMT fusion output.

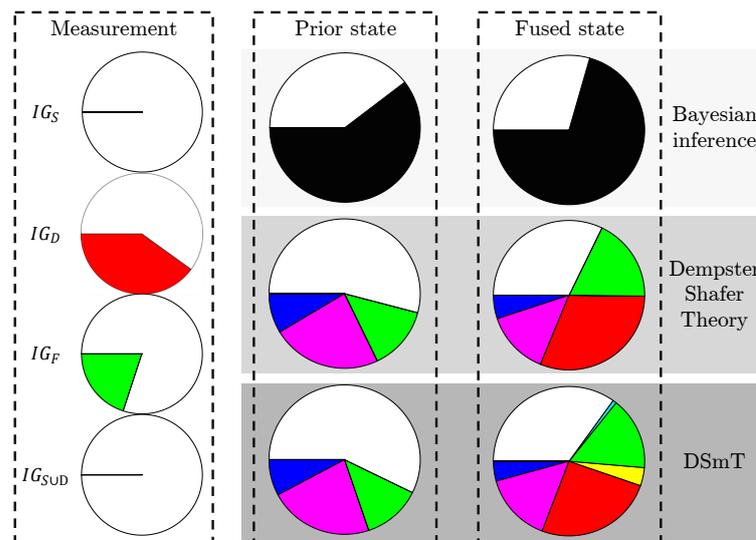


Figure 4.8. Graphical visualization of fusion example – Iteration 2: Multi sensor conflict.

Cell state values analysis

Graphical visualization of the fusion presents some insight into the different architectures, however, to complete the picture, the exact values of each cell are presented in Tables 4.2, 4.3 and 4.4. Additionally, the fusion results can be analyzed for the quantitative values of sensor conflict in Table 4.5.

Bayesian cell holds only a single occupancy probability value (Table 4.2). This straightforward representation loses information about the type of occupancy and sensor conflicts.

In the Dempster-Shafer theory, the power set for the three-elements frame of discernment consists of 8 propositions. Although due to the structure of the input evidences, the masses corresponding to propositions $\alpha_5 = \{S \cup F\}$ and $\alpha_6 = \{D \cup F\}$ will be always empty in the DST approach, which is highlighted in table as 'N/A' value. The mass of an empty set ($\alpha_0 = \{\emptyset\}$) is zero by definition of the Dempster model, but it can be valid using Smets' model in the first step of fusion. As presented in

Table 4.2. Cell probabilities for the Bayesian fusion example

Cell probability	Initial probability	First iteration		Second iteration		Third iteration	
		1 st step	2 nd step	1 st step	2 nd step	1 st step	2 nd step
p	0.5	0.65	0.65	0.45	0.6	0.61	0.7

Table 4.3 four intermediate grid inputs are converted into 6 possible beliefs in the first fusion step, and then normalized into valid 5 states using Dempster rule of combination in the 2nd step.

Table 4.3. Mass values for the DST fusion example. 1st step – conjunctive rule, 2nd step – Dempster rule. 'N/A' refers to propositions equal to 0 by the definition of fusion rule or model.

Cell proposition	Initial masses	First iteration		Second iteration		Third iteration	
		1 st step	2 nd step	1 st step	2 nd step	1 st step	2 nd step
$\alpha_0 = \{\emptyset\}$	N/A	0	N/A	0.02	N/A	0.08	N/A
$\alpha_1 = \{S\}$	0	0	0	0.08	0.09	0	0.05
$\alpha_2 = \{D\}$	0	0	0	0	0	0.18	0.31
$\alpha_3 = \{F\}$	0	0	0	0.18	0.14	0.12	0.18
$\alpha_4 = \{S \cup D\}$	0	0.3	0.3	0	0.23	0	0.14
$\alpha_5 = \{S \cup F\}$	N/A	N/A	N/A	N/A	N/A	N/A	N/A
$\alpha_6 = \{D \cup F\}$	N/A	N/A	N/A	N/A	N/A	N/A	N/A
$\alpha_7 = \{S \cup D \cup F\}$	1	0.7	0.7	0.72	0.54	0.48	0.32

The Dezert-Smarandache fusion architecture extends the capabilities of the DST. The conflicting information can be presented as 5 propositions $\{\alpha_{11}, \alpha_{12}, \alpha_{15}, \alpha_{16}, \alpha_{17}\}$ instead on a single α_0 in Dempster framework. These additional information layer enables more accurate conflict redistribution in the 2nd step and the DSmH fusion rule can properly model all propositions of the Dempster frame including the α_5 and α_6 .

Table 4.4. Mass values for the DSMT fusion example. 1st step – DSMT, 2nd step – DSMH. 'N/A' refers to propositions equal to 0 by the definition of fusion rule or model. Propositions beliefs, which are always zero regardless of the input data, has been omitted.

Cell proposition	Initial masses	First iteration		Second iteration		Third iteration	
		1 st step	2 nd step	1 st step	2 nd step	1 st step	2 nd step
$\alpha_1 = \{S\}$	0	0	0	0.08	0.08	0	0.04
$\alpha_2 = \{D\}$	0	0	0	0	0	0.32	0.25
$\alpha_3 = \{F\}$	0	0	0	0.18	0.13	0.12	0.15
$\alpha_4 = \{S \cup D\}$	0	0.3	0.3	0	0.22	0	0.15
$\alpha_5 = \{S \cup F\}$	0	N/A	0	N/A	0	N/A	0.01
$\alpha_6 = \{D \cup F\}$	0	N/A	0	N/A	0	N/A	0.04
$\alpha_7 = \{S \cup D \cup F\}$	1	0.7	0.7	0.72	0.57	0.48	0.35
$\alpha_{11} = \{S \cap D \cap F\}$	N/A	0	N/A	0	N/A	0	N/A
$\alpha_{12} = \{(S \cup D) \cap F\}$	N/A	0	N/A	0	N/A	0	N/A
$\alpha_{15} = \{S \cap D\}$	N/A	0	N/A	0	N/A	0	N/A
$\alpha_{16} = \{S \cap F\}$	N/A	0	N/A	0.02	N/A	0	N/A
$\alpha_{17} = \{D \cap F\}$	N/A	0	N/A	0	N/A	0.08	N/A

Quantitative evaluation of fusion example

Major drawback of the Bayesian fusion is that it is not able to describe neither conflicts nor uncertainty of the measured cell state. Therefore, comparison between the Bayesian and other two fusion architectures can be performed only on the level of occupancy probability measurement. For that purpose, a pignistic transformation is used, which for the Dempster model of probabilistic state takes form:

$$\text{betP}(\alpha_4 = \{S \cup D\} = \{O\}) = m(\alpha_1) + m(\alpha_2) + \frac{1}{2}m(\alpha_5) + \frac{1}{2}m(\alpha_6) + \frac{2}{3}m(\alpha_7) \quad (4.3.15)$$

The values for the occupancy probability values obtained using equation (4.3.15) are presented in Table 4.5. For the first and second iteration, the Bayesian and Dempster frame shows different occupancy value estimates. In the third iteration, however, the pignistic occupancy probability is almost the same for all fusion rules. Generally, when the overall cell uncertainty is high (i.e., the ignorance mass $m(\alpha_7)$ is high), the occupancy probability estimates for Bayesian and Dempster fusion may differ, but as more information is collected, the occupancy probability and pignistic transformation tend to the same value.

Comparison between DST and DSMT fusion can be measured using Entropy, Specificity and Auto conflict values, which are displayed in the Table 4.5. For the first iteration, DST and DSMT frameworks are identical, because both of them have the same definition of full ignorance state (VBA). The internal sensor conflict in the second iteration is solved similarly in both architectures, but in the DSMT the entropy and auto conflict values are slightly lower. The first step of the DSMT correctly identifies the meager conflict between static and free evidences, which affects the result by 0.01 for each evidence.

Difference between DST and DSMT frameworks is clearly present in the third fusion iteration. Entropy and auto conflict measurements are significantly lower for the DSMT fusion, meaning the result is less conflicted there. Specificity of the third iteration's output is a bit higher for the Dempster frame, which suggests that the Dempster rule is able to extract more information from the evaluated input data. Nevertheless, the fusion example presents highly conflicting information, therefore from the practical point of view, it might be better to extract less, but more precise information than overfit the Basic Belief Assignment.

Table 4.5. Quantitative measurement of probability and conflict in different fusion frameworks

Value		First iteration	Second iteration	Third iteration
Pignistic occupancy probability	Bayesian	0.65	0.60	0.70
	DS	0.77	0.68	0.71
	DSmT	0.77	0.68	0.70
Entropy	DS	0	0.10	0.27
	DSmT	0	0.09	0.20
Specificity	DS	0.38	0.52	0.71
	DSmT	0.38	0.51	0.67
Auto conflict	DS	0	0.09	0.21
	DSmT	0	0.08	0.17

Summary of the example analysis

From the three investigated fusion architectures, the Bayesian approach is the least complex one, providing no more information than the occupancy probability. Dempster and Dezert-Smarandache fusion architectures deliver a similar occupancy estimate as the Bayesian fusion when there is plenty of evidences and information delivered. In the case of sparse data, their uncertainty modeling capabilities enable the resolution of both internal and multi-sensor conflicts. The DSmT framework can estimate all possible combinations of conflicts separately, providing the most accurate conflict redistribution methods. On the other hand, DST fusion can estimate only a single conflict parameter, but characterizes with less computational complexity than the DSmT rules of combination. Furthermore, experiments comparing these architectures, features and complexity are presented in the experiments section 7.4.

5 Occupancy grid noise filtering

The occupancy grid algorithm is tailored to work as a connection node, where different sensor measurements are aggregated into a single map of probabilistic states representing the vehicle's environment. During the process of information fusion, multiple types of uncertainties can be present and have to be dealt with appropriately.

The occupancy grid is an input source of information for the downstream components of the software stack. Based on the map, they have to make decisions about the future vehicle actions. Therefore, the development and tuning of the algorithm require reliable metrics to measure the quality of the occupancy grid.

As a comprehensive solution for perception, the mapping algorithm can be tuned to the desired application by a variety of calibration factors, ranging from grid size and resolution, through to different sensor modeling possibilities and decay mean lifetime adjustment. Understanding of the parameters which drives the occupancy grid algorithm is a key to deliver a versatile mapping solution.

Classification of the common uncertainty sources present in the algorithm is described in Section 5.1. In order to assess the filtering capabilities of the mapping solution, an overview of available and novel quality metrics is presented in Section 5.2. Finally, Section 5.3 draws forward four main grid filtering capabilities, whose effects are measured in the next chapters.

5.1 Types of uncertainty

The occupancy grid algorithm aggregates heterogeneous sensor measurements into a single representation of the vehicle's environment. The measurement unification process comes with the disadvantage that all measurements have to be transformed into a grid frame of reference prior to the fusion step. As presented in Figure 5.1 the transformation from the detection coordinate to grid frame consists of two conversions: from the sensor frame to vehicle and from vehicle to grid frame. Positioning of the intermediate sensor and vehicle frames is reported as sensor mounting and vehicle position. Measurement uncertainty for these attachment points adds up to the total grid uncertainty level.

Moreover, the occupancy grid is fusing time series data. Synchronization issues or latency delays will result in errors in the correct detection placement.

Overall, the sources of uncertainty can be separated into four groups as presented in Figure 5.2. Each of these uncertainty types is described in detail in the following section.

5.1.1 Data synchronization delays

The process of data collection and processing requires accurate time measurements which are synchronized together. Ideally, all the vehicle's components should be aligned with the time master clock as specified in, for example, the AUTOSAR Time Synchronization Protocol [AUTOSAR, 2019].

In reality, however, even if the time is synchronized around the vehicle ECUs, every sensor might process measurements for different amounts of time, therefore producing detections with nonidentical

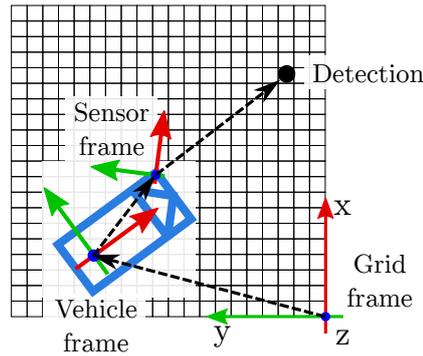


Figure 5.1. Coordinate transformation chain connecting sensor detection and grid measurement representation.

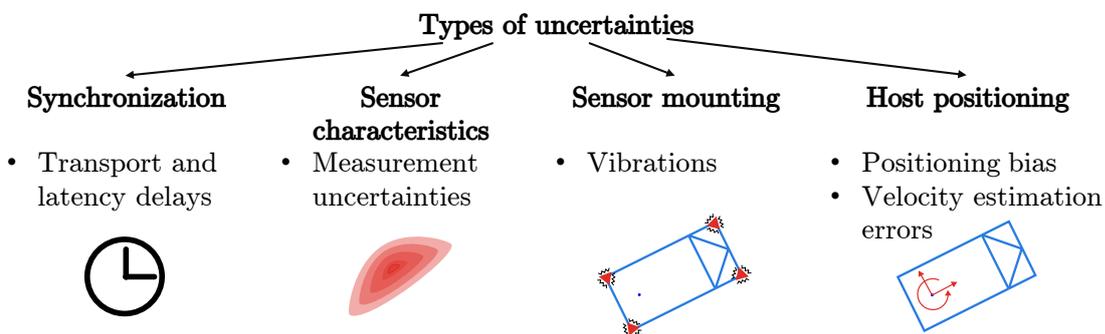


Figure 5.2. Schematic representation of four types of uncertainties present in the occupancy grid algorithm.

latencies. Measurement processing delays might amplify when the detections have to go through post-processing to validate or improve their quality. An example of sensor measurement postprocessing might be the extraction of the free space envelope from the camera image or the radar detection motion classification based on their range rate measurements.

Components of an automated vehicle are connected through CAN or FlexRay networks. Transmission of data packets via these networks is fast, but not instantaneous. When the detection processing chain is long or the network load is high, the transportation delay might even overcome the sensor processing latency. Fortunately, system can compensate and synchronize sensor measurements at the cost of increasing the total system latency.

5.1.2 Sensor detection characteristics

Each sensor measurement has uncertainty associated with spatial position of the detection. The level of uncertainty depends mostly on the type of the sensor and the detection position, i.e., its range and azimuth.

The minimal detection uncertainty levels could be achieved using LIDAR sensors. LIDAR provides accurate detections from the whole field of view.

Within automotive sensors, radars characterize with relatively high angular resolution errors and considerably smaller range errors. Radar detection uncertainty distribution forms a lens, or a bean shape as conceptually presented in Figure 5.2. Radar uncertainty depends on the range of detection – objects further away from the sensor might have higher uncertainty levels.

The camera sensor used to measure the free space envelope in front of the vehicle has high range measurement uncertainty which, similarly to the radar sensor, increases with the measured object distance.

5.1.3 Sensor mounting vibrations

The third type of uncertainties are sensor mounting vibrations. Sensors are mounted on the vehicle's body using holders to ensure accurate sensor placement. The holders are designed to withstand harsh road conditions, during drive it is subject to various high frequency vibrations which might tilt or misplace the sensor. For automotive applications, perception sensors are designed to detect objects from far ranges. Even a slight noise in the angular positioning of the sensor might result in increased spread of detections.

As a solution to that problem, some sensors may periodically run an alignment routine, which measures the accurate sensor mounting position and orientation relative to the vehicle frame of reference.

5.1.4 Host positioning uncertainties

The last but not the least source of sensor uncertainties are host positioning inaccuracies. The vehicle might measure its position in the world frame either using the external source of positioning such as GPS or by integrating internal linear and angular velocities to compute the relative position change. Both these methods are utilized in automotive and have their own advantages and disadvantages.

Nevertheless, as the occupancy grid in automotive applications represents only the closest surrounding area, the global position shift does not impact the grid quality. Nevertheless, even a slight bias in the measurement of linear or angular velocities might result in changing position misalignment, which leads to wrong detection placement on the grid map. This type of uncertainty behaves similarly to the synchronization issue, where the velocity estimation error shifts the vehicle frame and the whole grid to the wrong position.

5.2 Assessing occupancy grid mapping quality

A common approach for the evaluation of a map is visual inspection combined with the algorithm's expert knowledge. Nevertheless, the currently used visualization methods may lead to ambiguous conclusions as mentioned in the previous chapter in Section 4.2.5. This often leads to subjective results of the evaluation [Balaguer et al., 2009; Wagan et al., 2008]. In order to enable the systematic validation of the occupancy map, binary classification key performance indicators (KPIs) such as false positive rate, precision and recall are applied against the reference ground truth map.

One option to assess the matrix of cells within the grid is to compare cell by cell with the reference map. In order to enable the systematic validation of the occupancy map, binary classification KPIs such as false positive rate, precision, and recall are applied to the reference ground truth map. Application of continuous classification KPIs, such as a map score and covariance, also does not give satisfactory results [Grewe et al., 2012]. As presented in [Grewe et al., 2012], the existing quality measures from robotics are not adequate for automotive applications. The different goals in robotic and vehicle mapping limit the quality of an automotive map from a robotic viewpoint. The other downside of cell-wise grid evaluation is the requirement of highly accurate and therefore expensive ground truth information. Determining the complete ground truth for a stationary environment poses major practical challenges, and the data collection and labeling will contribute to the overall cost of the algorithm. Such metrics can be conveniently applied in simulated scenarios [Markiewicz and Porębski, 2020; Skruch et al., 2015].

Another approach for the evaluation of a grid map is the extraction of specific object representations and validation of their quality. This approach was first proposed by [Wagan et al., 2008] for robotic applications. The procedure was extended into automotive applications by [Laconte et al., 2019; Ledent et al.,

2019] for collision risk estimation and by [Steyer et al., 2019] for grid-based object tracking. In these evaluation applications, the occupancy grid is compared with a labeled list of reference dynamic objects or with an accurate scene description. This approach still requires well-described test scenarios or precisely labeled experimental data, but the reference information might be usable also in the development of different components (such as multi-object tracking), which lower their total cost.

As mentioned before, the automotive occupancy grid evaluation method should be able to assess grid quality using sparse reference scene description. The objects evaluated using this method should be frequently accessible during the test drives, and their labeling should be minimal to limit the cost of evaluation. This assessment criteria are met for signs and guardrails, which can be found very often on the road and whose metal elements effectively act as scattering centers for radar. Other object representations also might be used as evaluation metrics for the occupancy grid; however, they would require a complex assessment procedure. If the object has a complicated shape (as buildings and vegetation do), the grid representation will be different, based on the object's angle of incidence. Moreover, arbitrarily shaped objects might be subject to occlusions, which make the assessment even harder.

The specifications of a pole and its representation make it an ideal candidate for the evaluation reference point. Thanks to the simple cross-section of the pole, its representation will be independent of the vehicle's relative direction. Furthermore, poles are one of the most common objects in the highway environment. They are also accurately mapped using high-definition maps of roads and are clearly visible to other sensors, such as LiDARs or cameras, which makes the reference mapping easier. This feature evaluation can work on the sparse reference scene description. The ground-truth information used in the procedure can be easily obtained from an independent parallel perception setup, thereby limiting the effort required for the implementation of new algorithm versions [Porębski and Kogut, 2021]. This evaluation approach is further extended in the present chapter.

5.2.1 Extraction of interesting occupancy grid features

On the occupancy grid, the pole-like objects generate circular or elliptical grid map representations, as presented in Figure 5.3 for signs, bollards, or even guardrails. For guardrails (Figure 5.3c), all the detections are reflected from the poles supporting the guardrail, not from the railing itself, resulting in the characteristic dotted line of occupancy.

Extraction and identification of relevant objects from the occupancy grid should be performed prior to the KPI calculation step. In order to automate object detection, a clustering algorithm or a neural network solution could be used. From the family of clustering algorithms, the agglomerative methods such as DBSCAN algorithm show the most benefits for the occupancy grid application. The DBSCAN clustering is able to discard noise, it can detect any number of clusters, and its simple structure allows various optimizations for grid-based data approach [Feng et al., 2017]. Nevertheless, the out-of-the-shelf clustering algorithms can tackle only binary maps, therefore any occupancy grid type should be converted into single value grid map and binarized using manual or automatic thresholding such as Otsu's method.

There are different methods to simplify the occupancy grid state into a single number representation. To properly detect the stationary poles, this single value grid should emphasize static evidences. For the Bayesian type of occupancy grid, the only choice is to use the probability value for clustering. For the Dempster-Shafer grid representation, the stationary belief map will extract only evidences related to the evaluated landmark representation. An example of the clusters extracted from different types of occupancy grids is presented in Figure 5.4.

In the following thesis, the objects were identified using DBSCAN clustering algorithm and compared with the ground truth feature placement in order to associate only relevant clusters. For the clustering, a manual threshold value was utilized based on the probability value for the Bayesian occupancy grid types and stationary belief for the Dempster-Shafer ones.

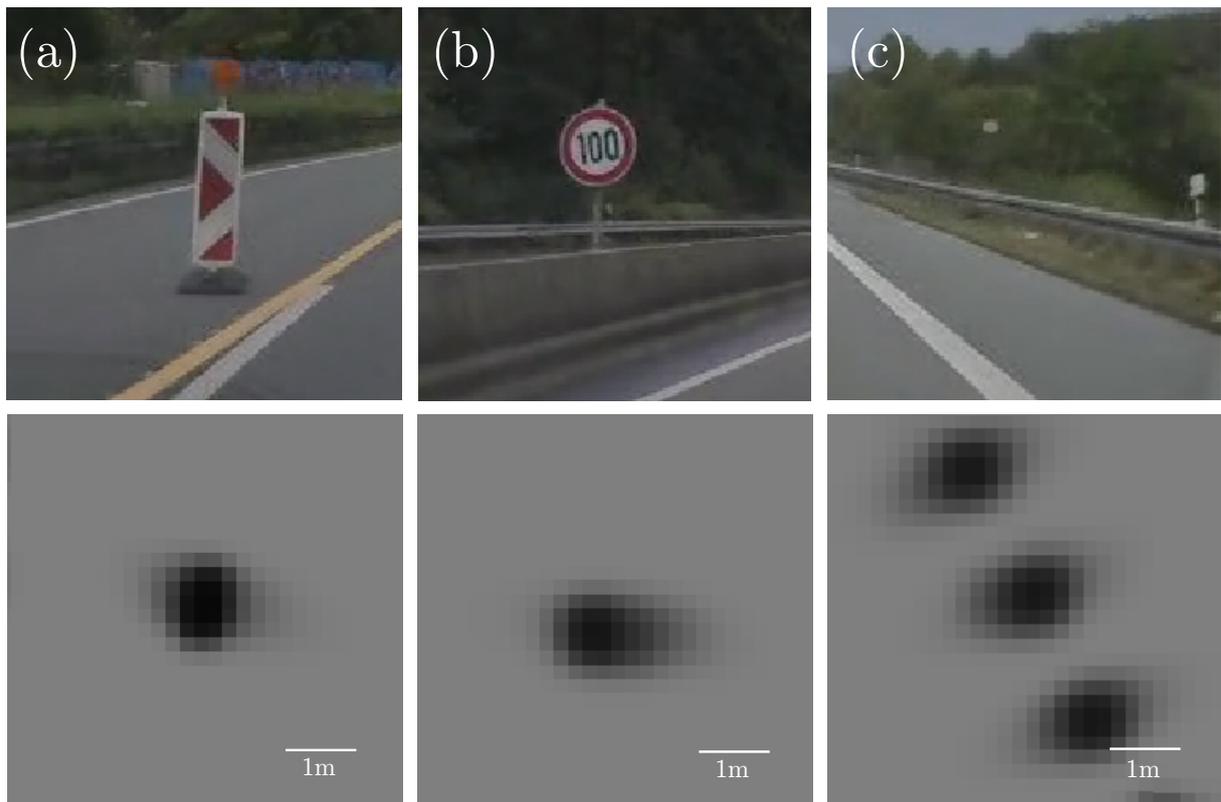


Figure 5.3. Examples of the occupancy grid representations of highway pole-like objects with corresponding video frames: (a) traffic bollard; (b) sign; (c) guardrail. The occupancy grid is generated using a single automotive-graded radar located in front of the vehicle (Image reused from: [Porębski and Kogut, 2021]).

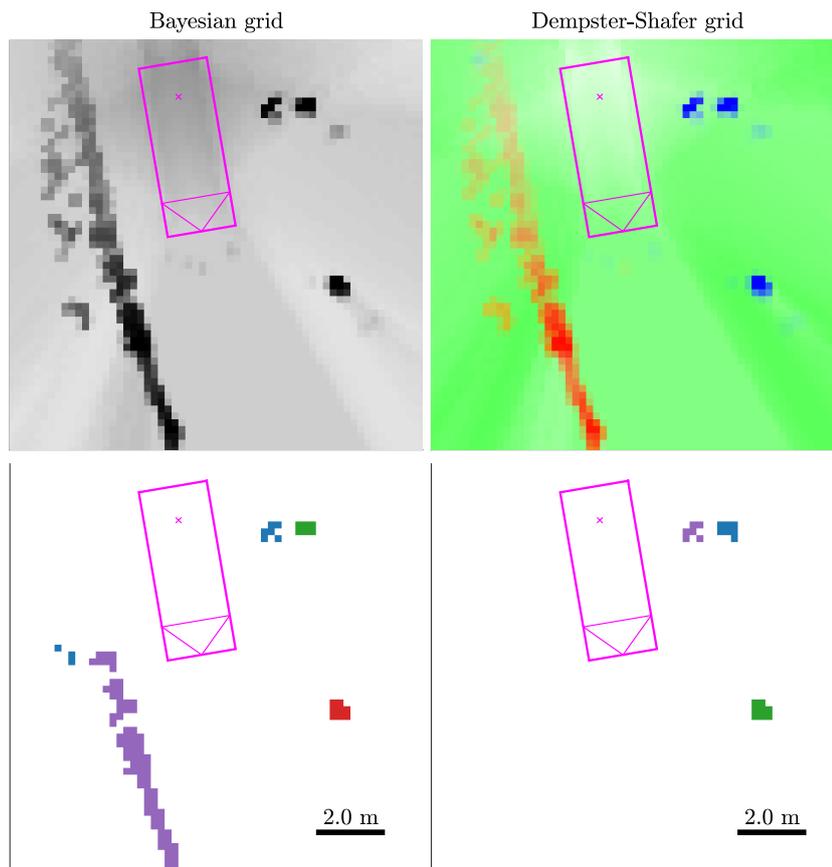


Figure 5.4. Comparison of clusters extracted from the Bayesian and Dempster-Shafer occupancy grid using the same binarization threshold levels. Clustering based on probabilities in Bayesian grid cannot deal with the trail of dynamic occupancy (violet cluster on the left). Clustering based on only stationary belief values in the DS grid (right) filters out dynamic and ambiguous occupancy from the cluster representation.

5.2.2 Definition of grid quality metrics

The grid representation of a pole should preserve the number of objects in the real world, limit their variance, and maintain the object's shape. To meet these requirements, the pole-like object's representation must be compact (without any gaps), occupy the finest space, and have low sensor conflict associated (low entropy). These attributes are the basis of a KPI definition.

As the occupancy grid can be constructed without the access to a global positioning sensor such as GPS, the localization of the object does not impact the KPI values. Only the shape of the representation is considered for the algorithm performance estimator.

5.2.2.1 Consistency

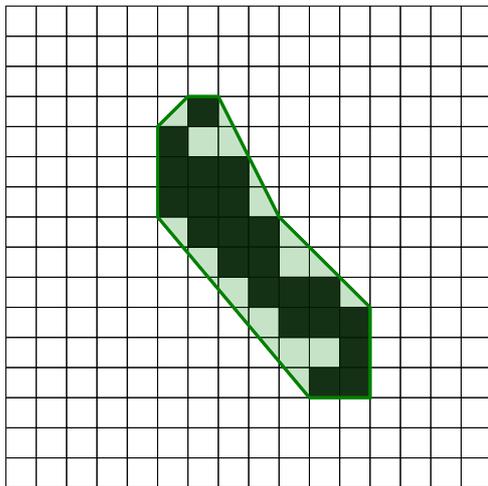
A consistency describes whether the pole's image on the occupancy grid can be expressed as one solid and convex object. Consistency measures the density of a representation, and in image processing may be called the solidity of an object. A measure of consistency can be obtained as a ratio of the object's area to the area of the convex hull of the object:

$$\text{Consistency} = \frac{\text{Cluster cell number}}{\text{Convex area}} \quad (5.2.1)$$

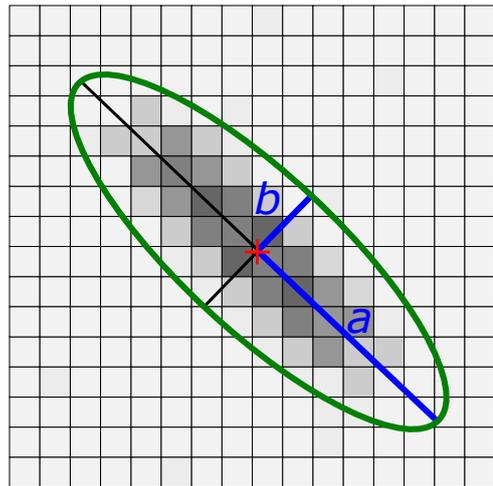
where "Cluster cell number" is the number of occupied cells classified for the object and "Convex area" is the number of cells which lie within the convex contour around the selected occupied cells. The convex hull of the cluster can be computed easily using different cluster envelope algorithms.

For evaluation purposes, every updated cell is taken into account; the convex hull size is computed over the measured pole representation area. Visualization of this KPI measure is presented in Figure 5.5a.

The best value of compactness should be as close to 1 as possible. If the consistency of the representation is close to zero, the object is sparse and should be divided into smaller ones. Computation of this KPI value is a validation check, enabling further quality assessment.



(a) Consistency calculation schematic. Black cells are occupied. In green is the computed convex hull.



(b) Area and circularity (eccentricity) ellipse. Blue lines mark major (a) and minor (b) ellipse axes which correspond to 2σ values of the corresponding distribution.

Figure 5.5. Schematic visualization of key performance indicators (KPI). (Image reused from: [Porębski and Kogut, 2021])

5.2.2.2 Area of the object representation

Consistency KPI considers only the binary image as a basis for the object's area measure. Such an approximation, however, does not consider individual cell weights and can be used only for rough object validation and not for the estimation of the object's variance.

The pole-like object representation on the occupancy grid plane is generated by radar scattering centers located in the pole's section. Hence, the grid object may be approximated as a Gaussian distribution. The estimated pole variance is measured as the area of distribution stretching over the standard deviation σ range.

The weighted maximum likelihood estimator [Gebru et al., 2016] is used to estimate Gaussian distribution for each set of cells taking into consideration their probabilities as weights for each data point.

$$\hat{\mu}_{ML} = \frac{\sum_{i=1}^N w_i x_i}{\sum_{i=1}^N w_i} \quad \hat{\Sigma}_{ML} = \frac{\sum_{i=1}^N w_i (x_i - \hat{\mu}_{ML})(x_i - \hat{\mu}_{ML})^T}{\sum_{i=1}^N w_i} \quad (5.2.2)$$

where:

- $\hat{\mu}_{ML}$ – estimated center point of the Gaussian distribution.
- $\hat{\Sigma}_{ML}$ – estimated covariance matrix of the Gaussian distribution.
- N – number of selected object's cells.
- w_i – i-th cell probability.
- x_i – i-th cell coordinates in 2D plane.

The area and circularity measurement require only information about Gaussian covariance matrix eigenvectors λ , which represent the variance of the distribution

$$\det(\Sigma - \lambda I) = 0 \quad (5.2.3)$$

The occupancy grid always has limited resolution of the object representation. To reflect that discrete parameter, the computed variance has some minimal value.

To estimate this minimal value, let us consider a single cell object. This object distribution should fit entirely into the cell area. For the Gaussian distribution, the 99.7 % of evidence lies inside the 3σ ellipse, thus the minimum variance is equal to:

$$\lambda > \frac{(\text{grid resolution})^2}{9\pi} \quad (5.2.4)$$

If any of the measured eigenvalues is smaller than the acceptable minimum value, it should be artificially increased to match this condition.

The standard deviation σ can be easily computed by taking the square root of the Gaussian variance λ .

$$\sigma = \sqrt{\lambda} \quad (5.2.5)$$

The approximated distribution area for given standard deviations is an ellipse with major and minor axes equal to the computed values (Figure 5.5b). Measured area of the object should cover at least 95% of the total's object evidence, therefore the area A_{95} is an ellipse stretching over the 2σ range (5.2.6).

$$A_{95} = 4\pi \cdot \sigma_a \cdot \sigma_b \quad (5.2.6)$$

where σ_a is the standard deviation over the semi-major axis of the fitted distribution, and σ_b is the standard deviation over the semi-minor axis.

In order to minimize the pole-like object representation spread, the area of occupancy should be minimized as well.

5.2.2.3 Entropy

The Dempster-Shafer occupancy grid framework enables qualitative measurements to determine the presence of sensor conflict. As presented in Section 4.2.6 entropy measurement can be used to measure the inconsistency level of the cell state.

In order to measure the level of sensor conflict associated with the pole object representation on the occupancy grid, the maximum cell entropy is computed for the cells corresponding to the object. This KPI can be used only in the Dempster grid framework and is useful only in cases when the sensor conflict might be present.

5.3 Occupancy grid filtering capabilities

The occupancy grid algorithm data flow consists of four steps as presented in Section 3.4: sensor modeling, fusion, decay, and grid shift. Each of these steps has its own calibration parameters which might be changed to tune the occupancy grid filter. A variety of parameters provide great flexibility for the algorithm application, but dependencies between different calibrations can be hard to observe.

This section describes how the sensor modeling, decay, and grid shift calibration influence the uncertainty filtering process of the occupancy grid. Fusion step has been extensively described in Chapter 4, therefore its influence will be omitted here.

5.3.1 Sensor detection modeling

Each occupancy grid input sensor has unique properties, which can be utilized to extract more information about the environment. The sensor modeling part of the occupancy grid algorithm aims to improve the accumulation procedure by transferring as much information as possible from the detection model to the occupancy grid. The sensor model has to consider all types of uncertainties present in the occupancy grid. Usually in the experimental setup, many uncertainty characteristics are not directly measurable, and the sensor model has to approximate the overall grid detection uncertainty, not only the sensor parameters itself.

The modeling can be performed via forward or inverse sensor modeling. Forward methods optimize the occupancy distribution based on accurate physical models of sensors. This type of modeling requires a lot of computational power and is not considered in the presented evaluation analysis.

The second type of modeling—the inverse sensor model—spreads the detection point into the occupancy probability based on a probabilistic distribution. This type of representation is commonly employed in occupancy grid algorithms. Sensor models may be differentiated based on the number of dimensions in which the distribution is computed.

The simplest and most widely utilized sensor model is the zero-dimensional hit-point model (Figure 5.6a). Its distribution assigns maximum occupancy evidence to the cell where the detection is located, and no evidence to other cells. This model is widely exploited due to its minimal computational and implementation requirements.

The other commonly used inverse sensor model is based on a two-dimensional probability distribution. The model computes evidence values in Cartesian or polar coordinates around the detection (Figure 5.6b). The evidence value is usually approximated as a Gaussian distribution with the constraint that the sum of all evidences should be equal to the detection's existence probability [Thrun et al., 2005].

Tuning of the hit point inverse sensor model consists of a single parameter, which determines the evidence assigned to the associated cell. This parameter is called the detection existence probability, and it describes the correctness of the measurement (probability of true positive).

For the 2D sensor model, the existence probability determines the total evidence assigned to the single detection. This sensor model can be calibrated using the range and cross-range standard deviation, which describe the distributions produced by the model.

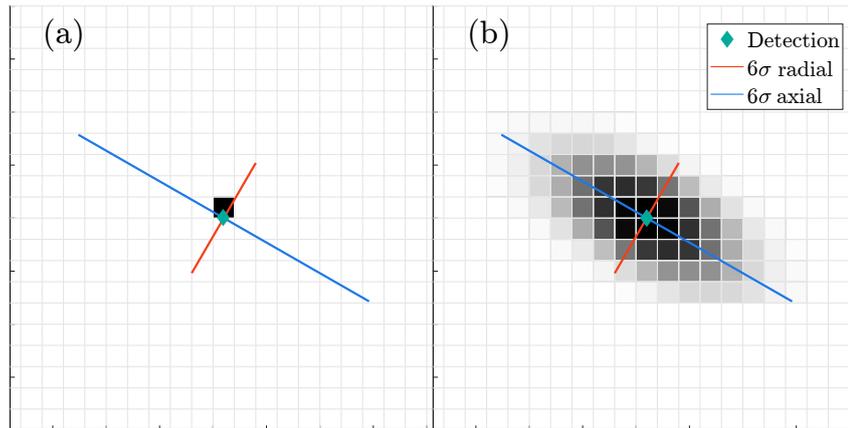


Figure 5.6. Types of sensor models: (a) hit point; (b) 2D approximation of detection uncertainty. Darker areas represent cells with higher probabilities.

5.3.2 Free space estimation filtering

Free space modeling exploits the fact that range sensor detections provide information not only about the obstacle, but also about empty areas along the space traversed by detection. This determination is developed by casting rays on the detections. The traversed space is then updated to increase the free space probability along the ray.

This type of filtering is extensively utilized for LIDAR occupancy grids, where the overall detection density is higher [Yguel et al., 2008]. The ray casting technique in such applications leads to Moiré artifacts or false free space determination.

This thesis implements the triangle ray casting method presented in prior research [Porębski, 2020], which solves the problem of artifacts by using a wider area for updates. Every cell in the triangle ray is filled with the same value, thereby forming a uniform free space probability distribution as presented in Figure 5.7.

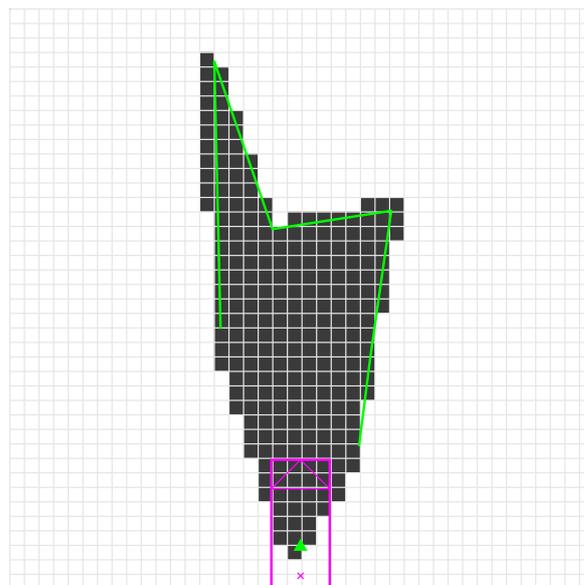


Figure 5.7. Example of the segment free space sensor model. Area inside the envelope is filled uniformly with free evidence equal to the free space gain.

This uniform free space sensor model can be calibrated using a single parameter: free space gain. This gain coefficient describes the free space evidence assigned to every cell in the sensor model step. As presented in Figure 5.7, every cell inside the envelope is assigned with the free space gain evidence value.

5.3.3 Dealing with modeling overconfidence

The key limitation of sensor and free space modeling is the simplification of many physical dependencies—the algorithm assumes data completion and cell independence. During the process of integrating multiple pieces of evidence, hidden dependencies are omitted. Consequently, the results quickly become overconfident [Thrun et al., 2005]. In order to deal with this overconfidence, the decay step is used to artificially diminish the evidence on the entire grid over time.

The overconfidence of occupancy modeling may be handled using different methods, but the most common one is the exponential decay described in Section 3.4.4. The calibration of this functionality consists of a single mean lifetime value τ . Nevertheless, the mean lifetime tuning is counter-intuitive – setting higher values of the mean lifetime slows down the decay. For the tuning and evaluation presented in this thesis, the inverse of the mean lifetime value called decay rate r will be used:

$$r = \frac{1}{\tau} \quad (5.3.1)$$

The decay rate r directly represents the speed of evidence forgetting, increasing the decay rate makes the decay work faster.

5.3.4 Sensor synchronization

Measurement lag and transport delays are ubiquitous in automated systems and cannot be removed completely. Data collection, processing, and transmission is not instantaneous, the final perception system always has some overall latency. Accurate time measurement and ECU synchronization might limit the impact of this uncertainty on the occupancy grid processing, but these systems are expensive and do not solve all the synchronization problems completely.

Handling sensor time differences within the occupancy grid is still rarely described in research publications. Many of the real occupancy grid applications omit this issue and assume no synchronization delay for their systems, forwarding the uncertainties to the sensor modeling step. The other approach is to use the so-called spatio-temporal detection alignment, where different sensor timestamps are aligned, and the detections are artificially moved to predicted positions using some motion models [Tanzmeister and Steyer, 2016]. This approach enables the processing of multiple sensor scans at the same time, improving the algorithm performance. On the other hand, the application of motion model to detection might lead to increasing their uncertainties, especially for dynamic object measurements.

In order to minimize the impact of the sensor synchronization and timing delays in the occupancy grid, every sensor scan is fused into the map independently. For every sensor scan, the occupancy grid is shifted to reflect the sensor timestamp, so the data might be fused to the grid without any spatio-temporal shifting needed.

6 Simulation evaluation

Advanced Driver Assistance Systems (ADAS) are complex structures combining multiple independent Electronic Control Units (ECUs), which communicate together using different communication protocols. As the number of important road scenarios for the system grows, mathematical modeling and computer simulation become important engineering tasks that aim to assure the required quality and compliance with safety standards.

The virtual validation environments are important for ADAS systems to quickly and reliably test an increasing number of important road scenarios required for quality assurance and safety standards [Markiewicz et al., 2018; Skruch et al., 2015].

For the evaluation presented in this thesis, the simulation environment offers full control and monitoring of all relevant parameters: the road scenery, sensors, their uncertainties, and timing. Simulations allow isolating the influence of some effects, therefore enabling their detailed evaluation.

As described in previous chapter 5 there are different uncertainty sources which might affect the occupancy grid and the grid can filter the information using various filtering features. This chapter defines the simulation environment in Section 6.1 and then describes in detail how different noise types and their levels impact the occupancy grid in Section 6.2. Handling noise level is presented in Section 6.3

6.1 Simulation environment

The simulation has been designed using the *Driving Scenario Designer* toolbox available in MathWorks MATLAB. The Driving Scenario Designer app enables designing synthetic driving scenarios for testing autonomous driving systems. The application is capable of creating different road scenarios based on real map snapshots, nevertheless, for the purpose of tuning the occupancy grid algorithm, a simple driving scenario was prepared.

6.1.1 Driving Scenario Designer

The Driving scenario presented in Figure 6.1(a) consists of a 250 m road segment with a narrow S-shaped turn which emphasizes the impact of heading angle and velocities. The vehicle is moving along the selected trajectory with the constant linear speed of $20 \frac{\text{m}}{\text{s}}$.

The simulated vehicle is equipped with five ideal radar sensors. Four of them are corner radars with 150° field of view (FOV) and 50 m maximum measurement range. The last sensor is a front radar with 100° FOV and 100 m maximum range. Schematic of the bird's eye view of the sensor FOVs is presented in Figure 6.1(b). The simulated radar sensors are producing the new sensor scan every 50 ms (20 Hz frequency). Information about the host position and velocity is provided with 100 Hz frequency.

Along the vehicle there are placed 130 poles with dimensions $10 \times 10 \times 200$ cm. Obstacles are placed every 10 m along the road and are distanced -12 , -2 , 2 , 6 and 16 m away from the vehicle. Pole placement reflects the possible position of real structures placed on the boundaries of the surrounding lanes of the highway. The obstacles visible from the vehicle point of view at the start of the simulation are presented in Figure 6.1(c).

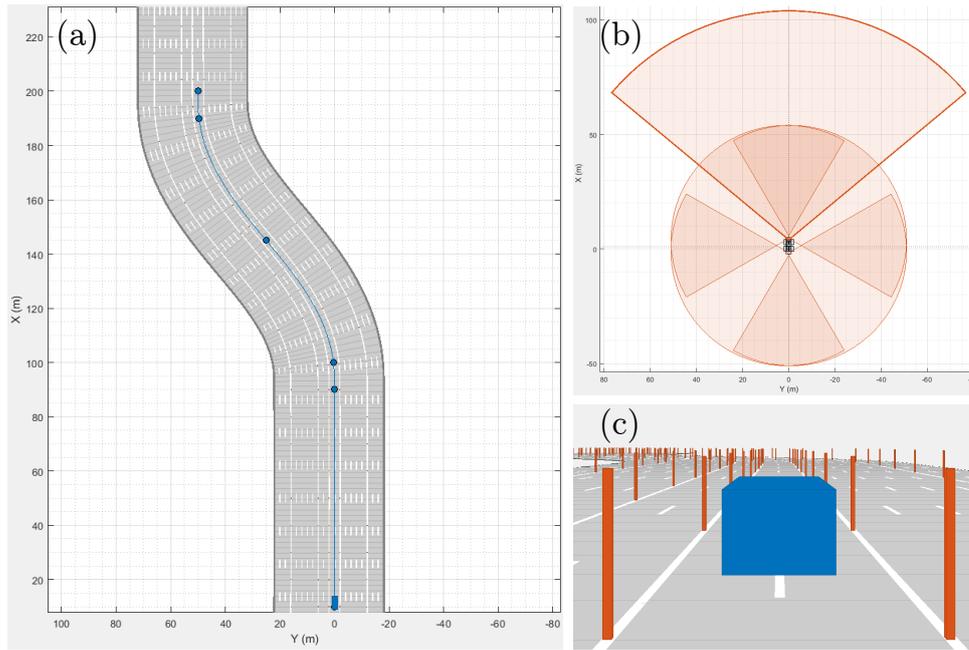


Figure 6.1. Simulation environment used for the evaluation of the occupancy grid. All simulations were prepared using *Driving Scenario Designer* application in MathWorks MATLAB.

6.1.2 Sensor uncertainty simulation

The simulation scenario is utilized to generate the baseline data set without any sensor inaccuracies. Sensor uncertainties are added as a post-processing step for more flexibility. All detections reported by the simulated sensors are true positive measurements, there are no false detections or missed objects. For the random uncertainties, the zero-mean Gaussian noise is assumed with variable standard deviation.

The generated noised sensor files are passed to the occupancy grid algorithm in order to generate the occupancy grids. The difference between maps generated using the ideal sensor data and noised detections is presented in Figure 6.2.

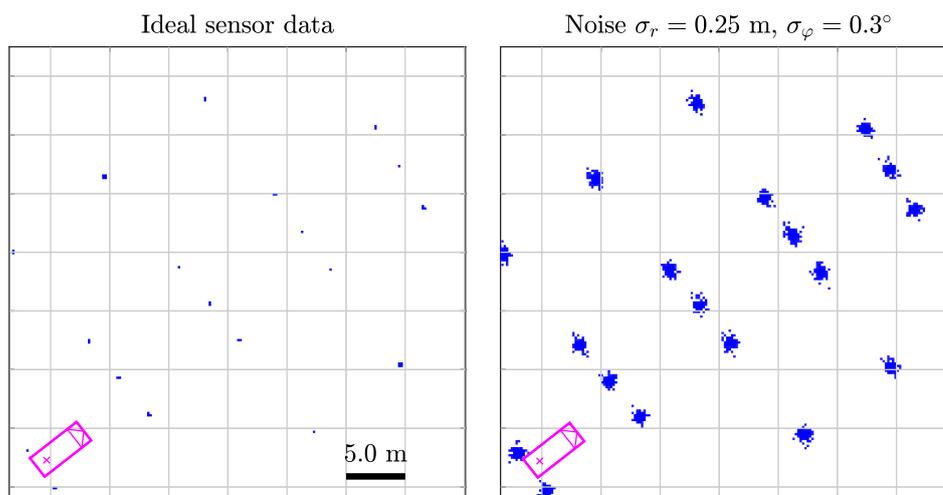


Figure 6.2. Snapshots of the two occupancy grids generated using the simulation environment without any filtering enabled. Figure on the right presents a grid with applied example noise parameters.

6.1.3 Occupancy grid configuration used for the simulation evaluation

The occupancy grid used for the simulation utilized cells with 20 cm resolution, and the total grid size of 100x100 m. The host position on the grid was adjusted to focus on the front area of the grid by placing the vehicle on the circle with radius of 45 m from the grid center (see Figure 6.3 for details). Occupancy grid algorithm operates with 20 Hz; however, every sensor scan is still fused independently on the map, without any spatio-temporal shifting.

In the simulated scenario vehicle is moving with the constant linear speed of $20 \frac{\text{m}}{\text{s}}$, therefore the 100 m span of environment depicted by the occupancy map is accumulated over 5 s of movement. In this time sensor working with the 20 Hz frequency produces 100 scans of measurements. Based on that every pole representation on the occupancy grid is constructed from approximately 100 to 200 detections from all sensors (assuming that a pole produces single radar reflection in every sensor scan).

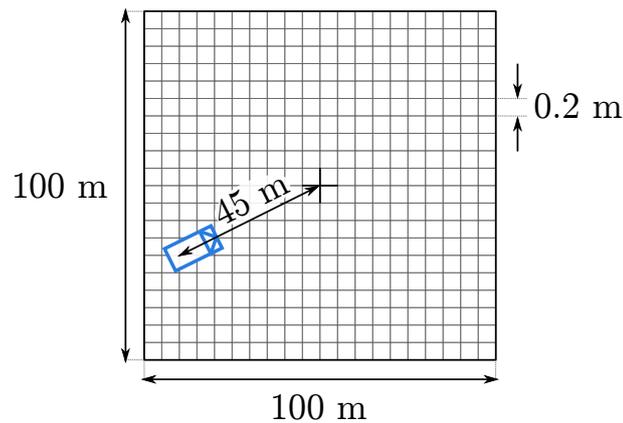


Figure 6.3. Grid dimensions and host placement in simulation scenario.

6.2 Comparison of different uncertainty types on the occupancy grid

The occupancy grid algorithm has to deal with different sources and levels of uncertainty as presented in Chapter 5. In order to compare the influence of each type of uncertainty, isolated simulation tests were performed. In every of these test scenarios, only a single uncertainty parameter was modified to emphasize its influence on the occupancy grid.

All evaluations were performed on the Dempster-Shafer occupancy grid framework using the DSmH fusion rule. Decay and free space filtering methods were disabled and for the sensor modeling, the zero-dimensional hit point sensor model with 90% existence probability was utilized.

The four main types of sensor noise presented in Section 5.1 define the main sources of uncertainties. Nevertheless, these sources can be further divided into subcategories of individual independent control parameters. In total, 12 different noise types were tested, which consist of:

- (a) ideal — Baseline simulation without any noise,
- (b) σ_r — Detection range uncertainty,
- (c) σ_φ — Detection azimuth uncertainty,
- (d) $\sigma_{\varphi_{origin}}$ — Uncertainty in sensor origin yaw measurement,
- (e) $\sigma_{\varphi_{host}}$ — Uncertainty in host yaw measurement,
- (f) σ_{sync} — Synchronization uncertainty,
- (g) $\sigma_{xy_{origin}}$ — Sensor origin positioning uncertainty along both X and Y axes of the sensor frame,
- (h) $\sigma_{xy_{host}}$ — Host positioning uncertainty both X and Y axes of the vehicle frame,
- (i) $\text{bias}_{\varphi_{host}}$ — Bias in host yaw angle measurement,
- (j) $\text{bias}_{xy_{host}}$ — Bias in host position measurement along both X and Y axes of the vehicle frame,
- (k) $\text{bias}_{v_{host}}$ — Bias in host linear velocity measurement,
- (l) $\text{bias}_{\omega_{host}}$ — Bias in host yaw rate measurement.

Uncertainties marked as σ are a zero mean Gaussian noises, where the σ value describes the standard deviation of this noise level. For the offset uncertainties marked with 'bias' name, the corresponding value is shifted by the 'bias' value for the whole scenario, mimicking, for example, the velocity estimation bias error.

The following sections present how each type of sensor uncertainty affects the simulated object representation and propose a qualitative and quantitative comparison for different noise amplitudes relative to system characteristics.

6.2.1 Qualitative evaluation of noise types

In order to measure the influence of each noise level qualitatively, each noise level was tuned to have the same mean pole object area throughout the simulation. Considered uncertainty levels are equal to:

- | | | |
|---|---|--|
| (a) 0 (ideal detections), | (e) $\sigma_{\varphi_{host}} = 0.3 \text{ deg}$, | (i) $\text{bias}_{\varphi_{host}} = 1.5 \text{ m}$, |
| (b) $\sigma_r = 0.25 \text{ m}$, | (f) $\sigma_{sync} = 10 \text{ ms}$, | (j) $\text{bias}_{xy_{host}} = 10 \text{ deg}$, |
| (c) $\sigma_\varphi = 0.3 \text{ deg}$, | (g) $\sigma_{xy_{origin}} = 0.13 \text{ m}$, | (k) $\text{bias}_{v_{host}} = 0.5 \frac{\text{m}}{\text{s}}$, |
| (d) $\sigma_{\varphi_{origin}} = 0.3 \text{ deg}$, | (h) $\sigma_{xy_{host}} = 0.13 \text{ m}$, | (l) $\text{bias}_{\omega_{host}} = 0.46 \frac{\text{deg}}{\text{s}}$. |

Firstly, to get the overall view of the generated pole representation for each type of noise, snapshots of the sample pole in the occupancy grid are presented in Figure 6.4. The images are recorded from the middle of the simulation when the vehicle is performing the S-turn, when each pole accumulated around 100 radar detections. The arrow in the first image points the direction to the current host position on the map.

Secondly, all objects in the whole simulation scenario are analyzed for the estimated area of the ellipse corresponding to the landmark representation as described in Section 5.2. The area measurements

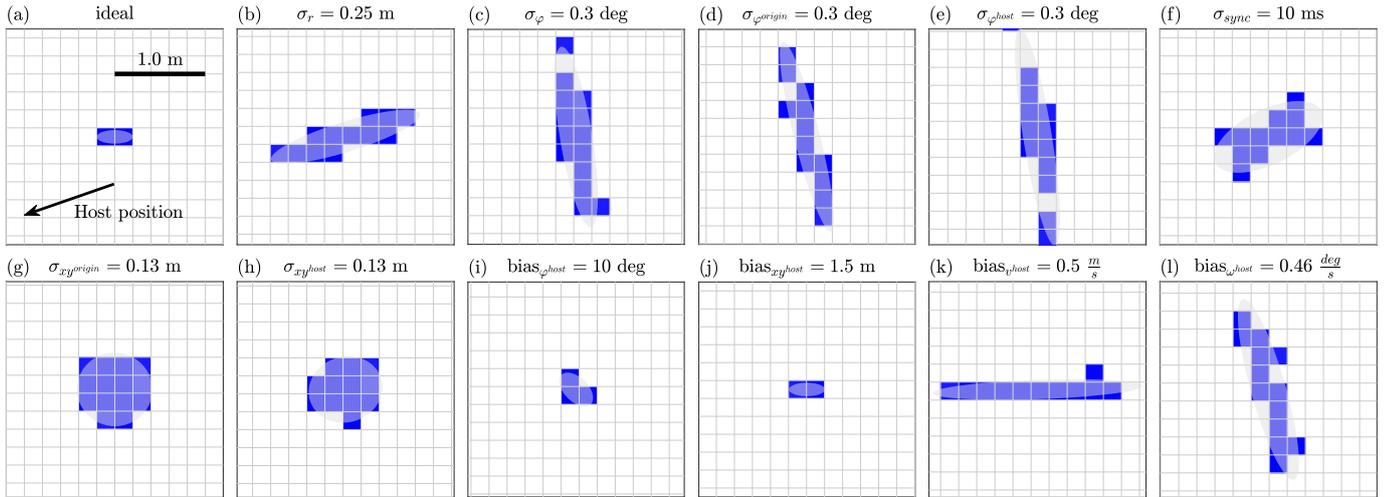


Figure 6.4. Grid snapshots of pole representation with different sensor noise types. Images represent the pole on the left side of the vehicle (host position is pointed by the arrow). Host is in the middle of the S-turn maneuver and is moving from the left to right (relative to the images). The transparent ellipses depict the area of the landmark's Gaussian representation.

are collected and presented in the form of violin plots in Figure 6.5. This analysis considers only poles closer than 80 m from the vehicle to limit poles with only a small number of evidences accumulated.

Violin plot is an extended version of the box plot, where the shape of the width of the violin in any point represents the frequency of the data. In the middle of the violin there is a box plot, where the middle line describes the median and the bold gray box depicts a range between 25% and 75% quantile of the data. The point in the middle of the violin describes the mean area of the object.

Based on the qualitative evaluation presented in Figure 6.5, constant bias on the host vehicle position and orientation does not influence the occupancy grid quality. Even high errors such as $bias_{xy^{host}} = 10^\circ$ and $bias_{\varphi^{host}} = 1.5$ m produced results close to the ideal occupancy grid. These types of noises are present mostly in the low-cost global positioning modules, where the satellite signal might be easily disturbed by the terrain structures. These errors do not impact the occupancy grid quality but might affect some downstream components if they require accurate map positioning in the world frame. Based on that, the constant bias on the host vehicle position and orientation is not omitted in the following sections.

The shape of the snapshots and corresponding landmark ellipse distributions shows that the angular uncertainties σ_φ , $\sigma_{\varphi^{origin}}$ and $\sigma_{\varphi^{host}}$ produce similar results. In the case of these three error types, the pole representation is heavily elongated in the direction perpendicular to the host position. The bias on the angular host velocity ($bias_{\omega^{host}}$) produce similar snapshot results, but since the angular aspect of the velocity is amplified only during turns, the resulting distribution of the landmark's area is different.

Bias in the linear host velocity ($bias_{v^{host}}$) causes the pole's representation to be elongated along the vehicle's trajectory. Similarly, the detection range uncertainty σ_r lengthen landmarks along the radial direction to the vehicle.

Finally, synchronization σ_{sync} , sensor mounting $\sigma_{xy^{origin}}$ and host position $\sigma_{xy^{host}}$ uncertainties affect the landmark representation similarly in all directions, resulting in mostly circular shapes. Moreover, the distribution plots for sensor mounting and host position noises have almost identical shapes, which permit the assumption that these two types of uncertainties have similar effects on the grid quality.

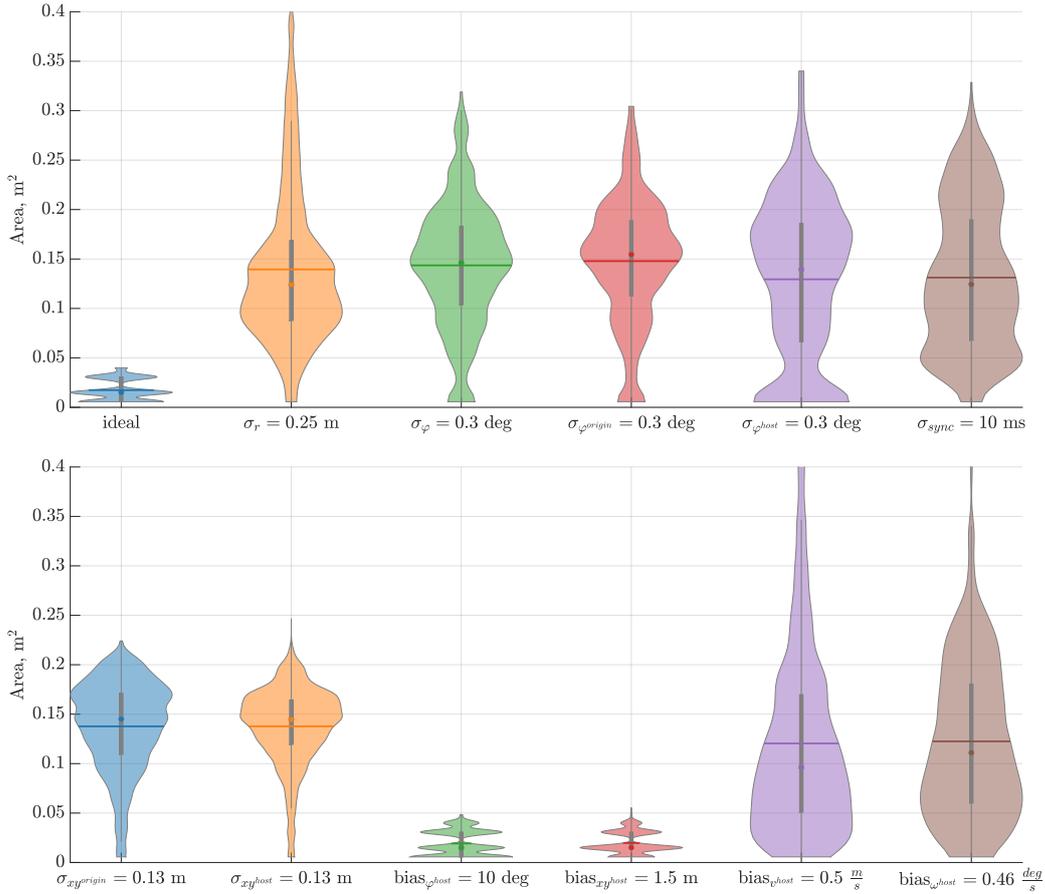


Figure 6.5. Distributions of landmark representation areas for different types of sensor uncertainties. When possible, uncertainty levels were selected manually to align the mean value of each representation.

6.2.2 Quantitative evaluation of noise types

The second step of the noise type comparison is to measure how the change of noise level influences the average area of the landmark representation. In order to do that, multiple simulations for different noise types and levels were prepared. For this analysis, biases in host position and heading were omitted, because their influence on the landmark representation is negligible. The results of the evaluation are presented in Figure 6.6.

All analyzed individual uncertainty types present a positive linear correlation with the mean landmark representation area in the simulated range of uncertainty levels. Nevertheless, not all presented noise levels are commonly present in the real vehicle. For example, if the sensor is properly mounted on the vehicle's body, its high-frequency vibrations cannot reach levels of 10 cm standard deviation. Similarly, if the host position is filtered out, but some inertial measurement unit oscillations should be much lower than the reference levels presented in Figure 6.6.d. These uncertainty types will have minimal or even negligible impact on the occupancy grid quality.

The angular uncertainty types presented in Figure 6.6.b show a lot of similarities with each other, which means for further analysis they could be simplified into a single uncertainty level.

Linear and angular velocity estimation levels presented in the plots 6.6.e and 6.6.f may be present in the real vehicle in the cases of quick acceleration or rapid turning. In most of the time, these measurements have to be more accurate, because a lot of downstream components rely on them and their

reliability. Therefore, the occupancy grid quality can be affected by the host velocity estimation errors in some cases, but when their values stabilize, the impact on the map should be minimal.

Based on the common levels of uncertainty levels present in real vehicles, the detection and synchronization uncertainty have a major impact on the occupancy grid quality.

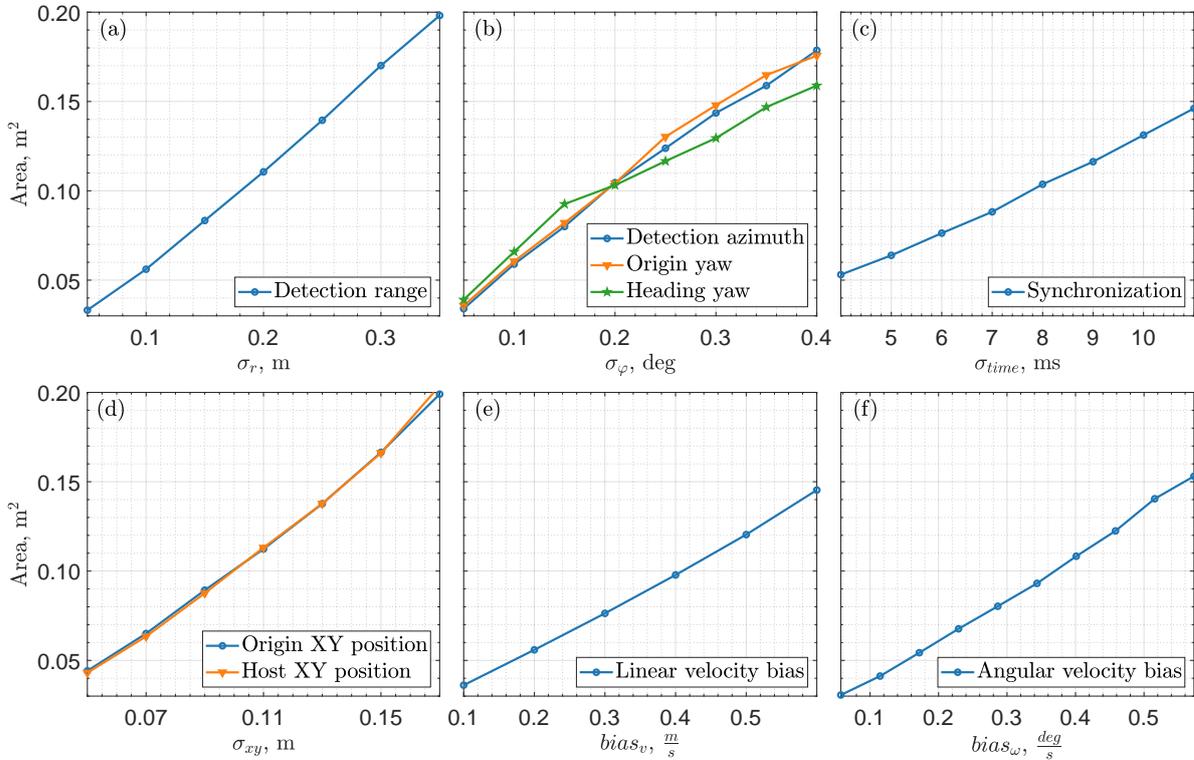


Figure 6.6. Plots showing mean landmark representation area for different noise types and levels.

6.3 Uncertainty filtering assessment

The occupancy grid has four main components which filter the uncertainties in the algorithm: occupancy and free space modeling, decay, and fusion rules (explained in detail in Chapter 5). The current section analyses the process of filtering noised occupancy grids in order to improve the grid quality. The evaluation process is a continuation and extension of single pole evaluation, which the results are published in [Porębski and Kogut, 2021]

The assessment of the uncertainty filtering capabilities of the grid is based on the aforementioned simulated scenario, with 130 poles instead of a single one as presented in [Porębski and Kogut, 2021]. For simplicity, only the detection uncertainties are allowed for this analysis. Detection uncertainty levels of $\sigma_r = 0.25$ m, $\sigma_\varphi = 0.3^\circ$, correspond to the average levels of the automotive grade radar [Autonomous Stuff, 2021].

All results presented in this section are based on the same simulated scenario described in Section 6.1. The example grid pole representations are recorded in the middle of the simulation, when the vehicle is performing the S-turn. The arrow in the snapshot image points the direction to the current host position on the map, and the transparent ellipse corresponds to the area of an estimated Gaussian pole representation.

For the analysis, only the landmarks closer than 80 m from the vehicle were considered as valid, in order to limit new poles with a few detections accumulated. The key performance indicators were measured using 0.3 evidence and corresponding 0.65 probability thresholds. Values of the threshold

were selected experimentally to limit the landmark area without discarding any existing landmarks as false objects. Detailed explanation of the threshold tuning is described in Appendix B.

6.3.1 Inverse sensor model types

The first experiments performed on the grid measure the impact of the sensor modeling on the pole representation on the occupancy grid. For this test, all decay and free space modeling are disabled to limit their influence. For the fusion method, the hybrid Dezert-Smarandache rule of combination was used, however all three fusion rules presented similar results for this case.

For this case, two basic ISMs were tested – the zero-dimensional hit point model, which always marks only a single cell and the Cartesian 2D model, which approximates the occupancy probability distribution as 2D Gaussian in Cartesian coordinates. The calibration for the Cartesian 2D ISM is equal to the sensor uncertainty levels provided by the simulation ($\sigma_r = 0.25$ m, $\sigma_\varphi = 0.3^\circ$).

The sample output pole image for the two types of inverse sensor models is presented in Figure 6.7 and the distribution of all available snapshots areas and consistency values are presented in Figure 6.8 as violin plots.

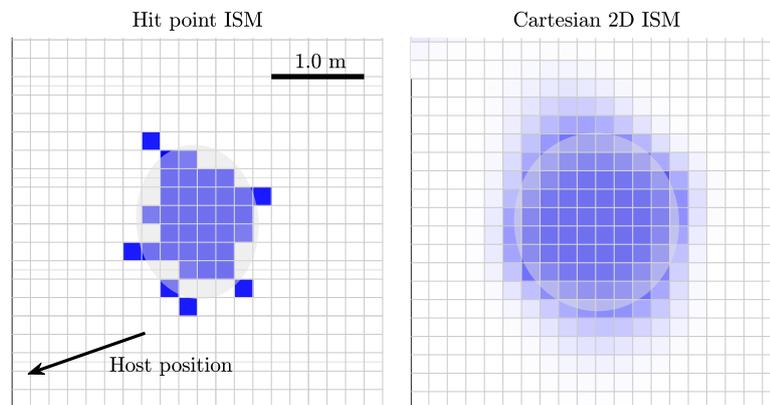


Figure 6.7. Grid snapshots of the sample pole representation with different types of ISM and uncertainty levels equal to $\sigma_r = 0.25$ m, $\sigma_\varphi = 0.3^\circ$. Grid is generated using DSmH fusion, decay and free space filtering are disabled.

The hit point snapshot is particularly important for the analysis, as it represents the baseline noised occupancy grid level without any filtering except fusion. An example snapshot of the pole representation shows a lot of irregularities (Figure 6.7) which corresponds to an average consistency of 0.69. The mean area of this raw representation is equal to 0.5 m². All further filtering assessments presented in this section try to improve this initial pole representation.

The Cartesian 2D ISM increases the area of the pole on the occupancy grid which is visible on both snapshot image 6.7 and on the violin plot 6.8(a). Nevertheless, the main advantage of the Cartesian 2D ISM over the hit point occupancy modeling is that it significantly improves the consistency value for almost all assessed poles.

More complex modeling utilized for the two-dimensional distribution can encode more information in cell occupancy probabilities and may yield better results after further filtering steps. Based on that observation, for the next parameter's evaluation, the 2D probability model was selected.

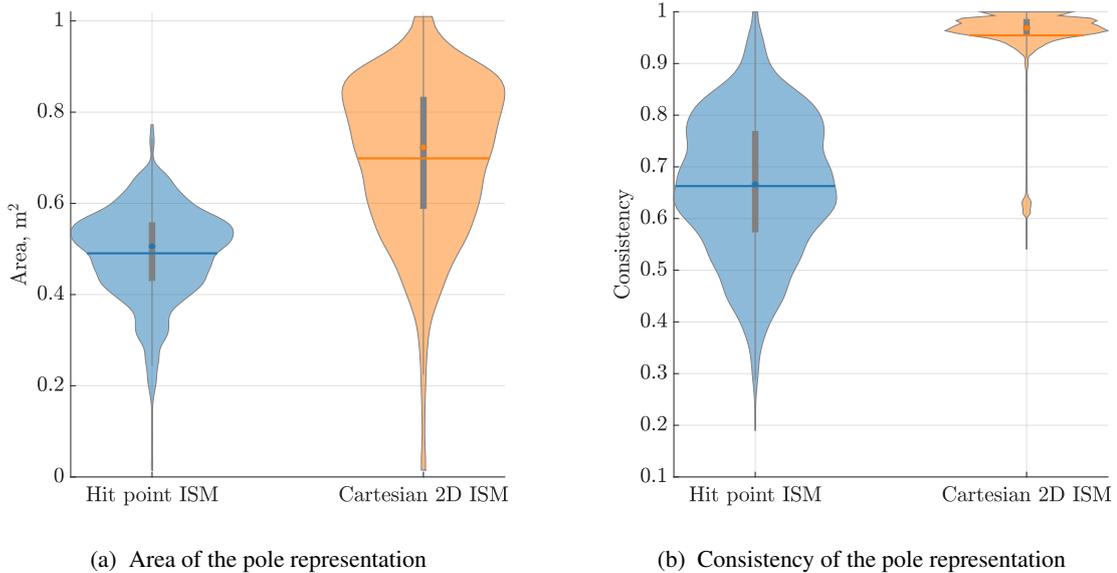


Figure 6.8. Distribution of (a) Area and Consistency (b) values depending on the type of inverse sensor model applied.

6.3.2 Decay filtering

The simulations for the second evaluated filtering method (decay) were performed similarly to the previous ones. The decay factor is driven by the decay rate ($\frac{1}{\lambda}$) value as described in Section 5.3.3. It is estimated that the larger the rate $\frac{1}{\lambda}$, the faster the decay, which decreases the probability.

The results for different decay factors show some interesting traits (see Figure 6.9). As the decay speeds up, the area of the pole representation decrease (Figure 6.10). Decay, however, was only applied to decrease fusion overconfidence, and higher values may result in object placement instability. This could be observed by measuring the consistency of the pole representation, which begin to slowly decrease as the decay rate goes up (Figure 6.10).

Decay filtering can easily counteract the inflation of landmarks' representation of areas caused by the usage of Cartesian 2D ISM. Even the smallest presented decay rate of $\frac{1}{\lambda} = 0.5 \text{ s}^{-1}$ can decrease the average area of the initial levels of the baseline grid, at the same time preserving the high consistency of the image.

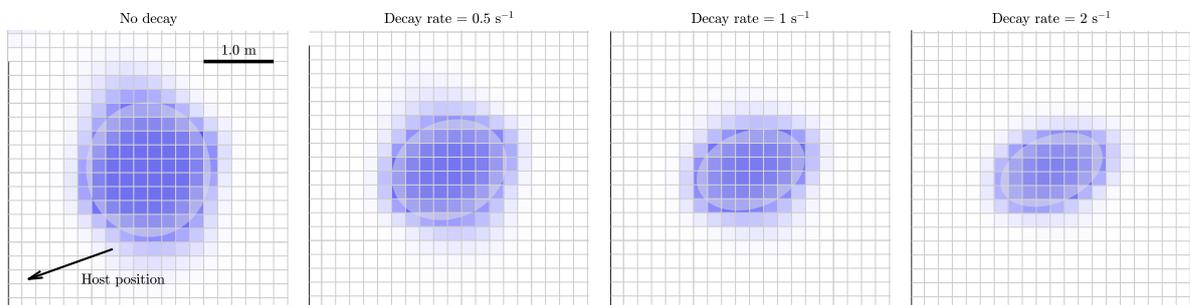


Figure 6.9. Grid snapshots of the sample pole representation with different decay rates and uncertainty levels equal to $\sigma_r = 0.25 \text{ m}$, $\sigma_\varphi = 0.3^\circ$. Grid is generated using DS_MH fusion and with Cartesian 2D ISM.

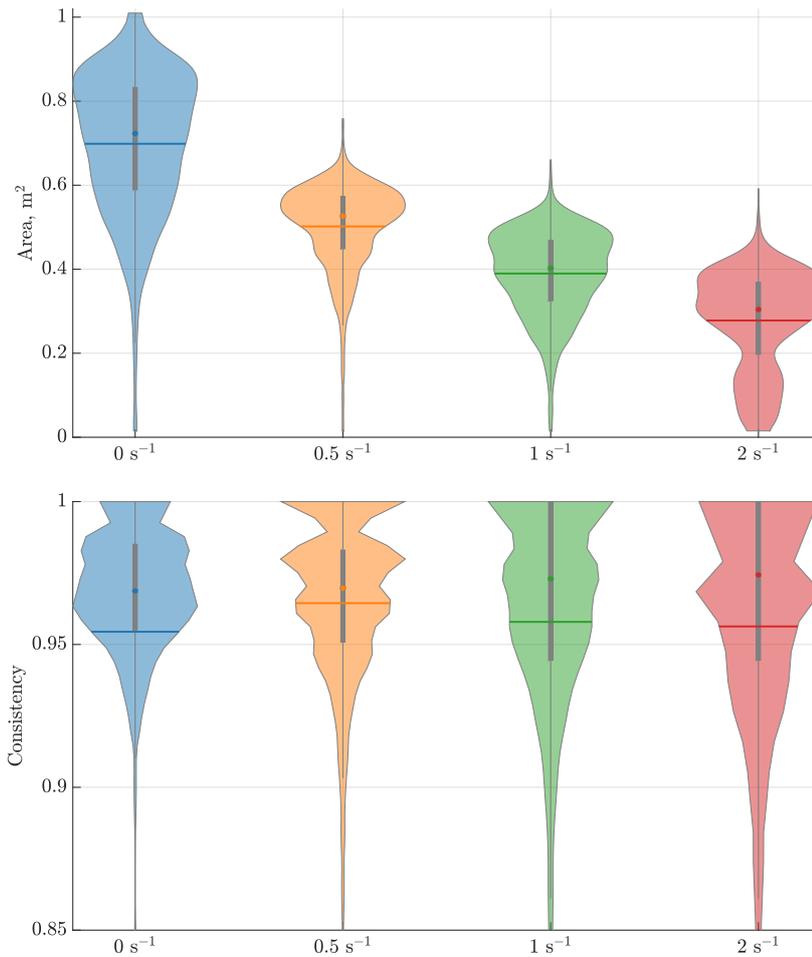


Figure 6.10. Area and consistency distributions of landmark representation generated with different decay rates.

6.3.3 Free space modeling

Free space modeling is intended to eliminate mostly false positives and sharpen the vehicle's immediate surroundings. The free space gain factor is defined as the maximum value of the free space probability assigned to cells updated in the free space determination process (see Section 5.3.2 for details).

The simulation of free space was conducted similarly to the decay influence evaluation with DS_mH combination rule for fusion and Cartesian inverse sensor model for occupancy estimation. The free space modeling provides a second type of evidences for fusion. As the free space evidence, described as the free space gain, goes up, more rays sweep nearby the stationary evidence reported for the pole. This new contradicting evidence increases cell state uncertainty measurement, commonly known as sensor conflict.

The sample pole representations presented in Figure 6.12 present how the landmark area shrinks with the increasing influence of free space.

Increasing the free space gain slowly decreases the estimated area of the object, but it increases the maximum entropy measurement for the poles' representations. The consistency of the landmark is unaffected by the free space gain as presented in the violin distribution plots in Figure 6.11. It is worth noting that cells closer to the vehicle are more likely to be affected by the free space modeling, therefore, some landmarks are filtered faster than the others. It can be observed in the area distribution plot (Figure 6.11), where for free space gains of 5% and 10% the two modalities are emerging.

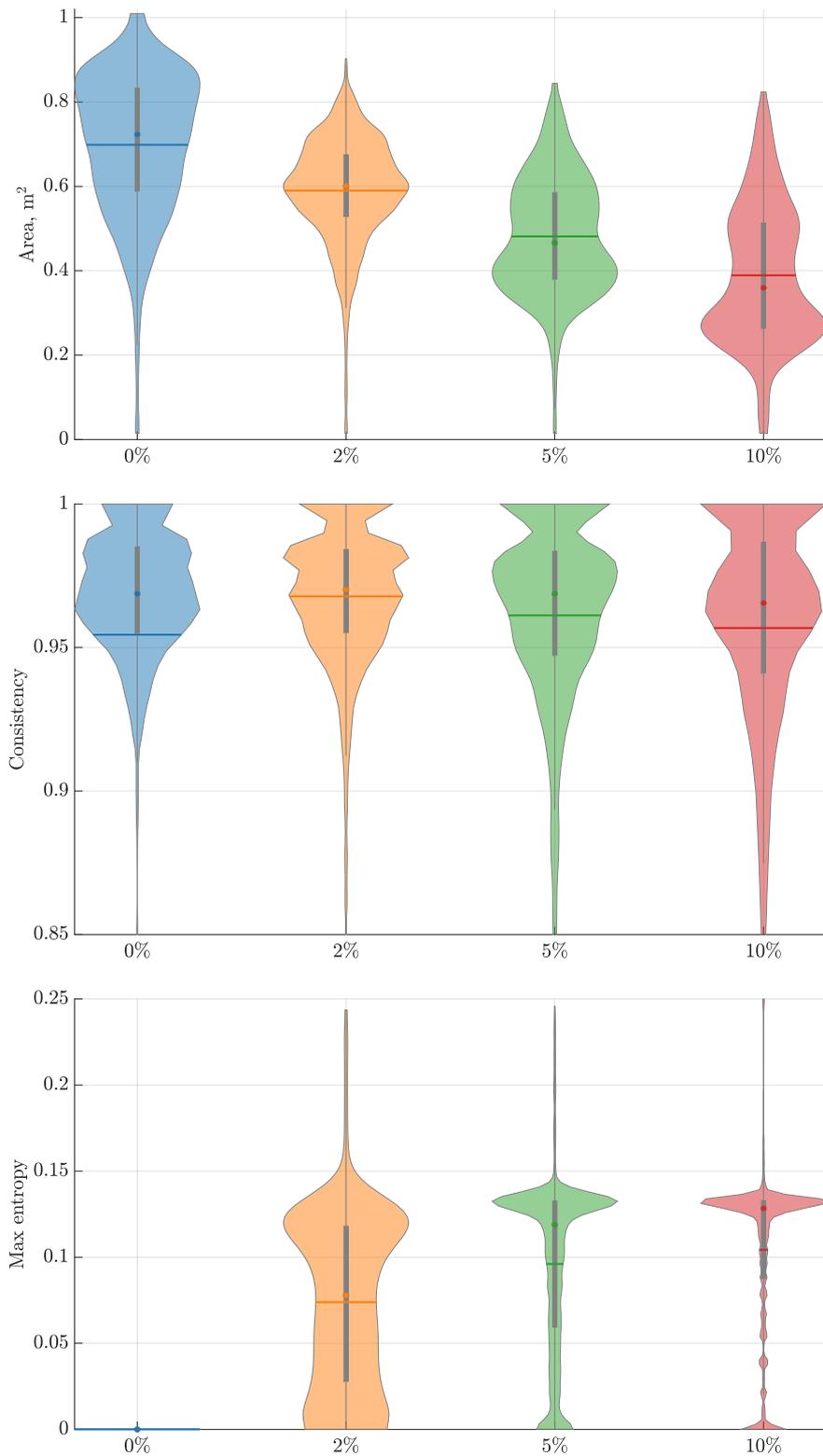


Figure 6.11. Area and entropy distributions of landmark representation generated with different free space gains.

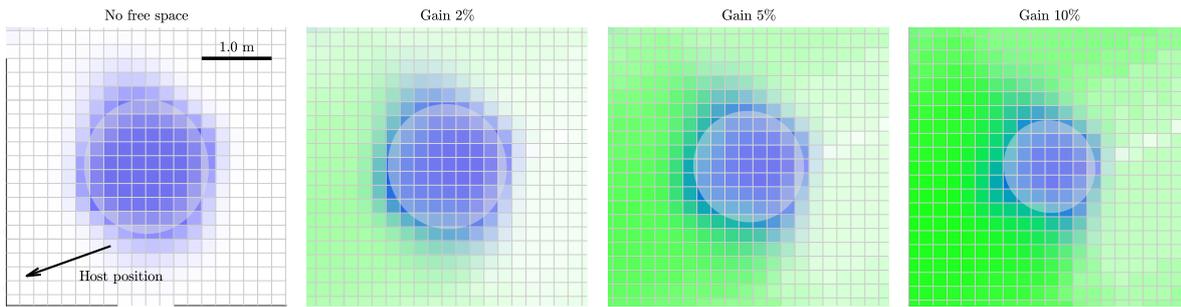


Figure 6.12. Grid snapshots of the sample pole representation with different free space gains and uncertainty levels equal to $\sigma_r = 0.25$ m, $\sigma_\varphi = 0.3^\circ$. Grid is generated using DSmH fusion and with Cartesian 2D ISM.

6.3.4 Fusion rules comparison

The fourth filtering method of the occupancy grid is fusion. The three evaluated fusion rules are Bayesian, Dempster and Hybrid Dezert-Smarandache (DSmH) methods, which are described in Chapter 4. Fusion rules cannot address the area or object representation directly, but they can solve sensor conflicts differently and with different speeds, leading to smaller entropy of the object. In order to fully utilize the fusion capabilities, all previously mentioned filtering options are enabled for this analysis. The occupancy grids in this scenario were generated using Cartesian ISM with the decay rate of $\frac{1}{\lambda} = 1$ s⁻¹ and 5% free space gain.

The sample pole representation images are presented in Figure 6.13. For this snapshot, all objects are visually similar, however, their details and underlying cell uncertainties are significantly different.

The distribution of KPI values presented in Figure 6.14 shows that for every fusion rule the object area distributions are comparable with a slight advantage of the DSmH combination rule. The consistencies of representations are almost the same for all fusion rules. The major difference is visible in the maximum entropy value of the poles' image. Entropy values for the DSmH rule of combination are almost two times smaller than for the Dempster fusion. Bayesian grid is not estimating any uncertainty; therefore, the entropy cannot be measured by this fusion rule.

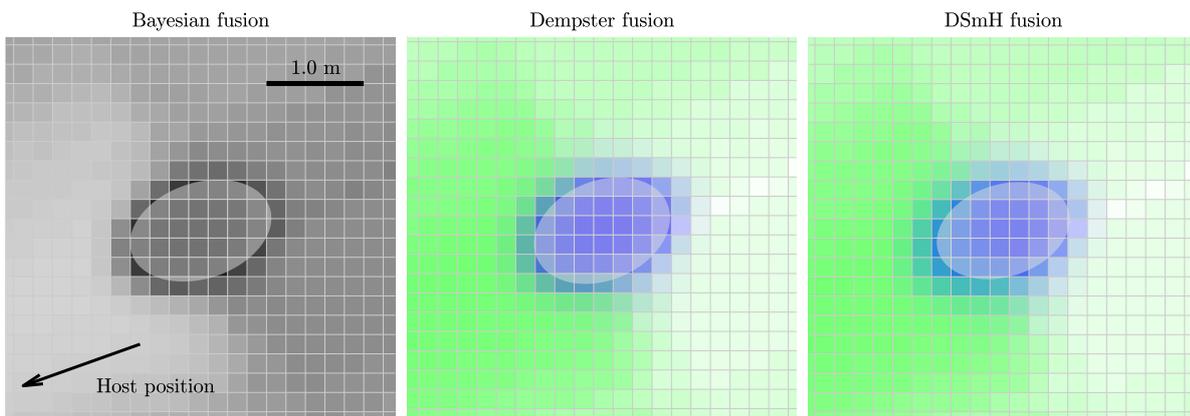


Figure 6.13. Grid snapshots of the sample pole representation with different combination rules and uncertainty levels equal to $\sigma_r = 0.25$ m, $\sigma_\varphi = 0.3^\circ$. Grid is generated using Cartesian ISM with the decay rate of 1 s⁻¹ and 5% free space gain.

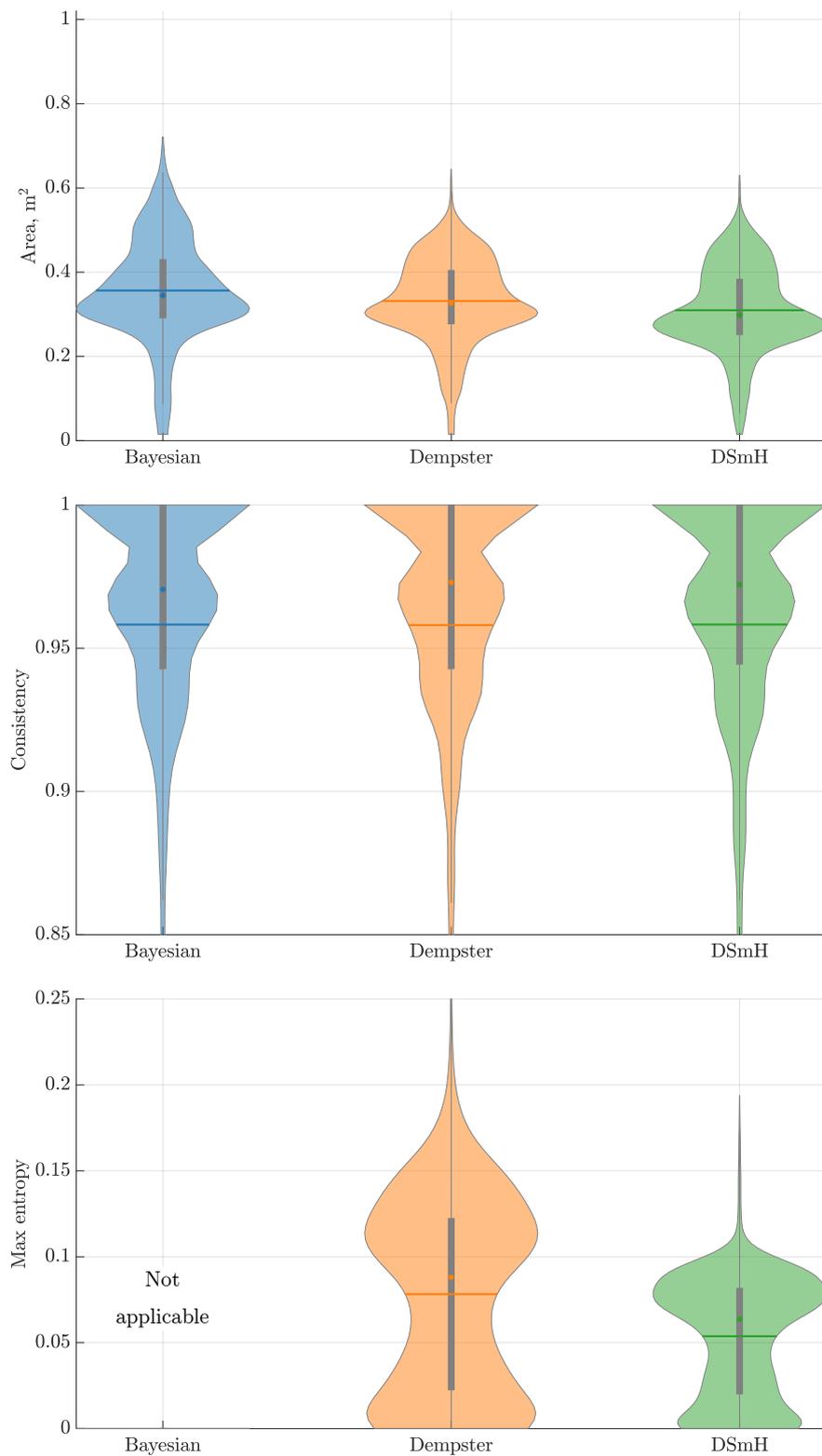


Figure 6.14. Area, consistency, and entropy distributions of landmark representation generated with different fusion rules. The occupancy grid used Cartesian ISM with the decay rate of 1 s^{-1} and 5% free space gain.

6.3.5 Combination of filtering methods

The evaluation of the filtering capabilities of the occupancy grid algorithm would not be completed without measuring how the aforementioned filtering methods cooperate with each other when combined. This simulation evaluation evaluates all combinations of the two inverse sensor models, three fusion rules with free space gain and decay rate values. For clarity of the presentation, this section presents only the average KPI values for every tested combination, but a more detailed presentation with pole representation snapshots is presented in Appendix B.

For every type of fusion and ISM, a mesh of free space gains and decay rates was tested and evaluated. The result mean KPIs values are presented in the form of contour plots describing: mean area of the pole representation – Figure 6.15, consistency of the object – Figure 6.16 and its maximum entropy – Figure 6.17.

Looking at the combination plots, the impact of the inverse sensor model type is clearly visible on the mean consistency plots in Figure 6.16. For all tested fusion rules, free space gains and decay rates, only ISM type changes the solidity of the pole representation. For the hit point ISM, the average consistency value fluctuates around 0.7, while for the 2D ISM it reaches 0.96. Application of the Cartesian ISM artificially spreads the evidence values on the occupancy grid, therefore smoothing out any uncertainties present in the input measurements. On the downside, the application of the 2D ISM increases the mean area of the occupancy grid.

For the mean pole representation area, the influence of decay and free space gain adds up easily, limiting the pole object area to less than 0.3 m^2 . Furthermore, for the 2D ISM with Bayesian and DSmH fusion rules, the synergy effect between free space and decay amplifies, and the area decrease effect is bigger than for the decay and free space itself (Figure 6.15).

On the maximum entropy graphs (Figure 6.17) the DSmH fusion rule delivers approximately two times smaller cell state uncertainty measures than the Dempster rule of combination. As described in previous sections, free space gain increases entropy levels by introducing contradicting evidences. Nevertheless, the decay may also help in decreasing the maximum entropy rates of the objects' representation.

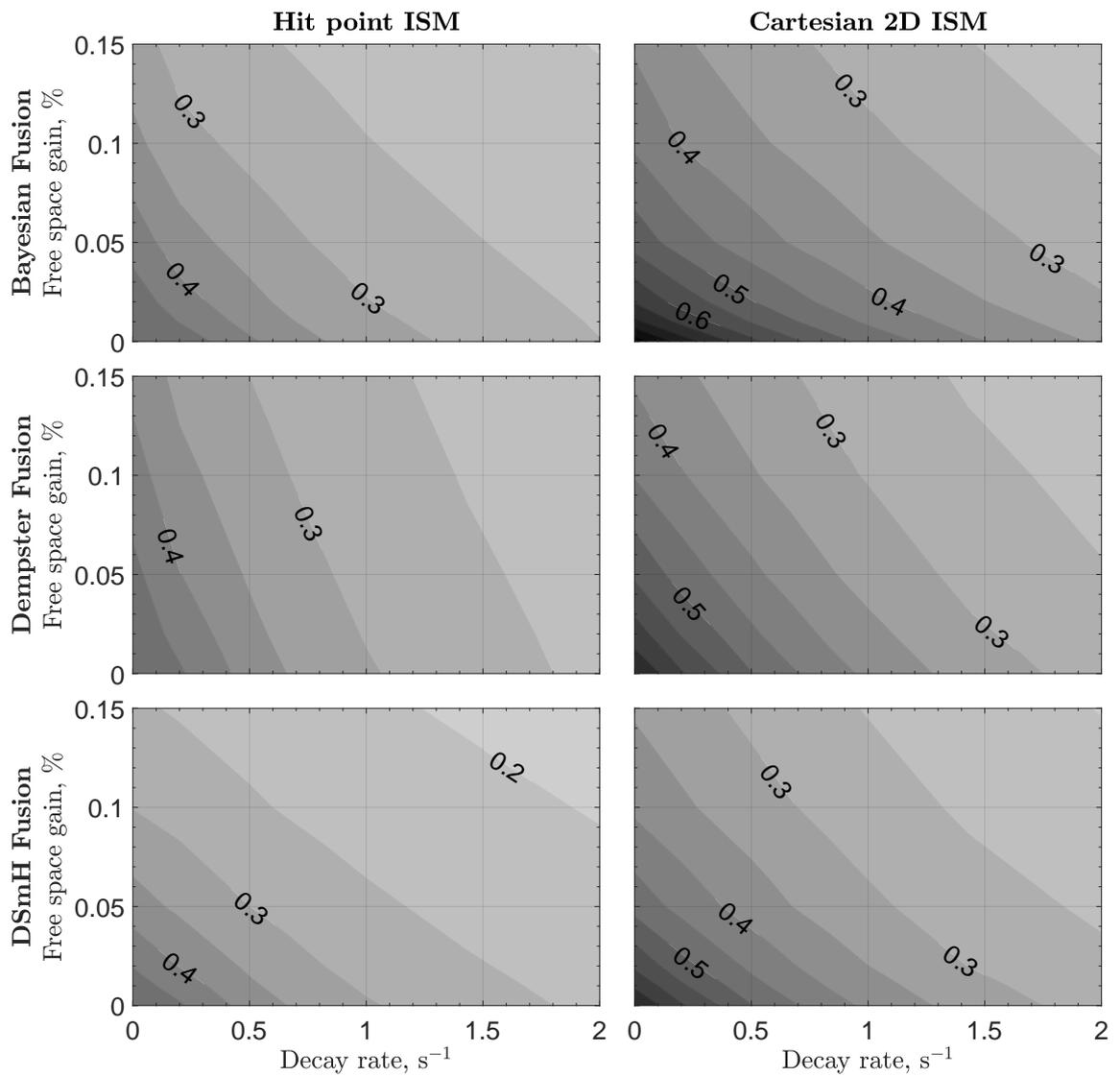


Figure 6.15. Mean area of the pole representation with different combination rules. Light gray areas represent better pole representation.

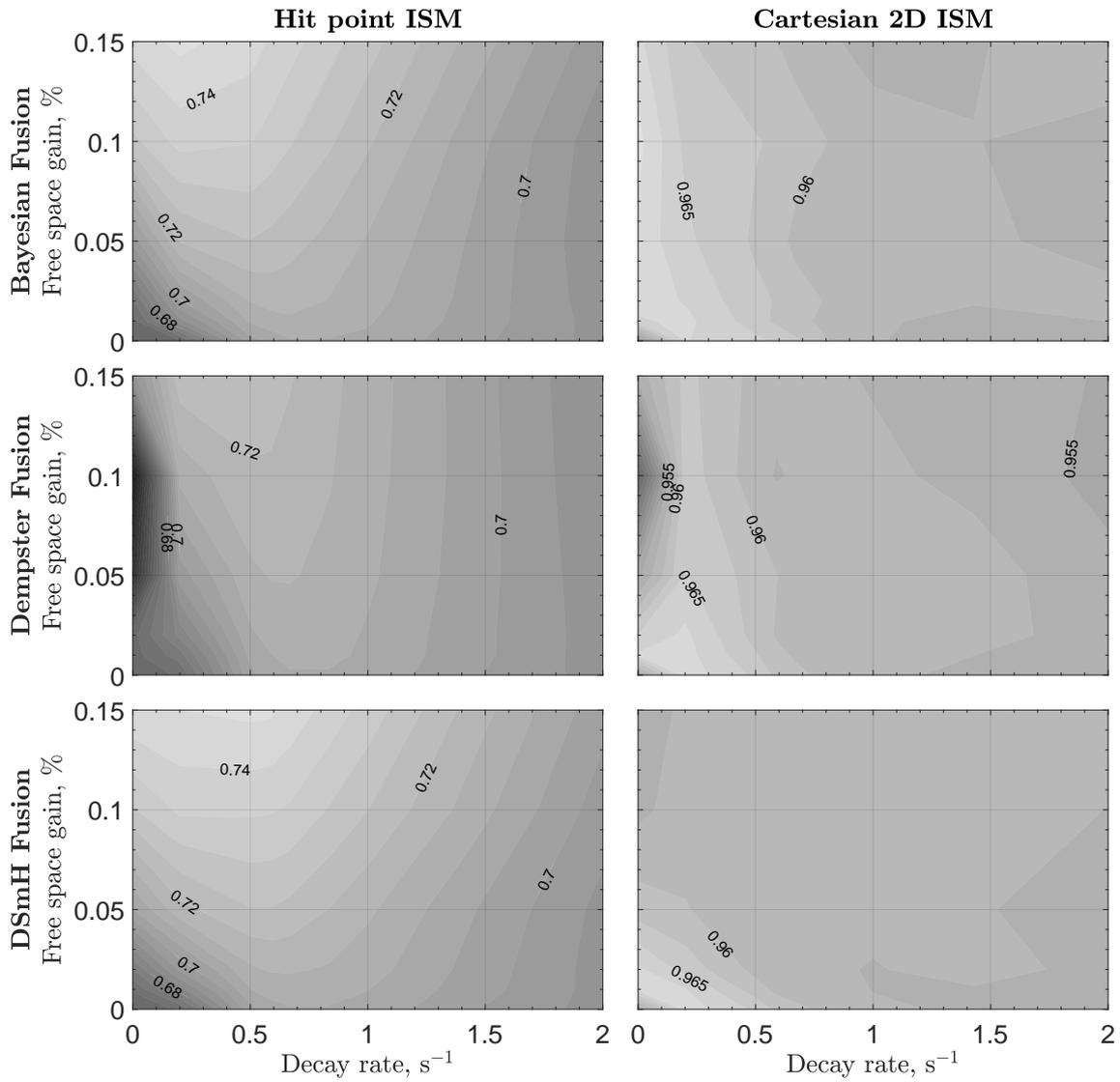


Figure 6.16. Consistency of the pole representation with different combination rules. Light gray areas represent better pole consistency, but the average consistency level for the hit point ISM is 0.7, while in the Cartesian 2D ISM it oscillates around 0.96.

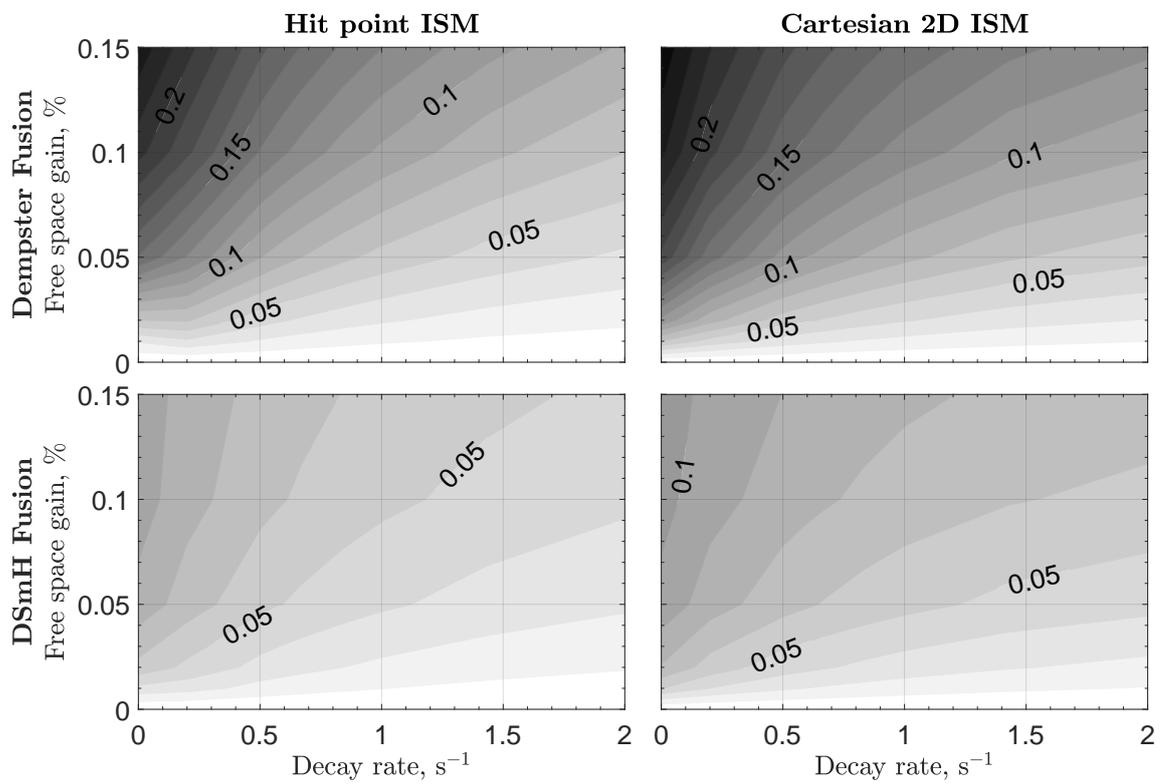


Figure 6.17. Distribution of each pole maximum entropy with different combination rules. White areas of the plot represent configurations with the lowest entropy (minimal sensor conflict).

6.3.6 Grid uncertainty filtering conclusions

The occupancy grid algorithm offers four main ways to decrease uncertainty levels in the output map: sensor modeling, decay, free space estimation, and data fusion rule. All these procedures can affect the output grid simultaneously, and their effects can interfere with each other.

Sensor modeling in the occupancy grid mostly affects the consistency of the objects' representation. Application of even small uncertainty spread on the input sensor detections improves this factor significantly. However, if a sensor is accurate or the number of detections is high enough to produce consistent objects, sensor modeling should be disabled, because it might artificially extend object area (see Figure 6.8).

From the quality factors' perspective, decay and free space modeling behave similarly to each other. Increasing the value of the decay or free space gain parameter decreases the object's representation area. These methods can be coupled with each other and their effects will add up affecting the grid quality even more. Nevertheless, decay and free space modeling should not be abused as they limit the accumulation capabilities of the algorithm. Too big free space gain or too fast decay leads to an increase in the entropy of the occupancy grid and may result in instability of the object distribution.

The choice of the fusion rule for the algorithm has no significant effect the visible area of the object representation. The combination rule, however, can distribute sensor conflicts differently which result in varying levels of entropy measurement. On the other hand, more sophisticated fusion rules tend to be computationally more complex. The comparison of fusion rules is continued in the next chapter of this dissertation.

Eventually, each filtering method in the occupancy grid algorithm has some drawbacks and the parameter selection for these methods is relative and depends on many conditions. In order to provide tuning guidelines for the occupancy grid algorithm, this work focused on applying the algorithm in an experimental environment, and the results are presented in the next chapter of this thesis.

7 Experimental evaluation

This chapter presents the experiments that have been carried out in order to evaluate the proposed implementation of the occupancy grid algorithm based on the actual sensor data. The structure of this chapter is as follows: the vehicle and test drive setup are described in Section 7.1. Section 7.2 presents guidelines how to tune radar-based occupancy grids. Real examples of different sensor conflicts are presented in Section 7.3 and Section 7.4 depict differences between the three proposed fusion rules for the occupancy grid. The following Section 7.5 describes the implementation details of the presented algorithm and its real-time computation capabilities and finally Section 7.6 shows how the presented architecture integrates with heterogeneous sensor measurement sources.

7.1 Experimental setup

The experimental setup consists of a test vehicle mounted with four corner radars, a vehicle positioning system, and a reference camera. A photo of the vehicle is presented in Figure 7.1. Selected experimental data samples are collected during test drives on motorway and semi-urban roads in the usual traffic conditions and good weather.



Figure 7.1. Experimental vehicle Peugeot 3008 used to test capabilities of presented algorithm. Vehicle is equipped with four corner radars and reference camera system. High resolution LIDAR visible on the roof was disabled for presented experiments.

The occupancy grid experiments use four corner radars with a 150° field of view and a maximum range of 100 m mounted on the vehicle in a way which covers the whole area around the host as presented in Figure 7.2.

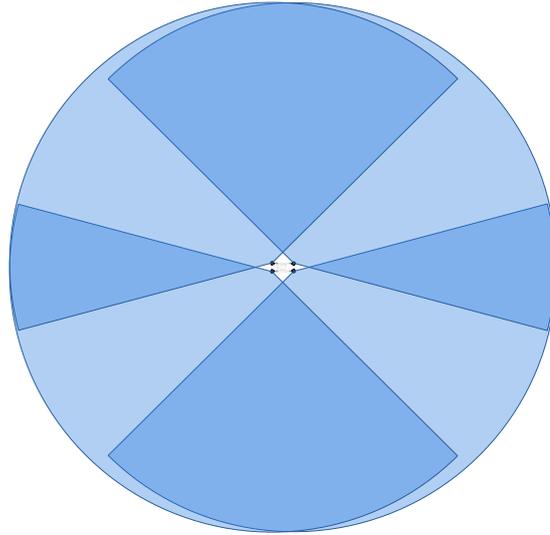


Figure 7.2. Sensor coverage of the test vehicle. For readability, only half of the radar range is presented. Plotted coverage is up to 50 m, while actual radars can detect objects up to 100 m.

All radar sensors used are automotive grade frequency-modulated continuous-wave radar sensors operating in a millimeter-wave bandwidth. Radars available in the commercial automotive market operate with around 50 ms update rates and produce on average 60–150 detections (point targets) from the sensed field of view [Autonomous Stuff, 2021].

Each of the utilized radar sensors can be configured independently in order to match the required application, however, for the simplicity of the experimental evaluation, all radars are treated uniformly with the same calibration for every sensor.

Apart from sensor data, the occupancy grid module requires a reliable positioning system which is able to deliver accurate and actual changes of the vehicle position. For the experimental evaluation, the motion and position of the ego are obtained using a high precision GPS and an IMU unit, which uses an online correction method to maintain its accuracy. The positioning data are available with the 100 Hz frequency.

The experimental vehicle is also equipped with a front reference camera used to record the test drive. Data from the camera are used only for visualization of the test results, the information provided by the recorded video does not influence in any way generation of the presented maps.

7.2 Example occupancy grid tuning procedure

The occupancy grid algorithm with a wide range of filtering methods, fusion rules, and sensor model might be quite complicated to tune in order to achieve good quality results. In this section, the developed guidelines for tuning the occupancy grid algorithm are presented.

The presented tuning procedure focuses on the quality of the signs and poles representations. For better visualization, the tuning procedure will be based on analysis of the two signs presented on the foreground of the camera snapshot displayed in Figure 7.3(a).

7.2.1 Selection of proper grid dimensions

The tuning of the occupancy grids starts from the selection of the general grid parameters such as map size, cell resolution, and vehicle placement. These requirements are usually well defined by the



(a) Front camera frame presenting two signs on the edge of motorway. The third sign visible in the background will be used at the end of this section.

(b) Occupancy grid representation of the two poles with hit point ISM and DSuH fusion rule. Decay and free space filtering are disabled.

Figure 7.3. Presentation of the two motorway's signs used for the grid tuning guidelines. Image (a) presents the real photo of the signs, and (b) shows the raw occupancy grid image of the poles without any filtering.

downstream components, which for example, might require constant monitoring of 100 m area in front of the vehicle. For this tuning scenario, the baseline occupancy grid parameters will be defined as the same as in the simulation evaluation presented in Section 6.1. Each cell of the occupancy grid has 20 cm resolution and the total grid size is 100x100 m. The host position on the grid is adjusted to focus on the front area of the grid by placing the vehicle on the circle with radius of 45 m from the grid center. This host placement ensures that at least 95 m corridor in front of the vehicle is mapped by the occupancy grid. The occupancy grid algorithm is executed every 50 ms therefore it operates with 20 Hz frequency.

This configuration of the occupancy grid area ensures that even with a host speed of $25 \frac{\text{m}}{\text{s}}$ the occupancy grid will accumulate detections for at least 4 s of movement, when each radar sensor can deliver around 80 measurement scans. Considering the overlapping sensors fields of view, the pole representations presented in this chapter are accumulated 100 – 150 independent sensor detections.

7.2.2 Selection of inverse sensor model parameters

After the selection of the general occupancy grid parameters, following this step, it to observe the occupancy grid with disabled filtering capabilities of the algorithm. This step allows to sanity check if the whole system is operating correctly and if all sensors are delivering a constant detection stream. This kind of raw occupancy grid for the analyzed signs is presented in Figure 7.3(b).

In the same manner as in the simulation evaluation in Chapter 6 raw occupancy grid is a baseline; all the following filtering capabilities of the algorithm should improve the selected object representation from the Figure 7.3(b).

The first filtering parameter to tune is the uncertainty level of the inverse sensor model. Unlike in simulation, the uncertainty levels are unknown in the experimental environment. The immeasurable uncertainty such as latency or transportation delay has to be approximated by the inverse sensor model of the occupancy grid. In order to measure the overall uncertainty of the occupancy grid which should be applied for the ISM, the recorded experimental sensor data are resimulated for different radial and azimuth standard deviations. Snapshots of the pole representations for the variable detection uncertainties of the Cartesian 2D ISM are present in Figure 7.4.

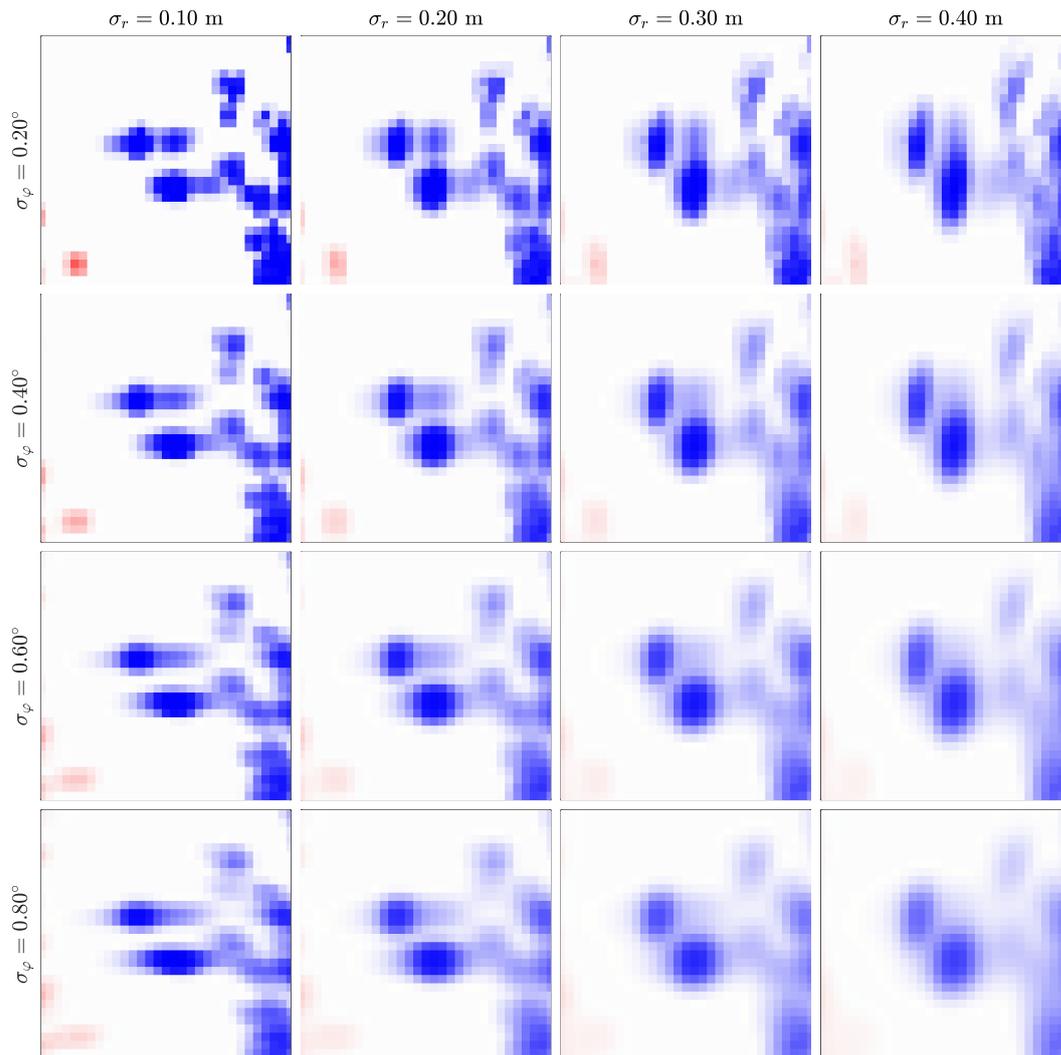


Figure 7.4. Example sings grid representations produced using different ISM uncertainty levels. For further processing the values of $\sigma_r = 0.2$ m, $\sigma_\varphi = 0.4^\circ$ were selected.

As identified in the simulation Chapter 6 the inverse sensor model increases the area of the object representation on the grid, but it is an ideal tool to deliver a highly consistent image of the environment with solid objects. The approximated values of the uncertainties used for ISM modeling should be the smallest of the values which already produce a smooth representation. This condition limits the inflation of signs' areas and significantly improves their solidity. Based on the results presented in Figure 7.4 uncertainty values of $\sigma_r = 0.2$ m and $\sigma_\varphi = 0.4^\circ$ are selected for further analysis.

7.2.3 Tuning of decay and free space modeling filters

After setting up the ISM to produce consistent grid images, the decay and free space modeling can be enabled to sharpen the grid representation of the objects. The procedure of tuning the decay rate and free space gain can be similar to the ISM uncertainty approximation. For the recorded scenario, options with different values can be tested, and the output grid could be analyzed.

Figure 7.5 present the pole representations generated using different filtering options for decay rate and free space modeling gain. In this step, both free space and decay filter the uncertainties and decrease the area of signs' representations. As described in the simulation assessment (Section 6.3.5) combination of both free space modeling and decay filtering might yield better results than each of these methods alone. This leads to a situation, where different calibrations can produce similar grid qualities. For choosing the filtering option in this step, it is best to balance the impact of decay and free space. For example, the results of resimulation with decay rate 1.0 s^{-1} and free space gain of 2% are almost indistinguishable from the sample with decay rate 2.0 s^{-1} and free space gain of 5%. In this case, the more balanced sample is the one with a decay rate 1.0 s^{-1} and free space gain of 2%; these parameters are selected for further analysis.

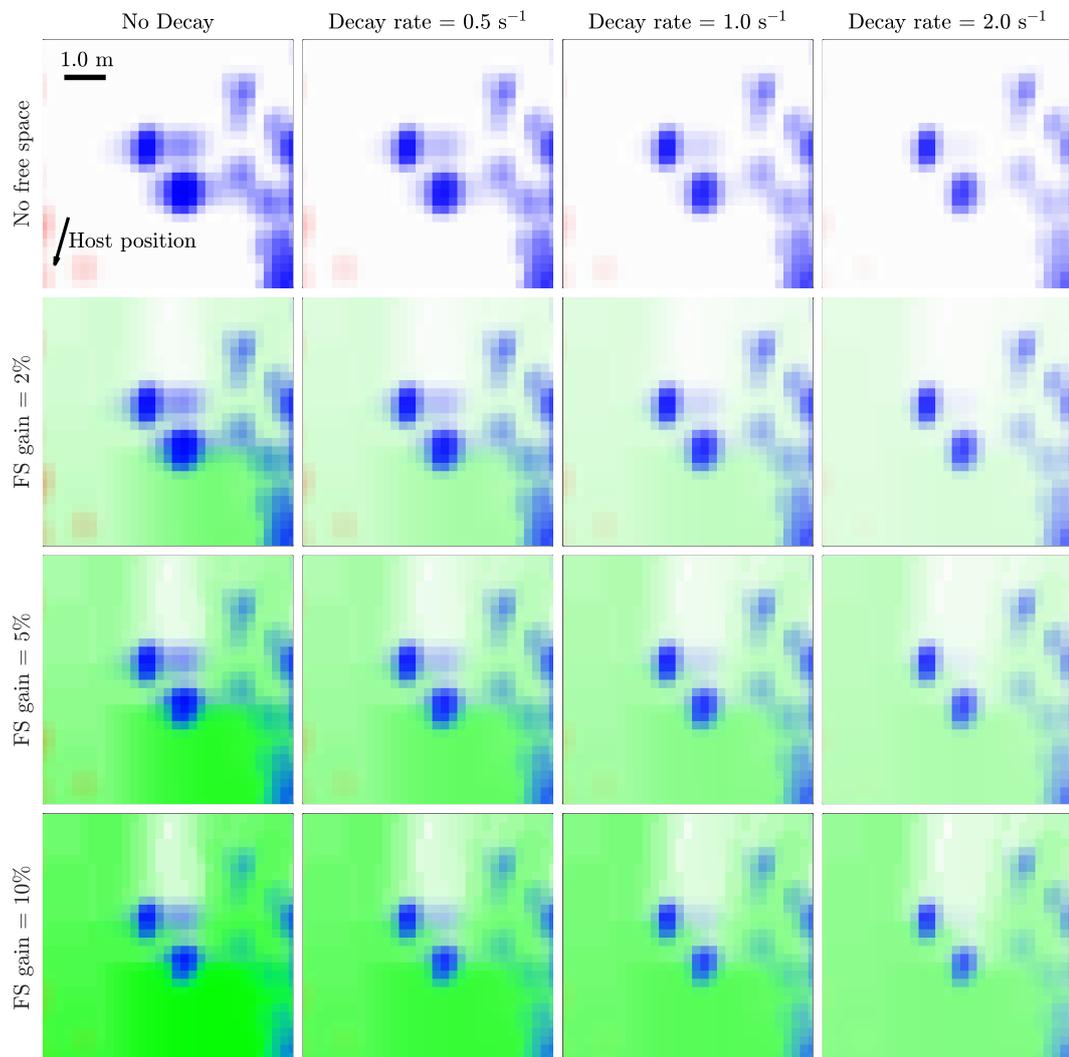


Figure 7.5. Comparison of different filtering options for the pole object. Grid snapshots were generated using DS_mH fusion rule and Cartesian 2D ISM with uncertainties $\sigma_r = 0.2 \text{ m}$, $\sigma_\varphi = 0.4^\circ$. From the evaluation, a sample with decay rate 1.0 s^{-1} and free space gain of 2% was selected for further analysis.

7.2.4 Occupancy grid before and after tuning

Proper selection of the general grid calibration, ISM uncertainties, decay, and filtering levels does not conclude the tuning procedure of the occupancy grid. Nevertheless, further parameters' selection can be based on the initial approximations delivered by the presented guidelines.

The presented tuning procedure is based on the analysis of a grid representation of the two signs. After tuning the occupancy grid parameters, it is required to verify if other landmarks and the whole occupancy grid preserve the tuned capabilities.

On the background of the camera image presented at the beginning of this section (Figure 7.3(a)) there is a third sign indicating a nearby motorway exit. Comparison of the grid representation for this sign with the raw and tuned occupancy grid is presented in Figure 7.6. Tuning the occupancy grid allows to clearly distinguish the sign (in the middle) from the vegetation (three smaller objects nearby).

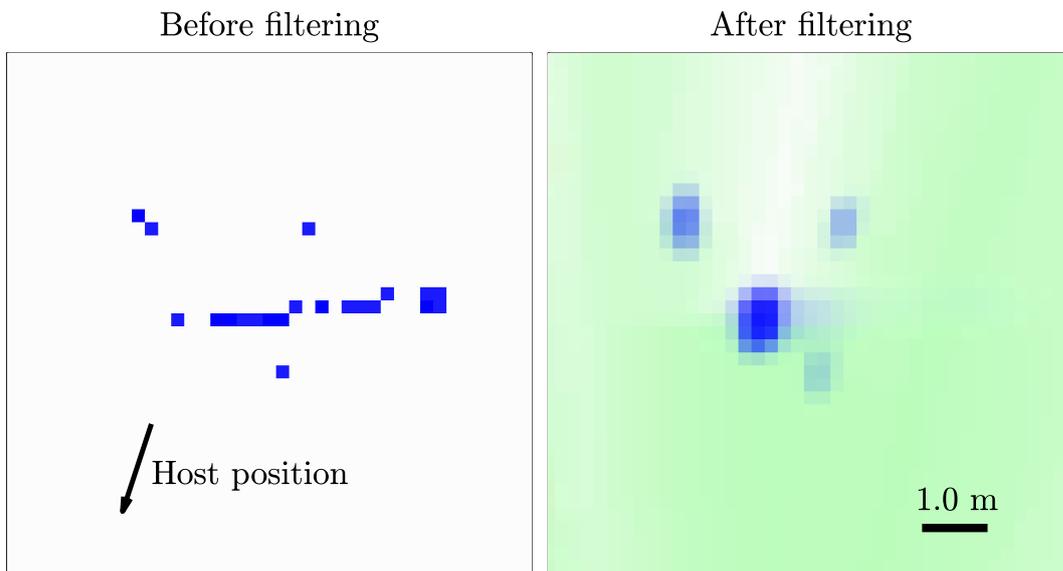


Figure 7.6. Presentation of the grid representation of the sign before and after tuning. Visualized sign is the third sign in the video frame presented in Figure 7.3(a). Presented grid snapshots depict the sign as it looks like closer to the vehicle (100 ms after the camera frame).

The overall quality of the occupancy grid is also improved after parameters' tuning as presented in Figure 7.7. The dynamic residuals (red) from other road users are effectively removed by free space modeling and decay. Application of the 2D ISM solidified the vegetation behind the road edge and cleaned up the guardrail occupancy representation on the opposite side of the road.

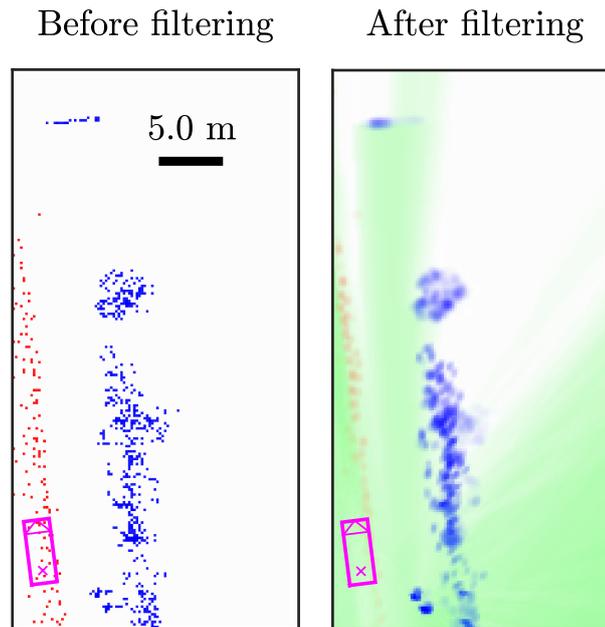


Figure 7.7. Presentation of the grid before and after tuning. All three signs evaluated in this section can be identified on the right side of the road. Red trail represents dynamic occupancy residuals.

7.3 Sensor conflict on different occupancy grids

The automotive occupancy grid has to deal with the management and combination of uncertain, imprecise, or highly conflicting sources of information. Different occupancy grid frameworks are developed to model probabilistic evidence.

The following paragraph presents a single snapshot of the highway radar occupancy grid created using the Bayesian and Dempster Shafer processing frameworks. The purpose of that example is to present the visualization capabilities of the occupancy grid and to examine how sensor conflict can be observed on the occupancy grid.

The presented scenario consists of a selected part of the occupancy grid in a highway scenario. In this example of three-lane highway, the host vehicle is on the middle lane. The right lane is empty with a bollard on the road edge, on the left lane another vehicle is overtaking the host. The resolution of the occupancy grid in both examples is 20 cm, and it is created using four radar setup. Other specific sensor setup and fusion tuning parameters are not important for this example due to its illustrative purpose.

7.3.1 Bayesian occupancy grid

Each cell in the Bayesian occupancy grid is represented as a single value, which can be visualized using colormap (4.2.17) as presented on the left in Figure 7.8.

Zoomed fragment of Figure 7.8 resents the dynamic object residual as an example of multisensory conflict. The Bayesian framework can operate only on a binary state of the cell; therefore, the dynamic object is resembled as any other occupancy object.

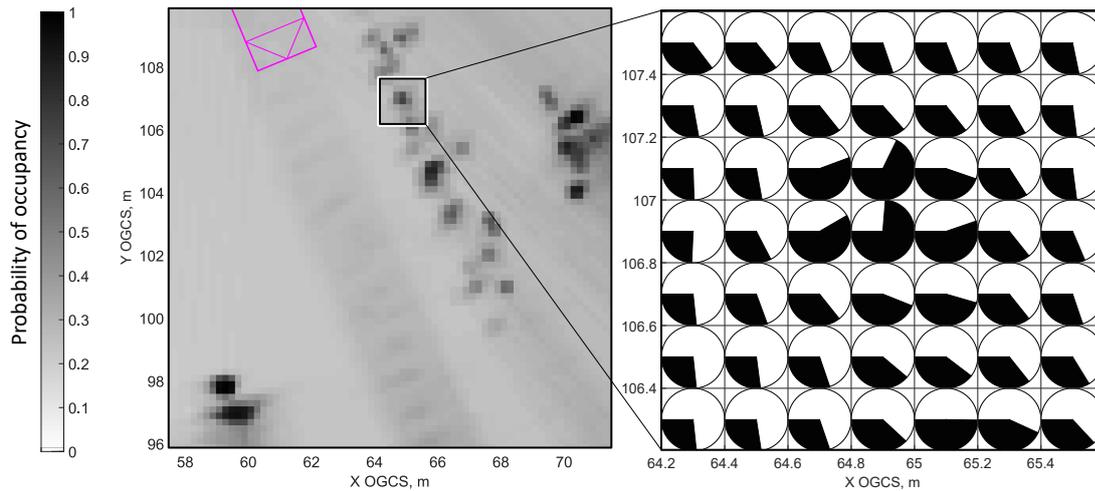


Figure 7.8. Example highway radar Bayesian occupancy grid presenting the overtaking dynamic object (zoomed area). The zoomed area presents the grid in form of a charts array.

7.3.2 Dempster-Shafer occupancy grid

Cells in the Dempster-Shafer framework are described as a 7-value BBA (the eight value is always zero ($m(\emptyset) = 0$)). Using colormap (4.2.19) the Dempster occupancy grid can be visualized as an image (Figure 7.9).

The DST is able to model sensor uncertainty and conflict. On the zoomed in example of the dynamic object residual (Figure 7.9(b)) cell BBA consists some dynamic evidence (red), but also free (green) and free or dynamic state (yellow). On the other hand, the bollard presented in Figure 7.9(a) contains mostly stationary occupancy evidence (blue) with some free space beliefs on the side of the object.

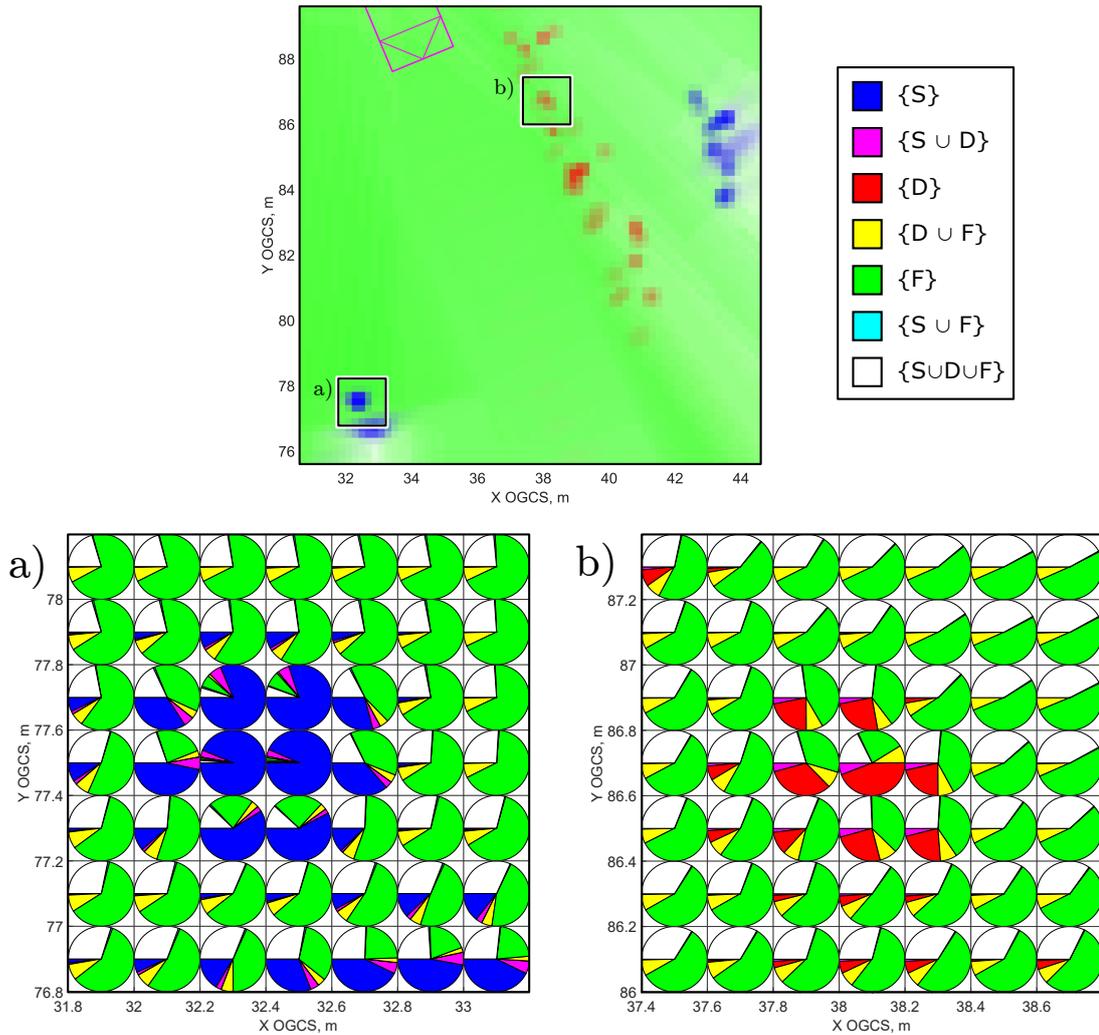


Figure 7.9. Example highway radar Dempster-Shafer occupancy grid. Top image presents the colormap grid representation. Zoomed area (a) presents grid representation of a stationary bollard. The dynamic object and its residual are visible in red in part (b) of the Figure.

7.4 Experimental evaluation of different fusion rules

Simulation evaluation of different fusion rules presents that the type of combination method does not have significant impact neither on the object's area nor consistency. Only different ways of handling contradicting sensor information differentiate the proposed three fusion rules: Bayesian, Dempster and Dezert-Smarandache Hybrid method.

Each analyzed occupancy grid fusion rule differently measures and handles sensor conflicts. On the crowded road, the test vehicle captured another road user who is attempting to turn left. As the road clears, the vehicle starts moving forward in order to perform the left turn. Camera snapshot of this situation is presented in Figure 7.10.

On the radar occupancy grid, the vehicle is represented by a group of detections which can be classified into stationary or dynamic ones. As the observed vehicle begins its turning maneuver, the sensor detections change their classification from stationary to moving. The time lapse of first 500 ms of this process is presented in Figure 7.11



Figure 7.10. Front camera video frame presenting a stopped vehicle on the left turn of a crowded crossroad. As the road clears, the vehicle starts moving forward in order to perform the left turn.

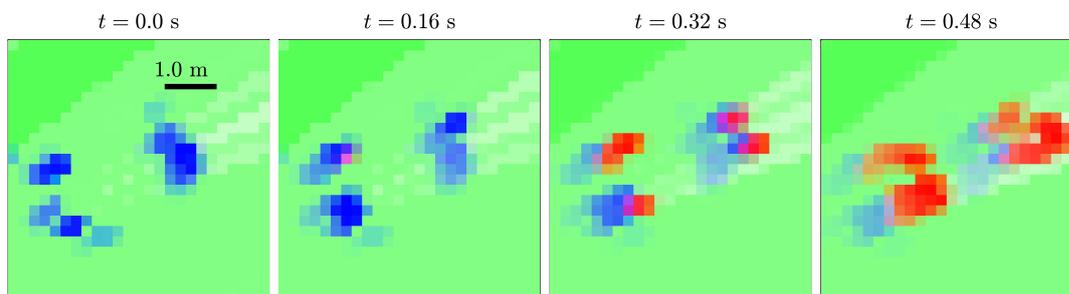


Figure 7.11. Time-lapse of the starting vehicle presented on the occupancy grid. Stationary car wheels (blue) start moving, and the vehicle will be classified as dynamic (red).

As soon as the target vehicle starts moving, the fusion on the occupancy grid has to decide if the underlying cells' states should contain stationary or dynamic evidences. Detailed look into this case is presented in Figure 7.12.

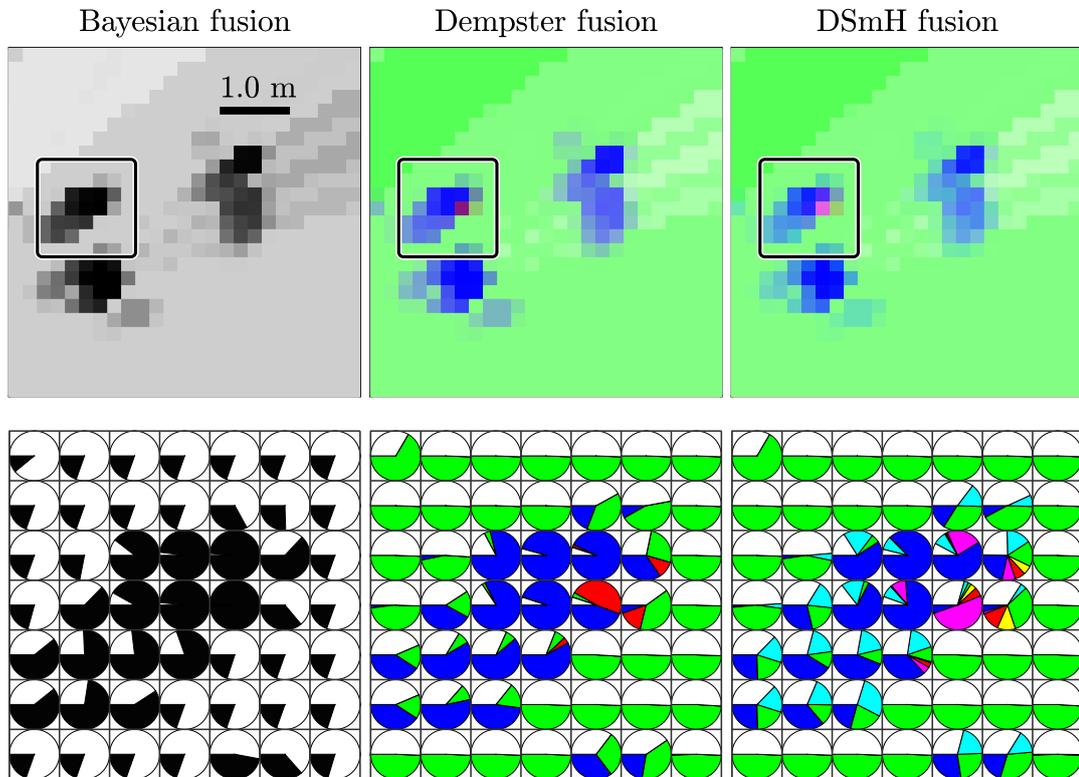


Figure 7.12. Comparison of Bayesian, Dempster and DSmH fusion rules based on the starting vehicle example.

The three grids presented in Figure 7.12 are similar to each other. The shape of the occupancy is preserved for every fusion type, but the internal cell states are different. In the middle of the left right wheel of the vehicle, there is a single cell which shows symptoms of sensor conflict.

The Bayesian occupancy grid combines stationary and dynamic evidences into a generic occupancy state. This way, sensor conflict is neglected, as the Bayesian grid relies only on the binary occupied-free cell state representation.

The Dempster fusion rule identifies the presence of sensor conflict by accumulating both stationary and dynamic evidences inside the cell. The unknown evidence, denoted as a white segment, is almost completely depleted, and as a result the cell is also fully occupied. The Dempster fusion describes occupancy for this cell as half stationary, half dynamic using elemental hypotheses' evidences. This results in the anomaly that the cell can be at the same time both static and dynamic, which is forbidden by the definition of exclusive elemental states. Future evidence fusion has to work with the contaminated cell state, which may give not expected results.

Finally, on the DSmH grid, the whole displayed part has lower confidence that the two grid predecessors. For the conflicting cells, the fusion rule converted 'stationary' evidence (blue) into 'stationary or dynamic' belief (pink). Unlike the other fusion rules, the DSmH method is able to measure the conflict value and accurately represent it, preserving the exclusivity constrains of the framework. Usage of the state 'stationary or dynamic' intuitively provides information about the conflict origin and how this type of evidence can be handled in the future.

The subtle differences between Dempster and DSmH fusion rules can be further evaluated by looking at the entropy of the occupancy grids generated for the whole scenario as presented in Figure 7.13. Visually, Dempster and DSmH fusion rules are quite similar to each other. The DSmH fusion rule, however, shows much lower entropy levels for the whole starting vehicle example.

The reason for the lower entropy of the DSmH combination rule depends directly on the different power sets used in Dempster and Dezert-Smarandache frameworks. Dempster fusion rules distribute the conflict measure uniformly to already existing evidences inside the occupancy grid cell. If the conflicting grid cell does not contain even residual 'stationary or dynamic' evidence, it cannot move the conflict mass in this hypothesis. As a result, the cell state has to rely only on elemental hypotheses, which increases the entropy levels.

On the other hand, the DSmH fusion can use a wider variety of intermediate conflicting states. This allows better redistribution of the conflict mass into conjunctive evidence of 'stationary or dynamic' hypothesis. Therefore, the proposed DSmH rule of combination is able to detect and properly resolve sensor conflicts, addressing the conflict origin.

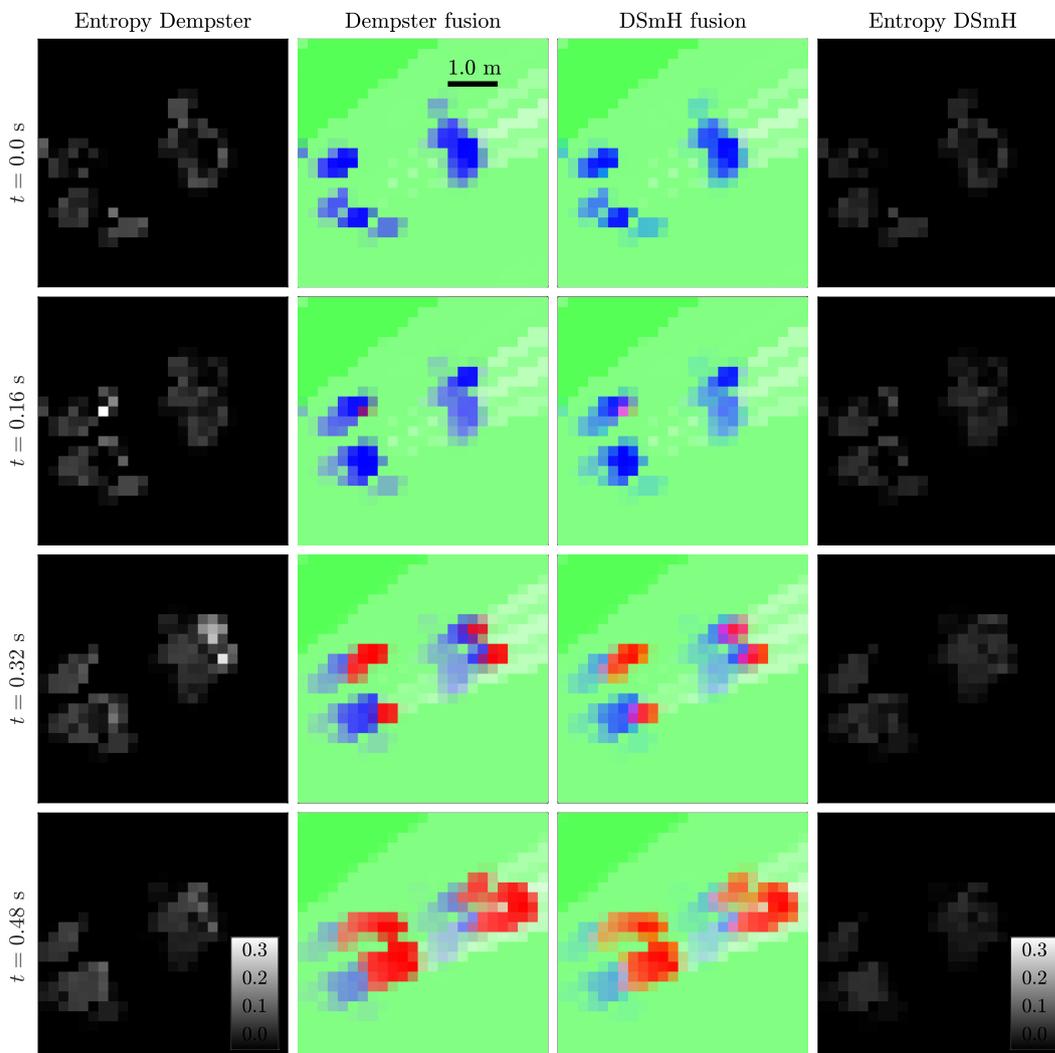


Figure 7.13. Comparison of Dempster and DSmH fusion rules based on the starting vehicle example.

7.5 Performance evaluation of the algorithm

Vehicle perception system has to operate in real-time to ensure reliability of the overall solution. The occupancy grid algorithm presented in this thesis has to deliver a new environment with 20 Hz frequency which corresponds to 50 ms refresh rate.

In order to meet these requirements, the occupancy grid algorithm is implemented in a C++ language with the help of MATLAB for prototyping and validation of the solution. The application is designed to operate on a single thread on a medium class CPU.

In the experimental vehicle, the algorithm is running in real time on a proprietary computing unit. In order to present the algorithm performance, additional tests on an easily accessible CPU were performed. For the evaluation platform, a minicomputer Raspberry Pi Model 4B was selected. It is equipped with a quad-core Cortex-A72 CPU with 1.5GHz's frequency on each core.

Performance evaluation tests were performed using a resimulation technique. A highway scenario sensor data from four corner radars and a positioning system was collected in a vehicle and replayed as a time series of data on a Raspberry Pi computer. The occupancy grid algorithm was compiled for the ARM v8, 32-bit architecture using g++10 compiler. Performance evaluation results: execution time and memory consumption are presented in Table 7.1.

Table 7.1. Occupancy grid algorithm performance measurements for a highway scenario running on a single core of Raspberry Pi 4B computer (1.5GHz).

	Bayes	Dempster-Shafer	Dezert-Smarandache
Mean execution time, ms	11.7 ± 3.9	15.2 ± 4.35	14.6 ± 4.14
Memory consumption, MB	10.10	24.40	24.40

The evaluated software implementation was developed mainly as a prototype solution in order to showcase different occupancy grid capabilities. The application may use only one of the four available CPU cores; therefore, the computational load of the computer is capped at 25 %. Moreover, the solution is not fully optimized against performance and memory consumption and the actual production implementation of this algorithm should yield better performance metrics.

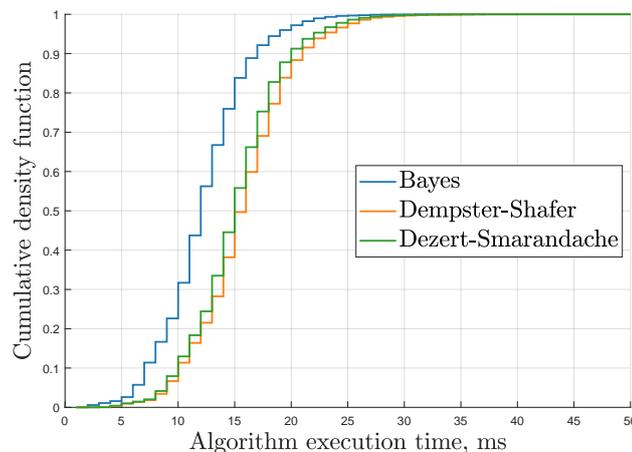


Figure 7.14. Algorithm execution profiling cumulative density function collected for different occupancy grid scans on the Raspberry Pi 4B computer.

Nevertheless, the application achieved real-time operation throughout the whole tested scenario. The worst algorithm execution time reached 40 ms from the acceptable 50 ms limit. Detailed distribution of the execution times of the algorithm is presented in Figure 7.14.

The performance evaluation utilized the same grid calibration as presented in previous sections with the 100x100 m occupancy grid with 0.2 m resolution, Cartesian 2D ISM, 1.0 s^{-1} decay rate and 2% free space gain. The only difference between the presented options was the fusion framework used to deliver the output occupancy map: Bayesian, Dempster-Shafer and Dezert-Smarandache.

Bayesian occupancy grid has the lowest execution time and memory consumption requirements from the tested grid frameworks. This is because the Bayesian cell state is represented as a single numeric parameter and the data fusion equation can be easily optimized. In the presented results, a traditional version of the Bayesian inference was used (see equation 4.3.1), which can be further optimized using the log-it version.

Memory consumption for Dempster-Shafer and Dezert-Smarandache frameworks is the same, because both of them use the same seven Basic Belief Assignments to describe the occupancy grid cell state. However, the extended cell state implied two times more memory consumption than in Bayesian map. The Dezert-Smarandache combination rule has slightly better execution time results than the Dempster-Shafer fusion. This effect can be caused by the specific implementation details of these fusion rules or by the fact that DSmH combination rule utilizes only addition and multiplication operations, while the DS combination rule uses a division operation (equations 4.3.14 and 4.3.7).

7.6 Execution on different combinations of automotive sensors

The occupancy grid architecture presented in this thesis is designed to operate with all possible combinations of automotive sensors sources. Following that, the prepared algorithm implementation can support radar, vision, free space, and LIDAR sensors. An example of the occupancy grid output generated using a typical combination of these sensor sources is presented in Figure 7.15.

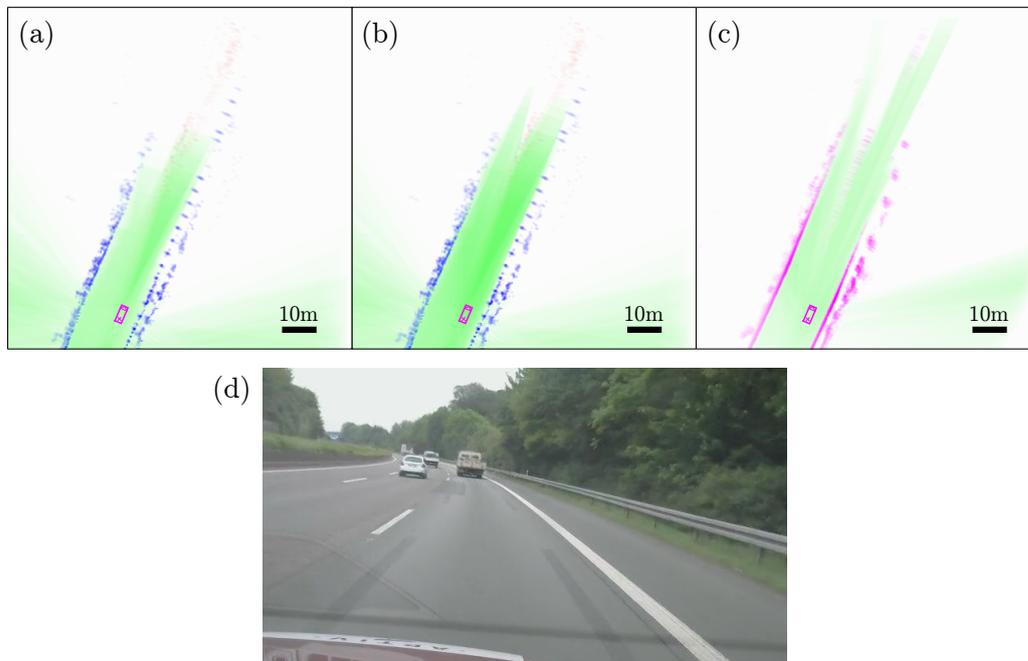


Figure 7.15. Grid output using different sensor sets: four corner radars (a), corner radars and front vision free space system (b) and LIDAR sensor (c). Reference camera image of the scene is presented at the bottom (d).

The Figure 7.15(a) presents the radar only occupancy grid. Combination of radar and vision sensors delivers a better free space estimation visible in the Figure 7.15(b). LIDAR sensor cannot differentiate stationary and dynamic detections, therefore the occupancy grid consists only of unknown occupancy evidences depicted in magenta color in Figure 7.15(c).

8 Conclusions and future work

This chapter concludes the presented manuscript. It summarizes the discussed automotive occupancy grid perception architecture, followed by a conclusion. Insights about perspectives and future works are listed thereafter.

8.1 Summary

In this manuscript, a modular and scalable architecture for the automotive occupancy grid framework is introduced. The latter enables the application of multiple seminal fusion rules, including Dezert-Smarandache rule of combination. The presented architecture is assessed against different sensor uncertainty sources. For this purpose, a new occupancy grid quality assessment method was proposed, which has been tested on both simulation and experimental environments. Respectively, the work presents an analysis of the occupancy grid filtering capabilities and how these methods interfere with each other. A grid filtering capabilities investigation allowed preparation of guidelines for tuning the algorithm for a specific vehicle setup. Lastly, the real-time and multi-sensor domain capabilities of the proposed solution are described.

8.2 Conclusions

This study aimed to investigate the capabilities of the occupancy grid framework for applications in automotive perception. Research results indicate that the occupancy grid algorithm is a versatile and reliable solution for the fusion of various automotive sensors inputs into a single representation of the environment. Customization and scalability of the algorithm allows it to be used in many vehicle setups with different automation levels.

Nowadays, car manufacturers have to offer a wide variety of vehicle options, which may differ in mounted sensors or driving comfort features. The occupancy grid algorithm delivers a single point of reference for planning and controlling downstream components and can work with almost any set of sensors. Therefore, the application of the proposed algorithm architecture allows easy and cost-effective reusability of vehicle features. Car manufacturers need to only tune the occupancy grid to the new conditions, without developing a new solution for a different vehicle option.

Secondary aim of the presented thesis is to provide a detailed description of the occupancy grid algorithm architecture, splitting it into a set of independent components. This way, the presented occupancy grid algorithm can be reimplemented from the description, enabling further experiments based on the proposed solution.

The unique part of the presented occupancy grid fusion framework architecture is its modular fusion approach. Sensor evidences are separated into four streams of independent information. Each evidence is temporarily stored on an intermediate grid, allowing better resolution of sensor conflicts during fusion.

The fusion methodology presented in this thesis allows usage of the Dezert-Smarandache fusion rule on the static-dynamic-free space occupancy grid. Dezert-Smarandache theory was invented in 2004, and

it offers better sensor conflict management, which leads to better quality of the occupancy grid maps. The thesis presents all fusion equations with their derivation in the Appendix A.

Simulations and experiments presented in this thesis deliver a better understanding of the sensor filtering capabilities of the algorithm. Based on that, the author proposes a tuning procedure with a set of key performance indicators. Described method helps to obtain a measurable mapping quality which meets the perception system requirements.

Finally, the solution proposed in this thesis has been successfully deployed in the automated vehicle, where it operates in real-time. Additional execution time evaluation shows that in the analyzed fusion rules, Bayesian inference is the fastest option, and the Dezert-Smarandache and Dempster-Shafer frameworks require more computational resources. From the presented options, the Dezert-Smarandache fusion is the best for sensor conflict redistribution. Depending on the available resources, the customer can choose the fusion method which matches requirements.

Nevertheless, the occupancy grid has some limitations. Foremost, the algorithm can produce easily visualized and understandable maps of the environment only if it is provided with meaningful sensor data. If the measurements are noised or the data quality is bad, the output will contain some irregularities and uncertain regions. Application of proper tuning of filtering capabilities can decrease the uncertainty levels, but some residuals might still be present.

Simulation findings show that the algorithm is especially vulnerable to the synchronization of sensor data. Therefore, the author suggests that the occupancy grid framework should be used only in automated vehicles, where the vehicle's communication network is synchronized, and sensors deliver measurements with correct time stamps.

8.3 Suggestions for future work

The occupancy grid framework is a well-known solution in robotics, with a lot of automotive focus in recent years. The presented thesis describes the overall architecture of the automotive solution and proposes one of the options how this algorithm can be implemented. Nevertheless, there are some research areas which can be developed further to improve the proposed algorithm.

This work presents a handful of filtering options for the occupancy grid application. This set of methods can be extended, and various new capabilities needs further testing. From the sensor modeling perspective, the ultrasonic sensor model can be adapted from the robotic occupancy grid applications [Liu et al., 2018]. The simple segment-based free space estimator can be enhanced using machine learning methodologies [Bauer et al., 2019]. Available combination rules set can be broadened with the Probability Conflict Redistribution rules (PCR) [Moras et al., 2015] and the grid decay might be computed using Hidden Markov Models.

Computational evaluation results presented in this thesis describe the overall trend of the algorithm complexity. In this case, the performance assessment results depend heavily on the development details and the available in-vehicle hardware specifications. To limit the implementation specific influence, the computational performance should be measured for each of the occupancy grid components. To further improve the algorithm performance and memory consumption, an integer implementation of the evidences' fusion should be investigated [Andriamahefa, 2017].

In the end, the dynamic occupancy determined by the proposed framework can be easily extended with a velocity estimation layer using state-of-the-art random finite set or particle filter solutions [Nuß, 2017; Steyer et al., 2018]. This solution requires a graphical processing unit (GPU), which is not available in the basic ADAS automated vehicles. However, with the cutting-edge advancements in the field of artificial intelligence and machine learning, various GPUs are suggested for the new generation of automated vehicles. This processing unit would enable further refinement of the occupancy grid solution with an ability to detect and track dynamic road users and their velocities.

A Fusion matrices and equations

This chapter presents some detailed explanations on the derivation of the fusion rules examples referred in Section 4.3. Equations presented in this chapter are described in a way that enables the implementation and deployment of those rules in combination in the automotive perception system.

A.1 Basic definitions

Based on the types of evidences which may be derived from automotive perception sensors, this thesis investigates the three-element frame of discernment:

$$\Theta = \{\{S\}, \{D\}, \{F\}\} \quad (\text{A.1.1})$$

where $\{S\}$ stands for static evidence, $\{D\}$ for dynamic occupancy and $\{F\}$ for free space. For short, this frame of reference will be called SDF frame from now on.

For the SDF frame, the elemental hypotheses $\theta_j : j \in \{1, 2, 3\}$ are defined as:

$$\theta_1 = \{S\} \quad \theta_2 = \{D\} \quad \theta_3 = \{F\} \quad (\text{A.1.2})$$

Combinations of elemental hypotheses form probabilistic propositions which are described in detail in Chapter 4, Section 4.1.2.

A.1.1 Dempster-Shafer model definition

The Dempster-Shafer power set 2^Θ of $\Theta = \{\theta_1, \theta_2, \theta_3\}$, is given by the set of the following $|2^\Theta| = 8$ irreducible propositions:

$$2^\Theta = \{\alpha_0, \alpha_1, \dots, \alpha_7\} \quad (\text{A.1.3})$$

where each $\alpha_i : i \in \{0, \dots, 7\}$ corresponds to a specific subset of hypotheses $\theta_j : j \in \{1, 2, 3\}$:

$$\begin{array}{ll} \alpha_0 := \emptyset & \alpha_4 := \theta_1 \cup \theta_2 \\ \alpha_1 := \theta_1 & \alpha_5 := \theta_1 \cup \theta_3 \\ \alpha_2 := \theta_2 & \alpha_6 := \theta_2 \cup \theta_3 \\ \alpha_3 := \theta_3 & \alpha_7 := \theta_1 \cup \theta_2 \cup \theta_3 \end{array} \quad (\text{A.1.4})$$

Applying notation from (A.1.2) the equation (A.1.4) can be rewritten as:

$$\begin{array}{ll} \alpha_0 := \emptyset & \alpha_4 := \theta_1 \cup \theta_2 = \{S\} \cup \{D\} \\ \alpha_1 := \theta_1 = \{S\} & \alpha_5 := \theta_1 \cup \theta_3 = \{S\} \cup \{F\} \\ \alpha_2 := \theta_2 = \{D\} & \alpha_6 := \theta_2 \cup \theta_3 = \{D\} \cup \{F\} \\ \alpha_3 := \theta_3 = \{F\} & \alpha_7 := \theta_1 \cup \theta_2 \cup \theta_3 = \{S\} \cup \{D\} \cup \{F\} \end{array} \quad (\text{A.1.5})$$

A.1.2 Dezert-Smarandache model definition

The Dezert-Smarandache hyper-power set D^Θ of $\Theta = \{\theta_1, \theta_2, \theta_3\}$, is given by the set of the following $|D^\Theta| = 19$ irreducible propositions:

$$D^\Theta = \{\alpha_0, \alpha_1, \dots, \alpha_{18}\} \quad (\text{A.1.6})$$

where subsets α_i are defined as:

$$\begin{aligned} \alpha_0 &:= \emptyset & \alpha_{10} &:= (\theta_2 \cap \theta_3) \cup \theta_1 \\ \alpha_1 &:= \theta_1 & \alpha_{11} &:= \theta_1 \cap \theta_2 \cap \theta_3 \\ \alpha_2 &:= \theta_2 & \alpha_{12} &:= (\theta_1 \cup \theta_2) \cap \theta_3 \\ \alpha_3 &:= \theta_3 & \alpha_{13} &:= (\theta_1 \cup \theta_3) \cap \theta_2 \\ \alpha_4 &:= \theta_1 \cup \theta_2 & \alpha_{14} &:= (\theta_2 \cup \theta_3) \cap \theta_1 \\ \alpha_5 &:= \theta_1 \cup \theta_3 & \alpha_{15} &:= \theta_1 \cap \theta_2 \\ \alpha_6 &:= \theta_2 \cup \theta_3 & \alpha_{16} &:= \theta_1 \cap \theta_3 \\ \alpha_7 &:= \theta_1 \cup \theta_2 \cup \theta_3 & \alpha_{17} &:= \theta_2 \cap \theta_3 \\ \alpha_8 &:= (\theta_1 \cap \theta_2) \cup \theta_3 & \alpha_{18} &:= (\theta_1 \cap \theta_2) \cup (\theta_1 \cap \theta_3) \cup (\theta_2 \cap \theta_3) \\ \alpha_9 &:= (\theta_1 \cap \theta_3) \cup \theta_2 \end{aligned} \quad (\text{A.1.7})$$

In order to apply the DSMT fusion rules on top of the Dempster-Shafer model, the hybrid DSMT needs to be defined with the constrain that all possible elemental hypothesis intersections are empty sets. This constrained hyper power set is defined in [Smarandache and Dezert, 2004, Chapter 4.] as:

$$\begin{aligned} \alpha_0 &:= \emptyset & \alpha_{10} &:= (\theta_2 \cap \theta_3) \cup \theta_1 = \alpha_1 \neq \emptyset \\ \alpha_1 &:= \theta_1 \neq \emptyset & \alpha_{11} &:= \theta_1 \cap \theta_2 \cap \theta_3 = \emptyset \\ \alpha_2 &:= \theta_2 \neq \emptyset & \alpha_{12} &:= (\theta_1 \cup \theta_2) \cap \theta_3 = \emptyset \\ \alpha_3 &:= \theta_3 \neq \emptyset & \alpha_{13} &:= (\theta_1 \cup \theta_3) \cap \theta_2 = \emptyset \\ \alpha_4 &:= \theta_1 \cup \theta_2 \neq \emptyset & \alpha_{14} &:= (\theta_2 \cup \theta_3) \cap \theta_1 = \emptyset \\ \alpha_5 &:= \theta_1 \cup \theta_3 \neq \emptyset & \alpha_{15} &:= \theta_1 \cap \theta_2 = \emptyset \\ \alpha_6 &:= \theta_2 \cup \theta_3 \neq \emptyset & \alpha_{16} &:= \theta_1 \cap \theta_3 = \emptyset \\ \alpha_7 &:= \theta_1 \cup \theta_2 \cup \theta_3 \neq \emptyset & \alpha_{17} &:= \theta_2 \cap \theta_3 = \emptyset \\ \alpha_8 &:= (\theta_1 \cap \theta_2) \cup \theta_3 = \alpha_3 \neq \emptyset & \alpha_{18} &:= (\theta_1 \cap \theta_2) \cup (\theta_1 \cap \theta_3) \cup (\theta_2 \cap \theta_3) = \emptyset \\ \alpha_9 &:= (\theta_1 \cap \theta_3) \cup \theta_2 = \alpha_4 \neq \emptyset \end{aligned} \quad (\text{A.1.8})$$

The frame D^Θ defined with constrains (A.1.8) has now $|D^\Theta| = |2^\Theta| = 8$ has different elements and corresponds to Shafer's model. This proposition set definition will be used in the application of the DSMT fusion rules in the occupancy grids in Section A.3.

A.2 Dempster-Shafer conjunctive rule

The Dempster-Shafer conjunctive rule is defined as:

$$m_{\cap}(A) = (m_1 \odot m_2)(A) = \sum_{X \cap Y = A} m_1(X) m_2(Y) \quad \forall X, Y \in 2^\Theta \quad (\text{A.2.1})$$

In the preparation step for the Dempster-Shafer conjunctive rule equation, it is best to prepare a helper table of all possible set intersections within the power set 2^Θ – Table A.1.

Table A.1. Operator $\alpha_i \cap \alpha_j$ for $i, j = 0, \dots, 7$ for the power set 2^Θ .

\cap	α_0	α_1	α_2	α_3	α_4	α_5	α_6	α_7
α_0								
α_1	α_0	α_1	α_0	α_0	α_1	α_1	α_0	α_1
α_2	α_0	α_0	α_2	α_0	α_2	α_0	α_2	α_2
α_3	α_0	α_0	α_0	α_3	α_0	α_3	α_3	α_3
α_4	α_0	α_1	α_2	α_0	α_4	α_1	α_2	α_4
α_5	α_0	α_1	α_0	α_3	α_1	α_5	α_3	α_5
α_6	α_0	α_0	α_2	α_3	α_2	α_3	α_6	α_6
α_7	α_0	α_1	α_2	α_3	α_4	α_5	α_6	α_7

Number of the combination of mass multiplication may make it harder to present the fusion rule equations. For clarity of the presented fusion rules, the following notation is assumed:

$$m_1^i = m_1(\alpha_i) \qquad m_2^i = m_2(\alpha_i) \qquad (\text{A.2.2})$$

Using the values stored in Table A.1 the conjunctive rule equation can be easily derived by writing down the sum of all pairs of masses that produce particular subset α_i :

$$\begin{aligned}
m_{\cap}(\alpha_0) &= m_1^0 + m_2^0 - m_1^0 m_2^0 + m_1^1(m_2^2 + m_2^3 + m_2^6) + m_2^1(m_1^2 + m_1^3 + m_1^6) + \\
&\quad + m_1^2 m_2^3 + m_1^3 m_2^2 + m_1^2 m_2^5 + m_1^5 m_2^2 + m_1^3 m_2^4 + m_1^4 m_2^3 \\
m_{\cap}(\alpha_1) &= m_1^1(m_2^1 + m_2^4 + m_2^5 + m_2^7) + m_2^1(m_1^4 + m_1^5 + m_1^7) + m_1^4 m_2^5 + m_1^5 m_2^4 \\
m_{\cap}(\alpha_2) &= m_1^2(m_2^2 + m_2^4 + m_2^6 + m_2^7) + m_2^2(m_1^4 + m_1^6 + m_1^7) + m_1^4 m_2^6 + m_1^6 m_2^4 \\
m_{\cap}(\alpha_3) &= m_1^3(m_2^3 + m_2^5 + m_2^6 + m_2^7) + m_2^3(m_1^5 + m_1^6 + m_1^7) + m_1^5 m_2^6 + m_1^6 m_2^5 \\
m_{\cap}(\alpha_4) &= m_1^4 m_2^4 + m_1^4 m_2^7 + m_1^7 m_2^4 \\
m_{\cap}(\alpha_5) &= m_1^5 m_2^5 + m_1^5 m_2^7 + m_1^7 m_2^5 \\
m_{\cap}(\alpha_6) &= m_1^6 m_2^6 + m_1^6 m_2^7 + m_1^7 m_2^6 \\
m_{\cap}(\alpha_7) &= m_1^7 m_2^7
\end{aligned} \qquad (\text{A.2.3})$$

Assuming that the initial state $m_{1(\text{init})}$ is a VBA:

$$\forall_{i \in \{0, \dots, 6\}} m_{1(\text{init})}^i = 0 \qquad m_{1(\text{init})}^7 = 1 \qquad (\text{A.2.4})$$

And all IG BBAs defined as (4.2.2) m_2 are fused with constrains:

$$\forall_{i \in \{1, 2, 3, 4, 7\}} m_2^i \geq 0 \qquad \forall_{i \in \{0, 5, 6\}} m_2^i = 0 \qquad (\text{A.2.5})$$

The simplified conjunctive rule equation can be derived:

$$\begin{aligned}
m_{\cap}(\alpha_0) &= m_1^0 + m_1^1(m_2^2 + m_2^3) + m_2^1(m_1^2 + m_1^3) + m_1^3(m_2^1 + m_2^2 + m_2^4) + m_1^4 m_2^3 \\
m_{\cap}(\alpha_1) &= m_1^1(m_2^1 + m_2^4 + m_2^7) + m_2^1(m_1^4 + m_1^7) \\
m_{\cap}(\alpha_2) &= m_1^2(m_2^2 + m_2^4 + m_2^7) + m_2^2(m_1^4 + m_1^7) \\
m_{\cap}(\alpha_3) &= m_1^3(m_2^3 + m_2^7) + m_1^7 m_2^3 \\
m_{\cap}(\alpha_4) &= m_1^4(m_2^4 + m_2^7) + m_1^7 m_2^4 \\
m_{\cap}(\alpha_5) &= 0 \\
m_{\cap}(\alpha_6) &= 0 \\
m_{\cap}(\alpha_7) &= m_1^7 m_2^7 \quad \square
\end{aligned} \qquad (\text{A.2.6})$$

A.3 DS_m common rule of combination for free-DS_m models

The common DS_m rule of combination is defined similarly to the DS conjunctive rule, but on the body of the hyper-power set:

$$m_{\cap}(A) = (m_1 \odot m_2)(A) = \sum_{X \cap Y = A} m_1(X) m_2(Y) \quad \forall X, Y \in D^{\ominus} \quad (\text{A.3.1})$$

The procedure for deriving an equation for this fusion rule is the same. First, it is needed to draw a table of all possible set intersections ($\alpha_i \cap \alpha_j$ for $i, j = 0, \dots, 18$) in the hyper-power set D^{\ominus} . Using the classical intersection operator on sets, the results can be summarized in the symmetric table A.2.

Table A.2. Operator $\alpha_i \cap \alpha_j$ for $i, j = 0, \dots, 18$ for the hyper-power set D^{\ominus}

\cap	α_0	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}	α_{11}	α_{12}	α_{13}	α_{14}	α_{15}	α_{16}	α_{17}	α_{18}
α_0	α_0	α_0	α_0	α_0	α_0	α_0	α_0	α_0	α_0	α_0	α_0	α_0	α_0	α_0	α_0	α_0	α_0	α_0	α_0
α_1	α_0	α_1	α_{15}	α_{16}	α_1	α_1	α_{14}	α_1	α_{14}	α_{14}	α_1	α_{11}	α_{16}	α_{15}	α_{14}	α_{15}	α_{16}	α_{11}	α_{14}
α_2	α_0	α_{15}	α_2	α_{17}	α_2	α_{13}	α_2	α_2	α_{13}	α_2	α_{13}	α_{11}	α_{17}	α_{13}	α_{15}	α_{15}	α_{11}	α_{17}	α_{13}
α_3	α_0	α_{16}	α_{17}	α_3	α_{12}	α_3	α_3	α_3	α_3	α_{12}	α_{12}	α_{11}	α_{12}	α_{17}	α_{16}	α_{11}	α_{16}	α_{17}	α_{12}
α_4	α_0	α_1	α_2	α_{12}	α_4	α_{10}	α_9	α_4	α_{18}	α_9	α_{10}	α_{11}	α_{12}	α_{13}	α_{14}	α_{15}	α_{16}	α_{17}	α_{18}
α_5	α_0	α_1	α_{13}	α_3	α_{10}	α_5	α_8	α_5	α_8	α_{18}	α_{10}	α_{11}	α_{12}	α_{13}	α_{14}	α_{15}	α_{16}	α_{17}	α_{18}
α_6	α_0	α_{14}	α_2	α_3	α_9	α_8	α_6	α_6	α_8	α_9	α_{18}	α_{11}	α_{12}	α_{13}	α_{14}	α_{15}	α_{16}	α_{17}	α_{18}
α_7	α_0	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}	α_{11}	α_{12}	α_{13}	α_{14}	α_{15}	α_{16}	α_{17}	α_{18}
α_8	α_0	α_{14}	α_{13}	α_3	α_{18}	α_8	α_8	α_8	α_8	α_{18}	α_{18}	α_{11}	α_{12}	α_{13}	α_{14}	α_{15}	α_{16}	α_{17}	α_{18}
α_9	α_0	α_{14}	α_2	α_{12}	α_9	α_{18}	α_9	α_9	α_{18}	α_9	α_{18}	α_{11}	α_{12}	α_{13}	α_{14}	α_{15}	α_{16}	α_{17}	α_{18}
α_{10}	α_0	α_1	α_{13}	α_{12}	α_{10}	α_{10}	α_{18}	α_{10}	α_{18}	α_{18}	α_{10}	α_{11}	α_{12}	α_{13}	α_{14}	α_{15}	α_{16}	α_{17}	α_{18}
α_{11}	α_0	α_{11}																	
α_{12}	α_0	α_{16}	α_{17}	α_{12}	α_{11}	α_{12}	α_{17}	α_{16}	α_{11}	α_{16}	α_{17}	α_{12}							
α_{13}	α_0	α_{15}	α_{13}	α_{17}	α_{13}	α_{11}	α_{17}	α_{13}	α_{15}	α_{15}	α_{11}	α_{17}	α_{13}						
α_{14}	α_0	α_{14}	α_{15}	α_{16}	α_{14}	α_{11}	α_{16}	α_{15}	α_{14}	α_{15}	α_{16}	α_{11}	α_{14}						
α_{15}	α_0	α_{15}	α_{15}	α_{11}	α_{15}	α_{11}	α_{11}	α_{15}	α_{15}	α_{15}	α_{11}	α_{11}	α_{15}						
α_{16}	α_0	α_{16}	α_{11}	α_{16}	α_{11}	α_{16}	α_{11}	α_{16}	α_{11}	α_{16}	α_{11}	α_{16}							
α_{17}	α_0	α_{11}	α_{17}	α_{11}	α_{17}	α_{17}	α_{11}	α_{11}	α_{11}	α_{17}	α_{17}								
α_{18}	α_0	α_{14}	α_{13}	α_{12}	α_{18}	α_{11}	α_{12}	α_{13}	α_{14}	α_{15}	α_{16}	α_{17}	α_{18}						

Writing down the full equation of the DS_m classic (DS_mC) fusion rule will produce complicated and long results. Although, for the fusion architecture presented in Section 4.3 this fusion rule is applied for the fusion of constrained intermediate grid BBAs (Equations (4.2.2)), which allow the supposition of the following two prepositions.

Firstly, the initial cell state is a VBA, which for hyper-power set takes form:

$$\forall_{i \in \{0, \dots, 18\} \setminus \{7\}} m_1^i(\text{init}) = 0 \quad m_1^7(\text{init}) = 1 \quad (\text{A.3.2})$$

Secondly, the fused IG BBAs (m_2) have constrains:

$$\forall_{i \in \{1, 2, 3, 4, 7\}} m_2^i \geq 0 \quad \forall_{i \in \{0, 5, 6, 8, 9, \dots, 18\}} m_2^i = 0 \quad (\text{A.3.3})$$

Based on these assumptions, a simplified equation for the DS_mC rule of combination can be obtained:

$$\begin{aligned}
m_{DSmC}(\alpha_0) &= 0 \\
m_{DSmC}(\alpha_1) &= m_1^1 m_2^1 + m_1^1 m_2^4 + m_2^1 m_1^4 + m_1^1 m_2^7 + m_2^1 m_1^7 \\
m_{DSmC}(\alpha_2) &= m_1^2 m_2^2 + m_1^2 m_2^4 + m_2^2 m_1^4 + m_1^2 m_2^7 + m_2^2 m_1^7 \\
m_{DSmC}(\alpha_3) &= m_1^3 m_2^3 + m_1^3 m_2^7 + m_2^3 m_1^7 \\
m_{DSmC}(\alpha_4) &= m_1^4 m_2^4 + m_1^4 m_2^7 + m_2^4 m_1^7 \\
m_{DSmC}(\alpha_5) &= 0 \\
m_{DSmC}(\alpha_6) &= 0 \\
m_{DSmC}(\alpha_7) &= m_1^7 m_2^7 \\
m_{DSmC}(\alpha_8) &= 0 \\
m_{DSmC}(\alpha_9) &= 0 \\
m_{DSmC}(\alpha_{10}) &= 0 \\
m_{DSmC}(\alpha_{11}) &= m_1^{11} (m_2^1 + m_2^2 + m_2^3 + m_2^4 + m_2^7) + m_2^3 m_1^{15} + m_2^2 m_1^{16} + m_2^1 m_1^{17} \\
m_{DSmC}(\alpha_{12}) &= m_1^{12} m_2^3 + m_1^{12} m_2^4 + m_1^{12} m_2^7 + m_1^3 m_2^4 + m_2^3 m_1^4 \\
m_{DSmC}(\alpha_{13}) &= 0 \\
m_{DSmC}(\alpha_{14}) &= 0 \\
m_{DSmC}(\alpha_{15}) &= m_1^{15} m_2^1 + m_1^{15} m_2^2 + m_1^{15} m_2^4 + m_1^{15} m_2^7 + m_1^1 m_2^2 + m_2^1 m_1^2 \\
m_{DSmC}(\alpha_{16}) &= m_1^{16} m_2^1 + m_1^{16} m_2^3 + m_1^{16} m_2^4 + m_1^{16} m_2^7 + m_1^1 m_2^3 + m_2^1 m_1^3 + m_2^1 m_1^{12} \\
m_{DSmC}(\alpha_{17}) &= m_1^{17} m_2^2 + m_1^{17} m_2^3 + m_1^{17} m_2^4 + m_1^{17} m_2^7 + m_2^2 m_1^3 + m_2^2 m_1^3 + m_2^2 m_1^{12} \\
m_{DSmC}(\alpha_{18}) &= 0 \quad \square
\end{aligned} \tag{A.3.4}$$

As presented in the equation A.3.4 only 10 subsets from the hyper-power set are used in the defined free DS_m model. This allows some further optimization and reduces significantly the complexity of the following DS_mH fusion rule.

A.4 DS_m rule of combination for hybrid-DS_m models

Hybrid DS_mT fusion rule is defined as:

$$m_{DSmH}(X) = \phi(X) [S_1(X) + S_2(X) + S_3(X)] \tag{A.4.1}$$

where $\phi(X)$ is a characteristic non-emptiness function of a set X , i.e., $\phi(X) = 1$ if $X \notin \emptyset$ and $\phi(X) = 0$ otherwise.

For two fusion sources components $S_1(A)$, $S_2(A)$ and $S_3(A)$ are defined as:

$$S_1(A) = m_{DSmC}(X) = \sum_{X \cap Y = A} m_1(X) m_2(Y) \quad \forall X, Y \in D^\ominus \tag{A.4.2}$$

$$S_2(A) = \sum_{\substack{[(u(X) \cup u(Y)) = A] \vee \\ [(u(X) \cup u(Y)) \in \emptyset \wedge A = \emptyset]}} m_1(X) m_2(Y) \quad \forall X, Y \in \emptyset \tag{A.4.3}$$

$$S_3(A) = \sum_{X \cup Y = A, X \cap Y = \emptyset} m_1(X) m_2(Y) \quad \forall X, Y \in D^\ominus \tag{A.4.4}$$

Let us consider the Dempster frame of reference as presented in Chapter 4, Section 4.1.3. In this frame, the characteristic non-emptiness function is defined as:

$$\phi(X) = \begin{cases} 1, & \text{for } X \in \{\alpha_1, \dots, \alpha_{10}\} \\ 0, & \text{for } X \in \{\alpha_0, \alpha_{11}, \dots, \alpha_{18}\} \end{cases} \quad (\text{A.4.5})$$

Based on that, calculations of values of $S_1(A)$, $S_2(A)$ and $S_3(A)$ for subsets $\{\alpha_0, \alpha_{11}, \dots, \alpha_{18}\}$ are not needed. For the non-zero subsets $\{\alpha_1, \dots, \alpha_{10}\}$ direct equations of the components $S_1(A)$, $S_2(A)$ and $S_3(A)$ have to be computed.

A.4.1 Component S_I

The component $S_1(\cdot)$ can be computed similarly as the m_{DSmC} , based on the values from Table A.2.

$$\begin{aligned} S_1(\alpha_1) &= m_1^1 m_2^1 + m_1^1 m_2^4 + m_2^1 m_1^4 + m_1^1 m_2^5 + m_2^1 m_1^5 + m_1^1 m_2^7 + m_2^1 m_1^7 + m_1^1 m_2^{10} + m_2^1 m_1^{10} \\ S_1(\alpha_2) &= m_1^2 m_2^2 + m_1^2 m_2^4 + m_2^2 m_1^4 + m_1^2 m_2^6 + m_2^2 m_1^6 + m_1^2 m_2^7 + m_2^2 m_1^7 + m_1^2 m_2^9 + m_2^2 m_1^9 \\ S_1(\alpha_3) &= m_1^3 m_2^3 + m_1^3 m_2^5 + m_2^3 m_1^5 + m_1^3 m_2^6 + m_2^3 m_1^6 + m_1^3 m_2^7 + m_2^3 m_1^7 + m_1^3 m_2^8 + m_2^3 m_1^8 \\ S_1(\alpha_4) &= m_1^4 m_2^4 + m_1^4 m_2^7 + m_2^4 m_1^7 \\ S_1(\alpha_5) &= m_1^5 m_2^5 + m_1^5 m_2^7 + m_2^5 m_1^7 \\ S_1(\alpha_6) &= m_1^6 m_2^6 + m_1^6 m_2^7 + m_2^6 m_1^7 \\ S_1(\alpha_7) &= m_1^7 m_2^7 \\ S_1(\alpha_8) &= m_1^8 m_2^8 + m_1^5 m_2^6 + m_2^5 m_1^6 + m_1^8 m_2^5 + m_2^8 m_1^5 + m_1^8 m_2^6 + m_2^8 m_1^6 + m_1^8 m_2^7 + m_2^8 m_1^7 \\ S_1(\alpha_9) &= m_1^9 m_2^9 + m_1^4 m_2^6 + m_2^4 m_1^6 + m_1^9 m_2^4 + m_2^9 m_1^4 + m_1^9 m_2^6 + m_2^9 m_1^6 + m_1^9 m_2^7 + m_2^9 m_1^7 \\ S_1(\alpha_{10}) &= m_1^{10} m_2^{10} + m_1^4 m_2^5 + m_2^4 m_1^5 + m_1^{10} m_2^4 + m_2^{10} m_1^4 + m_1^{10} m_2^5 + m_2^{10} m_1^5 + m_1^{10} m_2^7 + m_2^{10} m_1^7 \end{aligned} \quad (\text{A.4.6})$$

The DS_mH fusion rule is applied to fuse the output of the DS_mC rule m_1 together with the Dempster BBA m_2 . Based on that information, the following constrains apply:

$$\forall_{i \in \{1,2,3,4,7,11,12,15,16,17\}} m_1^i \geq 0 \quad \forall_{i \in \{0,5,6,8,9,10,13,14,18\}} m_1^i = 0 \quad (\text{A.4.7})$$

$$\forall_{i \in \{1,2,3,4,5,6,7\}} m_2^i \geq 0 \quad \forall_{i \in \{0,8,9,\dots,18\}} m_2^i = 0 \quad (\text{A.4.8})$$

After applying the aforementioned constraints, the component S_1 simplifies into:

$$\begin{aligned} S_1(\alpha_1) &= m_1^1 m_2^1 + m_1^1 m_2^4 + m_2^1 m_1^4 + m_1^1 m_2^5 + m_1^1 m_2^7 + m_2^1 m_1^7 \\ S_1(\alpha_2) &= m_1^2 m_2^2 + m_1^2 m_2^4 + m_2^2 m_1^4 + m_1^2 m_2^6 + m_1^2 m_2^7 + m_2^2 m_1^7 \\ S_1(\alpha_3) &= m_1^3 m_2^3 + m_1^3 m_2^5 + m_2^3 m_1^5 + m_1^3 m_2^6 + m_2^3 m_1^6 + m_1^3 m_2^7 + m_2^3 m_1^7 \\ S_1(\alpha_4) &= m_1^4 m_2^4 + m_1^4 m_2^7 + m_2^4 m_1^7 \\ S_1(\alpha_5) &= m_2^5 m_1^7 \\ S_1(\alpha_6) &= m_2^6 m_1^7 \\ S_1(\alpha_7) &= m_1^7 m_2^7 \\ S_1(\alpha_8) &= 0 \\ S_1(\alpha_9) &= m_1^4 m_2^6 \\ S_1(\alpha_{10}) &= m_1^4 m_2^5 \end{aligned} \quad (\text{A.4.9})$$

A.4.2 Component S_2

The component $S_2(A)$ represents the mass of all relatively and absolutely empty sets. In order to compute, it is helpful to define the union of all singletons θ_j that compose α_i :

$$\begin{aligned}
 u(\alpha_0) &= \emptyset = \alpha_0 & u(\alpha_{10}) &= \theta_1 \cup \theta_2 \cup \theta_3 = \alpha_7 \\
 u(\alpha_1) &= \theta_1 = \alpha_1 & u(\alpha_{11}) &= \theta_1 \cup \theta_2 \cup \theta_3 = \alpha_7 \\
 u(\alpha_2) &= \theta_2 = \alpha_2 & u(\alpha_{12}) &= \theta_1 \cup \theta_2 \cup \theta_3 = \alpha_7 \\
 u(\alpha_3) &= \theta_3 = \alpha_3 & u(\alpha_{13}) &= \theta_1 \cup \theta_2 \cup \theta_3 = \alpha_7 \\
 u(\alpha_4) &= \theta_1 \cup \theta_2 = \alpha_4 & u(\alpha_{14}) &= \theta_1 \cup \theta_2 \cup \theta_3 = \alpha_7 \\
 u(\alpha_5) &= \theta_1 \cup \theta_3 = \alpha_5 & u(\alpha_{15}) &= \theta_1 \cup \theta_2 = \alpha_4 \\
 u(\alpha_6) &= \theta_2 \cup \theta_3 = \alpha_6 & u(\alpha_{16}) &= \theta_1 \cup \theta_3 = \alpha_5 \\
 u(\alpha_7) &= \theta_1 \cup \theta_2 \cup \theta_3 = \alpha_7 & u(\alpha_{17}) &= \theta_2 \cup \theta_3 = \alpha_6 \\
 u(\alpha_8) &= \theta_1 \cup \theta_2 \cup \theta_3 = \alpha_7 & u(\alpha_{18}) &= \theta_1 \cup \theta_2 \cup \theta_3 = \alpha_7 \\
 u(\alpha_9) &= \theta_1 \cup \theta_2 \cup \theta_3 = \alpha_7 & &
 \end{aligned} \tag{A.4.10}$$

Having defined $u(\alpha)$, the next helper table of $u(\alpha_i) \cup u(\alpha_j)$ for the empty sets can be created (Table A.3).

Table A.3. Operator $u(\alpha_i) \cup u(\alpha_j)$ for $i, j = 0, 11, \dots, 18$ for the hyper-power set D^Θ

\cup	$u(\alpha_0)$	$u(\alpha_{11})$	$u(\alpha_{12})$	$u(\alpha_{13})$	$u(\alpha_{14})$	$u(\alpha_{15})$	$u(\alpha_{16})$	$u(\alpha_{17})$	$u(\alpha_{18})$
$u(\alpha_0)$	α_0	α_7	α_7	α_7	α_7	α_4	α_5	α_6	α_7
$u(\alpha_{11})$	α_7	α_7	α_7	α_7	α_7	α_7	α_7	α_7	α_7
$u(\alpha_{12})$	α_7	α_7	α_7	α_7	α_7	α_7	α_7	α_7	α_7
$u(\alpha_{13})$	α_7	α_7	α_7	α_7	α_7	α_7	α_7	α_7	α_7
$u(\alpha_{14})$	α_7	α_7	α_7	α_7	α_7	α_7	α_7	α_7	α_7
$u(\alpha_{15})$	α_4	α_7	α_7	α_7	α_7	α_4	α_7	α_7	α_7
$u(\alpha_{16})$	α_5	α_7	α_7	α_7	α_7	α_7	α_5	α_7	α_7
$u(\alpha_{17})$	α_6	α_7	α_7	α_7	α_7	α_7	α_7	α_6	α_7
$u(\alpha_{18})$	α_7	α_7	α_7	α_7	α_7	α_7	α_7	α_7	α_7

For the set of constraints A.4.7 and A.4.8 the component $S_2(\cdot)$ is defined as:

$$\begin{aligned}
 S_2(\alpha_1) &= 0 & S_2(\alpha_6) &= 0 \\
 S_2(\alpha_2) &= 0 & S_2(\alpha_7) &= 0 \\
 S_2(\alpha_3) &= 0 & S_2(\alpha_8) &= 0 \\
 S_2(\alpha_4) &= 0 & S_2(\alpha_9) &= 0 \\
 S_2(\alpha_5) &= 0 & S_2(\alpha_{10}) &= 0
 \end{aligned} \tag{A.4.11}$$

Component S_2 corresponds to complete conflict states from the input data such as disjoint frames of discernment. In the defined Dempster frame of discernment, all unions of elemental hypotheses are allowed, therefore the state S_2 is always zero and can be omitted.

A.4.3 Component S_3

The component $S_3(\cdot)$ describes the hybrid fusion conflict redistribution mechanism. This part of DS_mH equation defines how the resulting belief mismatch should be divided into existing states in the hybrid model.

In order to create this fusion rule equation, two conditions have to be met. Firstly, the component $S_3(\cdot)$ considers only the constrained/forbidden beliefs of the hybrid model. This condition is met when the intersection of two subsets α_i and α_j is empty (Table A.4).

The second condition defines where the conflict should be redistributed. This set can be obtained by taking into consideration only the union of subsets α_i and α_j which are equal to the evaluated prepositions. This step can be defined using (Table A.5).

Table A.4. Operator $\alpha_i \cap \alpha_j$ for $i, j = 0, \dots, 18$ for the hyper-power set D^Θ with Smets constrains.

\cap	α_0	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}	α_{11}	α_{12}	α_{13}	α_{14}	α_{15}	α_{16}	α_{17}	α_{18}
α_0	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
α_1	\emptyset	α_1	\emptyset	\emptyset	α_1	α_1	\emptyset	α_1	\emptyset	\emptyset	α_1	\emptyset							
α_2	\emptyset	\emptyset	α_2	\emptyset	α_2	\emptyset	α_2	α_2	\emptyset	α_2	\emptyset								
α_3	\emptyset	\emptyset	\emptyset	α_3	\emptyset	α_3	α_3	α_3	α_3	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
α_4	\emptyset	α_1	α_2	\emptyset	α_4	α_{10}	α_9	α_4	\emptyset	α_9	α_{10}	\emptyset							
α_5	\emptyset	α_1	\emptyset	α_3	α_{10}	α_5	α_8	α_5	α_8	\emptyset	α_{10}	\emptyset							
α_6	\emptyset	\emptyset	α_2	α_3	α_9	α_8	α_6	α_6	α_8	α_9	\emptyset								
α_7	\emptyset	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}	\emptyset							
α_8	\emptyset	\emptyset	\emptyset	α_3	\emptyset	α_8	α_8	α_8	α_8	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
α_9	\emptyset	\emptyset	α_2	\emptyset	α_9	\emptyset	α_9	α_9	\emptyset	α_9	\emptyset								
α_{10}	\emptyset	α_1	\emptyset	\emptyset	α_{10}	α_{10}	\emptyset	α_{10}	\emptyset	\emptyset	α_{10}	\emptyset							
α_{11}	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
α_{12}	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
α_{13}	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
α_{14}	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
α_{15}	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
α_{16}	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
α_{17}	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
α_{18}	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset

Table A.5. Operator $\alpha_i \cup \alpha_j$ for $i, j = 0, \dots, 18$ for the hyper-power set D^\ominus

\cup	α_0	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}	α_{11}	α_{12}	α_{13}	α_{14}	α_{15}	α_{16}	α_{17}	α_{18}
α_0	α_0	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}	α_{11}	α_{12}	α_{13}	α_{14}	α_{15}	α_{16}	α_{17}	α_{18}
α_1	α_1	α_1	α_4	α_5	α_4	α_5	α_7	α_7	α_5	α_4	α_{10}	α_1	α_{10}	α_{10}	α_1	α_1	α_1	α_{10}	α_{10}
α_2	α_2	α_4	α_2	α_6	α_4	α_7	α_6	α_7	α_6	α_9	α_4	α_2	α_9	α_2	α_9	α_2	α_9	α_2	α_9
α_3	α_3	α_5	α_6	α_3	α_7	α_5	α_6	α_7	α_8	α_6	α_5	α_3	α_3	α_8	α_8	α_8	α_3	α_3	α_8
α_4	α_4	α_4	α_4	α_7	α_4	α_7	α_7	α_7	α_7	α_4	α_4	α_4	α_4	α_4	α_4	α_4	α_4	α_4	α_4
α_5	α_5	α_5	α_7	α_5	α_7	α_5	α_7	α_7	α_5	α_7	α_5								
α_6	α_6	α_7	α_6	α_6	α_7	α_7	α_6	α_7	α_6	α_6	α_7	α_6							
α_7	α_7	α_7	α_7	α_7	α_7	α_7	α_7	α_7	α_7	α_7	α_7	α_7	α_7	α_7	α_7	α_7	α_7	α_7	α_7
α_8	α_8	α_5	α_6	α_8	α_7	α_5	α_6	α_7	α_8	α_6	α_5	α_8							
α_9	α_9	α_4	α_9	α_6	α_4	α_7	α_6	α_7	α_6	α_9	α_4	α_9							
α_{10}	α_{10}	α_{10}	α_4	α_5	α_4	α_5	α_7	α_7	α_5	α_4	α_{10}								
α_{11}	α_{11}	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}	α_{11}	α_{12}	α_{13}	α_{14}	α_{15}	α_{16}	α_{17}	α_{18}
α_{12}	α_{12}	α_{10}	α_9	α_3	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}	α_{12}	α_{12}	α_{18}	α_{18}	α_{18}	α_{12}	α_{12}	α_{18}
α_{13}	α_{13}	α_{10}	α_2	α_8	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}	α_{13}	α_{18}	α_{13}	α_{18}	α_{13}	α_{18}	α_{13}	α_{18}
α_{14}	α_{14}	α_1	α_9	α_8	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}	α_{14}	α_{18}	α_{18}	α_{14}	α_{14}	α_{14}	α_{18}	α_{18}
α_{15}	α_{15}	α_1	α_2	α_8	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}	α_{15}	α_{18}	α_{13}	α_{14}	α_{15}	α_{14}	α_{13}	α_{18}
α_{16}	α_{16}	α_1	α_9	α_3	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}	α_{16}	α_{12}	α_{18}	α_{14}	α_{14}	α_{16}	α_{12}	α_{18}
α_{17}	α_{17}	α_{10}	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}	α_{17}	α_{12}	α_{13}	α_{18}	α_{13}	α_{12}	α_{17}	α_{18}
α_{18}	α_{18}	α_{10}	α_9	α_8	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}	α_{18}							

For the set of constraints A.4.7 and A.4.8 the component $S_3(\cdot)$ is defined as:

$$\begin{aligned}
S_3(\alpha_1) &= m_2^1(m_1^{11} + m_1^{15} + m_1^{16}) \\
S_3(\alpha_2) &= m_2^2(m_1^{11} + m_1^{15} + m_1^{17}) \\
S_3(\alpha_3) &= m_2^3(m_1^{11} + m_1^{12} + m_1^{16} + m_1^{17}) \\
S_3(\alpha_4) &= m_2^4(m_1^{11} + m_1^{12} + m_1^{15} + m_1^{16} + m_1^{17}) + m_2^2 m_1^1 + m_2^1 m_1^2 \\
S_3(\alpha_5) &= m_2^5(m_1^{11} + m_1^{12} + m_1^{15} + m_1^{16} + m_1^{17}) + m_2^3 m_1^1 + m_2^1 m_1^3 \\
S_3(\alpha_6) &= m_2^6(m_1^{11} + m_1^{12} + m_1^{15} + m_1^{16} + m_1^{17}) + m_2^3 m_1^2 + m_2^2 m_1^3 \\
S_3(\alpha_7) &= m_2^7(m_1^{11} + m_1^{12} + m_1^{15} + m_1^{16} + m_1^{17}) + m_2^6 m_1^1 + m_2^1 m_1^6 + m_2^5 m_1^2 + m_2^2 m_1^5 + m_2^4 m_1^3 + m_2^3 m_1^4 \\
S_3(\alpha_8) &= m_2^3 m_1^{15} \\
S_3(\alpha_9) &= m_2^2 m_1^{12} + m_2^2 m_1^{16} \\
S_3(\alpha_{10}) &= m_2^1 m_1^{12} + m_2^1 m_1^{17}
\end{aligned} \tag{A.4.12}$$

A.4.4 Compression to non-empty propositions

By application of constraints A.4.7 and A.4.8 the component $S_2(\cdot)$ can be ruled out as defined in Section A.4.2.

Therefore, the hybrid DSm combination rule can be defined as:

$$m_{DSmH}^*(X) = \begin{cases} S_1(X) + S_3(X), & \text{for } X \in \{\alpha_1, \dots, \alpha_{10}\} \\ 0, & \text{for } X \in \{\alpha_0, \alpha_{11}, \dots, \alpha_{18}\} \end{cases} \tag{A.4.13}$$

$$\begin{aligned}
m_{DSmH}^*(\alpha_1) &= m_1^1(m_2^1 + m_2^4 + m_2^5 + m_2^7) + m_2^1(m_1^4 + m_1^7 + m_1^{11} + m_1^{15} + m_1^{16}) \\
m_{DSmH}^*(\alpha_2) &= m_1^2(m_2^2 + m_2^4 + m_2^6 + m_2^7) + m_2^2(m_1^4 + m_1^7 + m_1^{11} + m_1^{15} + m_1^{17}) \\
m_{DSmH}^*(\alpha_3) &= m_1^3(m_2^3 + m_2^5 + m_2^6 + m_2^7) + m_2^3(m_1^7 + m_1^{11} + m_1^{12} + m_1^{16} + m_1^{17}) \\
m_{DSmH}^*(\alpha_4) &= m_2^4(m_1^4 + m_1^7 + m_1^{11} + m_1^{12} + m_1^{15} + m_1^{16} + m_1^{17}) + m_1^1 m_2^2 + m_2^1 m_1^2 + m_1^4 m_2^7 \\
m_{DSmH}^*(\alpha_5) &= m_2^5(m_1^7 + m_1^{11} + m_1^{12} + m_1^{15} + m_1^{16} + m_1^{17}) + m_1^1 m_2^3 + m_1^3 m_2^1 \\
m_{DSmH}^*(\alpha_6) &= m_2^6(m_1^7 + m_1^{11} + m_1^{12} + m_1^{15} + m_1^{16} + m_1^{17}) + m_2^1 m_2^3 + m_1^3 m_2^2 \\
m_{DSmH}^*(\alpha_7) &= m_2^7(m_1^7 + m_1^{11} + m_1^{12} + m_1^{15} + m_1^{16} + m_1^{17}) + \\
&\quad + m_1^1 m_2^6 + m_1^6 m_2^1 + m_2^1 m_2^5 + m_1^5 m_2^2 + m_1^3 m_2^4 + m_1^4 m_2^3 \\
m_{DSmH}^*(\alpha_8) &= m_1^{15} m_2^3 \\
m_{DSmH}^*(\alpha_9) &= m_1^4 m_2^6 + m_1^{12} m_2^2 + m_1^{16} m_2^2 \\
m_{DSmH}^*(\alpha_{10}) &= m_1^4 m_2^5 + m_1^{12} m_2^1 + m_1^{17} m_2^1
\end{aligned} \tag{A.4.14}$$

Computed masses for each equivalent proposition have to be compressed after fusion to reduce the power set D^\ominus to 2^\ominus :

$$\begin{aligned}
m_{DSmH}(\alpha_0) &= 0 \\
m_{DSmH}(\alpha_1) &= m_{DSmH}(\alpha_1) + m_{DSmH}(\alpha_{10}) \\
m_{DSmH}(\alpha_2) &= m_{DSmH}(\alpha_2) + m_{DSmH}(\alpha_9) \\
m_{DSmH}(\alpha_3) &= m_{DSmH}(\alpha_3) + m_{DSmH}(\alpha_8) \\
m_{DSmH}(\alpha_4) &= m_{DSmH}(\alpha_4) \\
m_{DSmH}(\alpha_5) &= m_{DSmH}(\alpha_5) \\
m_{DSmH}(\alpha_6) &= m_{DSmH}(\alpha_6) \\
m_{DSmH}(\alpha_7) &= m_{DSmH}(\alpha_7)
\end{aligned} \tag{A.4.15}$$

Which delivers the final equation for the DS_mH fusion rule in the Dempster model of evidences:

$$\begin{aligned}
m_1^z &= m_1^7 + m_1^{11} + m_1^{12} + m_1^{15} + m_1^{16} + m_1^{17} \\
m_{DSmH}(\alpha_1) &= m_1^z m_2^1 + m_1^1 (m_2^1 + m_2^4 + m_2^5 + m_2^7) + m_1^4 m_2^1 + m_1^4 m_2^5 \\
m_{DSmH}(\alpha_2) &= m_1^z m_2^2 + m_1^2 (m_2^2 + m_2^4 + m_2^6 + m_2^7) + m_1^4 m_2^2 + m_1^4 m_2^6 \\
m_{DSmH}(\alpha_3) &= m_1^z m_2^3 + m_1^3 (m_2^3 + m_2^5 + m_2^6 + m_2^7) \\
m_{DSmH}(\alpha_4) &= m_1^z m_2^4 + m_1^4 m_2^4 + m_1^1 m_2^2 + m_1^2 m_2^1 + m_1^4 m_2^7 \\
m_{DSmH}(\alpha_5) &= m_1^z m_2^5 + m_1^1 m_2^3 + m_1^3 m_2^1 \\
m_{DSmH}(\alpha_6) &= m_1^z m_2^6 + m_1^2 m_2^3 + m_1^3 m_2^2 \\
m_{DSmH}(\alpha_7) &= m_1^z m_2^7 + m_1^1 m_2^6 + m_1^6 m_2^1 + m_1^2 m_2^5 + m_1^5 m_2^2 + m_1^3 m_2^4 + m_1^4 m_2^3
\end{aligned} \tag{A.4.16}$$

□

B Additional simulation evaluation results

Simulation evaluation results presented in Chapter 6 represent only a small group of performed simulations and evaluations of the occupancy grid algorithm. This Appendix chapter presents some supplementary grid representations for different types of fusion and inverse sensor models.

Section B.1 present the sequence of simulations which allowed to tune the optimal threshold level for the automatic landmark extraction procedure used in KPI calculation. Section B.2 present sample occupancy grid snapshots for all combinations of fusion rules, ISM types, decay and free space rates. This presentation can be useful in case of application of the DS_mH fusion framework is prohibited by the existing occupancy grid architecture.

B.1 Threshold dependency

The decision-making process, which extracts binary information from the occupancy grid, is crucial for the downstream components to perform correct actions and apply control to the vehicle. In order to automatically label and evaluate landmark representations, the occupancy grid has to be binarized and then clustered into a list of objects. The binarization process should focus on the extraction of the stationary objects from the occupancy grid. Depending on the grid type, the threshold levels applied for evaluation apply to:

- Bayesian occupancy grid — probability p
- Evidential occupancy grid — stationary evidence $e = m(\alpha_1) = m(\{S\})$

Independently of the grid type, the threshold levels should be comparable, therefore the equation used to convert intermediate grid evidences into Bayesian probabilities can be applied:

$$p = 0.5 \cdot e + 0.5 \tag{B.1.1}$$

In order to determine the best threshold levels which could be used for the decision-making process, simulations with different threshold levels for all types of fusion were performed. Presentations of a sample grid snapshot with the extracted ellipse representation are presented in Figure B.1. Snapshots presented in Figure B.1 clearly shows that the ellipse area strongly depends on the threshold level. In order to present this dependency, the average area for all detected snapshots is measured and is presented in Figure B.2(a).

Increasing the threshold level decreases the estimated landmark areas, but on the other hand, landmarks with low evidences might be missed in the process. Percentage of missed poles closer than 80 m from the vehicle is presented in Figure B.2(b). Occupancy grid used for the simulation scenario has a dimension of 100×100 m and the maximum sensor range is also equal to 100 m. The occupancy grid accumulation process requires at least a couple of scans in order to display the landmark evidences properly, therefore the maximum range of landmark evaluation is limited to 80 m discarding landmarks with small evidence levels.

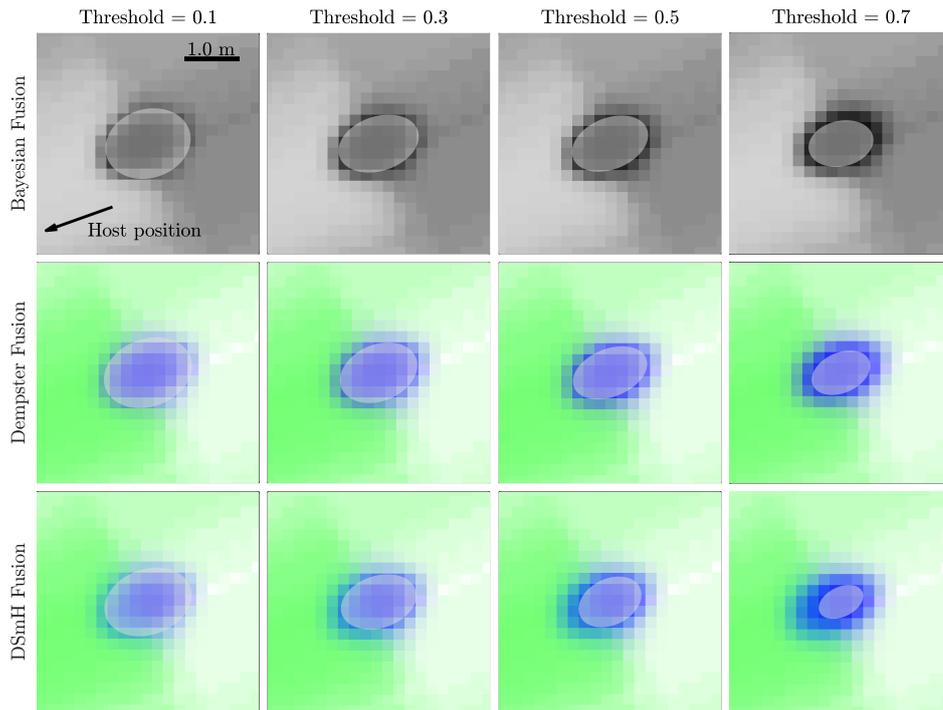
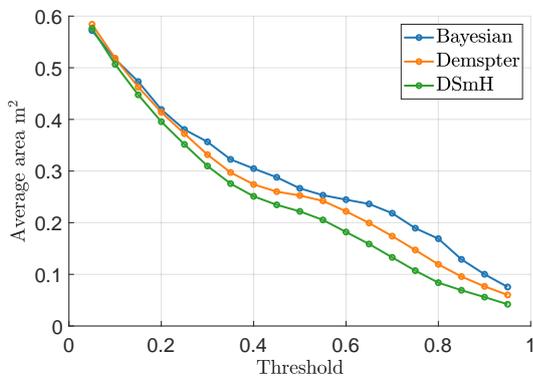
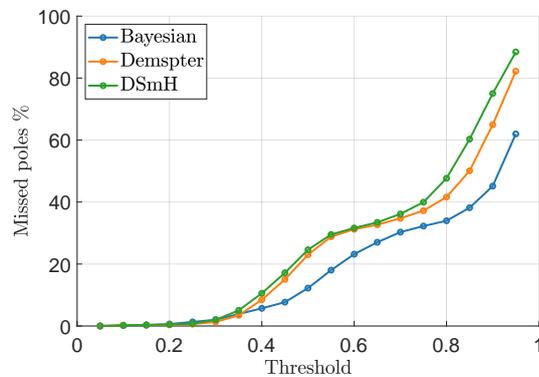


Figure B.1. Ellipse snapshots for different threshold values. For generation of the occupancy grid, Cartesian ISM was used with decay rate 1.0 s^{-1} and 2% free space gain.



(a) Average landmark representation area as a function of threshold.



(b) Percentage of missed poles closer that 80 m from the vehicle as a function of threshold value.

Figure B.2. Plots showing mean area of pole representation and percentage of missed poles as a function of binarization threshold.

The percent of missed poles starts is almost 0% up to the threshold level of 0.3. Binarization thresholds above 0.3 characterizes with growing missed landmark rate. Based on that, the borderline value of 0.3 was selected for the binarization threshold for the whole simulation evaluation presented in Chapter 6. The value of evidence threshold $e = 0.3$ has its equivalent in a probability values as $p = 0.65$ based on equation (B.1.1).

B.2 Additional results for the combination of filtering methods

Combination of filtering methods used to improve the occupancy grid quality might result in different shapes of the pole object representation. This section is an extension of Section 6.3.5, where example snapshots of the grid pole representation are presented for different fusion rules, types of ISM, decay rates and free space gains.

B.2.1 Bayesian grid fusion with hit point inverse sensor model

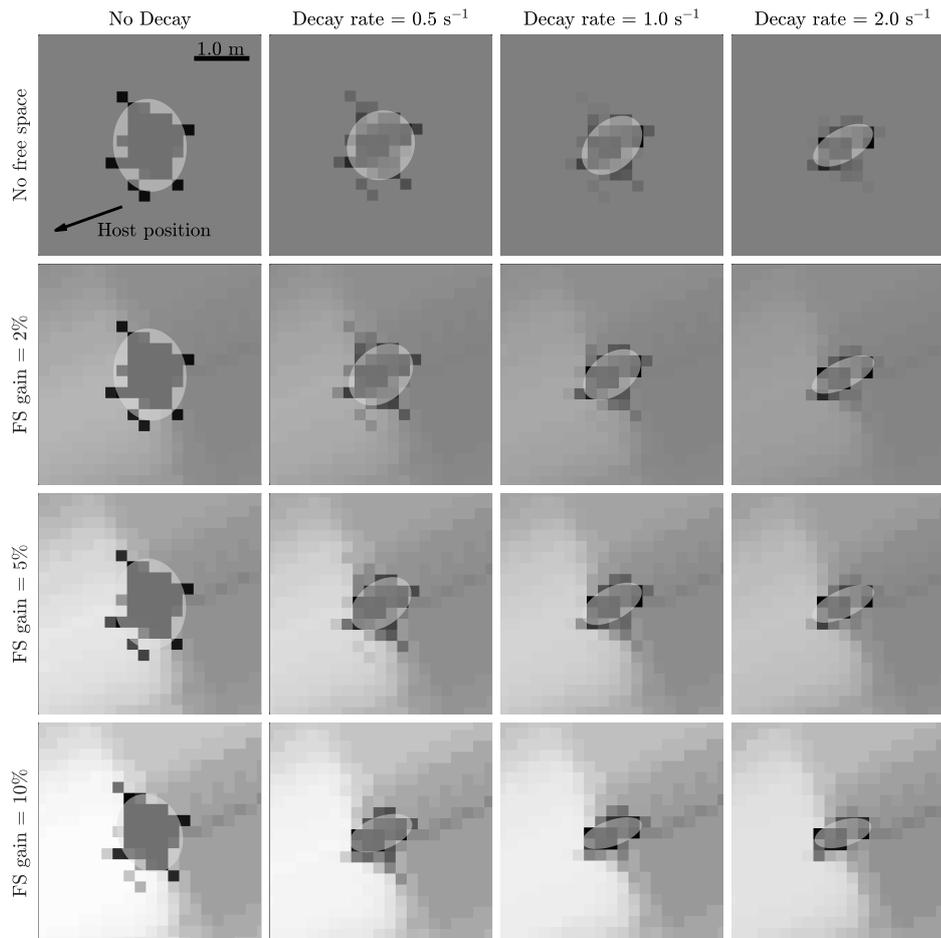


Figure B.3. Snapshots of the example landmark representation on the occupancy grid, generated using Bayesian fusion and the hit point inverse sensor model.

B.2.2 Bayesian grid fusion with two-dimensional inverse sensor model

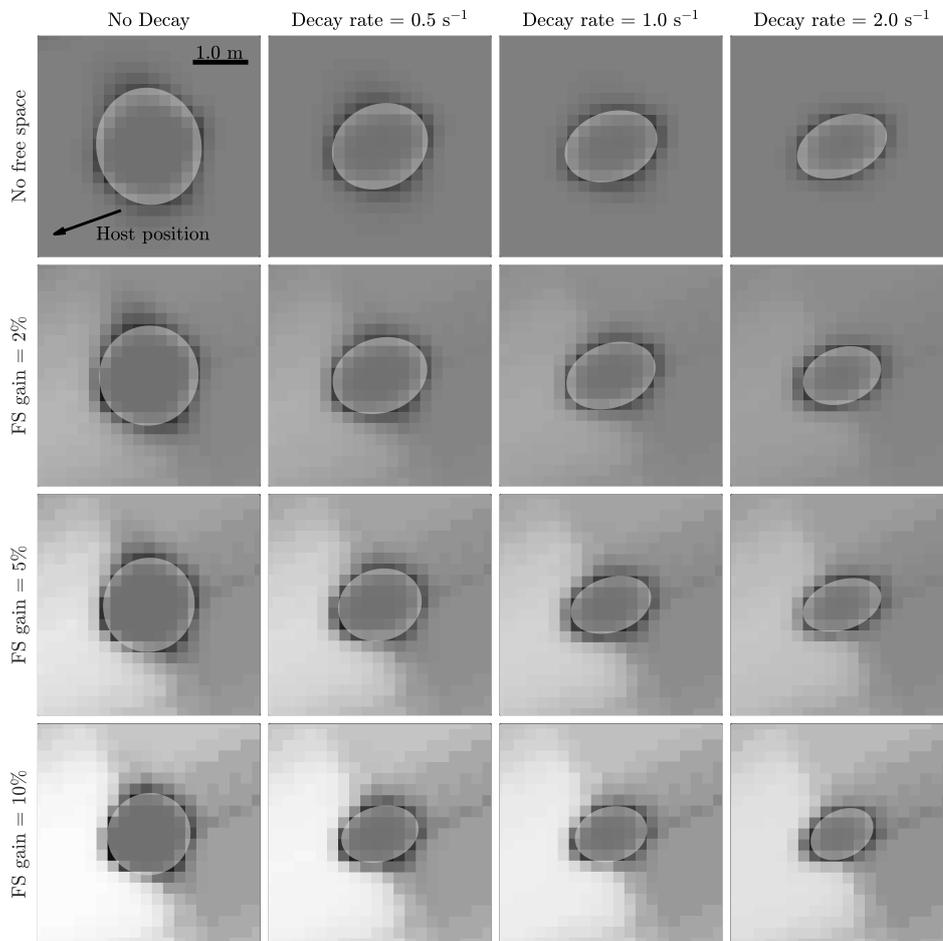


Figure B.4. Snapshots of the example landmark representation on the occupancy grid, generated using Bayesian fusion and the two-dimensional inverse sensor model.

B.2.3 Dempster combination rule with hit point inverse sensor model

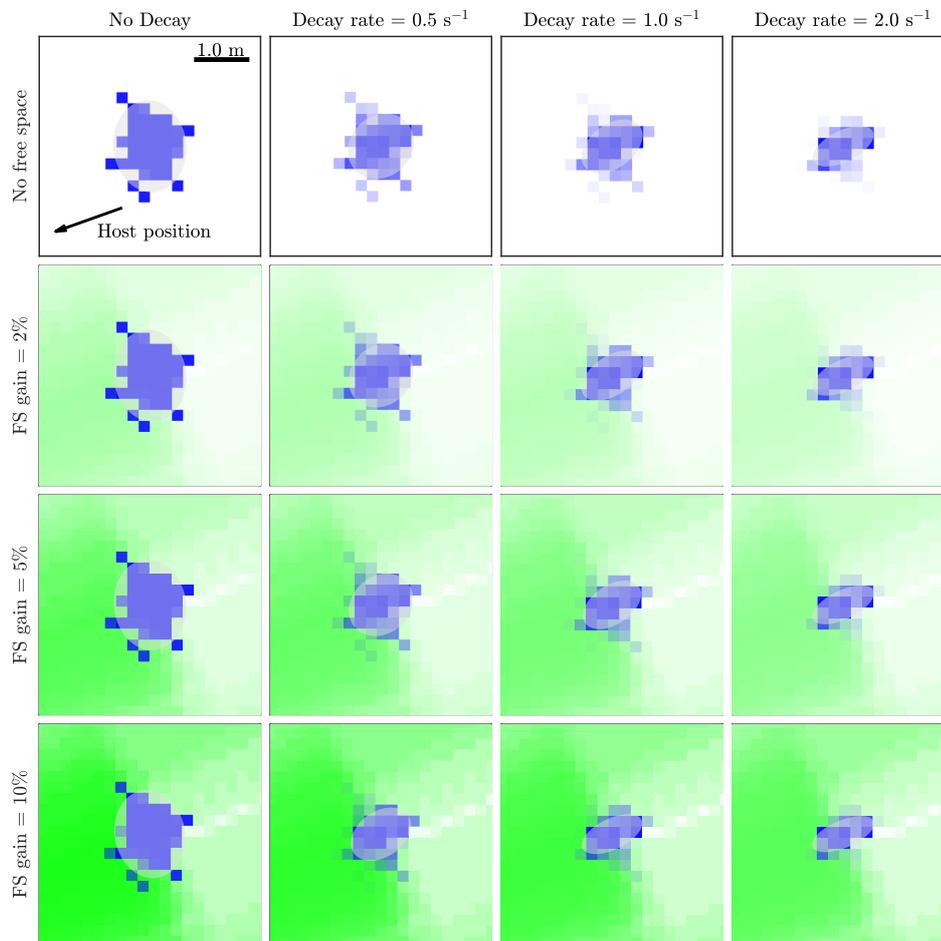


Figure B.5. Snapshots of the example landmark representation on the occupancy grid, generated using Dempster combination rule and the hit point inverse sensor model.

B.2.4 Dempster combination rule with two-dimensional inverse sensor model

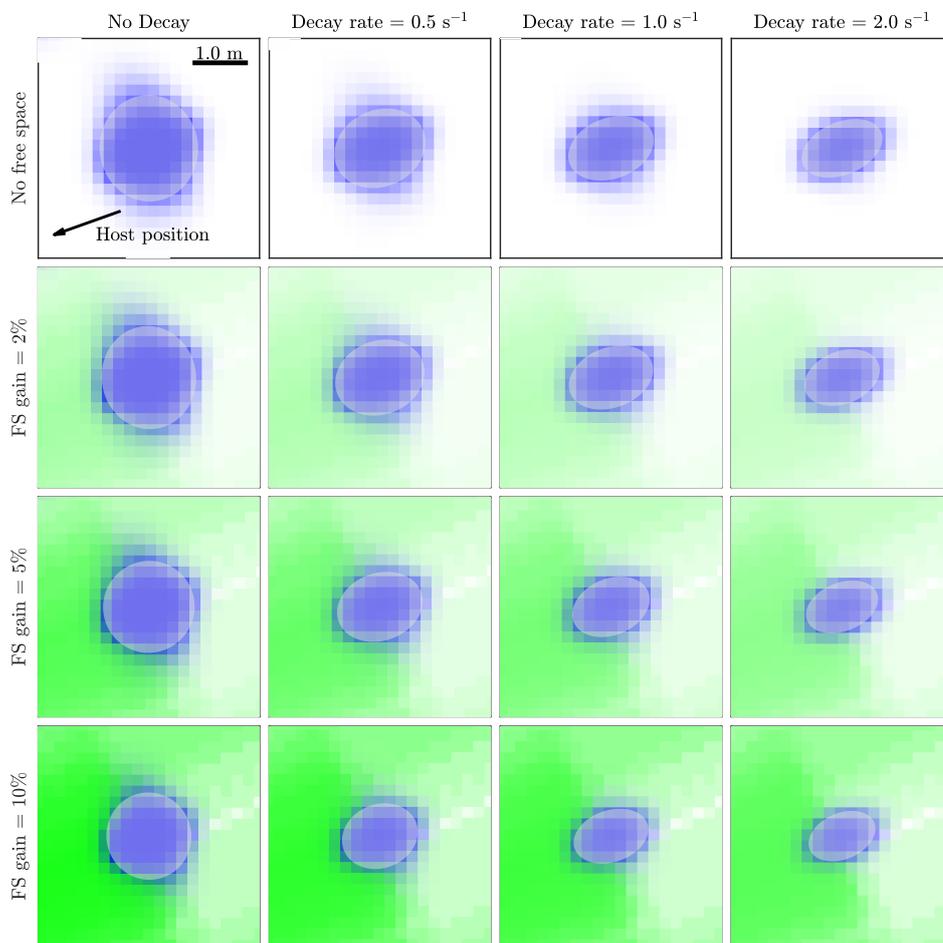


Figure B.6. Snapshots of the example landmark representation on the occupancy grid, generated using Dempster combination rule and the two-dimensional inverse sensor model.

B.2.5 Hybrid Dezert-Smarandache combination rule with hit point inverse sensor model

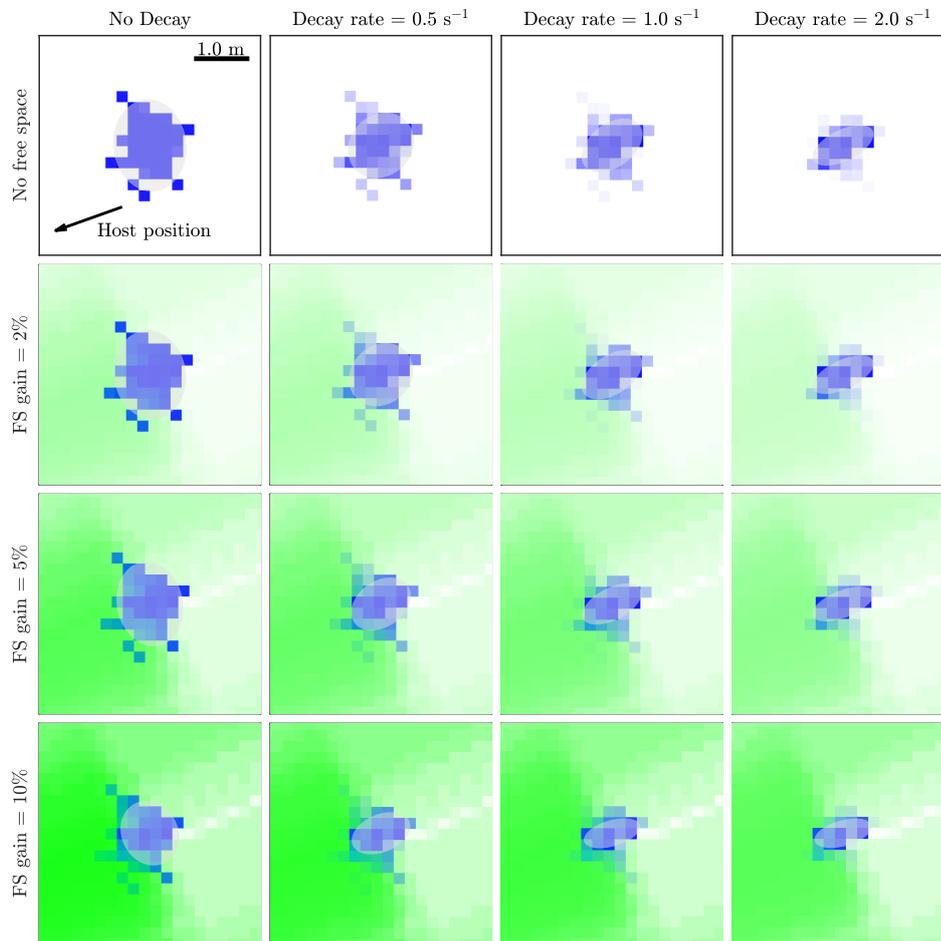


Figure B.7. Snapshots of the example landmark representation on the occupancy grid, generated using Hybrid Dezert-Smarandache combination rule and the hit point inverse sensor model.

B.2.6 Hybrid Dezert-Smarandache combination rule with two-dimensional inverse sensor model

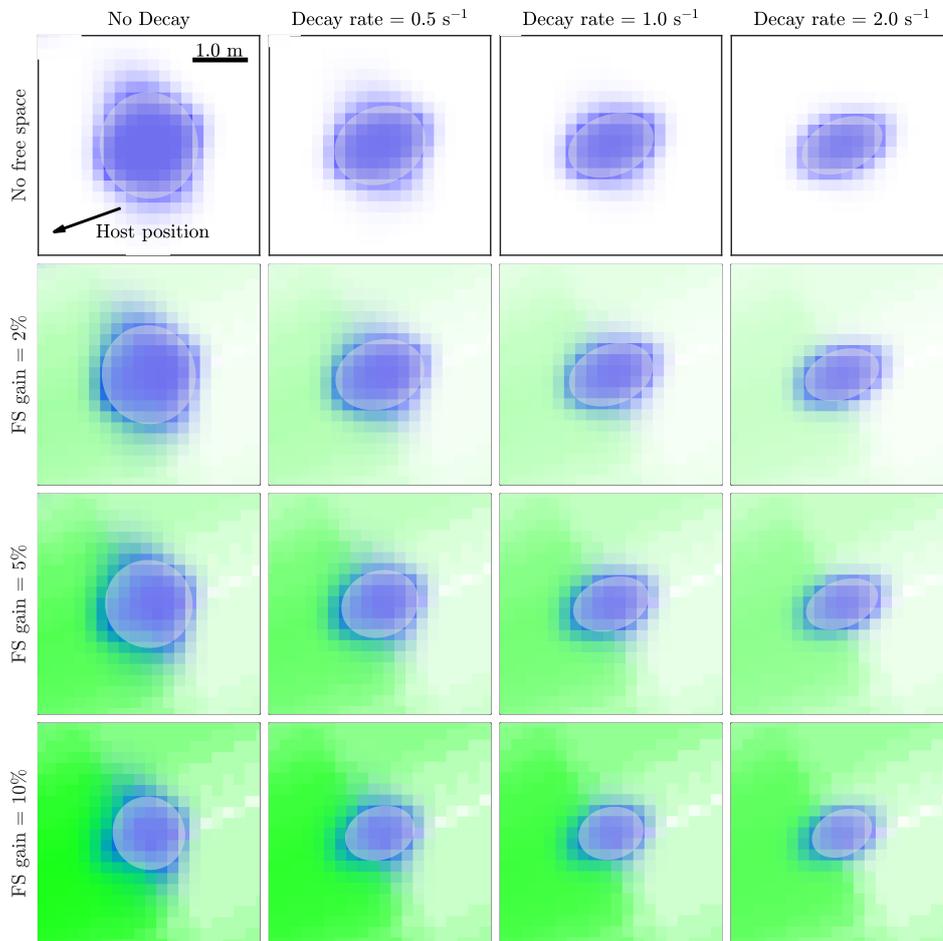


Figure B.8. Snapshots of the example landmark representation on the occupancy grid, generated using Hybrid Dezert-Smarandache combination rule and the two-dimensional inverse sensor model.

Bibliography

- T. R. Andriamahefa. *Integer Occupancy Grids: a probabilistic multi-sensor fusion framework for embedded perception*. PhD thesis, Université Grenoble Alpes, 2017.
- Anktiengesellschaft Volkswagen. Ces: From tech gadgets to digital mobility. URL www.volkswagenag.com, 1 2021. Accessed: 2021-08-07.
- Aptiv, Audi, BaiduApollo, BMW, Continental, FCA, Here, Infineon, Intel, Volkswagen, and Daimler. Safety First for Automated Driving. Technical report, SaFAD, 2019.
- Autonomous Stuff. Radar comparison chart. URL autonomoustuff.com, 3 2021. Accessed: 2021-03-08.
- AUTOSAR. Time Synchronization Protocol Specification. URL www.autosar.org, Nov. 2019. Accessed: 2021-09-28.
- AUTOSAR. Autosar (automotive open system architecture). specifications, release 4.2. URL www.autosar.org, 2021. Accessed: 2021-08-07.
- B. Balaguer, S. Balakirsky, S. Carpin, and A. Visser. Evaluating maps produced by urban search and rescue robots: Lessons learned from RoboCup. *Autonomous Robots*, 27(4):449–464, 11 2009. ISSN 0929-5593, 1573-7527.
- D. Bauer, L. Kuhnert, and L. Eckstein. Deep, spatially coherent inverse sensor models with uncertainty incorporation using the evidential framework. *arXiv preprint arXiv:1904.00842*, 2019.
- E. Bieńkowska. Road safety: Commission welcomes agreement on new EU rules to help save lives. URL ec.europa.eu, Mar. 2019. Accessed: 2021-06-08.
- J. Carvalho and R. Ventura. Comparative evaluation of occupancy grid mapping methods using sonar sensors. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 889–896. Springer, 2013. doi: 10/gjmkwh.
- S. Challa and D. Koks. Bayesian and dempster-shafer fusion. *Sadhana*, 29(2):145–174, 2004. doi: 10/fr758h.
- D. Cieslar, K. Kogut, M. Rózewicz, and M. Orłowski. Dynamic input generation for the development of active safety perception algorithms. Technical report, SAE Technical Paper, 2016.
- R. Danescu, F. Oniga, and S. Nedevschi. Modeling and tracking the driving environment with a particle-based occupancy grid. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1331–1342, 2011.
- M. Daniel. What is it Hybrid DS_m Rule for Combination of Belief Functions? URL www.utia.cas.cz, 2014. Accessed: 2021-10-05.

- M. Daniel and V. Kratochvíl. Hidden Conflicts of Belief Functions. *International Journal of Computational Intelligence Systems*, 14(1):438, 2020. ISSN 1875-6883. doi: 10/gjvhkc.
- J. Dezert. Foundations for a new theory of plausible and paradoxical reasoning. *Information and Security*, 9:13–57, 2002.
- J. Dezert, P. Wang, and A. Tchamova. On The Validity of Dempster-Shafer Theory. In *15th International Conference on Information Fusion (Fusion)*, Singapur, Singapore, July 2012. Accessed: 2021-12-29.
- J. Dezert, J. Moras, and B. Pannetier. Environment perception using grid occupancy estimation with belief functions. In *18th International Conference on Information Fusion (Fusion)*, pages 1070–1077. IEEE, 2015.
- J. Dickmann, N. Appenrodt, J. Klappstein, H.-L. Bloecher, M. Muntzinger, A. Sailer, M. Hahn, and C. Brenk. Making bertha see even more: Radar contribution. *IEEE Access*, 3:1233–1247, 2015.
- K. C. Dietmayer, S. Reuter, and D. Nuss. Representation of fused environment data. *Handbook of Driver Assistance Systems: Basic Information, Components and Systems for Active Safety and Comfort*, pages 1–30, 2014.
- W. Ding, L. Zhang, J. Chen, and S. Shen. Safe Trajectory Generation for Complex Urban Environments Using Spatio-temporal Semantic Corridor. *arXiv:1906.09788 [cs]*, June 2019. Accessed: 2019-09-16.
- A. Elfes. Occupancy grids: A probabilistic framework for robot perception and navigation. *Ph. D Thesis, Carnegie-Mellon University*, 1989.
- A. Elfes and L. Matthies. Sensor integration for robot navigation: combining sonar and stereo range data in a grid-based representataion. In *Decision and Control, 1987. 26th IEEE Conference on*, volume 26, pages 1802–1807. IEEE, 1987.
- A. Eskandarian. Fundamentals of driver assistance. *Handbook of intelligent vehicles*, pages 491–535, 2012.
- P. Fankhauser and M. Hutter. A universal grid map library: Implementation and use case for rough terrain navigation. In *Robot Operating System (ROS)*, pages 99–120. Springer, 2016.
- L. Feng, K. Liu, F. Tang, and Q. Meng. GO-DBSCAN: Improvements of DBSCAN Algorithm Based on Grid. *International Journal of Computer Theory and Engineering*, 9(3):5, 2017. doi: 10/gkx3d7.
- D. Filliat and J.-A. Meyer. Map-based navigation in mobile robots:: I. a review of localization strategies. *Cognitive Systems Research*, 4(4):243–282, 2003.
- M. Foroughi, U. Iurgel, A. Ioffe, and W. Doerr. Free space grid for automotive radar sensors. In *FAST-zero'15: 3rd International Symposium on Future Active Safety Technology Toward zero traffic accidents, 2015*, 2015.
- M. Galvani. History and future of driver assistance. *IEEE Instrumentation & Measurement Magazine*, 22(1):11–16, Feb. 2019. ISSN 1094-6969, 1941-0123. doi: 10/gmfgr3.
- C. Gálvez del Postigo Fernández. Grid-based multi-sensor fusion for on-road obstacle detection: Application to autonomous driving, 2015.
- I. D. Gebru, X. Alameda-Pineda, F. Forbes, and R. Horaud. EM Algorithms for Weighted-Data Clustering with Application to Audio-Visual Scene Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(12):2402–2415, Dec. 2016. ISSN 0162-8828, 2160-9292. doi: 10/gg7ptt.

- F. Gies, A. Danzer, and K. Dietmayer. Environment perception framework fusing multi-object tracking, dynamic occupancy grid maps and digital maps. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3859–3865. IEEE, 2018.
- A. Gómez-Corral, D. R. Insua, F. Ruggeri, and M. Wiper. Bayesian Inference of Markov Processes. In *Wiley StatsRef: Statistics Reference Online*, pages 1–15. American Cancer Society, 2015. ISBN 978-1-118-44511-2. doi: 10.1002/9781118445112.stat07837.
- J. Godoy, V. Jiménez, A. Artuñedo, and J. Villagra. A Grid-Based Framework for Collective Perception in Autonomous Vehicles. *Sensors*, 21(3):744, Jan. 2021. ISSN 1424-8220. doi: 10/gjkrfx.
- R. Grewe, M. Komar, A. Hohm, S. Lueke, and H. Winner. Evaluation method and results for the accuracy of an automotive occupancy grid. In *2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012)*, pages 19–24, Istanbul, Turkey, 7 2012. IEEE. ISBN 978-1-4673-0993-6 978-1-4673-0992-9 978-1-4673-0991-2. doi: 10.1109/ICVES.2012.6294297.
- J. Guo, U. Kurup, and M. Shah. Is it safe to drive? an overview of factors, challenges, and datasets for driveability assessment in autonomous driving. *arXiv preprint arXiv:1811.11277*, 2018.
- J. Happian-Smith, editor. *An introduction to modern vehicle design*. Butterworth-Heinemann, Oxford, nachdr. edition, 2006. ISBN 978-0-7506-5044-1.
- F. Homm, N. Kaempchen, J. Ota, and D. Burschka. Efficient occupancy grid computation on the gpu with lidar and radar for road boundary detection. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 1006–1013. IEEE, 2010.
- International Electrotechnical Commission. Iec 61508 functional safety. URL www.iec.ch, 2021. Accessed: 2021-08-07.
- International Organization for Standardization. ISO/IEC 15504-5:2012 Information technology — Process assessment — Process assessment model for software life cycle processes. URL www.automotivespice.com, 2012. Accessed: 2021-12-07.
- International Organization for Standardization. Iso/iec 27002: Information technology — security techniques — code of practice for information security controls. URL www.iso.org, 2013. Accessed: 2021-08-07.
- International Organization for Standardization. Iso/pas 21448: Road vehicles — safety of the intended functionality. URL www.iso.org, 2019. Accessed: 2021-08-07.
- International Organization for Standardization. ISO/DIS 23150: Road vehicles - data communication between sensors and data fusion unit for automated driving functions - logical interface, 2021.
- D. Joubert. *Adaptive occupancy grid mapping with measurement and pose uncertainty*. PhD thesis, Stellenbosch: Stellenbosch University, 2012.
- A.-L. Jousselme, D. Grenier, and E. Bosse. A new distance between two bodies of evidence. *Information Fusion*, 2(2):91–101, June 2001. ISSN 15662535. doi: 10/bf7234.
- A. Jøsang, S. Pope, and M. Daniel. Conditional Deduction Under Uncertainty. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, volume 3571, pages 824–835. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. ISBN 978-3-540-27326-4 978-3-540-31888-0. doi: 10.1007/11518655_69.

- M. N. Khan and S. Anwar. Paradox Elimination in Dempster–Shafer Combination Rule with Novel Entropy Function: Application in Decision-Level Multi-Sensor Fusion. *Sensors*, 19(21):4810, Nov. 2019. ISSN 1424-8220. doi: 10.3390/s19214810.
- K. Konolige. Improved occupancy grids for map building. *Autonomous Robots*, 4(4):351–367, 1997.
- J. Korta, P. Skruch, and K. Holon. Reliability of Automotive Multidomain Controllers: Advancements in electronics cooling technologies. *IEEE Vehicular Technology Magazine*, 16(2):86–94, June 2021. ISSN 1556-6072, 1556-6080. doi: 10.1109/MVT.2020.3038588.
- F. Kunz, D. Nuss, J. Wiest, H. Deusch, S. Reuter, F. Gritschneider, A. Scheel, M. Stübler, M. Bach, P. Hatzelmann, et al. Autonomous driving at ulm university: A modular, robust, and sensor-independent fusion approach. In *2015 IEEE intelligent vehicles symposium (IV)*, pages 666–673. IEEE, 2015.
- J. Laconte, C. Debain, R. Chapuis, F. Pomerleau, and R. Aufrère. Lambda-Field: A Continuous Counterpart of the Bayesian Occupancy Grid for Risk Assessment. *arXiv:1903.02285 [cs]*, 3 2019.
- P. Ledent, A. Paigwar, A. Renzaglia, R. Mateescu, and C. Laugier. Formal Validation of Probabilistic Collision Risk Estimation for Autonomous Driving. In *CIS-RAM 2019 - 9th IEEE International Conference on Cybernetics and Intelligent Systems (CIS) Robotics, Automation and Mechatronics (RAM)*, pages 1–6, Bangkok, Thailand, 11 2019. IEEE. doi: hal-02355551.
- Z. Liu, E. Blasch, G. Bhatnagar, V. John, W. Wu, and R. S. Blum. Fusing synergistic information from multi-sensor images: an overview from implementation to performance assessment. *Information Fusion*, 42:127–145, 2018.
- P. Markiewicz and J. Porębski. Developing Occupancy Grid with Automotive Simulation Environment. *Applied Sciences*, 10(21):16, 2020. doi: 10.3390/app10217629.
- P. Markiewicz, K. Kogut, M. Rózewicz, P. Skruch, and R. Starosolski. Occupancy grid fusion prototyping using automotive virtual validation environment. In *Proceedings of the 6th International Conference on Control, Mechatronics and Automation*, pages 81–85. ACM, 2018.
- A. Martin, A.-L. Jousselme, and C. Osswald. Conflict measure for the discounting operation on belief functions. In *2008 11th International conference on information fusion*, pages 1–8. IEEE, 2008.
- J.-A. Meyer and D. Filliat. Map-based navigation in mobile robots:: Ii. a review of map-learning and path-planning strategies. *Cognitive Systems Research*, 4(4):283–317, 2003.
- A. Milstein. Occupancy grid maps for localization and mapping. In *Motion planning*. IntechOpen, 2008.
- C. MISRA. Misra c:2012 (misra c3). URL www.misra.org.uk, 2021. Accessed: 2021-08-07.
- J. Moras, J. Dezert, and B. Pannetier. Grid occupancy estimation for environment perception based on belief functions and PCR6. In *SPIE Defense + Security Conference*, volume 9474, pages 208 – 220, Baltimore, Maryland, United States, 2015. SPIE. doi: 10/gjs9t8.
- H. P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI magazine*, 9(2):61, 1988.
- A. Nègre, L. Rummelhard, and C. Laugier. Hybrid sampling bayesian occupancy filter. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 1307–1312. IEEE, 2014.
- D. Nuß. *A random finite set approach for dynamic occupancy grid maps*. PhD thesis, Universität Ulm, 2017.

- B. Odgerel and C.-H. Lee. An Improved Dempster-Shafer Algorithm Using a Partial Conflict Measurement. *The International Journal of Fuzzy Logic and Intelligent Systems*, 16(4):308–317, Dec. 2016. ISSN 1598-2645, 2093-744X. doi: 10.5391/IJFIS.2016.16.4.308.
- F. Oniga and S. Nedeveschi. Processing dense stereo data using elevation maps: Road surface, traffic isle, and obstacle detection. *IEEE Transactions on Vehicular Technology*, 59(3):1172–1182, 2010.
- C. Osswald and A. Martin. Understanding the large family of Dempster-Shafer theory’s fusion operators – a decision-based measure. In *2006 9th International Conference on Information Fusion*, pages 1–7, Florence, July 2006. IEEE. ISBN 978-1-4244-0953-2 978-0-9721844-6-5. doi: 10/fnwp6f.
- D. S. Pae, Y. S. Jang, S. K. Park, and M. T. Lim. Track Compensation Algorithm Using Free Space Information with Occupancy Grid Map. *International Journal of Control, Automation and Systems*, 19(1):40–53, Jan. 2021. ISSN 1598-6446, 2005-4092. doi: 10/gjh3v7.
- W. Payre, S. Birrell, and A. M. Parkes. Although autonomous cars are not yet manufactured, their acceptance already is. *Theoretical Issues in Ergonomics Science*, 22(5):567–580, Sept. 2021. ISSN 1463-922X, 1464-536X. doi: 10.1080/1463922X.2020.1836284.
- D. Pfeiffer and U. Franke. Modeling dynamic 3d environments by means of the stixel world. *IEEE Intelligent Transportation Systems Magazine*, 3(3):24–36, 2011.
- S. Pietzsch, T. D. Vu, J. Bulet, O. Aycard, T. Hackbarth, N. Appenrodt, J. Dickmann, and B. Radig. Results of a precrash application based on laser scanner and short-range radars. *IEEE Transactions on Intelligent Transportation Systems*, 10(4):584–593, 2009.
- J. Porębski and K. Kogut. Performance Evaluation of the Highway Radar Occupancy Grid. *Sensors*, 21(6):2177, Mar. 2021. ISSN 1424-8220. doi: 10/gjk3p6.
- J. Porębski, K. Kogut, P. Markiewicz, and P. Skruch. Occupancy grid for static environment perception in series automotive applications. *IFAC-PapersOnLine*, 52(8):148–153, 2019. ISSN 24058963. doi: 10.1016/j.ifacol.2019.08.063.
- J. Porębski. Customizable Inverse Sensor Model for Bayesian and Dempster-Shafer Occupancy Grid Frameworks. In *Advanced, Contemporary Control*, volume 1196, pages 1225–1236. Springer International Publishing, Cham, 2020. ISBN 978-3-030-50935-4 978-3-030-50936-1. doi: 10.1007/978-3-030-50936-1_102.
- T. Rakotovo, J. Mottin, D. Puschini, and C. Laugier. Multi-sensor fusion of occupancy grids based on integer arithmetic. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1854–1859. IEEE, 2016.
- SAE International. Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems, 2014.
- N. H. Saleem, H.-J. Chien, M. Rezaei, and R. Klette. Effects of ground manifold modeling on the accuracy of stixel calculations. *IEEE Transactions on Intelligent Transportation Systems*, 2018.
- M. Saval-Calvo, L. Medina-Valdés, J. M. Castillo-Secilla, S. Cuenca-Asensi, A. Martínez-Álvarez, and J. Villagrà. A review of the bayesian occupancy filter. *Sensors*, 17(2):344, 2017.
- C. Seeger, A. Müller, L. Schwarz, and M. Manz. Towards road type classification with occupancy grids. In *IEEE Intelligent Vehicles Symposium (IV) Workshop: DeepDriving-Learning Representations for Intelligent Vehicles*, pages 1–4, 2016.

- G. Shafer. *A mathematical theory of evidence*, volume 42. Princeton university press, 1976. ISBN 0-691-10042-X.
- P. Skruch, R. Długosz, K. Kogut, P. Markiewicz, D. Sasin, and M. Różewicz. The Simulation Strategy and Its Realization in the Development Process of Active Safety and Advanced Driver Assistance Systems. In *SAE 2015 World Congress & Exhibition*, pages 2015–01–1401, 4 2015. doi: 10.4271/2015-01-1401.
- F. Smarandache and J. Dezert. *Advances And Applications Of Dsmt For Information Fusion*. Zenodo, Apr. 2004. doi: 10.5281/ZENODO.8791.
- F. Smarandache and J. Dezert. Proportional Conflict Redistribution Rules for Information Fusion. *arXiv:cs/0408064*, Mar. 2005.
- P. Smets. The combination of evidence in the transferable belief model. *IEEE Transactions on pattern analysis and machine intelligence*, 12(5):447–458, 1990. doi: 10/d54bzt.
- C. Stachniss. *Exploration and mapping with mobile robots*. PhD thesis, University of Freiburg, 2006.
- J. Steinbaeck, C. Steger, G. Holweg, and N. Druml. Next generation radar sensors in automotive sensor fusion systems. In *2017 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pages 1–6. IEEE, 2017.
- P. Stepan, M. Kulich, and L. Preucil. Robust data fusion with occupancy grid. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 35(1):106–115, 2005.
- S. Steyer, G. Tanzmeister, and D. Wollherr. Object tracking based on evidential dynamic occupancy grids in urban environments. In *Intelligent Vehicles Symposium (IV), 2017 IEEE*, pages 1064–1070. IEEE, 2017.
- S. Steyer, G. Tanzmeister, and D. Wollherr. Grid-based environment estimation using evidential mapping and particle tracking. *IEEE Transactions on Intelligent Vehicles*, 2018.
- S. Steyer, C. Lenk, D. Kellner, G. Tanzmeister, and D. Wollherr. Grid-Based Object Tracking With Nonlinear Dynamic State and Shape Estimation. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–20, 2019. ISSN 1524-9050, 1558-0016. doi: 10.1109/TITS.2019.2921248.
- M. Szlachetka, D. Borkowski, and J. Was. Stationary environment models for Advanced Driver Assistance Systems. In *2020 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, pages 116–121, 2020. doi: 10/gjkrtd.
- K. Takahashi. Curvature Blindness Illusion. *i-Perception*, 8(6), Dec. 2017. ISSN 2041-6695. doi: 10/gcm3tb.
- G. Tanzmeister and S. Steyer. Spatiotemporal alignment for low-level asynchronous data fusion with radar sensors in grid-based tracking and mapping. In *2016 IEEE International Conference on Multi-sensor Fusion and Integration for Intelligent Systems (MFI)*, pages 231–237, Baden-Baden, Germany, Sept. 2016. IEEE, IEEE. ISBN 978-1-4673-9708-7. doi: 10.1109/MFI.2016.7849494.
- G. Tanzmeister, J. Thomas, D. Wollherr, and M. Buss. Grid-based mapping and tracking in dynamic environments using a uniform evidential environment representation. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 6090–6095. IEEE, 2014.
- S. Thrun. Learning occupancy grid maps with forward sensor models. *Autonomous robots*, 15(2):111–127, 2003.

- S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT Press, Cambridge, Mass., 2005. ISBN 0-262-20162-3 978-0-262-20162-9.
- R. Triebel, P. Pfaff, and W. Burgard. Multi-level surface maps for outdoor terrain mapping and loop closing. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 2276–2282. IEEE, 2006.
- C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, et al. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, 25(8):425–466, 2008.
- M. Valente, C. Joly, and A. De La Fortelle. Grid matching localization on evidential slam. In *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 1477–1483. IEEE, 2018.
- A. I. Wagan, A. Godil, and X. Li. Map quality assessment. In *Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems - PerMIS '08*, page 278, Gaithersburg, Maryland, 2008. ACM Press. ISBN 978-1-60558-293-1. doi: 10/cztdvx.
- X. Wang, W. Wang, X. Yin, C. Xiang, and Y. Zhang. A New Grid Map Construction Method for Autonomous Vehicles. *IFAC-PapersOnLine*, 51(31):377–382, 2018. ISSN 24058963. doi: 10.1016/j.ifacol.2018.10.077.
- T. Weiss, B. Schiele, and K. Dietmayer. Robust driving path detection in urban and highway scenarios using a laser scanner and online occupancy grids. In *Intelligent Vehicles Symposium, 2007 IEEE*, pages 184–189. IEEE, 2007.
- K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard. Octomap: A probabilistic, flexible, and compact 3d map representation for robotic systems. In *Proc. of the ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation*, volume 2, 2010.
- R. R. Yager. On the dempster-shafer framework and new combination rules. *Information Sciences*, 41(2):93–137, Mar. 1987. ISSN 00200255. doi: 10.1016/0020-0255(87)90007-7.
- R. R. Yager. Entropy and specificity in a mathematical theory of evidence. In *Classic Works of the Dempster-Shafer Theory of Belief Functions*, pages 291–310. Springer, 2008. ZSCC: 0000562.
- M. Yguel, O. Aycard, and C. Laugier. Efficient GPU-based construction of occupancy grids using several laser range-finders. *International Journal of Vehicle Autonomous Systems*, 6(1/2):48, 2008. ISSN 1471-0226, 1741-5306. doi: 10.1504/IJVAS.2008.016478.
- Z. Yi. *Multi-Ultrasonic Sensor Fusion for Mobile Robots in Confined Spaces*. PhD thesis, Nanyang Technological University, 2001.
- Z. Yi, Y. K. Ho, C. S. Chua, and X. W. Zhou. Multi-ultrasonic sensor fusion for autonomous mobile robots. In B. V. Dasarathy, editor, *Sensor Fusion: Architectures, Algorithms, and Applications IV*, volume 4051, pages 314–322, Orlando, FL, 2000. International Society for Optics and Photonics. doi: 10.1117/12.381644.
- L. A. Zadeh. A Simple View of the Dempster-Shafer Theory of Evidence and its Implication for the Rule of Combination. In *Advances in Fuzzy Systems — Applications and Theory*, volume 6, pages 674–679. WORLD SCIENTIFIC, May 1996. ISBN 978-981-02-2421-9 978-981-4261-30-2. doi: 10.1142/9789814261302_0033.

- L. Zheng, B. Li, B. Yang, H. Song, and Z. Lu. Lane-level road network generation techniques for lane-level maps of autonomous vehicles: A survey. *Sustainability*, 11(16):4511, Aug 2019. ISSN 2071-1050. doi: 10.3390/su11164511.
- J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller, E. Kaus, R. G. Herrtwich, C. Rabe, D. Pfeiffer, F. Lindner, F. Stein, F. Erbs, M.ENZweiler, C. Knoppel, J. Hipp, M. Haueis, M. Trepte, C. Brenk, A. Tamke, M. Ghanaat, M. Braun, A. Joos, H. Fritz, H. Mock, M. Hein, and E. Zeeb. Making berththa drive—an autonomous journey on a historic route. *IEEE Intelligent Transportation Systems Magazine*, 6(2):8–20, Summer 2014. ISSN 1939-1390. doi: 10.1109/MITS.2014.2306552.

List of Figures

2.1	A modern vehicle control system architecture based on MDCs (inspired by [Korta et al., 2021]).	26
2.2	Levels of driving automation (inspired by [Andriamahefa, 2017]).	27
2.3	V-model development process diagram.	30
2.4	Cognitive information flow of the ADAS features with elements used to perform each step.	31
2.5	Smooth and sharp trajectories. The most viable trajectory according to road scene can be directly executed by the vehicle controller [Urmson et al., 2008].	34
2.6	Examples of information extracted from the occupancy grid. The grid representation (top) could be transformed depending on application into, for example (from left): contours, objects, free space envelope or a trajectory validation tool.	35
3.1	Robot and automotive occupancy grids. (a) Robot grids depict whole operational area of robot (image adapted from [Stachniss, 2006]). (b) Automotive occupancy grid map only local environment and is shifted as the vehicle moves to the new area.	40
3.2	Coordinate systems used to position the occupancy grid in the world frame.	40
3.3	Examples of occupancy grid extensions for handling the dynamic objects and the multi-dimensional environment structure.	42
3.4	The tree structure of 2^d -trees with the corresponding spatial subdivision [Andriamahefa, 2017].	42
3.5	Comparison of the low- and high-level grid fusion architectures.	43
3.6	Occupancy grid data flow.	44
3.7	Sensor scan measurement types: (a) point (radar and LIDAR detections), (b) contour shape (vision free space envelope).	45
3.8	Graphical illustration of the sensor model data flow.	48
3.9	Types of independent evidences maintained in the proposed occupancy grid framework.	48
3.10	Decaying process of the occupied cell with $p = 0.9$ (solid lines) and free cell $p = 0.1$ (dotted lines) for different mean lifetimes τ . The decay is applied with 10 Hz frequency.	50
3.11	Graphical illustration of the grid shift procedure.	51
3.12	Types of host positioning (a) center, (b) constant radius [Porębski et al., 2019].	51
4.1	Fusion architectures designed for combination of intermediate grids into prior occupancy grid cell states. The symbol \otimes represents fusion operation.	59
4.2	Brightness contrast illusion in the gray-scale occupancy grid and proposed grid structured pie charts as a grid visualization.	64
4.3	Legend for the pie chart cell state visualization.	64

4.4	Presentation of quantitative conflict metrics for the example Dempster-Shafer's grid snapshot. The gray-scale for entropy and auto conflict is changed in a way that white areas represent more inconsistency in the cell evidence. Most of the conflicts are visible in the dynamic residual area (red area in the middle of the top left grid image and as an envelope of stationary object (blue area in the bottom of the top left grid image)).	65
4.5	Developed hybrid fusion architectures tailored for usage within different fusion frameworks. Symbol \odot represents conjunctive fusion rule, DS – Dempster rule of combination, $DSmC$ and $DSmH$ – classical and hybrid Dezert-Smarandache rules accordingly.	67
4.6	Graphical visualization of fusion example – Iteration 1: Fusion into initial cell state.	72
4.7	Graphical visualization of fusion example – Iteration 2: Instant sensor conflict.	73
4.8	Graphical visualization of fusion example – Iteration 2: Multi sensor conflict.	73
5.1	Coordinate transformation chain connecting sensor detection and grid measurement representation.	78
5.2	Schematic representation of four types of uncertainties present in the occupancy grid algorithm.	78
5.3	Examples of the occupancy grid representations of highway pole-like objects with corresponding video frames: (a) traffic bollard; (b) sign; (c) guardrail. The occupancy grid is generated using a single automotive-graded radar located in front of the vehicle (Image reused from: [Porębski and Kogut, 2021]).	81
5.4	Comparison of clusters extracted from the Bayesian and Dempster-Shafer occupancy grid using the same binarization threshold levels. Clustering based on probabilities in Bayesian grid cannot deal with the trail of dynamic occupancy (violet cluster on the left). Clustering based on only stationary belief values in the DS grid (right) filters out dynamic and ambiguous occupancy from the cluster representation.	82
5.5	Schematic visualization of key performance indicators (KPI). (Image reused from: [Porębski and Kogut, 2021])	83
5.6	Types of sensor models: (a) hit point; (b) 2D approximation of detection uncertainty. Darker areas represent cells with higher probabilities.	86
5.7	Example of the segment free space sensor model. Area inside the envelope is filled uniformly with free evidence equal to the free space gain.	86
6.1	Simulation environment used for the evaluation of the occupancy grid. All simulations were prepared using <i>Driving Scenario Designer</i> application in MathWorks MATLAB.	90
6.2	Snapshots of the two occupancy grids generated using the simulation environment without any filtering enabled. Figure on the right presents a grid with applied example noise parameters.	90
6.3	Grid dimensions and host placement in simulation scenario.	91
6.4	Grid snapshots of pole representation with different sensor noise types. Images represent the pole on the left side of the vehicle (host position is pointed by the arrow). Host is in the middle of the S-turn maneuver and is moving from the left to right (relative to the images). The transparent ellipses depict the area of the landmark's Gaussian representation.	93
6.5	Distributions of landmark representation areas for different types of sensor uncertainties. When possible, uncertainty levels were selected manually to align the mean value of each representation.	94
6.6	Plots showing mean landmark representation area for different noise types and levels.	95

6.7	Grid snapshots of the sample pole representation with different types of ISM and uncertainty levels equal to $\sigma_r = 0.25$ m, $\sigma_\varphi = 0.3^\circ$. Grid is generated using DSmH fusion, decay and free space filtering are disabled.	96
6.8	Distribution of (a) Area and Consistency (b) values depending on the type of inverse sensor model applied.	97
6.9	Grid snapshots of the sample pole representation with different decay rates and uncertainty levels equal to $\sigma_r = 0.25$ m, $\sigma_\varphi = 0.3^\circ$. Grid is generated using DSmH fusion and with Cartesian 2D ISM.	97
6.10	Area and consistency distributions of landmark representation generated with different decay rates.	98
6.11	Area and entropy distributions of landmark representation generated with different free space gains.	99
6.12	Grid snapshots of the sample pole representation with different free space gains and uncertainty levels equal to $\sigma_r = 0.25$ m, $\sigma_\varphi = 0.3^\circ$. Grid is generated using DSmH fusion and with Cartesian 2D ISM.	100
6.13	Grid snapshots of the sample pole representation with different combination rules and uncertainty levels equal to $\sigma_r = 0.25$ m, $\sigma_\varphi = 0.3^\circ$. Grid is generated using Cartesian ISM with the decay rate of 1 s^{-1} and 5% free space gain.	100
6.14	Area, consistency, and entropy distributions of landmark representation generated with different fusion rules. The occupancy grid used Cartesian ISM with the decay rate of 1 s^{-1} and 5% free space gain.	101
6.15	Mean area of the pole representation with different combination rules. Light gray areas represent better pole representation.	103
6.16	Consistency of the pole representation with different combination rules. Light gray areas represent better pole consistency, but the average consistency level for the hit point ISM is 0.7, while in the Cartesian 2D ISM it oscillates around 0.96.	104
6.17	Distribution of each pole maximum entropy with different combination rules. White areas of the plot represent configurations with the lowest entropy (minimal sensor conflict).	105
7.1	Experimental vehicle Peugeot 3008 used to test capabilities of presented algorithm. Vehicle is equipped with four corner radars and reference camera system. High resolution LIDAR visible on the roof was disabled for presented experiments.	107
7.2	Sensor coverage of the test vehicle. For readability, only half of the radar range is presented. Plotted coverage is up to 50 m, while actual radars can detect objects up to 100 m.	108
7.3	Presentation of the two motorway's signs used for the grid tuning guidelines. Image (a) presents the real photo of the signs, and (b) shows the raw occupancy grid image of the poles without any filtering.	109
7.4	Example signs grid representations produced using different ISM uncertainty levels. For further processing the values of $\sigma_r = 0.2$ m, $\sigma_\varphi = 0.4^\circ$ were selected.	110
7.5	Comparison of different filtering options for the pole object. Grid snapshots were generated using DSmH fusion rule and Cartesian 2D ISM with uncertainties $\sigma_r = 0.2$ m, $\sigma_\varphi = 0.4^\circ$. From the evaluation, a sample with decay rate 1.0 s^{-1} and free space gain of 2% was selected for further analysis.	111
7.6	Presentation of the grid representation of the sign before and after tuning. Visualized sign is the third sign in the video frame presented in Figure 7.3(a). Presented grid snapshots depict the sign as it looks like closer to the vehicle (100 ms after the camera frame).	112

7.7	Presentation of the grid before and after tuning. All three signs evaluated in this section can be identified on the right side of the road. Red trail represents dynamic occupancy residuals.	113
7.8	Example highway radar Bayesian occupancy grid presenting the overtaking dynamic object (zoomed area). The zoomed area presents the grid in form of a charts array.	114
7.9	Example highway radar Dempster-Shafer occupancy grid. Top image presents the colormap grid representation. Zoomed area (a) presents grid representation of a stationary bollard. The dynamic object and its residual are visible in red in part (b) of the Figure.	115
7.10	Front camera video frame presenting a stopped vehicle on the left turn of a crowded crossroad. As the road clears, the vehicle starts moving forward in order to perform the left turn.	116
7.11	Time-lapse of the starting vehicle presented on the occupancy grid. Stationary car wheels (blue) start moving, and the vehicle will be classified as dynamic (red).	116
7.12	Comparison of Bayesian, Dempster and DSmH fusion rules based on the starting vehicle example.	117
7.13	Comparison of Dempster and DSmH fusion rules based on the starting vehicle example.	118
7.14	Algorithm execution profiling cumulative density function collected for different occupancy grid scans on the Raspberry Pi 4B computer.	119
7.15	Grid output using different sensor sets: four corner radars (a), corner radars and front vision free space system (b) and LIDAR sensor (c). Reference camera image of the scene is presented at the bottom (d).	120
B.1	Ellipse snapshots for different threshold values. For generation of the occupancy grid, Cartesian ISM was used with decay rate 1.0 s^{-1} and 2% free space gain.	138
B.2	Plots showing mean area of pole representation and percentage of missed poles as a function of binarization threshold.	138
B.3	Snapshots of the example landmark representation on the occupancy grid, generated using Bayesian fusion and the hit point inverse sensor model.	139
B.4	Snapshots of the example landmark representation on the occupancy grid, generated using Bayesian fusion and the two-dimensional inverse sensor model.	140
B.5	Snapshots of the example landmark representation on the occupancy grid, generated using Dempster combination rule and the hit point inverse sensor model.	141
B.6	Snapshots of the example landmark representation on the occupancy grid, generated using Dempster combination rule and the two-dimensional inverse sensor model.	142
B.7	Snapshots of the example landmark representation on the occupancy grid, generated using Hybrid Dezert-Smarandache combination rule and the hit point inverse sensor model.	143
B.8	Snapshots of the example landmark representation on the occupancy grid, generated using Hybrid Dezert-Smarandache combination rule and the two-dimensional inverse sensor model.	144

List of Tables

2.1	Comparison of environment imaging sensor capabilities (extended from [Steinbaeck et al., 2017]). Symbol '✓✓' means very good performance, '✓' acceptable execution, '✗' and '✗✗' refers to low and very low measurement capability. Fields marked with '—' describe not available capabilities for the sensor. . . .	32
4.1	Intermediate grid input values for each fusion iteration.	72
4.2	Cell probabilities for the Bayesian fusion example	74
4.3	Mass values for the DST fusion example. 1 st step – conjunctive rule, 2 nd step – Dempster rule. 'N/A' refers to propositions equal to 0 by the definition of fusion rule or model. . . .	74
4.4	Mass values for the DSMT fusion example. 1 st step – DSMT, 2 nd step – DSMT. 'N/A' refers to propositions equal to 0 by the definition of fusion rule or model. Propositions beliefs, which are always zero regardless of the input data, has been omitted.	75
4.5	Quantitative measurement of probability and conflict in different fusion frameworks	76
7.1	Occupancy grid algorithm performance measurements for a highway scenario running on a single core of Raspberry Pi 4B computer (1.5GHz).	119
A.1	Operator $\alpha_i \cap \alpha_j$ for $i, j = 0, \dots, 7$ for the power set 2^Θ	127
A.2	Operator $\alpha_i \cap \alpha_j$ for $i, j = 0, \dots, 18$ for the hyper-power set D^Θ	128
A.3	Operator $u(\alpha_i) \cup u(\alpha_j)$ for $i, j = 0, 11, \dots, 18$ for the hyper-power set D^Θ	131
A.4	Operator $\alpha_i \cap \alpha_j$ for $i, j = 0, \dots, 18$ for the hyper-power set D^Θ with Smets constrains.	132
A.5	Operator $\alpha_i \cup \alpha_j$ for $i, j = 0, \dots, 18$ for the hyper-power set D^Θ	133