DEPARTMENT OF INFORMATION ENGINEERING

UNIVERSITÁ DEGLI STUDI DI PADOVA

Master's Thesis in Automation Enginerering

Map Validation for Autonomous Driving Systems

Andrea Fabris

DEPARTMENT OF INFORMATION ENGINEERING

UNIVERSITÁ DEGLI STUDI DI PADOVA

Master's Thesis in Automation Enginerering

Map Validation for Autonomous Driving Systems

Author:	Andrea Fabris
Supervisor:	Prof. Dr.-Ing. Angelo Cenedese
Advisor:	Dipl.-Inform. Sebastian Schneider, BMW AG
	Dr. Luca Parolini, BMW AG
Submission Date:	December 12, 2019

I confirm that this master's thesis in automation enginerering is my own work and I have documented all sources and material used.

Unterschleißheim, Germany, December 12, 2019 Andrea Fabris

Acknowledgments

First of all, I would like to express my sincere gratitude to my thesis advisors Dr. Luca Parolini and Dipl. Inform. Sebastian Schneider, for their motivation, suggestions, enthusiasm, and immense knowledge. They supported me during all the time of research and they also enabled me to collaborate in the writing an extend abstract for the *Computer Science in Car Symposium 2019*. Moreover, with Dr. Luca Parolini I had the pleasure of writing the patent *Use of probabilistc graphical model for map* filed at the *German Patent and Trademark Office* on August 30, 2019 (Application number: 10 2019 123 353.4)
Besides my advisors, I would like to thank a lot my academic supervisor Prof. Dr.-Ing. Angelo Cenedese for his encouragement, the useful advice, and all the help that made this internship possible. I am grateful to all the BMW Group for giving me the opportunity to contribute to the autonomous driving project and in particular I would like to thank all the members of the team Nash, who hosted me during my internship and have made the environment and work more enjoyable. Finally, I must express my very profound gratitude to my parents and to my girlfriend whose love and guidance are with me in whatever I pursue. I thank them for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis.

This accomplishment would not have been possible without all of them.

Abstract

High-definition mapping is one of the key technologies supporting self-driving vehicles. Especially in urban scenarios, the field of view of sensors is often limited, and so HD map data provides additional useful information about the road and its environment. Maps used in this field are high resolution with centimetre-level accuracy and their correctness is fundamental for decision making of autonomous-driving (*AD*) vehicles. In this thesis we describe and compare different approaches for online map validation (*OMV*) whose goal is to verify if reality and map data are inconsistent.
First of all, a probabilistic framework to perform the data sensor fusion is defined. After that a spatial correlation concept is introduced to interpolate the information and obtain a more reliable map validation. The result is a probabilistic representation of the map whose assumed values represent the probability with which the map is valid in every single point. Since, in a normal situation, the invalidation of the map will not allow an autonomous vehicle to continue driving, the problem of identifying which part of the map is valid (partially validate the map) becomes fundamental.

Keywords Autonomous driving, Sensor fusion, Bayesian network, Dempster-Shafer, Map validation

Contents

1. Introduction

Autonomous driving (*AD*) is an increasingly interesting area of research in both industry and University during the last years. Recently, autonomous cars (also called driverless cars, self-driving cars, or robotic cars) have become a concrete reality thanks to different companies that are working hard to be competitive on the market [1]. Autonomous driving vehicles (AVs) promise to provide numerous benefit to our society [2], ranging from a higher standard of transportation to traffic reduction.
Fully autonomous driving cars will potentially avoid thousands of car accidents every year since it could prevent many dangerous situations usually due to human mistakes. Moreover, autonomous driving technology would also improve mobility independence of those who cannot get a driver's license, including elderly and disabled people. Still, a critical point that has to be addressed is the trust in these vehicles. From a recent report of *Continental AG* [3] it appears that leaving control to the car is the first preoccupation of today's driver. In this study, 62% of the interviewed people, answered that they have concerns about automated driving; 57% do not believe in the reliability of AD systems and, approximately a quarter believes, that AVs will reduce the joy of driving a car. Otherwise, the same study also shows that the 53% of the people in Germany thinks that Automated driving is a sensible advancement and 64% of them believes that Automated driving could take over the task of driving in monotonous or stressful situations. Moreover, Autonomous vehicle will be able to plan optimal routes to save energy and so reduce polluting emission.
In industry, five distinct levels of automation have been defined [4], [5]. At each level corresponds a precise interaction between car and driver beside the task that the autonomous vehicle can provide. The autonomous driving levels go from 0 to 5: level 0, means "No Automation" and it is the simplest case where the driver controls the car without any support from a driver assistance system [6]. Following are listed the six levels of autonomous driving in details:

0. **No automation**: This is the fully human level. The driver has to accelerate, brake, steer, and negotiate traffic without assistance from any technological device.

1. **Driver Assistance**: At this level are considered driver assistance systems that support drivers on the road and help to ensure additional safety and comfort. Some examples are: the Adaptive Cruise Control [7] with Stop & Go function,

which independently adjusts the distance to the car in front of you and the Collision and Pedestrian Warning with City Brake Activation, which prevents collisions via automatic braking.

2. **Partly Automated Driving**: Semi-autonomous driving assistance systems characterize level two of driving automation. An example is Steering and Lane Control Assistant [8] including Traffic Jam Assistant: they can brake automatically, accelerate and take over steering.

3. **Highly Automated Driving** At this level, the car will be able to drive autonomously under certain traffic situations, such as on highways, but when those conditions are not valid anymore, the driver must be able to take over control within a few seconds.

4. **Fully Automated Driving** The vehicle can handle highly complex urban driving situations, such as the sudden appearance of construction sites, without any driver intervention although a human driver can still request control: indeed the car still has a cockpit. The driver, however, must remain fit to drive and capable of taking over control if needed.

5. **Full Automation** At the last level, drivers don't need to be fit to drive and don't even need to have a driving license. The car is completely autonomous and no cockpit is present.

As soon as the level of automated driving is increased, it becomes more important the availability of a high-resolution map for the area of interest since map information is fundamental for multiple autonomous-driving applications such as the computation of trajectories and path planning [9]. In the case of incorrectness, the safety of road users may be compromised. In order to enforce its safety requirements, the vehicle must be equipped with a software system that is able to recognize discrepancies between map data and the real-world, and in case of anomalies, to act according to the situation.
In the ideal scenario, the vehicle should be able to drive simply keeping information from the surrounding environment using the available sensors and a simple global map necessary for navigation purposes. In particular, this map might be rather old and without a very high resolution. Unfortunately, in order to reach the necessary precision in the information, the latter approach, leads to a very complex algorithms that today are hard to execute in a real-time framework. Furthermore, today's sensors are not able to provide the sufficient accuracy of measurements needed for guaranteeing the safety of the road users. For these reasons, nowadays, the most practical solution to approach the autonomous driving world is the use of *HD maps* and *OMV* algorithms. Depending on the complexity of an *OMV*, we can recognize three major use cases:

1. Detect discrepancies between map and reality. In this case, the *OMV* does not inform where the map is invalid. A simple boolean flag could be a reasonable output of an *OMV* block.

2. Detect locally if the map is respectively valid or invalid. If the are discrepancies in a certain area, invalidate the latter and provide information about how the map matches reality in the other areas.

3. If it is possible, correct the map using the information coming from the sensors.

The role of *OMV* becomes increasingly important as soon the level of autonomous driving increases. In some applications, the map can be directly obtained from sensors using simultaneous localization and mapping (*SLAM* [10]) algorithm. With *SLAM* is intended the process by which a mobile robot can build a map of an unknown environment and at the same time use this map to compute its location. These methods work quite well when the area of interest is limited, but as soon the scenario becomes larger the computational cost increase and they become prohibitive.

In this thesis, we propose a framework for performing the validation (or invalidation) of a map based on spatial and temporal correlations of sensor measurements. Probabilities for *Valid*, *Invalid*, and *Unknown* states of a map are represented in a grid-like structure and then interpolated exploiting the knowledge of correlated data.
The fusion of sensor measurements can be performed in different ways; one of them, discussed in detail, is through the use of *Bayesian Network* [11]. The latter can represent a set of variables and their conditional dependencies via a directed acyclic graph combining the information in a probabilistic framework. Using this approach, it is so possible to model the relation between multiple sources of information, directly, in a way, which is understandable for a human observer.

1.1. Thesis objectives

The work presented in this master's thesis aims to describe steps necessary to implement a model based approach to face the problem of map validation. More precisely, we will try to extend the concept of grid for the characterization of the validity of a map. In our case, the environment is discretized and the value of each cell represents the probability that the cell is valid rather then the probability that the cell is occupied as it happens in a normal occupancy grid. Commonly, for computational reasons, the single cells in a grid are considered independent from the others. Indeed, with this assumption, the probability distribution of the map is given by the product over the cells probabilities greatly simplifying the calculations. This hypotheses results very convenient in many situations but sometimes it is unrealistic and it can neglect some

useful information related to the reciprocal influence among cells. For this reason in the *Validity map* developed in this thesis (a map used for *OMV* purpose), it is also considered a spatial correlation among cells. In other words: a map can be considered as a surface where there is a sort of continuity on its validity. Besides being more realistic, this approach can be exploited to estimate the validity of the map in areas where limited data are available. The latter information can come from multiple source that, under certain assumptions, can be considered as nodes of a probabilistic network. This specific solution is described in details in this thesis.

1.2. State-of-the-art

To the best of our knowledge, algorithms for OMV typically focus on SLAM approaches: a real-time map is created from sensor data and then compared with the map to check.

Landsiedel and Wollherr [12] propose the integration of a 3D metric environment representation with the semantic knowledge from open data. The rectangle-based model idea used is a computational interesting and graphical-probabilistic model approach. It appears that the road model approach seems relevant for *OMV*.

Tanzmeister (et al.) [13], [14] focus on estimating the drivable area using vehicle motion in unknown environments. With this approach, using path planning algorithms it is possible to obtain an interesting road course estimation.

In [15], the authors provide a method for coarse map localization which uses noisy lane hypotheses from sensor inputs, noisy maps with lane-level accuracy and noisy global positioning to simultaneously improve the accuracy of all three. The problem is solved by applying Loopy Belief Propagation to a tailored factor graph which models the dependencies between observed and hidden variables. The approach is interesting but inapplicable to the OMV due to the high computational costs.

Hasberg and Hensel [16] describe an interesting approach for fusing noisy sensor data in a parametric representation of the lane using a probabilistic approach. This framework could be valid for modelling the geometry of the lanes and has to be tested in the urban scenario.

A promising alternative for *OMV* is the use of the machine learning approach to classify and estimate the road.
In [17], Fernandez (et al.) generates a model of the complex urban scenario using the information provided by a digital navigation map and a vision-based sensing module. In this approach, a road pixels segmentation is performed with the use of a Convolutional Neural Network (CNN) and fused with information map data. Interestingly, the simple model for describing the geometry of the intersection that this paper proposes is based on six parameters only. Mayer et al. [18] use a deep neural

network (DNN) applied to RGB images to extract lane boundaries for the ego lane including the semantic information about adjacent lanes. The evaluation results are rather comprehensive and comparable to other state-of-the-art approaches.

1.3. Thesis outline

The rest of this thesis is organized as follow: in chapter 2 are introduced the common characteristics of an *HD map*, the requirements that it has to satisfy and its standards. Furthermore, in the same chapter, the concept of Occupancy grid is explained and the sensors typically used in autonomous-driving vehicles, with their common fusion algorithms are introduced.
In chapter 3 is provided a quick description of the Probabilistic graphical model with the help of some example and in chapter 4 the concept of map validation with spatial correlation is defined. This chapter compares different approaches to solve the problem (Gaussian Approach, probabilistic approach and the use of Dempster-Shafer theory) and for each of them the benefits and the disadvantages are discussed.
In chapter 5 a more practical example of map validation is explained. The characteristics of some sensors usually used in the autonomous driving world are taken into account in order to obtain a more meaningful example. In chapter 6 a possible way to validate a map based on maneuvers of other traffic participants is mentioned and some examples are provided. Closing, there is a summary of the thesis and an outlook to future research topics.

2. Background

This chapter introduces some key concepts related with maps for *AV* and with map validation. Sections 2.1 describes the characteristics of the HD maps, more in detail, 2.1.1 defines their standards and 2.1.2 explains the requirements needed for an autonomous driving application. Section 2.2 introduces the concept of grid map. Sections 2.3 and 2.4 respectively describe the type of sensors and their fusion use in the autonomous driving framework .

2.1. HD maps

For a self-driving vehicle, the availability of sensor data combined with the maps used by standard commercial navigation systems, is not enough to guarantee the achievement of good performance. An autonomous vehicle needs multiple information sources in order to understand its surroundings and to reach the desired destination in a safe and efficient way. In complex scenarios like cities, it becomes necessary an advanced map that provides an extended field of vision for path and strategy planning. In general, an HD Map is a precise, reliable information resource. This support is a deeply detailed map that autonomous vehicles need in order to perform tasks normally provided by human memory and experience. The map contains a large variety of information about road geometry, such as the width, the height of the galleries, the curvature of the roads etc. with precision in centimetres. A sample of the visualized HD map is shown in Figure 2.1.

Using this source of data, a vehicle can see and take action in a more informed way respect to the one that could be possible by using only the range of its own sensors. In Figure 2.2 is reported the typical pipeline that autonomous driving systems use.

Sensors information, HD map data, and localization technologies such as GNSS [19] (Global Navigational Satellite System) and GPS [20] (Global Positioning System) are combined in order to plan the optimal trajectory and ensure the satisfaction of all the security requirements. Maps used for autonomous driving need to be much more accurate and reliable that the ones available in a *GPS* navigator. In fact, besides being high resolution maps, they contain other elements such as landmarks, street signs, traffic-light positions. Moreover, the map for autonomous driving usually contains semantic information like driving direction and right of way. The information typically

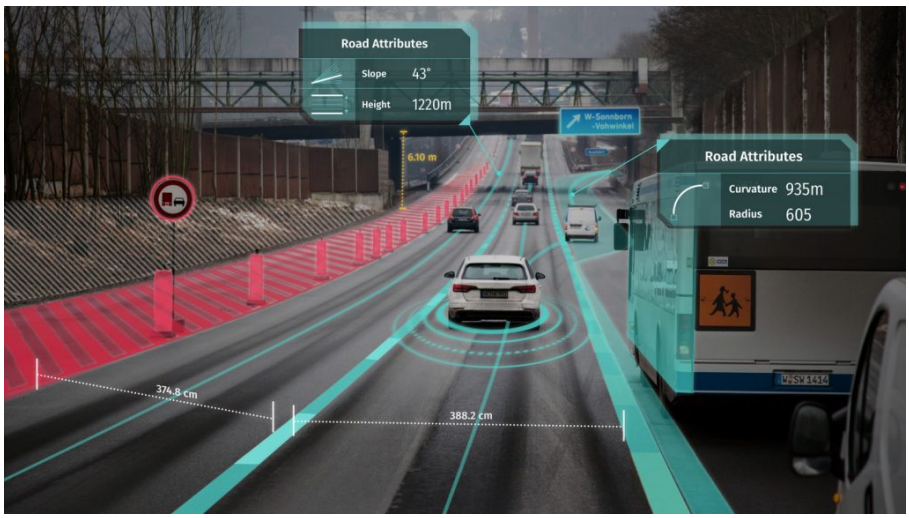


Figure 2.1.: HD Live Map Example. [1]

available in this kind of map can be divided into two big groups:

- **Geometric information**[21]: List of characteristics regarding: the shape and size of the road, the position of the lanes etc.
- **Semantic information**[22]: Set of information related to the behaviour that the vehicle has to maintain on the road; for example: driving direction, right of way, speed limits etc.

In order to evaluate the state of the map, it is necessary to consider both geometric and semantic information. Indeed, the classification of the map as invalid does not necessarily mean that it is unreliable and so useless for our goals. In practice, we need to distinguish between inconsistencies that may lead to unsafe situations and discrepancies which do not have safety-critical or functional implication. Each information element must be weighted in a proper way in order to ensure a meaningful and usable result in real application; so, for example, a missing street sign for touristic information, e.g., like the direction to reach a Hotel, will not invalidate the map if the road geometry and the other information necessary for a safe driving experience are still correct, but it can highlight the presence of an anomaly in the map that can become invalid in few meters (the road to the Hotel is closed).
Even if it is not possible to define an exact order of importance for the different map

[1]Source: https://www.autofreaks.com/2017/139073/lg-partners-next-gen-telematics/ (visited on 11/20/2019)

content, since it depends on the specific use case, it is appropriate to define a list of map elements that are common to most applications:

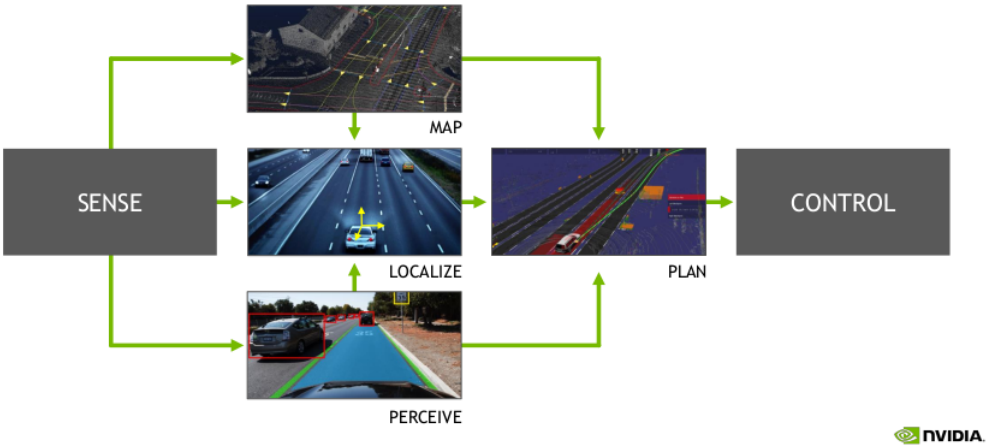


Figure 2.2.: Typical AD pipeline. [2]

- Ego lane geometry
- Course lane geometry
- Traffic lights on course lanes
- Driving direction on course lanes
- Oncoming lane geometry
- Oncoming lane driving direction
- Branches at intersections, not covered by course lanes
- Right of way at intersections
- Pedestrian crossings
- Parking lanes
- Traffic signs

[2]Source: http://www.imaging.org/Site/PDFS/Conferences/ElectronicImaging/EI2017/Keynotes/EI2017_AVM_Keynote_Justyna_Zander.pdf (visited on 11/20/2019).

2.1.1. NDS: A new data format for map

Since, in the future, the HD Map for AD vehicles should be globally applicable and always updated, collaboration and data sharing among different automotive industry have been recognized as critical. In order to make it possible, the standardized format *Navigation Data Standard* (NDS) have been created. The main objectives of the NDS standard are:

- the definition of a model that can be widely used in the automotive industry and adopted by suppliers of navigation maps;
- the flexibility and the usability in many different platforms;
- the availability of always up-to-date maps;
- the global applicability.

Its structure is characterized by the division in building blocks. Distinct blocks are defined for lane level data, localization data, obstacles data, and the routing building block with the link network and road topology. The fusion of all these elements create the ground truth for autonomous driving technology. One example is reported in Figure 2.3 where the map is divided into Road Model, HD Lane Model, and HD Localization Model. Each of them contains different geometrical and semantic clues that can be used severely by the vehicle to solve different assignments or combined together if the task requires it.

2.1.2. ASIL (Automotive Safety Integrity Level)

Two fundamental requirements that a safety-critical system such as an AV has to guarantee are the ability to correctly avoid hazardous situations and the capacity to detect and manage faults. These requirements are governed, for the automobile industries, by the *ISO 26262* functional safety standard, that defines the *Automotive Safety Integrity Levels* (ASIL). The latter establishes safety requirements for automotive component and can be distinguished in four different levels: A, B, C, D (Table 2.1). Level A is the one with the lowest requirements in terms of integrity, this level is required for example in the rear-light of the vehicle or in the vision ADAS (Advanced Driver Assistance Systems) system. ASIL Level B could be used, in the rear-view camera, headlights and break lights, but, as we move towards the more safety-critical applications an ASIL C or D can be required. An application of ASIL C is the radar cruise control and ASIL D, witch identify the highest standards, is used in Airbags, Antilock breaking and Electric power steering.

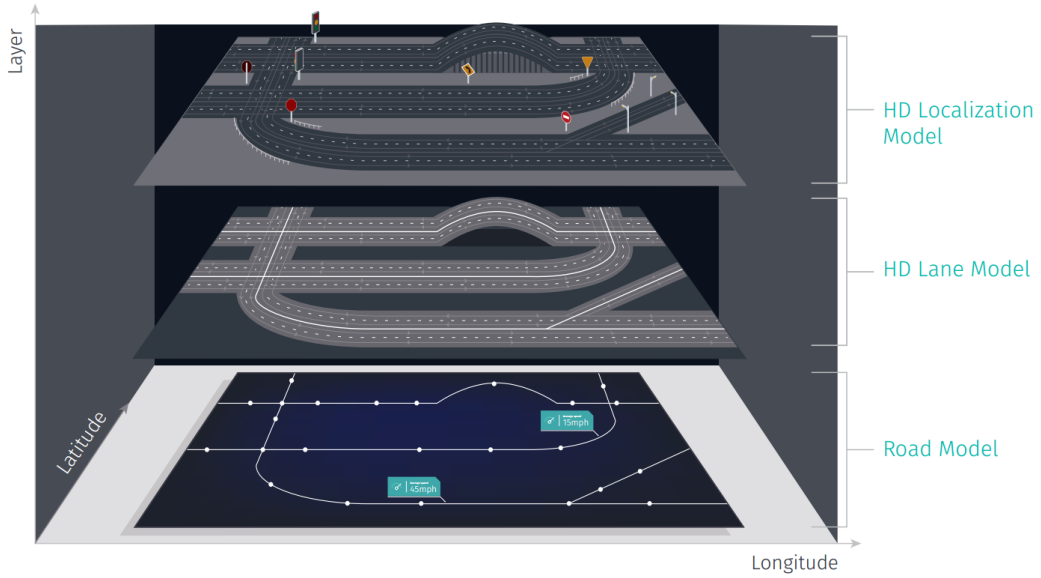


Figure 2.3.: Multiple layer HD Live Map. [3]

In ASIL D, designers will need to add system redundancy, develop policies and produce documentation strategies that meet the rigorous certification requirements of *ISO 26262* (Table 2.1). The difference among the different ASIL levels is quantified using some minimum target metrics [23] as shown in Table 2.1.

The Single-Point Faults Metric (SPFM) indicates the percentage of faults that are not

Metric	ASIL A	ASIL B	ASIL C	ASIL D
SPFM	N/A	$> 90\%$	$> 97\%$	$> 99\%$
LFM	N/A	$> 60\%$	$> 80\%$	$> 90\%$
PMHF	$\leq 10^{-6}$	$\leq 10^{-7}$	$\leq 10^{-7}$	$\leq 10^{-8}$

Table 2.1.: ASIL levels.

treated by a safety system and lead to a violation of the safety goal. Faults that which are not directly violating the safety goal, are considered in the Latent Fault Metric (LFM). This measure takes into account the possibility of a second fault which may be elevated by the undetected first fault to violate the safety requirement. The Probabilistic

[3]Source: https://360.here.com/tag/hd-live-map

Metric for Random Hardware Faults (PMHF) gives the frequency of failures per hour. The association to a certain ASIL level to a component follows a clear procedure defined in the ISO 26262 norm. For each possible failure of a certain component, the procedure considers the effect of the failure to the overall system and assess.

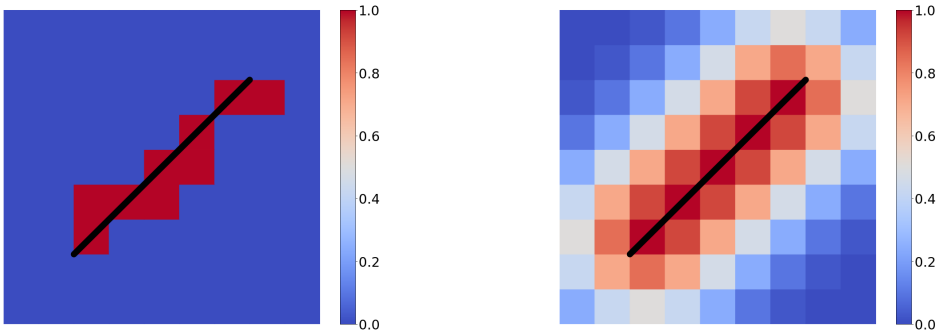
- **Severity** (the type of injuries to the driver and passengers)
- **Exposure** (how often the vehicle is exposed to the hazard)
- **Controllability** (how much the driver can do to prevent the injury)

by combining these values, the specific ASIL level is defined. In particular: severity has four classes ranging from "no injuries" (*S0*) to "life-threatening/fatal injuries" (*S3*); exposure has five classes covering the "incredibly unlikely" (*E0*) to the "highly probable" (*E4*) and controllability has four classes ranging from "controllable in general" (*C0*) to "uncontrollable" (*C3*). It is important to highlight that ASIL definitions are informative rather than prescriptive: for example, if a device is *S3* for the severity and *C3* for the controllability, it could still classify as ASIL A (low risk) because of the hazard probability is extremely low (*E0*).

2.2. Space distribution: grid

In robotics and similar related fields, the robot workspace is represented using the so called occupancy grids introduced by *A. Elfes* [24]. With this term, usually researcher refers to a family of probabilistic algorithms which address the problem of generating a map of the environment from noisy sensor measurements. These data come from Laser range finders [25], bump sensors [26], cameras [27], and other sensors that are used to identify obstacles in the robot's neighborhood and build the environment map. The world is so discretized in a finite number of cells whose size depends on the desired precision and the computational load that the system is able to manage. Indeed, by decreasing the size of the cells, their total number increases as well as the number of variables to be taken into account in the system.
An occupancy map can be of two types:

- **Binary occupancy map**: It uses an alphabet of two symbols for defining a cell of a map as either occupied or free. No information about the uncertainty of this information is provided. An example is reported in figure Figure 2.4a.
- **Probability occupancy map**: Each cell in the occupancy grid is a binary random variable and its value represent the probability that the cell is occupied. A cell value close to 1 represent a high certainty that the cell is occupied, conversely, a



(a) Binary occupancy grid (b) Probabilistic occupancy grid

Figure 2.4.: Comparison between binary and probabilistic occupancy grid in case of in the presence of an object (black). The color of the cells, from red to blue, represents with what probability the cell is occupied.

cell value close to 0 means that most likely the cell is empty. A cell value equal to 0.5 imply no knowledge about that specific cell. An example is reported in Figure 2.4b.

2.3. Sensors typically found in autonomous-driving vehicles.

The presence of different sensors is fundamental for reaching an high level of automation in the automotive field. Working together, the sensors provide the car visuals of its surroundings and help it detect the speed and distance of nearby objects (Figure 2.5). On the market, many type of sensors are available and each of them has its own strengths and weaknesses in terms of range, detection capabilities, and reliability.

Discant et al. in [28] classifies the main types of sensors in two big categories: *passive* and *active*. With the first class of sensors, they consider all the sensors that capture energy from the environment and the most common example in the field of autonomous driving are the cameras. Considering instead the *active* sensors, they are the ones which emit some sort of waveform and measure the reflections from the possible targets. The main member of this category of sensors in the field of autonomous driving are: Lidar, Radar, Sonar.

Following the characteristics of the sensors are explained more in details:

- **Cameras:** they are the most available type of sensors. Monocular cameras do not

[4]Source: https://blogs.nvidia.com/blog/2019/04/15/how-does-a-self-driving-car-see/ (visited on 12/10/2019).

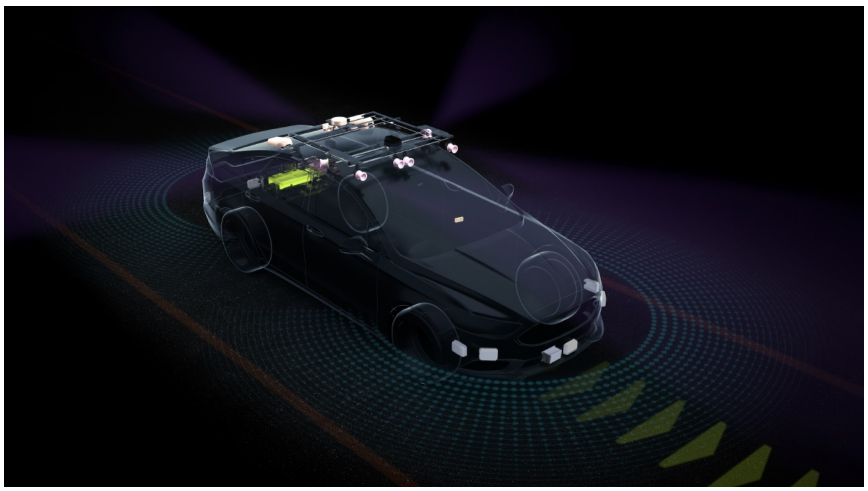


Figure 2.5.: System overview.[4]

provide depth information, since they see in two dimensions anyways, for this reason the stereo cameras also exist that solved the problem. Furthermore, they can sense visible light or the infrared one; in this case they are called normal or infrared cameras respectively.

- **Lidar**: a LIDAR (abbreviation of Light detection and Ranging), is a sensor that uses laser light, usually in the infrared band (900nm), to measure distances in all directions. This technology reaches an excellent angular resolution and range accuracy and it allows to generate a precise 3D map of the vehicles's surroundings. The main drawbacks are the following:
  - They generate huge amounts of data that can become expensive to process;
  - They have problems to detect object characterized by specular reflection.
  - They suffer from light scattering in case of bad weather conditions like rain and snow.
  - Usually they can not measure the object radial speed.
- **Radar**: a RADAR (abbreviation of Radio Detection and Ranging), is a device that uses radio waves to determine the velocity, range and angle of objects. Radars for automotive applications have many advantages respect to the other types of sensors: one of them is for sure the all weather capability. For this reason and many other, in literature it can be found many algorithm for the detection of road and obstacles [29].

Sensor	Camera	Lidar	Radar	Ultrasonic	**Fusion**
Field of View	Good	Excellent	Good	Poor	Excellent
Range	Good	Excellent	Excellent	Bad	Excellent
Velocity Resolution	Poor	Good	Excellent	Bad	Excellent
Angular Resolution	Excellent	Good	Poor	Bad	Excellent
Adverse weather	Poor	Poor	Good	Poor	Excellent
Light Disturbance	Good	Excellent	Excellent	Excellent	Excellent
Object classification	Excellent	Good	Poor	Bad	Excellent
Coverage of all surfaces	Good	Good	Good	Good	Excellent

Table 2.2.: Comparisons of sensors vehicles

Radars in the automotive field usually use a millimeter-wave band around $77GHz$, they have an high range resolution and using the *Doppler effect* [30] they can measure the radial speed of moving object. Unfortunately, they often have a poor angular resolution but the latter can be usually compensated by using the combination of phased-array receivers and high resolution algorithms with witch can be achieved performances, up to $\pm 1^{\circ}$ [31].

- **Sonar(ultrasonic)**: Sonar sensors are based on the same working principle of Radars but the using ultrasonic sound-wave instead. They are really cheap but their performances are strongly related with the environmental conditions, and for this reason, sonars are not commonly used for autonomous driving application.

As it can be observed in Table 2.2, every technology has its own strengths and weaknesses so, in order to reach the required level of performance and safety, the combination of multiple sensors become compulsory [32].

2.4. Algorithms for sensor fusion in automotive

Over time, a variety of fusion algorithms have been proposed in literature. In the following we provide a possible classification based on the work of McKendall and Mintz (1988) in [33] say: "*...the problem of sensor fusion is the problem of combining multiple measurements from sensors into a single measurement of the sensed object or attribute, called the parameter*". The sensors can be combined in different ways:

- **Competitive fusion**: This configuration is used to resolve the incompleteness of sensor data. In this situation the sensors complete each other in order to give a better overview of the phenomena. An example of this configuration could be the use of multiple cameras, each observing disjunctive part of the room. The multiple stream of images can be combined in order to obtain an overall vision of the environment.
- **Complementary fusion**: In this case each sensor provides independent measurements of the same property. The data to fuse can come from different sensors or can be provided by the same device but in different time instance as explain *Visser* and *Groen* in [34]. In this scenario the Kalman filter [35] and the Particle Filter [36] have to be mentioned. Both of them are algorithms that uses a series of measurements observed over time, containing statistical noise and other inaccuracies, to produces estimates of unknown variables. In case of a linear system with Gaussian noise, the Kalman filter is optimal. In a system that is nonlinear, the Kalman filter can still be used, but the particle filter may give better results at the price of additional computational cost.
- **Cooperative fusion**: In this configuration, the system uses the data provided by two (or more) independent sensor to obtain an information that would not be available from the single sensors.

2.5. Summary

In this chapter a background to easily follow the thesis is introduced. The *HD map* that has to be validated is described in details. Multiple elements can be used to evaluate the validity of an *HD map* since it is very detailed and full of information. The validation can be done using the different sensors available. Their data can be fused in multiple ways depending on the type of measurement and the specific information we want to extract from them. Different type of fusion algorithm can led to very different result in terms of final performance. In the next paragraph we discuss in detail the use of probabilistic graphical models for sensors fusion purpose.

3. Probabilistic graphical model

In this chapter it is discussed the approach we are proposing for fusing sensor data and to compare them against map data for the purpose of map validation. In order to evaluate the quality of a map, the knowledge of the surrounding environment is fundamental. Unfortunately, precise information can not be guaranteed for all the area of interest because, very often, there are estimation errors due to sensor imprecision and limited field of view. For this reason it is not possible to rely on one sensor only, and it is necessary to fuse information from multiple of them [37], [38]. By combining the input from various devices, individual systems complement each other and can achieve enhanced performances (Table 2.2). One interesting way to perform sensor fusion is the use of the so called *Probabilistic graphical model* (PGM) [39]. Formally, a PGM consists of a graph structure. Each node of the graph is associated with a random variable, and each edge that connect two nodes describes a conditioning relation between them. Compared with logical reasoning, that is concerned with absolutely certain truths and inferences the probabilistic reasoning used in PGM can arrive at rational descriptions of the problem even when there is not enough deterministic information on the functioning of the system. The PGM can be divided in two main categories depending on their structure. If the graph is direct the model become a Bayesian networks (BN) otherwise if the graph is undirected we refer to Markov networks (MN). In this thesis we focus only on Bayesian network; the idea of using them for map validaton purpose also led to the writing of the patent *Use of probabilistc graphical model for map* filed at the *German Patent and Trademark Office* on August 30, 2019 (Application number: 10 2019 123 353.4)

3.1. The Bayesian network in detail

Bayesian networks, are commonly used when an event occurs and it is necessary to predict the likelihood that any one of possible known causes was a contributing factor. An example of field of application is the medical one: BNs can be easily used to represent the probabilistic relationships between diseases and symptoms. Once the symptoms are visible, a *BN* can be used to compute the probabilities of the presence of different diseases.
In general, a BN is a graph in which the following properties hold:

- A set of random variables constitutes the nodes of the network;
- A set of oriented edges connects the pairs of nodes (an arrow from node A to node B is that A has a direct influence on B);
- Each node has a conditional probabilities that quantifies the effects that the "parents" have on the node, where "parents" means all those nodes that have arrows pointing to the node;
- The graph has no direct cycles.

In a BN, the random variables identified by the single nodes of the net can be discrete, continuous or both of them. If a node has relatives (i.e. one or more arrows pointing towards it) then the node contains a table of conditional probabilities. In this way, a BN allows a graphical representation of the probabilistic relation among random variables. Consider the simple case of two random variables, one dependent from the other. The relation between two events described by

$$P[A, B] = P[B|A] \cdot P[A] \tag{3.1}$$

becomes the following graph:

Figure 3.1.: Three events Bayesian network example.

In a *BN*, nodes whose values can be directly observed are called *observable nodes* (the node B in this specific case), instead nodes whose values cannot be seen are called *hidden nodes* (the node A in this framework).
At this stage, we do not need to specify anything further information about the interested variables, such as whether they are discrete or continuous. Indeed, one of the powerful aspects of graphical models is that a specific graph can make probabilistic statements for a broad class of distributions. Consider now the joint distribution $P[A, B, C]$ over three variables A, B, C defined as

$$P[A, B, C] = P[C|A, B] \cdot P[A, B] \tag{3.2}$$

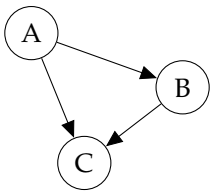


Figure 3.2.: Three events Bayesian network example

and applying the *chain rule* on the last term of the right hand side, we obtain:

$$P[A, B, C] = P[C|A, B] \cdot P[B|A] \cdot P[A] \tag{3.3}$$

The right hand side of Equation 3.3 is graphically represented in Figure 3.2. In the latter we can see a node for each random variable considered and some directed links (arrows) from the nodes corresponding to the variables on which their distribution is conditioned. Thus, for the factor $P[C|A, B]$, there are links from nodes A and B to node C, whereas for the factor $P[A]$ there are no incoming links. An interesting point to note about Equation 3.3 is that the left hand side is symmetrical with respect to the three variables A, B, C whereas the right end side it is not. Indeed, the decomposition of Equation 3.3 has been done choosing a specific ordering. Changing the latter we obtain a different decomposition and so a diverse graphical representation. Moreover, notice that analogous considerations are also valid for Equation 3.1.

3.2. Proposed modeling approach

The sensors available in an autonomous vehicle are heterogeneous and produce different types of measurements. We propose to connect observation in the vehicle such as the geometry of the road, or the behavior of other vehicle with hidden nodes which models the probability that the geometry of the road is valid, or that road rules are valid. In this way it is possible to combine all measurement types into one consistent framework. Figure 3.3 shows a possible example of this approach.

In the picture, the observable nodes are highlighted in blue and the hidden ones have an empty background. There are also two external factors (*Enforced traffic rules* and *Valid sensor measurements*) that affects the net and are represented without any bounding box. As previously explained with some examples, an arrow from a certain node A to a node B identify a dependence of the value of B from the value of A. The "weight" of this dependence is described by the conditional probability $P[B|A]$. In the case discussed in Figure 3.3 we can see that the node *Map Valid* affects all the

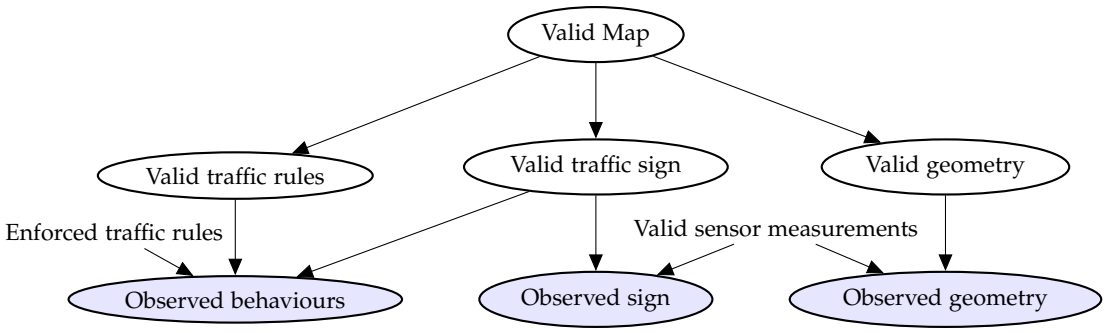


Figure 3.3.: Example of Bayesian network

nodes that represent a specific component of the map (semantic or geometric). The latter level affects the observed variables placed on the bottom of the network. This layer, also depends on other external factors such as the sensor characteristics and the enforcement of the traffic rules. The main goal of using a Bayesian network like the one reported in figure 3.3, is to exploit the value of the observed variables (that can be captured from the environment), to ascend the net from the bottom using the predefined dependence relations between nodes and to obtain as result a probabilistic value os the note *Valid Map*. This approach is relatively very flexible, since it can consider all possible observations vehicles make and can correlate them with relevant properties of the map. The graph reported in figure 3.3 is a simple example and it can easily modified (nodes considered and relation between them) in order to fit better the characteristics of the system considered. Furthermore, exploiting the *DBN* time evolution can also be included.

3.3. Example

This section discusses some results based on the probabilistic graphical network shown in 3.3. By selecting the values of the observed nodes (which in this case can be only *Valid* or *Invalid*), it is possible to infer the probability of the map being *Valid* or *Invalid*(root of the graph). An example are reported in Figure 3.4.
In the picture the height of each bar represents the probability of Map *Valid* when the observation written on the horizontal axis are *Valid*. With different color are underlined the situations with: no *Valid* measurements available (red), one *Valid* measurement (orange), two *Valid* measurements (light blue) and a three *Valid* measurements (blue). The exact numerical values of *Map Valid* depend on the conditional probabilities used in the network (which are reported in the app. D), but the trend is clear: increasing the number of *Valid* observations, the probability of map *Valid* tends to rise. Conversely,

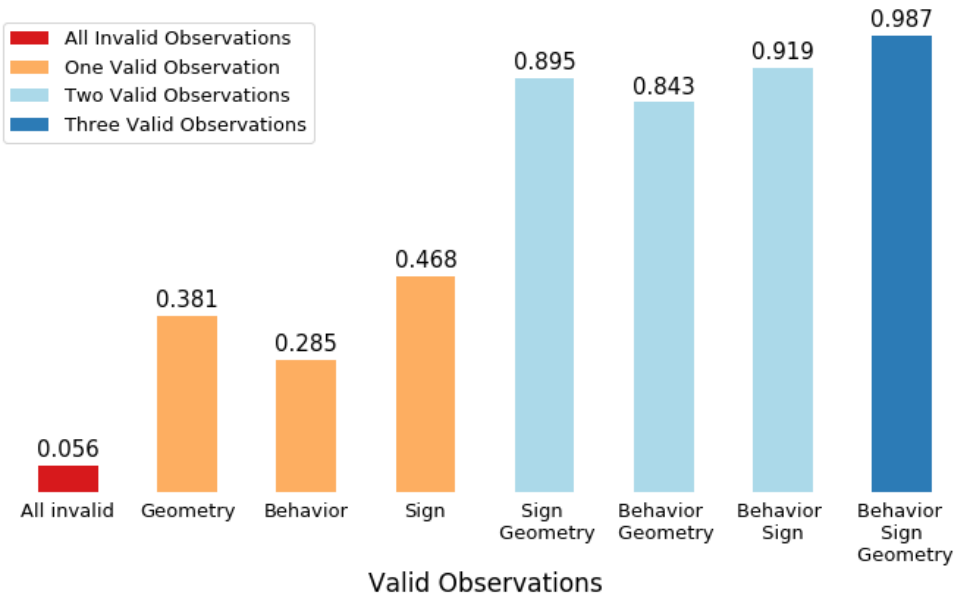


Figure 3.4.: Map validity as a function of the observed variables

if the measurements suggest an invalid map, the probability of the node *Map valid* assume a value close to zero.

3.4. Benefit of the proposed approach

The proposed approach is extremely flexible and it allows to neatly incorporate different kind of observations from the vehicle along with map properties that we want to validate. The problems provided by incorrect localization can also be included in the model. Furthermore, different properties of the map can be included and, where relevant, independently validated.

3.5. Drawback of the proposed approach

One of the main drawbacks of the proposed approach is the parameters that the model needs to know. Every edge needs to represent a dependency between random variables and the values of this dependency must be defined with a probability distribution. So the number of parameters increases exponentially as soon the quantity of connections among nodes rises. In case of complex models (which take into account many possible

relations), the number of parameters to estimate might be so height that the proper tuning of the net could become extremely difficult.

4. Validation with temporal-spatial correlation

The map validation problem can be divided into three main tasks:

1. detect the presence of discrepancies between available map and environment;
2. evaluate witch specific areas of the map are invalid.
3. update the map with the available information.

In order to allow the development of all these features, it is necessary to define a suitable and flexible environment. The sensor measurements can not completely cover all the relevant area, so an interpolation among sensor data is applied. Every infinitesimal point p_i is characterized by spatial/temporal coordinates and correlated with the rest remaining map. The relation between two points p_i and p_j depends on their distance; in particular:

- the condition $\|p_i - p_j\| \to \infty$ could occur if the spatial distance is big, if the temporal distance is big (p_i and p_j could correspond to the same point in two different temporal instants) or if these two situations occur simultaneously. In this case the correlation between p_i and p_j has to be high.
- the condition $\|p_i - p_j\| \to 0$ occur when both the spatial and temporal distance tends to 0. In this case the correlation between p_i and p_j has to be low.

Furthermore the following assumption are introduced:

- the discretization of the environment does not necessarily need to be uniform. For example, we may consider a case where lane geometries are discretized along their length, at relatively fine granularity, whereas areas around a road can have relatively rough discretization blocks;
- we assume the presence of *smart* sensors, i.e. sensors that are able to provide high level information on the map. The output of these advanced sensors takes values from a finite set of discrete labels (e.g. *Valid*, *Invalid*, *Unknown*) and can be directly used to determine the overall map validity.

Three approaches have been considered:

- **Gaussian approach**: Every single cells is considered as a Gaussian random variable and the update of the "probability" value of each cell is done using a Kalman-like statistical filter.
- **Probabilistic approach**: The problem is studied in a completely probabilistic framework and a *ad-hoc* correlation function among the single cells is used.
- **Dempster Shafer approach**: Using the framework described in the Dempster Shafer theory, a spatial-temporal correlation among the single cells of the map is defined.

In particular, the first approach is based on a published work [40] and is mainly used as comparison for the second approach which has been completely developed in this thesis. The third approach, instead, tries to replicate the second one using the alternative Dempster-Shafer Theory.
Initially, in all the cases, the environment is divided into a two-dimensional lattice where cells are treated as hidden variables to be estimated. Every cell can be only *Valid* or *Invalid* and we want to estimate with which probability the single cell can be considered *Valid*.

4.1. Continuous approach: Gaussian filtering

In this framework, every cell is associated to a continuous random variable, characterized by a specific mean and variance. By considering a spatial correlation between the cells (Guassian for example) and by using a specific implementation of a Kalman Filter, the validity of each single cell can be computed recursively.
The recursive approach has numerous computational benefit and lend itself easily to a realtime implementation with stringent timing and memory constraints. This type of approach has been used by Blaco, Monroy and Gonzales in [41]. They introduce a probabilistic method to learn a gas distribution model of planar environments, given a sequence of localized gas sensor readings. In this work, a simple sensor model, leads to the derivation of an efficient implementation of a Kalman Filter. The estimated model, also referred as the map of gas concentrations, keeps a density distribution of the expected concentration at each cell, including its uncertainty. An alternative approach is presented by Ke Sun et. al. in [40]. The focus of the latter is to provide a model that captures the occupancy correlation of map elements, providing an efficient approximation of Gaussian Process (GP) [42] classification with complexity that does not scale with time. The latter approach, implemented by using a Kalman-filter-like

structure, has been considered a starting point for this thesis and so, some simple evaluation have been performed.
By defining a grid map $\mathbf{m}$, abstracting each cell with its center position $\mathbf{x} \in \chi$ and, defining $o(\cdot)$ a function such that $o(\mathbf{x}) := -1$, if $\mathbf{x}$ is unoccupied, $o(\mathbf{x}) := 1$ if $\mathbf{x}$ is occupied and, $o(\mathbf{x}) := 0$ if the state of $\mathbf{x}$ is unknown, the problem treated by Ke Sun et. al. is formalized as follow:

> Given occupancy measurements $Z_{0:t}$, construct an estimate $\hat{o}_t \in \{-1, 0, 1\}^{|\chi|}$ of the true environment occupancy $\{o(\mathbf{x}) | \mathbf{x} \in \chi\}$

Namely, the task is to build a geometric map of a static environment having a sequence of occupancy measurements $Z_{0:t}$ available.
In the work of Ke Sun et. al., assuming the measurements within the pre-defined tessellation χ, their binary model is represented as

$$y_i = h(\mathbf{x}_i) \in \{-1, 1\} \tag{4.1}$$

Assuming then that the distribution of the map follows $\mathbf{m} \sim \mathcal{N}(\hat{\mathbf{m}}_t, \Sigma_t)$ at a time t, the mean and the covariance estimation for the map at $t+1$ is

$$\hat{\mathbf{m}}_{t+1} = \hat{\mathbf{m}}_t + y_i \cdot \Sigma_t \mathbf{v}_i \frac{\phi\left(y_i \cdot \frac{\mathbf{v}_i^T \mathbf{m}_t}{\sqrt{\mathbf{v}_i^T \Sigma \mathbf{v}_i + 1}}; 0, 1\right)}{\eta_{t+1} \sqrt{\mathbf{v}_i^T \Sigma \mathbf{v}_i + 1}} \tag{4.2}$$

$$\Sigma_{t+1} = \Sigma_t - (\hat{\mathbf{m}}_{t+1} - \hat{\mathbf{m}}_t)(\hat{\mathbf{m}}_{t+1} - \hat{\mathbf{m}}_t)^T - y_i \cdot \Sigma_t \mathbf{v}_i \mathbf{v}_i^T \Sigma_t^T \frac{\phi\left(y_i \cdot \frac{\mathbf{v}_i^T \mathbf{m}_t}{\sqrt{\mathbf{v}_i^T \Sigma \mathbf{v}_i + 1}}; 0, 1\right) \left(\mathbf{v}_i^T \hat{\mathbf{m}}_t\right)}{\eta_{t+1} (\mathbf{v}_i^T \Sigma \mathbf{v}_i + 1)^{3/2}} \tag{4.3}$$

where $\mathbf{v}_i$ is a vector containing all 0s except for a single 1 corresponding to the i^{th} cell, and η_{t+1} is the normalization factor,

$$\eta_{t+1} = \Phi\left(y_i \cdot \frac{\mathbf{v}_i^T \mathbf{m}_t}{\sqrt{\mathbf{v}_i^T \Sigma \mathbf{v}_i + 1}}; 0, 1\right). \tag{4.4}$$

In 4.2, 4.3, and 4.4 we have that $\Phi(x; 0, 1) = \int_{-\infty}^{x} \phi(u; 0, 1) du$, where $\phi(u; 0, 1)$ is the standard normal *pdf*. The occupancy of each cell is then recovered through the following:

$$o(\mathbf{x}) = \begin{cases} 1, & \Phi(m_i; 0, 1) > r_0 \\ 0, & \Phi(m_i; 0, 1) < r_f, \ \forall \mathbf{x}_i \in \chi \\ -1, & \text{otherwise} \end{cases} \tag{4.5}$$

where r_o and r_f are the thresholds for determining occupied and free cells respectively. The work of Ke Sun et. al. can be easily adapted to the purpose of this thesis by assigning a different meaning to the state of the single cells. In this way the labels 1, -1 and 0 will be respectively associated *Valid*, *Invalid* and *Unknown* states.

4.1.1. Example

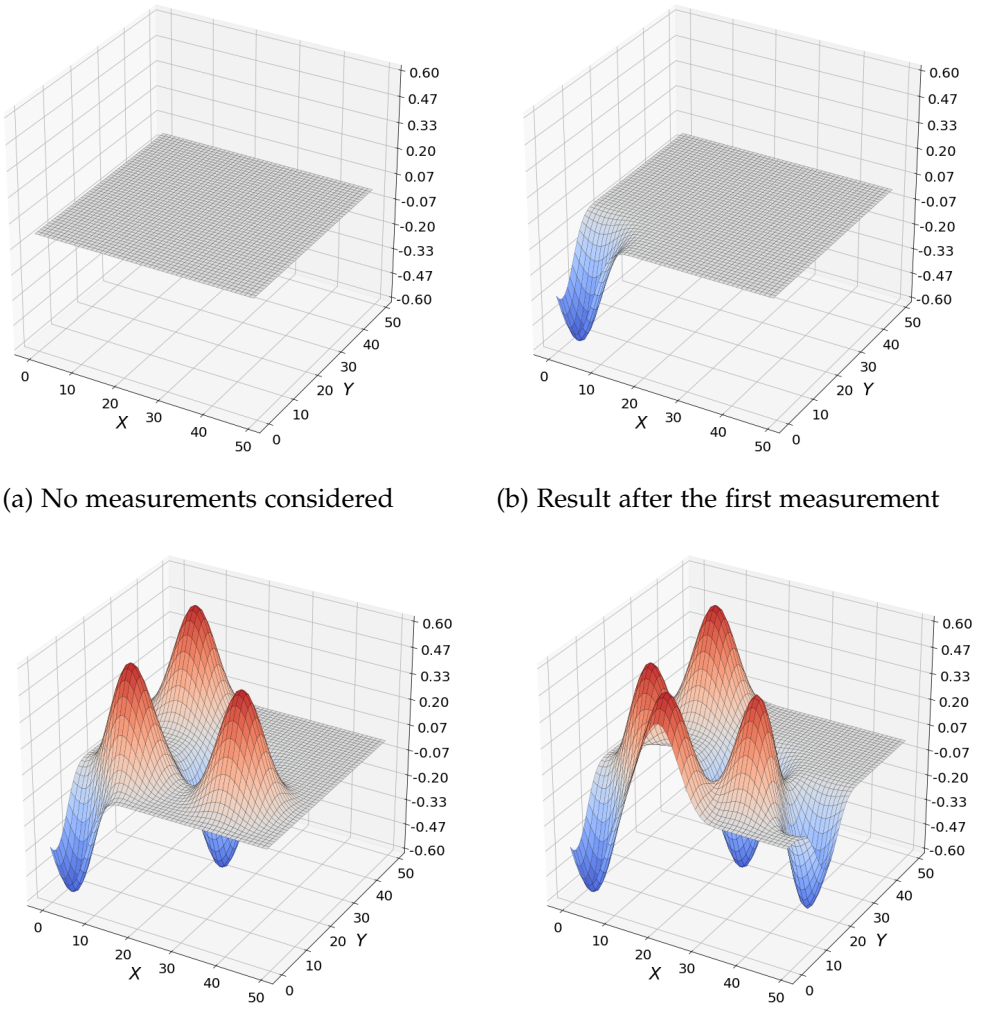
In this subsection is reported and described an example of application of the approach shown previously.
Consider the following assumptions:

- The map is described by squared lattice with $50x50$ cells.
- The value (*Valid* = 1 or *Invalid* = -1) of n isolated points is available. The points considered in this example are respectively placed in position $[4, 2]$, $[10, 40]$, $[12, 12]$, $[25, 25]$, $[36, 13]$, $[44, 16]$, $[21, 3]$. Three of them are *Invalid* measurements and the others are considered *Valid*.

The result of the application of the algorithm can be seen in Figure 4.1 where is reported the value assumed by (4.2) as the measures are added to the system. Indeed, the (4.2) is a recursive equation and the measurements have to be taken into account once at a time.
In Figure 4.1a there is the initial mean value of the map. In each cell of the grid the mean is equal to 0 (*Unknown* state). In Figure 4.1b one measurement (-1 in $[4, 2]$) has been added and the map validity surface has been manipulated. The highest value of the map is placed where the valid measurement is available and the rest of the map is updated in a continuous way, exploiting the spatial correlation among different cells. In Figure 4.1c and Figure 4.1d is reported the mean value of the map after that 5 and 7 measurements are respectively added to the system. As in the previous step, the desired interpolation and propagation effect of the information can be clearly seen.

In [40], the authors recover the occupancy of each cell using the (4.5); in our application, a similar function can be used for determining *Valid* and *Invalid* cell respectively, indeed, the range of values can be normalized in the interval $[0, 1]$ and the obtained values can be interpreted as probabilities.
Using this kind of approach it is possible to estimate a value of probability also in areas where no information is available. This is fundamental for Map Validation algorithm since it allows not only to determine if there is any part of the map invalid but also where this zone is.
This method is based on the conversion of probability values (which are limited in the interval $[0, 1]$) into variables characterized by a certain mean and variance; subsequently

it is also necessary the inverse conversion to be able to bring back everything in the desired range. All these transformations lead to obvious approximations that can affect the final performances; for this reason in the next section we propose a purely probabilistic approach that does not need the conversions defined above.



(a) No measurements considered

(b) Result after the first measurement

(c) Result after the first five measurements

(d) Result after the seven measurements

Figure 4.1.: Example Gaussian Filtering

4.2. Probabilistic approach

In this section we discuss a possible way to consider the spatial relations between the cells of a grid used for *OMV* purpose. In this step the Gaussian hypothesis of the random variables has been abandoned to leave space to a pure probabilistic approach. First of all, we try an *Ad hoc* probabilistic law based on a gaussian-like spread of information. After that we reproduce the same framework exploiting the *Dempster Shafer Theory* and we compare the result with the previous one (section 4.3).

4.2.1. Bayesian approach with sparse knowledge about the map state

For the sake of an uncluttered discussion, we consider at first the following assumptions:

- **Sparse measurement available**: Map **m** is known to be *Valid* (or *Invalid*) only on a limited number of points.
- **Spatial correlation between points**: The knowledge of the condition of one point p_i can provide some information regarding the surrounding points.
- $2+1$ **dimensional point coordinates**: Each point p_i is characterized by:
  - 2 spatial coordinates (x and y position)
  - a temporal dimension t that describe the age of the measure.

We further assume that:

- **Point sensors**: Sensor $s(p_i)$ provides knowledge about a single point p_i of the map **m**
- **Sensor values**: A sensor $s(p_i)$ assumes two different states:
  - $Valid$
  - $Invalid$

  and its value corresponds exactly to the actual state of the correspondent map point $m(p_i)$.
- **Binary map**: Each point of the map $m(p_i)$ assumes values:
  - $Valid$
  - $Invalid$

Probability Framework

In order to fuse the information coming from sensors and to create the surface described at the beginning, a probabilistic approach can be used. In particular, assuming the knowledge of some measurements of the map (*Valid, Invalid*) we want to compute the probability:

$$P\left[m(p_i)|s(p_{j1}),...s(p_{jn})\right] \tag{4.6}$$

Since, at this stage, we assumed that the value of each sensor $s(p_i)$ corresponds to the state of the map point $m(p_i)$, the (4.6) can be rewritten as

$$P\left[m(p_i)|m(p_{j1}),...m(p_{jn})\right], \tag{4.7}$$

that represents the probability (of validity) of a single point p_i given the knowledge on the points $p_{j1}...p_{jn}$. The (4.7) has to be characterized *Ad-hoc* in order to satisfy all the desired properties.
In subsection 4.2.2 it is considered the case of only one measurement available that is then generalized in subsection 4.2.3 where n measurements are used.

4.2.2. One measurement case

Let's consider the simple case with only one measurement available; in this case we want to determine

$$P\left[m(p_i)|m(p_j)\right] \tag{4.8}$$

that represent the probability which the point i of the map is *Valid*, given the point the knowledge of the single point state j.
This conditional probability must have the following two properties:

- $\lim_{\|p_i-p_j\|\to 0} P\left[m(p_i)|m(p_j)\right] = \gamma\left(m(p_i), m(p_j)\right)$

- $\lim_{\|p_i-p_j\|\to +\infty} P\left[m(p_i)|m(p_j)\right] = P\left[m(p_i)\right]$

where $\|p_i - p_j\|$ is distance between p_i and p_j, $P\left[m(p_i)\right]$ represent the prior knowledge on the point p_i and

$$\gamma(\cdot,\cdot) : \{Valid, Invalid\}^2 \to \{0,1\} \tag{4.9}$$

is defined as

$m(p_i)$	$m(p_j)$	$\gamma\left(m(p_i), m(mp_j)\right)$
Valid	Valid	1
Invalid	Valid	0
Valid	Invalid	0
Invalid	Invalid	1

Table 4.1.: Conditional probability of the map.

The (4.8) directly depends on the considered points and their distance. The effect of a measurement in p_j on point p_i will increase as soon the two points will be close to each others. Conversely, if p_i is further from p_j, there will be no interaction between them and (4.8) will collapse to the prior $P\left[m(p_i)\right]$. The desired properties can not be satisfied by any standard function and so an *Ad-hoc* one has to be defined.

A possible example is:

$$P\left[m(p_i)|m(p_j)\right] = \left(1 - k(\left\|p_i - p_j\right\|)\right) \cdot P\left[m(p_i)\right] + k(\left\|p_i - p_j\right\|) \cdot \gamma(\cdot, \cdot) \tag{4.10}$$

where $k(\cdot)$ is a monotonic function decreasing with the distance between p_i and p_j limited between 0 and 1. The latter can have different shapes and intuitively the most sensible idea is to choose a smooth function that varies continuously from 1 when its argument is 0, and 0 when its argument is $+\infty$ (in practice, after a certain value); an example of this behaviour is reported in Fig. 4.2.

Further details about $k(\cdot)$ and its possible definition are available in app. B.

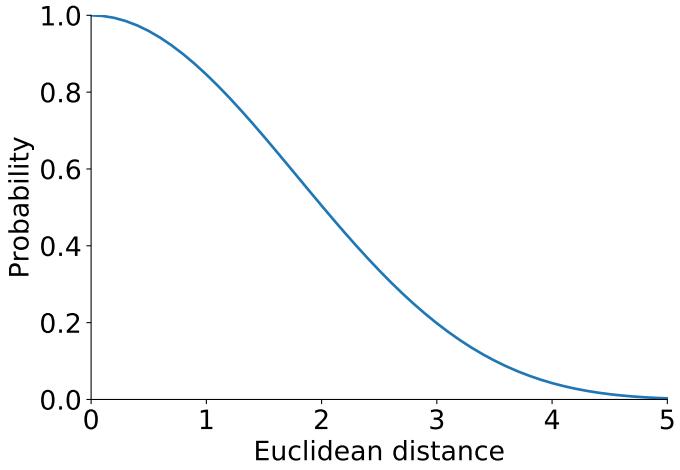


Figure 4.2.: $k(\cdot)$ function behavior

4.2.3. General case: n measurements available

Consider now the situation where n measurements are available. Since we suppose that the sensors returns exactly the actual state of the map in the point of interest, the probability to evaluate is (4.7).
Similarly to (4.8), the equation (4.7) has to take into account the different measurements available, giving more weigh to those that are closer to the point of interest.
More probabilistic information has to be fused simultaneously with the guarantee of respecting all the features of (4.8).
Multiple functions can be used to define (4.7): in case of prior probability of each point equal to 0.5, the solution proposed in this thesis is

$$P\left[m(p_i)|m(p_{j1}),...m(p_{jn})\right] = S\left(\sum_{h=j1}^{jn} L\left(P\left[m(p_i)|m(p_h)\right]\right)\right), \tag{4.11}$$

where $S(x)$ is the Sigmoid function

$$\begin{aligned} &S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1} \\ &\text{with} \\ &S : \mathbb{R} \to (0,1) \end{aligned} \tag{4.12}$$

and $L(x)$ is its inverse in the interval $(0,1)$: the Logit function

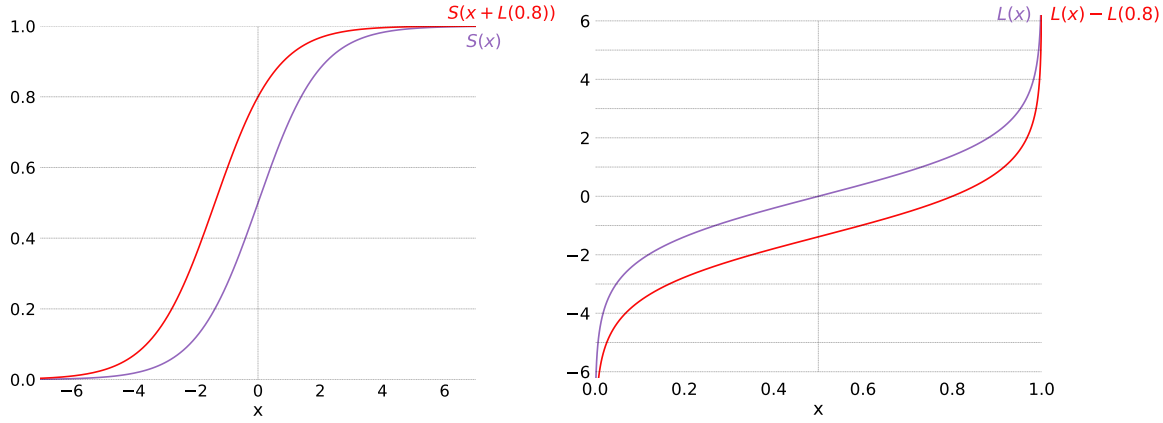
$$\begin{aligned} &L(x) = S(x)^{-1} = ln\left(\frac{x}{1-x}\right) \\ &\text{with} \\ &S : (0,1) \to \mathbb{R}. \end{aligned} \tag{4.13}$$

where $ln(\cdot)$ is the logarithm with base e.
These two functions are often used in the field of Machine Learning (ML) and their behavior is reported in Figure 4.3.

Using this combination of functions, it is possible to fuse the effect of different measurements (incorporated in $P\left[m(p_i)|m(p_h)\right]$), in a continuous way. Indeed, in case of only one measurement available, the (4.11) will simplify in the Equation 4.8 as desired. Furthermore, 4.11 strongly depends on the position of the measurements available: when $i \in \{j1...jn\}$, its value has to collapse in the function $\gamma(m(p_i)|m(p_i))$ i.e.

$$P\left[m(p_i)|m(p_{j1}),\ldots,m(p_i),\ldots,m(p_{jn})\right] \overset{(1)}{=} P\left[m(p_i)|m(p_i)\right] \overset{(2)}{=} \gamma(m(p_i)|m(p_i)) \tag{4.14}$$

where the first equality derives from the the conditional independence of the variables and the second equality comes from the (4.10). This property is correctly managed by



(a) $S(x)$ in violet and $S(x + L(0.8))$ in red

(b) $L(x)$ in violet and $L(x) - L(0.8)$ in red

Figure 4.3.: Sigmoid function and its inverse: the Logit function

4.11 as it is formally proven in app. C.

Consider now the situation where the prior probability value is not fixed and equal to 0.5. The approach saw so far can still be used after small improvements; indeed, calling $P[m_i]$ the prior probability of the considered point, it is sufficient to shift the Logit function along the y axis and the Sigmoid function along the x axis in such a way they across the point $(P[m_i], 0)$ and $(0, P[m_i])$ respectively. This process is shown in fig. 4.3 where the value $P[m_i] = 0.8$ is considered.
The final combination formula, that can manage any prior probability value in the interval $(0, 1)$, is reported in Equation 4.15.

$$P\left[m(p_i)|m(p_{j1}), ...m(p_{jn})\right] = S\left(\sum_{h=j1}^{jn}\left[L\left(P\left[m(p_i)|m(p_h)\right]\right) - L\left(P\left[m(p_i)\right]\right)\right] + L\left(P\left[m(p_i)\right]\right)\right) \tag{4.15}$$

It can be easily verified that in case of $P[m_i] = 0.5$, (4.15) coincides with (4.11) since $L\left(P\left[m(p_i)\right]\right) = 0$.
Furthermore, the property 4.14 it is also valid for 4.15 and it can be proven by some small modifications of the proof available in app. C.

4.2.4. Time effect

A possible extension of the approach involves the introduction of the aging temporal effect of the available information and a subsequent return to the prior. The aging of

the measurements can be easily interpreted as an additional distance to be taken into account in the approach. Further information can be found in the appendix E.

4.2.5. Summarize

The probability relation among points can be summarized by the following two relations:

$$P\left[m(p_i)|m(p_{j1}),...m(p_{jn})\right] = S\left(\sum_{h=j1}^{jn}\left[L\left(P\left[m(p_i)|m(p_h)\right]\right) - L\left(P\left[m(p_i)\right]\right)\right] + L\left(P\left[m(p_i)\right]\right)\right) \tag{4.16}$$

with:

$$P\left[m(p_i)|m(p_h)\right] = (1 - k(\|p_i - p_h\|)) \cdot P\left[m(p_i)\right] + k(\|p_i - p_h\|) \cdot \gamma(\cdot) \tag{4.17}$$

and:

$$P\left[m(p_i)|m(p_{j1}),...m(p_{jn})\right] = \gamma(\cdot) \quad \text{if } i \in \{j1,\dots jn\}\,. \tag{4.18}$$

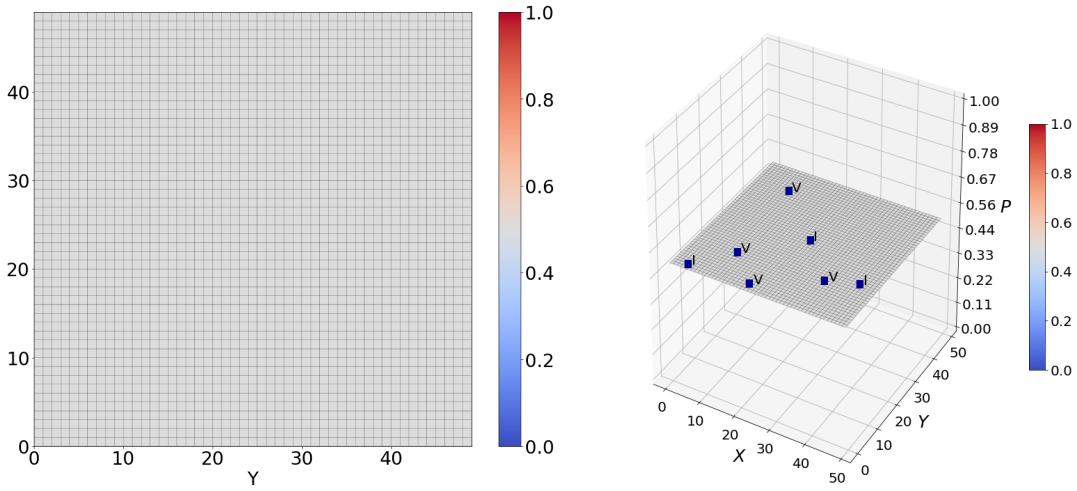
Using (4.16), (4.17) and (4.18) it is possible to fuse multiple validity measurements in a fully probabilistic way. This approach avoids many of the conversions and approximations that would be necessary with other algorithms (an example is the Kalman-like structure presented in section 4.1) and it allows to obtain a result in a more direct way. In subsection 4.2.6, some examples that better describe the behaviour of the developed algorithm are reported.

4.2.6. Examples

First example

Consider the following example:

- The map is discretized and described by squared lattice composed of 50 times 50 cells.
- For each cell is available no more than one measurement.
- Each cell assumes a single value of probability.
- The value (Valid or Invalid) of n isolated points is available and it is necessary to update the remaining points of the map. In particular, the points considered in this example are 7 and respectively placed in position [4, 2], [10, 40], [12, 12], [25, 25], [36, 13], [44, 16], [21, 3]. Three of them are Invalid measurements and the others are considered Valid.



(a) Flat representation

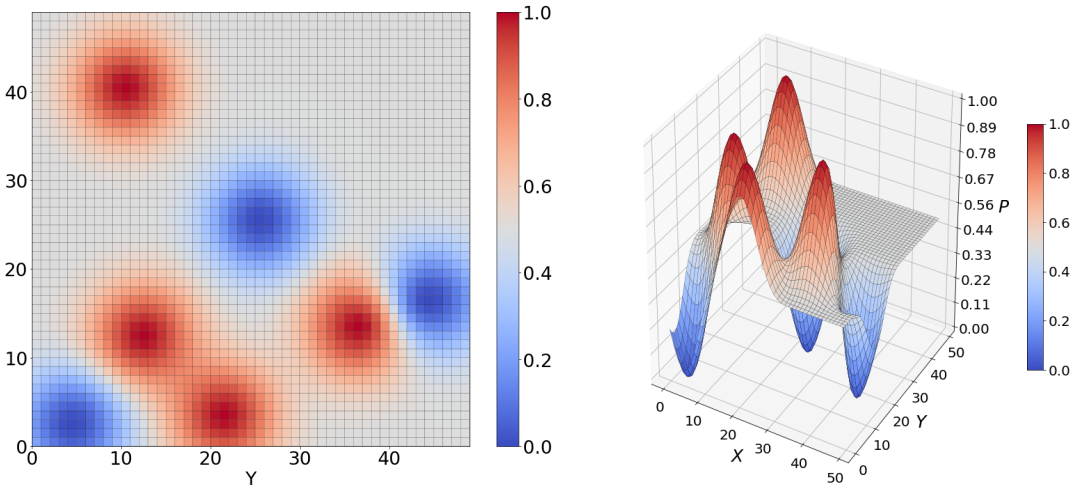
(b) 3D representation

Figure 4.4.: 50x50 grid with flat prior equal to 0.5

- The prior probability is considered completely flat on the surface analyzed and equal to 0.5 for each cell. This assumption corresponds to a complete ignorance of the map state (Valid or Invalid).

In Figure 4.4 is reported the initial conditions of the map (flat prior equal to 0.5) and the position of the known point. A label 0 means that in that point the map is *Invalid*, conversely, the label equal to 1 means map *Valid*. In Figure 4.5 is reported the validity map of the example after the update with the approach described previously.

Observing Figure 4.5, the interpolation exerted by the implemented approach is evident. We can see that the effect of the measurements is localized in a neighborhood (as we want) and decrease with the distance.
Since, at this stage, we suppose that the sensor measurements coincide exactly with the map states in the respective cells, the values assumed by these points can just be 0 (*Invalid*) or 1 (*Valid*) according with the function $\gamma(\cdot,\cdot)$ 4.1. If no data are available on a certain cell, the other measurements are taken into account and their influence will be as strong as the distance from the considered cell will be small. Furthermore, if the points are far from any measurement, their validity probability will remains at the prior value as it happen in the upper right corner of the picture. The transition from the value of one cell to the value of another cell is smooth since the function 4.16 is a combination of continuous functions.
In this case the value of β_{corr} used in Equation B.2 has been chosen equal to 0.4.



(a) Flat representation

(b) 3D representation

Figure 4.5.: 50x50 grid with flat prior after the spatial updating

Second example

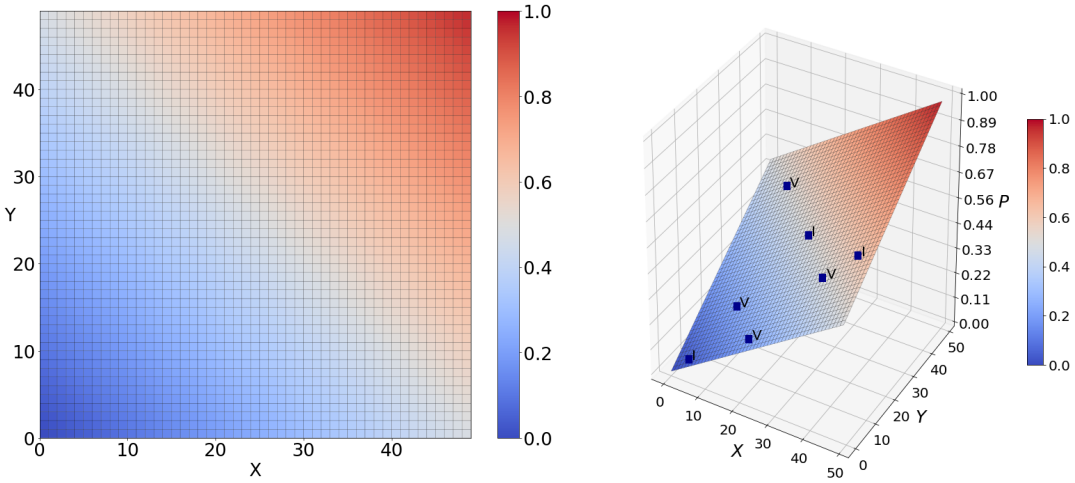
Consider now the case with tilted surface prior (fig. (4.6)). The prior probability is not equal to 0.5 in any cell and it varies from 0 to 1 depending on the location. The discretization of the area of interest and the measurements available are the same of the previous example.

The update with the measurements can be seen in Figure 4.7. Like in the previous example it is evident the interpolation effect of the algorithm. As can be seen, the final result is modulated according with the prior probability value. This is an important desired behavior since the prior probability represent a starting knowledge that can be exploited in order to have a better estimation of the state of the map.

4.2.7. Considering sensor uncertainties

Consider now an extension of the theory reported in subsection 4.2.1.
In this case the uncertainty of the sensors will be considered. Like in the previous case some assumptions are made:

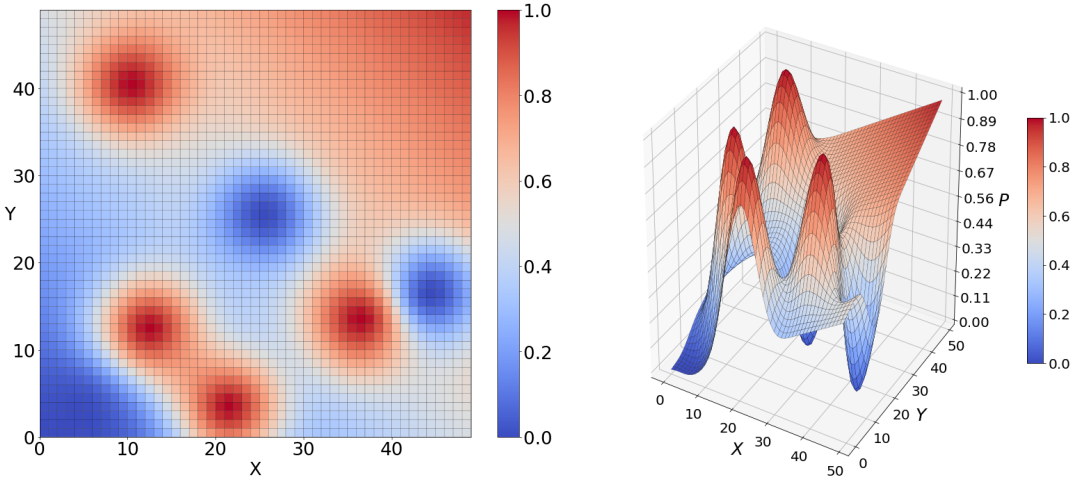
- We assume a spatial correlation between the points of the map.
- We suppose that the measurements came from sensors characterized by a proper



(a) Flat representation

(b) 3D representation

Figure 4.6.: 50x50 grid with tilted prior probability



(a) Flat representation

(b) 3D representation

Figure 4.7.: 50x50 grid with flat prior after the spatial updating

uncertainty. The sensors can assume three different values: *Valid*, *Invalid*, *Unknown* with a certain probability that depends from the sensor characteristics.

- Even if the map cells are correlated, the sensor measurements are considered independent from each other.

- The forward model of each sensor is considered known: this model, usually indicated with $p(z|m)$, specifies a probability distribution over sensor measurements z given a map m. In this thesis we consider this value available and independent from the position of the measurement.

In this framework, for every point of the map, we want to compute Equation 4.19

$$P\left[m(p_i)|s(p_{j1})\dots s(p_{jn})\right] \tag{4.19}$$

where $m(p_i)$ is the state of the map in the point i and $s(p_{j1})\dots s(p_{jn})$ represent the measurements available. In order to evaluate Equation 4.19

$$P\left[m(p_i), s(p_{j1}), \dots s(p_{jn})\right] \tag{4.20}$$

and

$$P\left[s(p_{j1}), \dots s(p_{jn})\right] \tag{4.21}$$

have to be evaluated. Consider Equation 4.20. Conditioning the sensor joint probability on the map value of the respective points we obtain:

$$\begin{aligned} P\left[s(p_{j1})\dots s(p_{jn})\right] &\overset{(1)}{=} \sum_{m=\{V,I\}^{jn}} \Big[P\left[s(p_{j1})\dots s(p_{jn})|m(p_{j1}),\dots m(p_{jn})\right] \\ &\cdot P\left[m(p_{j1}),\dots m(p_{jn})\right]\Big] \\ &\overset{(2)}{=} \sum_{m=\{V,I\}^{jn}} \left[\prod_{h=j1}^{jn} P\left[s(p_h) \mid m(p_h)\right] \cdot P\left[m(p_{j1})\dots m(p_{jn})\right]\right] \end{aligned} \tag{4.22}$$

where $m = \{V, I\}^{jn}$ represent all the possible combination of map validity values that the vector $m(p_{j1}), \dots m(p_{jn})$ can assume; the total number of possibilities is equal to 2^{jn} where jn is the number of points considered. Still considering the Equation 4.22, in (2) has been used the independence property of the sensors:

$$P\left[s(p_{j1})\dots s(p_{jn})|m(p_{j1}),\dots m(p_{jn})\right] = \prod_{h=j1}^{jn} P\left[s(p_h) \mid m(p_{j1}),\dots m(p_{jn})\right] \tag{4.23}$$

with the conditional independence hypothesis of the sensors given the map:

$$P\left[s(p_h) \mid m(p_{j1}),\dots m(p_{jn})\right] = P[s(p_h) \mid m(p_h)] \tag{4.24}$$

The problem is now to compute all the elements of the last term of (4.22).
Regarding $P\left[s(p_j) \mid m(p_j)\right]$, it is defined by the actual sensor

With reference to the joint probability $P\left[m(p_{j1})\dots m(p_{jn})\right]$, by applying the chain rule we have:

$$\begin{aligned} P\left[m(p_{j1})\dots m(p_{jn})\right] &= P\left[m(p_{j1}) \mid m(p_{j2}),\dots m(p_{jn})\right] \cdot P\left[m(p_{j2})\dots m(p_{jn})\right] \\ &= P\left[m(p_{j1}) \mid m(p_{j2})\dots m(p_{jn})\right] \cdot P\left[m(p_{j2}) \mid m(p_{j3})\dots m(p_{jn})\right] \cdot \\ &\quad \cdot P\left[m(p_{j3}),\dots m(p_{jn})\right]\dots \end{aligned} \tag{4.25}$$

Recalling the simplified case discussed in subsection 4.2.1 we have:

$$\begin{aligned} &P\left[m(p_i)|m(p_1),\dots m(p_{i-1}),m(p_{i+1}),\dots m(p_n)\right] = \\ &= S\left(\sum_{\substack{j=1 \\ i\neq j}}^{n}\left[L\left(P\left[m(p_i)|m(p_j)\right]\right) - L\left(P\left[m(p_i)\right]\right)\right] + L\left(P\left[m(p_i)\right]\right)\right) \end{aligned} \tag{4.26}$$

With

$$P\left[m(p_i)|m(p_j)\right] = (1 - k(\cdot)) \cdot P\left[m(p_i)\right] + k(\cdot) \cdot \gamma(\cdot) \tag{4.27}$$

Where $k(\cdot)$ is a correlation function depending on the distance between points and described in app. B. If $k(\cdot) = 0$ then $m(p_i)$ and $m(p_j)$ are spatially independents and $P\left[m(p_i)|m(p_j)\right] = P\left[m(p_i)\right]$.

Consider 4.20:

$$\begin{aligned} P\left[m(p_i), s(p_{j1})\dots s(p_{jn})\right] &\overset{(1)}{=} \sum_{m=\{V,I\}^{jn}}\Big[P\big[m(p_i), s(p_{j1})\dots s(p_{jn}) \mid m(p_{j1})\dots m(p_{jn})\big] \\ &\qquad \cdot P\left[m(p_{j1})\dots m(p_{jn})\right]\Big] \\ &\overset{(2)}{=} \sum_{m=\{V,I\}^{jn}}\Big[P\big[m(p_i) \mid s(p_{j1})\dots s(p_{jn}), m(p_{j1})\dots m(p_{jn})\big] \\ &\cdot P\big[s(p_{j1})\dots s(p_{jn}) \mid m(p_{j1})\dots m(p_{jn})\big] \cdot P\left[m(p_{j1})\dots m(p_{jn})\right]\Big] \end{aligned} \tag{4.28}$$

Like for the denominator, the first equality comes from the conditioning of the joint probability with the map states; the second equality comes from the use of the chain rule. We can now notice that the last part of Equation 4.28 coincide with Equation 4.22 and so:

$$P\left[m(p_i), s(p_{j1}) \dots s(p_{jn})\right] = \sum_{m=\{V,I\}^{jn}} \left[P\left[m(p_i) \mid s(p_{j1}) \dots s(p_{jn}), m(p_{j1}) \dots m(p_{jn})\right] \cdot \prod_{h=j1}^{jn} P\left[s(p_h) \mid m(p_h)\right] \cdot P\left[m(p_{j1}) \dots m(p_{jn})\right] \right] \tag{4.29}$$

Furthermore the term $P\left[m(p_i) \mid s(p_{j1}) \dots s(p_{jn}), m(p_{j1}) \dots m(p_{jn})\right]$ can be further simplified; consider the case with two points; we have:

$$P\left[m(p_i) \mid s(p_j), m(p_j)\right] = P\left[m(p_i) \mid m(p_j)\right] \tag{4.30}$$

namely we assume that a punctual sensor does not add any new information once we know the map on the same point. This fact can be generalized with n points and (4.20) become:

$$P\left[m(p_i), s(p_{j1}) \dots s(p_{jn})\right] = \sum_{m=\{V,I\}^{jn}} \left[P\left[m(p_i) \mid m(p_{j1}) \dots m(p_{jn})\right] \cdot \prod_{h=j1}^{jn} P\left[s(p_h) \mid m(p_h)\right] \cdot P\left[m(p_{j1}) \dots m(p_{jn})\right] \right] \tag{4.31}$$

Finally, combining 4.28 and 4.22 the results become:

$$P\left[m(p_i) | s(p_{j1}) \dots s(p_{jn})\right] = \frac{\sum_{m=\{V,I\}^{jn}} \left[P\left[m(p_i) \mid m(p_{j1}) \dots m(p_{jn})\right] \cdot \prod_{h=j1}^{hn} P\left[s(p_h) \mid m(p_h)\right] \cdot P\left[m(p_{j1}) \dots m(p_{jn})\right] \right]}{\sum_{m=\{V,I\}^N} \prod_{i=1}^{jn} \left[P\left[s(p_i) \mid m(p_i)\right] \cdot P\left[m(p_{j1}) \dots m(p_{jn})\right] \right]} \tag{4.32}$$

where:

$$P\left[m(p_i) | m(p_{j1}), \dots m(p_{jn})\right] = S\left(\sum_{h=j1}^{jn} \left[L\left(P\left[m(p_i) | m(p_h)\right]\right) - L\left(P\left[m(p_i)\right]\right) \right] + L\left(P\left[m(p_i)\right]\right) \right) \tag{4.33}$$

In this subsection it has been extended the probabilistic map validation framework, including the possible sensor uncertainties. The behaviour of the algorithm in presence of these uncertainties, is strongly related with the characteristics of the sensors (forward sensor model). In order to better understand the behaviour of the algorithm, some examples are available in subsection 4.2.8.

4.2.8. Examples

Here are reported some examples that better explain the approach just illustrated. In all the three considered cases the framework is the same of the first example in section 4.2.6, with the only difference that now the measurements can assume three possible values: Valid, Invalid, Unknown. The prior knowledge considered is completely flat and equal to 0.5, while $\beta_{corr} = 0.3$ (for more details about the parameter β_{corr} see app. B). For each

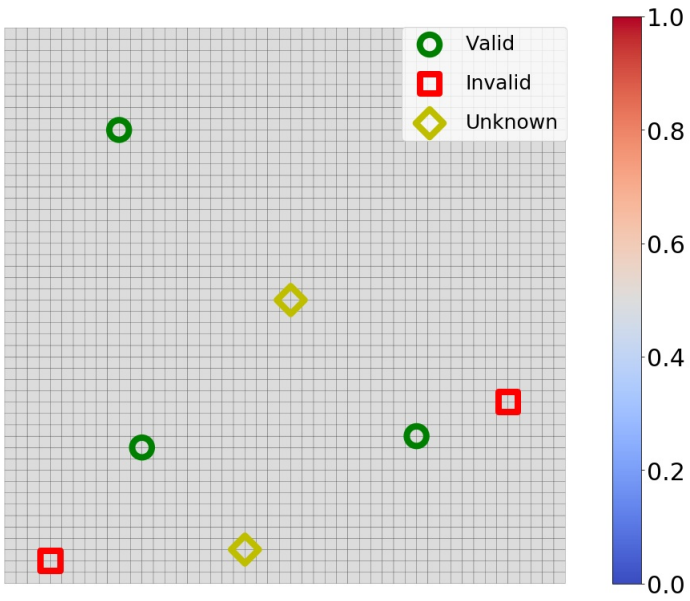


Figure 4.8.: 50x50 grid with flat prior and different measurements: green circle = *Valid*, red square = *Invalid*, yellow diamonds = *Unknown*

example is reported the image of the updated result (Figure 4.9, Figure 4.10, Figure 4.11) and a table that contain the values of $P\left[s(p_j) \mid m(p_j)\right]$ used (forward sensor model: Table 4.2, Table 4.3, Table 4.4).

As can be observed, the effect of the unknown measurements is strongly related with the defined values of $P\left[s(p_j) = Unknown \mid m(p_j)\right]$, relative to the specific example. Analyzing the result, we can see that, if $P\left[s(p_j) = Unknown \mid m(p_j) = Valid\right]$ is higher than $P\left[s(p_j) = Unknown \mid m(p_j) = Invalid\right]$, then, the effect of an Unknown measurement is comparable to the one of a Valid one.
Instead, if $P\left[s(p_j) = Unknown \mid m(p_j) = Valid\right] < P\left[s(p_j) = Unknown \mid m(p_j) = Invalid\right]$, the effect of Unknown measurements is similar to the case of the Invalid ones. In the end, if $P\left[s(p_j) = Unknown \mid m(p_j) = Valid\right]$ and $P\left[s(p_j) = Unknown \mid m(p_j) = Invalid\right]$ are similar, the effect of Unknown data, is equivalent to the lack of information. This

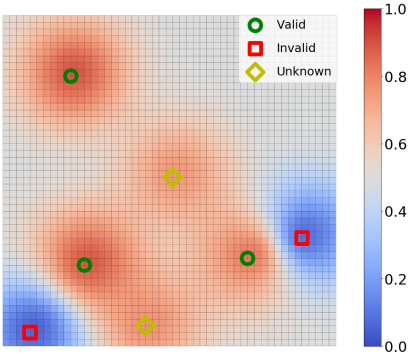


Figure 4.9.: First example

$s(p_i)$	$m(p_j)$	$P\left[s(p_j) \mid m(p_j)\right]$
Valid	*Valid*	0.8
Invalid	*Valid*	0.05
Unknown	*Valid*	0.15
Valid	*Invalid*	0.1
Invalid	*Invalid*	0.85
Unknown	*Invalid*	0.05

Table 4.2.: Forward sensor model, first example

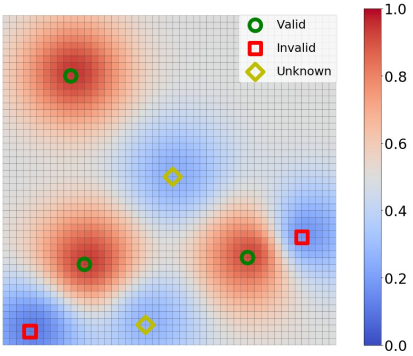


Figure 4.10.: Second example

$s(p_i)$	$m(p_j)$	$P\left[s(p_j) \mid m(p_j)\right]$
Valid	*Valid*	0.85
Invalid	*Valid*	0.1
Unknown	*Valid*	0.05
Valid	*Invalid*	0.05
Invalid	*Invalid*	0.8
Unknown	*Invalid*	0.15

Table 4.3.: Forward sensor model, second example

behavior can be easily explained by the fact that an Unknown measure is substantially different from the ignorance of the state of the map. In the first example, the unknown measure behaves similarly to a valid one since the state of map valid is more probable than map Invalid. Conversely, in the second example the state of map invalid is more likely than map valid when an Unknown measure occurs: for this reason Unknown and Invalid data behaves similarly. Finally, in the last example $P\left[s(p_j) = Unknown \mid m(p_j) = Valid\right]$ and $P\left[s(p_j) = Unknown \mid m(p_j) = Invalid\right]$ are comparable and so in that point the validity of the map tends to the prior. In general, with the Bayesian approach it is difficult to represent the total ignorance of an event since it is always necessary to define a proper value of probability to associate. In order to exceed this problem, different theories have been developed during the years: one of them is the so called **Dempster Shafer Theory**.

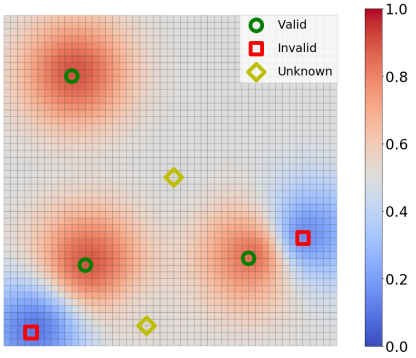


Figure 4.11.: Third example

$s(p_i)$	$m(p_j)$	$P\left[s(p_j) \mid m(p_j)\right]$
Valid	*Valid*	0.85
Invalid	*Valid*	0.1
Unknown	*Valid*	0.05
Valid	*Invalid*	0.1
Invalid	*Invalid*	0.85
Unknown	*Invalid*	0.05

Table 4.4.: Forward sensor model, third example

4.3. Dempster-shafer approach

One of the main drawback of the Bayesian theory is its inability to represent the ignorance of an event. In this section we will show how the *DST* (Dempester Shafer Theory) is a generalization of the Bayesian probability, where the use of a *Belief* function instead of the probability, lead to a greater flexibility in the representation of the uncertainty. To exaplain better the problem, let's introduce an example: consider a car sensor designed to recognize the presence of objects and assume that the precision of the sensor is 90%. In the Bayesian framework, if the sensor recognizes a car, this implicitly means that with 90% of probability there is a car and with the 10% of probability the area is free. With the *DST* instead, the 90% represents the *Belief* and the remaining 10% can be assigned to any other state of the system: for example, to another type of object (pedestrian, bycicle, dog, ecc), to a combination of them or to an unknown state that represent the total ignorance.

4.3.1. Characteristics

In order to characterize the *DST*, first of all, it is necessary to introduce the so called ***Frame of Discernment*** (*DOF*), Θ:

$$\Theta = \{X_1, X_2, \ldots, X_K\} \tag{4.34}$$

It is a finite set of all possible exlusive outcomes of a random variables.
In the case of a finite and discrete *DOF*, *DST* can be interpreted as a generalization of the classical Probability Theory in which the probabilities are assigned to sets and not to independent single events. In this way it assumes particular importance the *power*

set of the system were this probability is assigned. When the evident data are sufficient to assign the probabilities to the single events, then the *DST* collapses in the classical probabilistic theory.
Let's define now the functions that characterize the *DST*:

- ***Basic Probability Assignment (bpa)(m)***: This function describes a relation between *power set* and the interval $[0, 1]$ in wich the *bpa*, usually indicated with m, is defined. The value assigned to the empty set is 0 and the sum of the *bpa* in all the different subsets must be equal to 1. The properties of the *bpa* are reported in (4.35) and (4.36):

$$m : 2^{\Theta} \rightarrow [0, 1] \tag{4.35}$$

$$\sum_{A \in 2^{\Theta}} m(A) = 1 \qquad m(\emptyset) = 0 \tag{4.36}$$

  where 2^{Θ} represent the *power set* of Θ, $\emptyset$ is the empty set and A is a generic set of 2^{Θ}, i.e. $A \in 2^{\Theta}$. The values that the function m assumes are called *basic probability mass* (this is the reason why the *bpa* is indicated with m).

The use of the *DST* leads to a greater flexibility compared with the classical Bayesian theory thanks to the definition of a Lower bound and Upper bound on the probability. These two bounds are respectively called *Belief* and *Plausibility*:

- ***Belief (Bel)***: it represents the intensity of all the measurements that support a proper decision and the value that it assume is reported in Equation 4.37

$$Bel(X) = \sum_{A \subseteq X} m(A). \tag{4.37}$$

- ***Plausibility (Pl)***: it represent the intensity of all the tests that do not conflict with the hypotheses, namely the intensity of the tests that are not in favor of other hypotheses. The plausibility is usually indicated with Equation 4.38

$$Pl(X) = 1 - \sum_{A \cap X = \emptyset; A \in 2^{\Theta}} m(A). \tag{4.38}$$

The (4.37) and the (4.38) allow us to grasp the imprecision in attributing the probabilities to the events of our system and approximately define the range in which the probability values are defined (4.39):

$$Bel(X) \leq P(X) \leq Pl(X). \tag{4.39}$$

In case the range described by $Bel(X)$ and $Pl(X)$ collapses in a singleton, the *DST* converges to the Bayesian theory. In Figure 4.12 is reported a graphical representation of the relation between *Belief, Plausibility* and *Probability*.

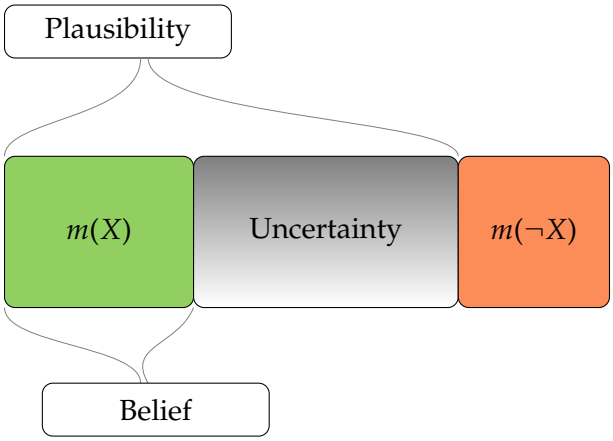


Figure 4.12.: Uncertainty interval

4.3.2. Combination rule

Another important aspect to define is the combination rule necessary to fuse information coming from different sources. This fusion can be done using three possible approaches:

- **Conjunctive Evidence**: if both the source information are considered reliable, they can be combined using an *AND* operation. The relation that provide this computation is reported in Equation 4.40

$$(m_1 \cap m_2)(A) = \sum_{B \cap C = A} m_1(B) \cdot m_2(C) \quad \forall A \subseteq \Theta \tag{4.40}$$

  where Θ is the considered frame of discernment.

- **Disjunctive Evidence**: if at least one of the two source information is considered reliable and we do not know exactly witch one is, it is reasonable to use an *OR* operation. In this case the relation is reported in Equation 4.41

$$(m_1 \cup m_2)(A) = \sum_{B \cup C = A} m_1(B) \cdot m_2(C) \quad \forall A \subseteq \Theta \tag{4.41}$$

- **Trade off Evidence**: it is an hybrid approach between the first two previously described and it takes into account different combination of *OR* and *AND* among the various input sources.

Even if this last strategy is more flexibility compared to the Bayesian approach, these additional degrees of freedom must be managed appropriately, since from their choice depends the goodness of the solution found.
The criterion originally used by Dempster and Shafer is the the so called *Dempster Shafer Rule* (*DSR*) and it is a generalization of the *Bayes rule*. This rule is based on the *Conjuction evidence* approach and so it works properly only when the conflict among sources is limited, otherwise it can bring to unexpected or, at least, difficult to understand results. In (4.42) is reported the *Dempster Shafer Rule*.

$$(m_{12})(A) = \frac{\sum_{B \cap C = A} m_1(B) \cdot m_2(C)}{K} \quad \forall A \subseteq \Theta \tag{4.42}$$

where K is

$$K = 1 - \sum_{B \cap C = \emptyset} m_1(B) \cdot m_2(C) = \sum_{B \cap C \neq \emptyset} m_1(B) \cdot m_2(C) \tag{4.43}$$

and it acts as normalization constant. It is important to notice that the numerator of the *DSR* is normalized with respect to the mass of the consistent tests of the two sources ($B \cup C \neq \emptyset$) and the rule itself assigns all the existing conflict between the different sources to the empty set, thus ignoring the contradiction present in the system. In order to quantify the conflict of the system, the following specific measurement can be defined:

$$Con(Bel_1, Bel_2) = \log\left(\frac{1}{K}\right) \tag{4.44}$$

$$= \log\left(\frac{1}{1 - \sum_{B \cap C = \emptyset} m_1(B) \cdot m_2(C)}\right) \tag{4.45}$$

$$= -\log\left(1 - \sum_{B \cap C = \emptyset} m_1(B) \cdot m_2(C)\right). \tag{4.46}$$

4.3.3. Decision

Once that the information fusion is performed, a decision rule must be defined (in our case we want to determine if the cell state is (*Valid* or *Invalid*). A possible solution is the use of the *pignistic transformation* [43]:

$$P(A) = \sum_{X \in 2^{\Theta}} m(X) \cdot \frac{|A \cap X|}{|X|} \tag{4.47}$$

where $|X|$ is the cardinal of the subset X. With the (4.47) it is also possible to compare the *DST* with the classical Bayesian theory.

4.3.4. Formalization of the map validation problem

As in the Bayesian case, we suppose that every map cell can assume only two opposed state: *Valid* and *Invalid*. So, the frame of discernment is characterized by only two symbols:

$$\Theta = \{V, I\}, \tag{4.48}$$

and the power set of the frame of discernment is:

$$2^{\Theta} = \{\emptyset, V, I, \{V, I\}\}. \tag{4.49}$$

Suppose that the sensors are reliable even if they can make mistakes. In this case the simplest choice is to update the *bpa* with the *Dempster rule*.

Grid initialization

During the initialization, the total absence of knowledge is represented assigning a zero mass distribution to both the *Valid* and Invalid states. This implicitly implies that the mass of $m_i(\{V, I\})$ is equal to 1 for each cell:

$$m_i(V) = m_i(I) = 0 \quad \forall i \tag{4.50}$$

$$m_i(\{V, I\}) = 1. \tag{4.51}$$

If, at the beginning, some prior information is available, the mass can be distributed in a different way among the various states according with this prior knowledge.

Spatial propagate sensor information

We suppose that the available sensors, can provide information of a limited number of isolated map points. In these cells the mass m_h of the $h - th$ point is directly updated according with following:

$$\text{if sensor output} = Valid$$

$$\begin{cases} m_h^s(V) = P\left[m(p_h) = Valid | s(p_h) = Valid\right] \\ m_h^s(I) = 0 \\ m_h^s(\{V, I\}) = 1 - m_h^s(V) - m_h^s(I) \end{cases}$$

$$\text{if sensor output} = Invalid$$

$$\begin{cases} m_h^s(V) = 0 \\ m_h^s(I) = P\left[m(p_h) = Invalid | s(p_h) = Invalid\right] \\ m_h^s(\{V, I\}) = 1 - m_h^s(V) - m_h^s(I) \end{cases}$$

$$\text{if sensor output} = Unknown$$

$$\begin{cases} m_h^s(V) = 0 \\ m_h^s(I) = 0 \\ m_h^s(\{V, I\}) = 1 - m_h^s(V) - m_h^s(I) \end{cases}$$

where $P\left[m(p_h) = Valid | s(p_h) = Valid\right]$ represent the probaility that the map is $Valid$ in the point p_h if the sensor in the same point measure $Valid$.
Instead, $P\left[m(p_h) = Invalid | s(p_h) = Invalid\right]$ represent the probability that the map is $Invalid$ in the point p_h if the sensor in the same point measure $Invalid$.
Since we want to spread this limited information on the whole considered area, a spatial correlation among cells is introduced.
For each cell it is so created a dummy measurement obtained as a weighted sum of the few data available. This step is performed with the following equation:

$$m_i^s = \frac{1}{\sum_{h=j1}^{jn} k\left(\|p_i - p_h\|\right)} \sum_{h=j1}^{jn} k\left(\|p_i - p_h\|\right) \cdot m_h^s \tag{4.52}$$

where m_i^s represent the dummy measurement of the $i-th$ cell obtained as combination of the few m_h^s available, and $k(\cdot)$ is a continuous function that takes into account the distance among points. A possible implementation of this function is described in app. B.

Update

Once a sensor measurement (dummy or not) is available for each point, the *Dempster Shafer Rule* can be applied:

$$m^{t+1}(V) = \frac{m^t(V)m^s(V) + m^t(V)m^s(\{V,I\}) + m^t(\{V,I\})m^s(V)}{1 - m^t(I)m^s(V) - m^t(V)m^s(I)} \tag{4.53}$$

$$m^{t+1}(I) = \frac{m^t(I)m^s(I) + m^t(I)m^s(\{V,I\}) + m^t(\{V,I\})m^s(I)}{1 - m^t(I)m^s(V) - m^t(V)m^s(I)} \tag{4.54}$$

Note that $m^{t+1}(\{V,I\})$ can be obtained as:

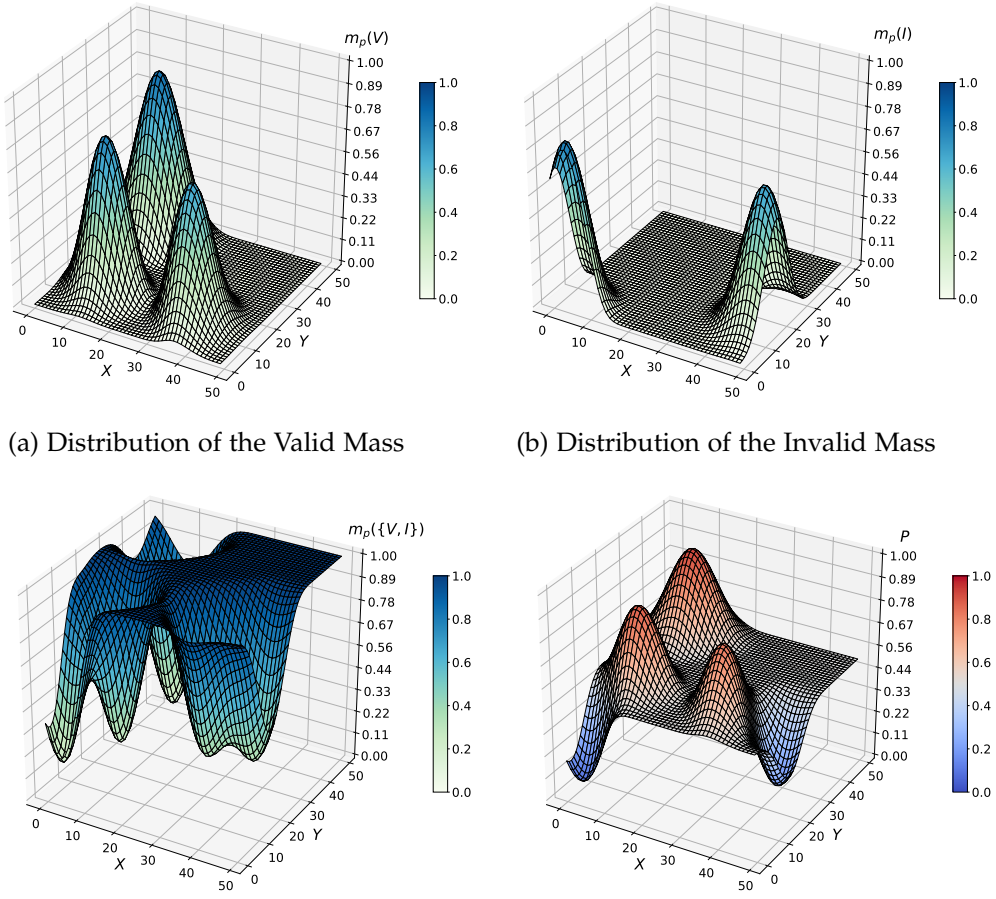
$$m^{t+1}(\{V,I\}) = 1 - m^{t+1}(V) - m^{t+1}(I) \tag{4.55}$$

4.3.5. Example

In this section it is reported an example of use of the *DST*.
The considered measurements are the same of the example reported in 4.2.8.
In this example in particular:

- $P\left[m(p_h) = Valid | s(p_h) = Valid\right] = 0.85$
- $P\left[m(p_h) = Invalid | s(p_h) = Invalid\right] = 0.85$

The results are reported in Figure 4.13.
Figure 4.13a, Figure 4.13b and Figure 4.13c represent respectively the distribution of *Valid* mass, *Invalid* mass and *Unknown* mass after the application of the propagation of the sensor information and the *Dempster-Shafer rule*. The mass transition from one cell and another is smooth and the sum of the three masses it is always equal to 1.
In Figure 4.13d the three masses are combined together using the Pignistic transform (4.47) in order to obtain the desired Validity map.
The result obtained with the *DST* can be compared with the third example in subsection 4.2.8. Indeed in both these example has been used similar parameters (same values of $P\left[m(p_h) = Valid | s(p_h) = Valid\right]$ and $P\left[m(p_h) = Invalid | s(p_h) = Invalid\right]$).
In Figure 4.14 is shown the two flat representation of the validity map obtained with the two approaches are placed side by side in order to be compared. From a qualitative point of view the results are very similar. Indeed, as can be seen in fig. 4.15, that represent the difference between the two previous maps, their discrepancy never exceed the value of 0.1.



(a) Distribution of the Valid Mass

(b) Distribution of the Invalid Mass

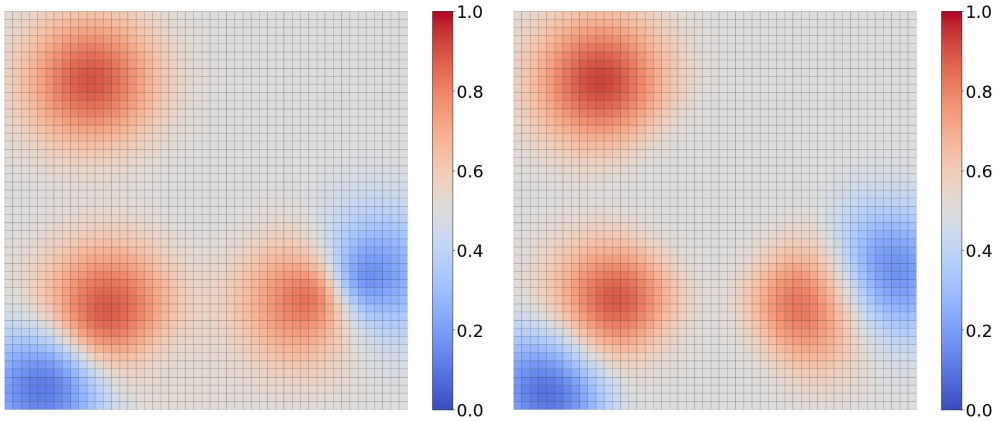
(c) Distribution of the Unknown mass

(d) Result of Pignistic transform

Figure 4.13.: Example of *DST* application

4.4. Summary

In this section we propose three different approaches to validate a Map using the spatial and temporal correlation of sparse information. The first approach, based on a Kalman-filter-like structure proposed in [40], allows a recursive solution of the problem that does not scale with time. Even if this approach seems promising, and in certain circumstances could be enough, this is approximated. So we have chosen to develop a second approach, based on a fully probabilistic relation among random variables. This was possible thanks to the definition of an ad-hoc function to combine the various measures. From a qualitative point of view, the obtained result is similar to the one



(a) Probabilistic approach (b) Dempster shafer

Figure 4.14.: Comparisong between maps obtained with probabilistic approach and dempster shafer

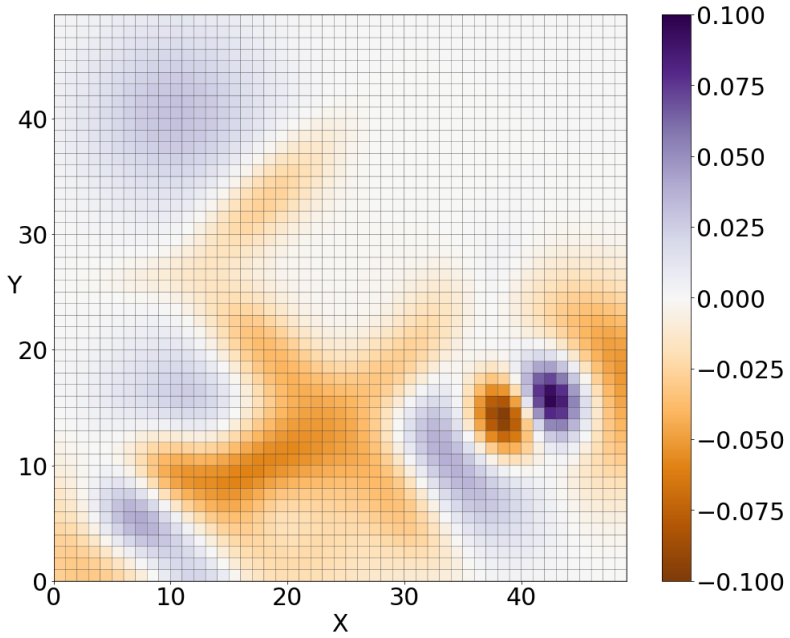


Figure 4.15.: Difference between the estimated map with probabilistic approach and with the DST (scale range: from -0.10 to 0.10)

of the previous approach, but in this case the information are directly represented as probability from the beginning and not with a certain mean ad variance that in this

framework can be difficult to understand. In the and, with the aim of introducing the concept of lack of information the theory of Dempster-Shafer is considered, defining, also in this case, a proper correlation among among variables. This last approach has been compared with the previous one and we can say that the resulting difference is negligible and dependent on the choice of the parameters.

5. Practical example

In this section we consider a simple scenario were the effect of three sensors: Radar, Lidar, and Camera is considered in the probabilistic framework described previously.

5.1. First example: planar space divided in multiple areas

As first example we consider the simple scenario defined by a car driving on a two-lanes road and where it is present a traffic sign on the border of the road. The sensors considered in this section are: Camera, Radar, and Lidar, characterized by different performances and field of view (FOV).

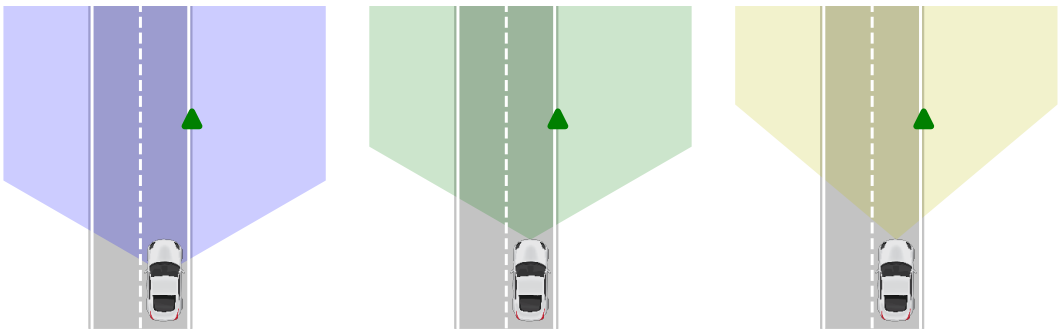
5.1.1. Assumption

We consider the following assumption:

- The camera is placed inside the windscreen (Figure 5.1a) and it is able to provide information about the position of road marking and traffic sign (red dot on the picture) with a field of view of 120°.
- The Lidar is placed in front of the vehicle (Figure 5.1b) and it is able to provide information about the position of road marking and traffic sign with a field of view of 120°.
- The radar is placed in front of the vehicle (same position of the lidar, Figure 5.1c) and it is able to provide information only about the position of the traffic sign with a field of view of 100°.
- the presence of detected lane marking or traffic sign is sure but their position can contain some errors.

Other hypothesis regard the partition of the area of interest considered:

- The area of interest is divided in four zones, delimited by the intersection of the different sensors FOVs (Figure 5.2a):



(a) Camera Field of View (b) Lidar Field of View (c) Radar Field of view

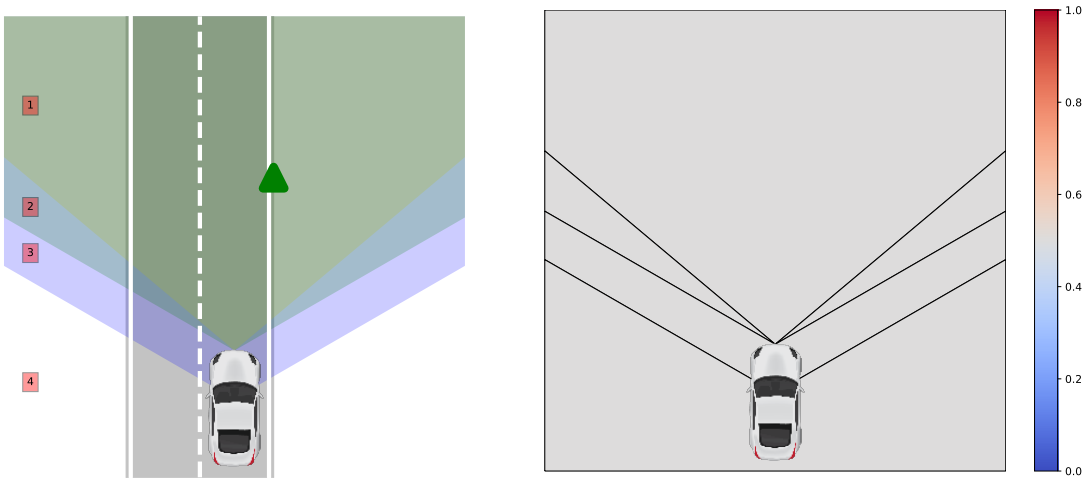
Figure 5.1.: Field of view of the different sensors

  - **Area 1**: it describes the common area where all the three sensors can provides information.
  - **Area 2**: it defines the zone where both Camera and Lidar can provide useful data and correspond to the difference between the field of view of the Lidar and the Radar.
  - **Area 3**: it is the area where the information can come only from the camera. It corresponds to the difference between the FOV of the camera and he FOV of the Lidar.
  - **Area 4**: it describes the zone were no measurements are available.

- The correlation among zones is evaluated with Equation B.2 where its argument is the distance between the centroids of the respective zones.
- The prior knowledge of the map validity is equal to 0.5 that correspond to the absence of prior information (Figure 5.2b)

5.1.2. Algorithm explanation

The main goal is to evaluate the reliability of the available map by comparing the latter with the sensor data. The map is assumed to be valid and, in case of sensors discrepancies, it is invalidated. The algorithm used in this section is the one proposed in 4.2.7. For each line (or traffic sign) detected by the sensors, the distance to the nearest line (or traffic sign) present in the map is calculated and used in the equation Equation B.2. The value assumed by Equation B.2, is considered the $P\left[sensor = Valid | map = Valid\right]$ and the parameter β (in this situation is denoted with β_{dist}) used in Equation B.2 acts as



(a) Areas of interest (b) Prior probability

Figure 5.2.: Experimental setup

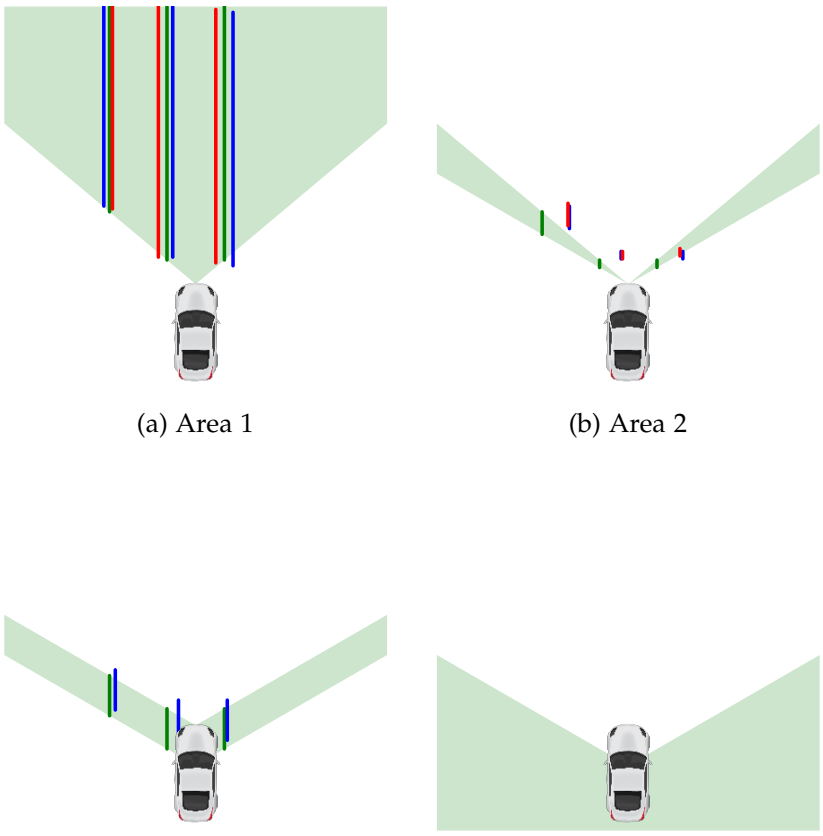
weigh parameter for the distance between area centroids. The sensor measurements are always considered valid and their effect in the validating procedure is modulated by the value that $P\left[sensor = Valid|map\right]$ assumes. For the correct functioning of the algorithm it is also necessary to define the probability of $P\left[sensor = Valid|map = Invalid\right]$. In this case the value is fixed and denoted with $invP$.

5.1.3. Data

Figure 5.3 shows the lane detection in the four areas of interest from two different sensors: the green lines represent the lane marking saved in the HD map, the blue and red lines describe the lane markings respectively detected by the Camera and the Lidar. From Figure 5.4 it can be seen the detection of a traffic sign represented by a triangle in the image. The different colors mean respectively:

- Green triangle: Traffic sign present in the map
- Blue triangle: Traffic sign detected by the Camera
- Red triangle: Traffic sign detected by the Lidar
- Yellow triangle: Traffic sign detected by the Radar

In order to test the algorithm, the displacement between map data and detected elements have been created adding a white Gaussian noise, zero mean and variance 0.5.



(a) Area 1 (b) Area 2

(c) Area 3 (d) Area 4

Figure 5.3.: Lane detection on the different areas

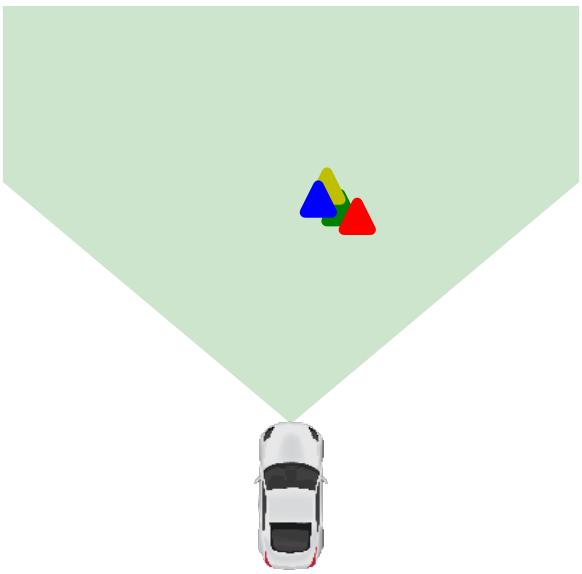


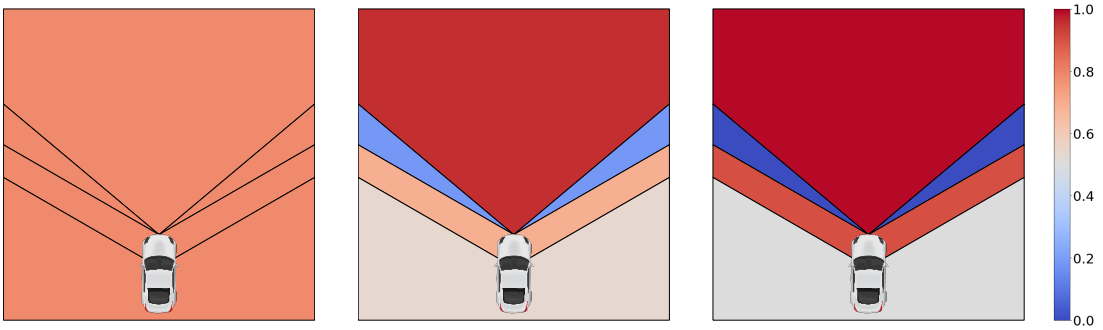
Figure 5.4.: Sign detection

5.1.4. Test

It is now possible to test the behavior of the approach by analyzing the different result obtained by varying the parameters that characterize the system. In order to better evaluate the effect of each parameter, they are made to vary one at a time.

First test $\beta_{dist} = 2.5$**,** $invP = 0.4$

In first test we try to fix $B_{dist} = 2.5$ and $invP = 0.4$ while β_{corr} can assume three different values: 0.01, 0.85 and 50 as reported in Figure 5.5. As we have already discussed in section 4.2, β_{corr} has the function to regulate the correlation between the different areas. An high value of β_{corr} corresponds to a complete independence of zones (Figure 5.5c): the measurements in one area do not affect the other and the probability transition among them can be very sharp. Conversely, if β_{corr} become smaller the relation between points increases (Figure 5.5b) to the point that all the cell assume the same probability value (Figure 5.5a). The validity value that each area assume is strongly dependent from their position in the space. In Figure 5.6 is shown the behavior of Validity probability when β_{corr} is varied in a continuous way from 0 to 2. The framework is the same of the previous example with $\beta_{dist} = 2.5$ and $invP = 0.4$.
Observing Figure 5.6 can be seen the convergence to the same point when β_{corr} tends to 0 and the fact that over a certain value of β_{corr} (approximately 1.75) the interaction



(a) $\beta_{corr} = 0.01$ (b) $\beta_{corr} = 0.85$ (c) $\beta_{corr} = 50$

Figure 5.5.: Comparison of the algorithm with different values of β_{corr}

between areas becomes negligible. An interesting behavior is present when β_{corr} varies in the interval $[0, 1]$. In that interval, the plot relative to *Area 3* and *Area 4* presents some oscillations before reaching the settling point. The main reason for the latter is the variation of the contribution of different terms as β_{corr} change: indeed with different values of spatial correlation the predominant area in the neighborhood (the area in the neighborhood that has more effect on the one under consideration) can be different.

Second test $\beta_{corr} = 50$**,** $\beta_{dist} = 2.5$

In the second test β_{corr} and β_{dist} are fixed equal to 50 and 2.5 respectively, instead $invP$ assume three different values: 0.12, 0.4 and 0.7. The various result are reported in Figure 5.7. As previously seen, $\beta_{corr} = 50$ correspond to a complete independence of the single areas; this choice permits to understand more easily the effect of varying other parameters. Observing the plots, it results that increasing the value of $invP$ (Figure 5.7c), the average validity probability decrease. Conversely, as soon $invP$ become smaller, the probability of the map increase. This behavior could be expected, since $invP$ represent $P\left[sensor = Valid | map = Invalid\right]$ and so the higher is the value the most unlikely the map is corrected.

Third test $\beta_{corr} = 50$**,** $\beta_{dist} = 2.5$

The last parameter to analyze is β_{dist} when β_{corr} and are fixed and respectively equal to 50 and 0.4. The algorithm result with three different values of β_{dist} $(1, 2.5, 5)$ is reported in Figure 5.8.

β_{dist} is the parameter that weight the effect of the discrepancies between map and measurements and it is fundamental for the definition of $P[sensor = Valid | map = Valid]$.

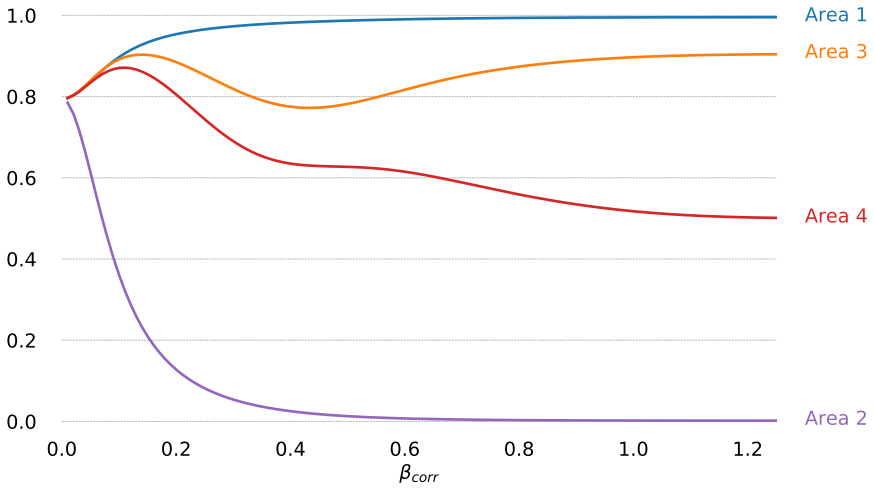
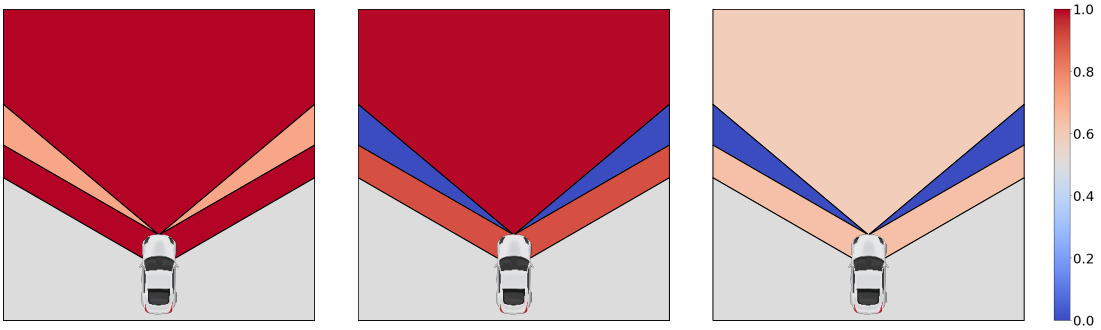


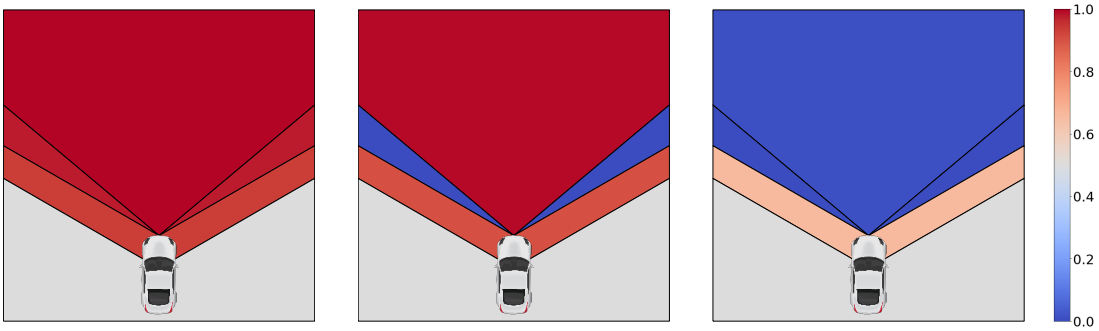
Figure 5.6.: Validity probability of individual areas as β_{corr} varies



(a) $invP = 0.12$ (b) $invP = 0.4$ (c) $invP = 0.7$

Figure 5.7.: Comparison of the algorithm with different values of $invP$

In particular, a relative high β_{dist} (Figure 5.8c) will weight more the anomalies with respect to a small value of the parameter (Figure 5.8a) with the result that the map will be easier considered invalid.



(a) β_{dist} = 1 (b) β_{dist} = 2.5 (c) β_{dist} = 5

Figure 5.8.: Comparison of the algorithm with different values of β_{dist}

6. Validation based on maneuvers of other traffic participants

As already mentioned, knowing the position and geometry of the road is essential for safety systems: spatial knowledge about the road is necessary for motion control systems, that need to know exactly were the ego-vehicle can move. Usually, the road detection is entrusted to the combination of different sensors such as Lidars, Cameras and Radars with a commercially available navigational road map. The accuracy of the information provided by the sensors depends on the environmental condition (light, presence of snow, rain, fog etc.) and on the availability of a free field of measurement. Usually, the main obstacles are other traffic participants which can hide not only the areas of the road under them, but also blocking large part of the FOV. However, in case of occlusion by other vehicles, if moving vehicles occlude parts of the road, they can still provide potentially significant information. For example, in [44], *Casapietra* et. al., use the observation and interpretation of other vehicles behavior to infer the presence and location of occluded road surface in a 2D grid-based map. Hence, predicting other traffic participants trajectories and understanding their maneuver can be particularly useful to highlight any anomalies in the available map, namely it can be used for *OMV*. In literature are available many possible approaches to recognize the driving maneuvers, for example: Andreas Lawitzky et. al. in [45] predicts the motion of vehicles on highways considering the interaction of the traffic participants. In this section we present a simple approach to obtain map validity information by exploiting the road characteristics observing the behavior of other vehicles.

6.1. Our approach

The vehicles moving on the road are subject to compliance with certain behaviors in order to guarantee a proper traffic flow and avoid accidents. One of the principal rules that affects all the vehicle on a road is the direction of travel to be respected, and if the common flow of cars does not respect the direction defined in the available map, an anomaly in the latter can be present. Hence, it is interesting to define a certain measure of *direction error* that can be converted in a probability measure and then fused with the other sensor information. This can be done by adding a proper layer on the

available map that describe the direction of travel as an oriented potential field. The latter can find many possible application in robotics like *Jan Vascak* did in [46] where the potential field framework is combined with neural networks for navigation purpose in an environment full of obstacles. In our case, the potential field attached to the map is defined according with the natural flow of cars allowed on the street. In Figure 6.1 is reported an example of potential field on a straight road with three possible maneuvers: to drive in accordance with the direction of travel of the lane, to overtake, to drive in the opposite direction of travel of the lane.

These tree different behaviors should lead to three results of a specific error function that characterize the behaviors themselves:

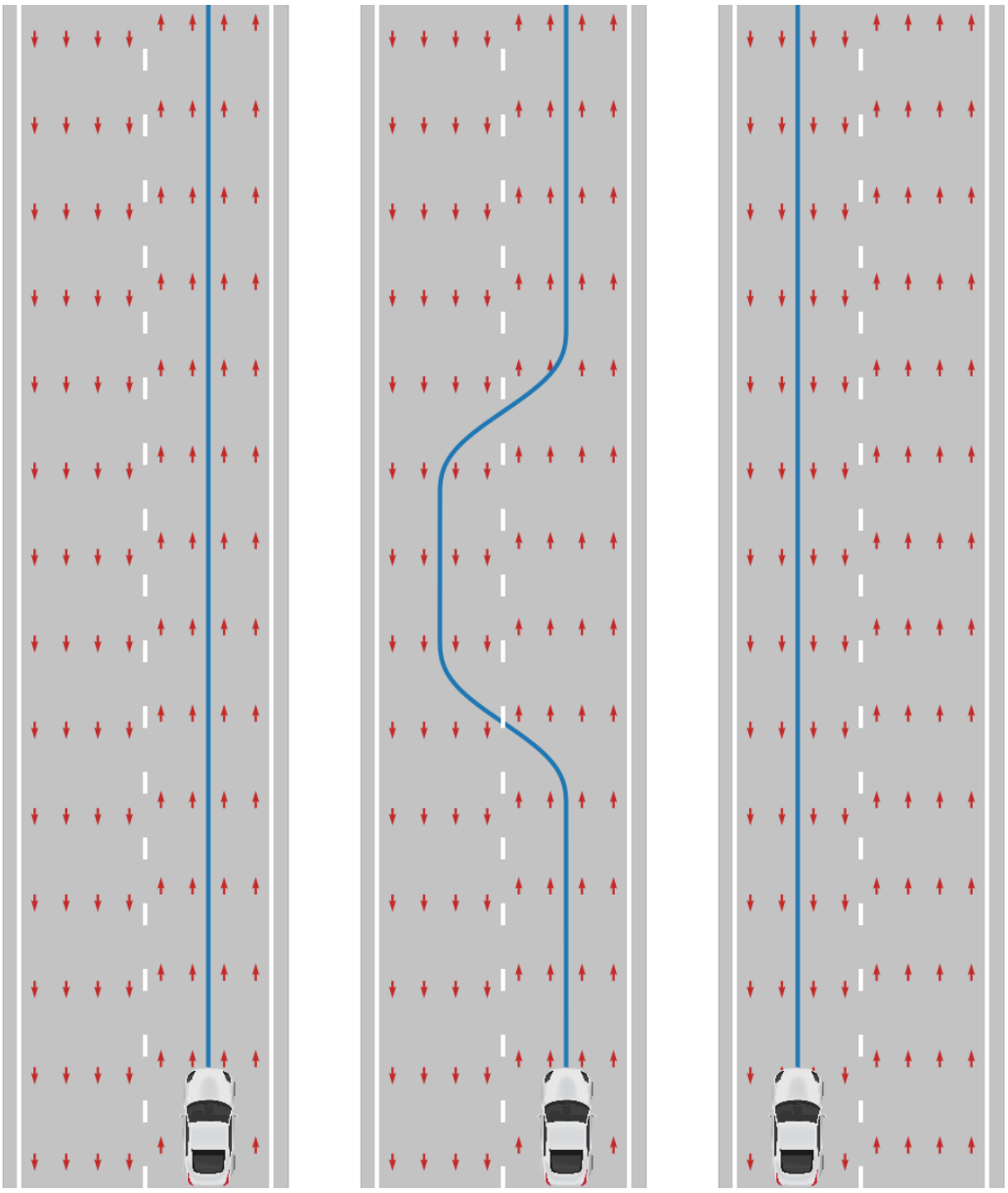
- in the first case the vehicle follows the rules of the road: this means that the error function should return a value as close as possible to zero.
- in the second case the car is performing an overtaking maneuver. Here two situations must be considered:
  1. The overtaking is allowed: the error has to be close to zero;
  2. The overtaking is prohibited: the error has to be high.
- in the third situation the user is driving in the wrong direction. This behavior should be highly penalized by the error function.

One of the simplest approach to define a proper *error function* is to compare the direction of the vehicle speed vector with the direction of the field in its position and accumulate it over time; in this way, the more the vehicle drives in a direction different from the on of the field, the higher the error will become. This simple *error function* is reported in Equation 6.1:

$$e(t+1) = e(t) + \alpha \cdot (\theta_{field} - \theta_{car}) \tag{6.1}$$

where θ_{field} and θ_{car} represent respectively the angle of the vector field and the car speed vector both refereed to the same global frame and α is a coefficient introduced to weight the effect of the angle variation. This approach could actually works in the situations where the overtaking maneuver is prohibited but can easily lead to a wrong result (and so wrong behavior interpretation) when the latter is allowed since it will create an error that will be propagated over time without ever being attenuated. Hence, it is necessary to include in the Equation 6.1 a proper forgetting factor that takes into account the needed *flexibility* of the rule and attenuate the effect of isolated wrong behaviors. The *error function* becomes the one reported in Equation 6.2.

$$e(t+1) = \gamma \cdot e(t) + \alpha \cdot (\theta_{field} - \theta_{car}) \tag{6.2}$$



(a) Right behavior (b) Overtaking (c) Wrong behavior

Figure 6.1.: Comparison of different behaviors

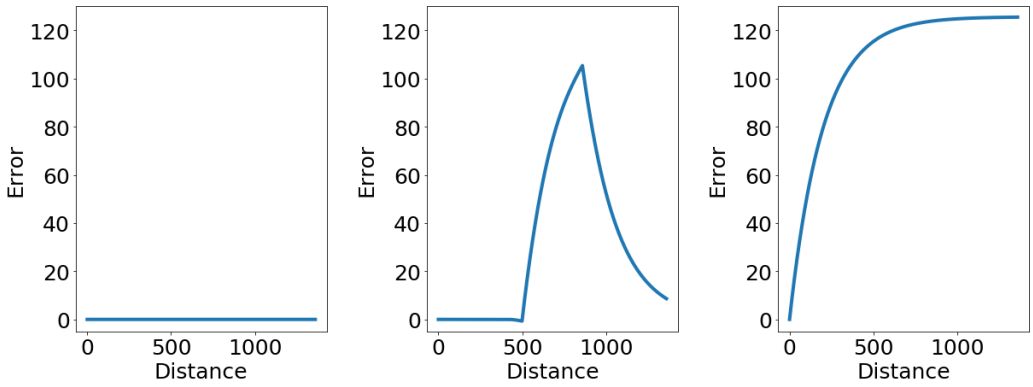


Figure 6.2.: Error function behaviour

where γ is the introduced forgetting factor. In Figure 6.2 is reported the output of Equation 6.2 in the three different cases: right direction, overtaking and wrong direction. In the first picture the error remains equal to zero for all the time, this is expected since the speed vector direction of the car is always equal to the one of the field. In the second case, the car direction is opposite to the field one during the overtaking maneuver. This lead to an increasing error during the first part of the plot and a successive decreasing when the car return back in its lane. In the latter, the shape of the curve is determined by the forgetting factor. In the last plot the error continues to increasing for all the time interval as expected (the car drives for all the time in the wrong direction). It is interesting the asymptomatic behavior of the curve that seems to stabilize about a specific value. This is due to the to the forgetting factor that compensates the increasing error. In Figure 6.3 is reported the comparison among the errors in the three different situations. In order to obtain a more understandable result, the error measurements can be converted in a value limited between 0 (high presence of error) and 1 (limited presence of error) using the normalized error function reported in Equation 6.1.

$$P = 1 - |tanh(k \cdot e(\cdot))| \tag{6.3}$$

where k is a multiplicative constant and $e(\cdot)$ is the error obtained from Equation 6.2. In Figure 6.4 is reported the behavior of the errors relative to the situations described previously after that Equation 6.1 is applied. The values obtained in this way can be interpreted as probability measures that specify how much the behavior of the vehicles is coherent with the traffic rules. Observing Figure 6.4, an unwanted result is shown; even if the overtaking maneuver can be allowed, the normalized error function return a value close to 0 in some instants; this is unavoidable since observing only the position and the speed, it is not possible to distinguish an overtaking maneuver from

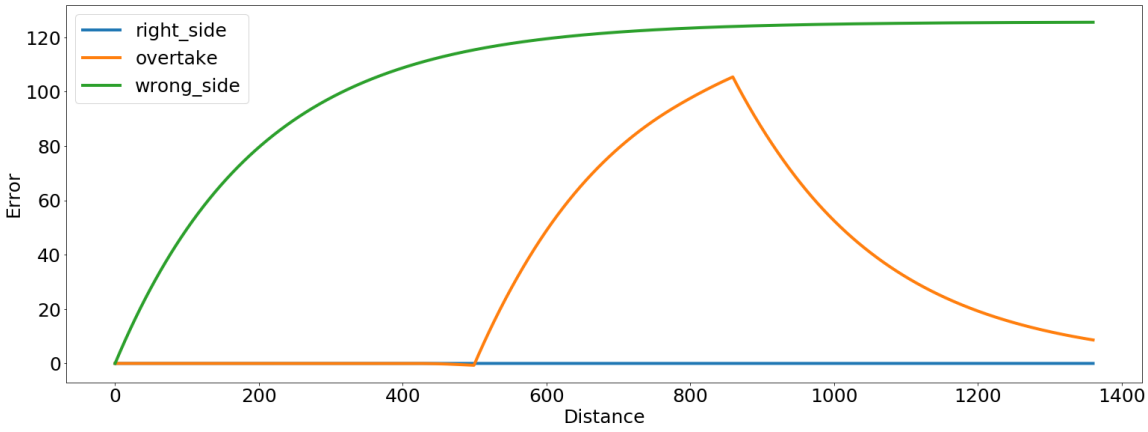


Figure 6.3.: Comparison of errors functions

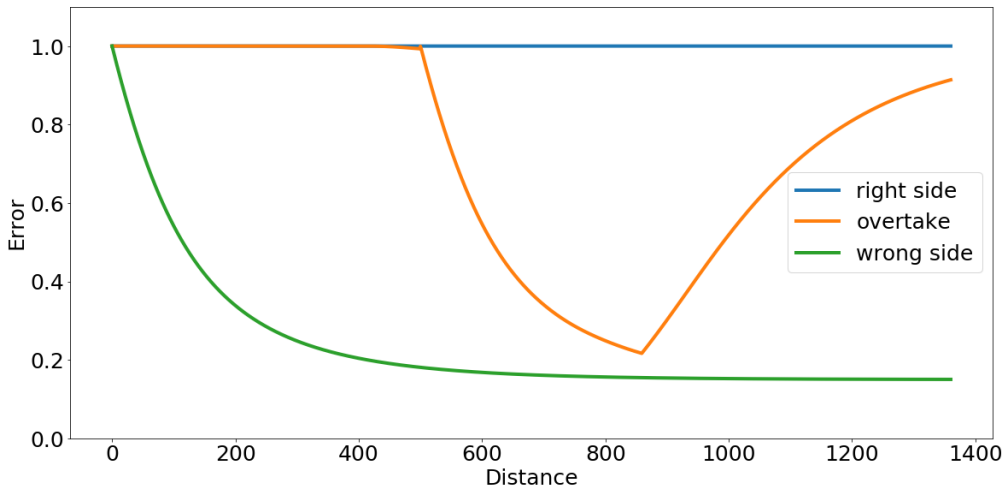


Figure 6.4.: Potential field on a straight road

a lane change or other similar action until all the maneuver is concluded. Obviously, the information so obtained is not enough to classify correctly the state of the map (Valid or Invalid) in all the cases, but it can provide some clues that can improve the performance of a map validation algorithm when they are fused with other information. In particular, all the reasoning based on the drivers behaviors, can become a specific node of the BN described in 3.2: the node *Observed behaviour* for example.

7. Conclusion

In this master thesis the goal is to develop a proper way to perform an Online Map Validation. This is needed for increasing driving comfort in AD system and mostly guarantee the safety. In this section the overall result of this thesis is summarized and briefly discussed.
Compared to other fields related with autonomous driving, the topic of map validation currently still lacks focus in research. For this reason, an extensive outlook for future work is given.

7.1. Summary

The two main ideas on which the work of this thesis is based are:

- a probabilistic representation of the information coming from the sensors to determine the validity of the map;
- the spatial-temporal interpolation of the knowledge in order to evaluate the probability of validity even in areas where no evidence is available.

The representation of experimental data in the form of probabilities allows us to completely detach ourselves from the physical characteristics of the sensors and combine the information at a higher level of abstraction. In this way the information can come not only from specific devices placed on the vehicle but also from semantic reasoning such as the respect of the traffic signs by road users. The transformation of an experimental measure into a probabilistic information is not always a direct process and therefore this operation must be defined case by case; an example of this is reported in chapter 6 where a possible, but certainly not unique, way to obtain a Valid Map probability from the behavior of the other vehicles is defined. In particular, in that specific example, the behavior considered is the possibility of vehicles in wrong direction on a straight path, but the used approach can be easily extended to other kind of roads. Indeed, the idea on which the approach developed chapter 6 is based is to associate with the road a sort of potential field that characterizes the admissible displacements of the vehicles. This field can be easily adapted to different environment and road pattern. The probability of a valid map will depend on the behavior of these vehicles and will be as close to

zero as the behavior of road users will be inconsistent with the potential field defined above. Furthermore, by introducing a forgetting factor, the duration of the incorrect behavior also has a decisive effect.
Therefore this idea allows a probabilistic interpretation of the behavior of road users based on a limited number of information; in our case we are supposed to know only the velocity vector of the surrounding vehicles.
In any case, in order to have a sufficiently precise evaluation of the validity of the map, it is necessary to consider different aspects of the surrounding environment. If multiple probabilistic information are available, they must to be fused in order to obtain a better result. In this thesis we have opted for the use of the Probabilistic Graphical Model and in particular the Bayesian Networks. The latter are oriented graphs that allow us to define dependency relationships between random variables that define our system. These variables can may also not be directly observable. In this way, starting from the observation of more general information (the validity of the road signs, the respect of the right of way, etc.) it is possible to go up in the network using the Bayesian inference and obtain a final probability measurement of validity. This value takes into consideration all the available information. The single Bayesian network can be so considered a sort of smart sensor that provides a final probability value based on the total value of the map.
Besides understanding if the map is invalid in general, for the map validation algorithms it is necessary to determine which precise part of the map is invalid and which (if possible) part is still usable. Unfortunately, the measures that can be used to determine the validity on the map are very often scattered and in very limited numbers. To address this problem, in this thesis we have proposed and compared different ways of interpolating this information using a correlation from the spatial and temporal point of view of the measurements (in this case we specifically consider the measures taken by the smart sensors defined above). More in detail, the first approach that was evaluated uses an alternative implementation of a Kalman filter and therefore it is based on the hypothesis of Gaussian random variables involved. Therefore, to use this method, it is necessary to convert average values and variance values into probabilities. This procedure can give rise to strong approximations. For these reasons, a purely probabilistic approach has been created. All the equations and relations necessary to face up to the problem, are developed step by step in this thesis and some Ad-Hoc function has been defined in order to characterize the relationship between areas of the map directly starting from probability values. The third approach developed, similar to the previous one, instead of using the Bayes theory, it is based on the Dempster-Shafer theory. Also in this case, some functions have been defined in order to spreed the limited information on all the area of interest.
The end result of these three approaches is very similar: a probabilistic surface that

specifies its probability of being valid at every point on the map. Even if from a purely qualitative point of view the three approaches are comparable, the second appears to have better characteristics than the others. First of all, the approach is purely probabilistic, therefore it does not need approximations that are inevitable with the Kalman-like structure. The probabilistic approach also turns out to be computationally more efficient than that one based on the Dempester Shafer theory since this last one requires the continuous updating of the masses of the Frame of Discernment.
Moreover the probabilistic approach allows to include in the algorithm a prior probability on the map validity that can be exploited to obtain better performances. For all these reasons this approach has been preferred to the others for the examples proposed in this thesis.
An interesting evolution of the algorithm has also been included in chapter chapter 5 where a more real example has been proposed. Three type of sensors have been taken into account and an alternative way of segmentation of the area of interest has been proposed. Indeed, the proposed approach is not fixed to the representation of the environment as a regular grid and a more clever segmentation of the map can be used. In the specific example proposed in the thesis, the area has been divided according to the intersection with the visual fields of the various sensors, but in general the possible segmentation are infinite. This allows in particular to reduce the computational complexity that for large areas can be relevant.
Closing, in this thesis we have proposed the various steps necessary to deal with the problem of map validation directly from a probabilistic point of view. This approach, although it needs improvements and clarifications, can represent a valid solution to the proposed problem as can be seen from the numerous examples proposed in this thesis.

7.2. Future Work

The idea developed in this thesis can undoubtedly be extended and improved. Possible fields of future research could be the use of more detailed sensor models in order to obtain more information from the surrounding environment and with better accuracy. More precisely, this research can be approached to an in-depth study of the conversion of measurable information directly from the sensors, in probability that can be used to validate the high resolution map. Some examples and approaches have been described in this thesis (in chapter 6 for example) but further developments must be made.
Another possible field of development can certainly concern the segmentation of the environment in order to limit the computational complexity (segmentation less fine than the surrounding area), while guaranteeing good performances. We have explained that the probabilistic proposed approach con be used without any problems on a map

segmented in an irregular way. By developing an algorithm capable of subdividing the environment in an optimal manner (for example, by segment the road used by the vehicle in a more refined way and the surrounding environment in a more coarse manner) could undoubtedly have improvements both in terms of performance and computational complexity.

Abbreviations

AD Autonomous-driving

ADAS Advanced Driver Assistance Systems

ASIL Automotive Safety Integrity Levels

BN Bayesian network

bpa basic probility assignement

CNN Convolutional neural network

DAG Directed acyclic graph

DOF Frame of Discernment

DS Dempster shafer

DSR Dempster shafer rule

DST Dempster shafer theory

FOV Field Of View

GPS Global Positioning System

ISO International Standard Organization

LFM Latent Fault Metric

LIDAR Light detection and Ranging

MDS Modified Dempster Shafer

ML Machine Learning

MN Markov Model

NDS Navigation Data Standard

OMV Online map validation

PGM Probabilistic Graphical Model

PMHF Probabilistic Metric for Random Hardware Faults

RADAR Radio detection and Ranging

SLAM Simultaneous localization and mapping

SPFM Single-Point Faults

Appendices

A. Recall to Bayes Theory

Given two events A and B, if they are in some way correlated, it is reasonable to think that the knowledge of one of them can give some information on the other one. A formal way to deal with this conditional probability is the Bayes theorem [47].
First of all we recall the conditional probability.
Given a probability space $(\Omega, \mathbf{A}, P(\cdot))$ where:

- Ω is a countable set and it contains all possible outcomes in classical sens.
- $\mathbf{A}$ event space that contains Ω
- $P(\cdot)$ probability function with domain $\mathbf{A}$ and codomain in $[0, 1]$

And, given two events A and B belonging to $\mathbf{A}$ we indicate with

$$P[A|B] \tag{A.1}$$

the probability of A occurring knowing that B has already occurred, ie the probability of A conditioned to B. Using the Bayes rule, Equation A.1 can be decomposed as:

$$P\left[A|B\right] = \frac{P\left[A, B\right]}{P\left[B\right]} \text{ with } P\left[B\right] \neq 0 \tag{A.2}$$

Moreover, for the chain rule we have:

$$P\left[A, B\right] = P\left[B|A\right] \cdot P\left[A\right] \tag{A.3}$$

And combining Equation A.2 with Equation A.3 we obtain the *Bayes Theorem*:

$$P\left[A|B\right] = \frac{P\left[B|A\right] \cdot P\left[A\right]}{P\left[B\right]} \text{ with } P\left[B\right] \neq 0 \tag{A.4}$$

As it is shown, we can calculate $P\left[A|B\right]$ as a function of the so called "inverse" probability $P\left[B|A\right]$ and this property is the basis of the Bayesian inference.
The Bayes theorem can be easily generalized.

Given the probability space $(\Omega, \mathbf{A}, P(\cdot))$ and be B_1, B_2,... B_n, elements of $\mathbf{A}$ with the following property:

- $\forall P\left[B_i\right] > 0$;
- $B_i \cdot B_j = 0$ with $i \neq j$;
- $\Omega = \bigcup_i B_i$ so, for every A belonging to $\mathbf{A}$

The Bayes theorem assumes the following form:

$$P\left[B_k|A\right] = \frac{P\left[A|B_k\right] \cdot P\left[B_k\right]}{\sum_i P\left[A|B_i\right] \cdot P[Bi]} \tag{A.5}$$

B. Local kernel correlation

B.1. The problem and the solution

There are multiple ways to define the correlation among different point of the map and finding a "good" function that is able to guarantee high performances with low computational cost can be challenging. The requested function must be smooth and decreasing with the distance; in particular if the distance between two points tends to infinity, the correlation should tend to zero. One possibility to solve this task is to use a Gaussian kernel. in Equation B.1:

$$k(\|p_i - p_j\|) = e^{\frac{-\beta_{corr}\|p_i - p_j\|^2}{2\pi}} \tag{B.1}$$

Even if it satisfy the required behaviour with the increasing distance, it tends to have an infinite range. This means that the value of the map at one point is (slightly) affected by points which are a huge distance away leading to an high computational cost of the algorithm. In order to minimize the latter it is important to define a correlation function that has non-zero value only for a small portion of the space. One example, is the following isotropic covariance function proposed by Storkey in [48]:

$$k(\|p_i - p_j\|) = \begin{cases} \frac{(2\pi - \Delta(\|p_i - p_j\|))\,(1 + (\cos\Delta(\|p_i - p_j\|))\,/2}{3\pi} + \\ \qquad + \frac{\frac{3}{2}\sin\Delta(\|p_i - p_j\|)}{3\pi}, & \text{if } \Delta < 2\pi \\ 0, & \text{otherwise} \end{cases} \tag{B.2}$$

where $\Delta = \beta_{corr}\|p_i - p_j\|_2$, for $\beta_{corr} > 0$. This function closely resemble a Gaussian kernel Equation B.1 but has zero value for distances larger than $2\pi/\beta_{corr}$.

In Figure B.1 the isotropic covariance function and the Gaussian Kernel are compared:

Varying the value of β_corr is possible to vary the range of action of the available measurements. In Figure B.2 are reported some examples with different values of β.

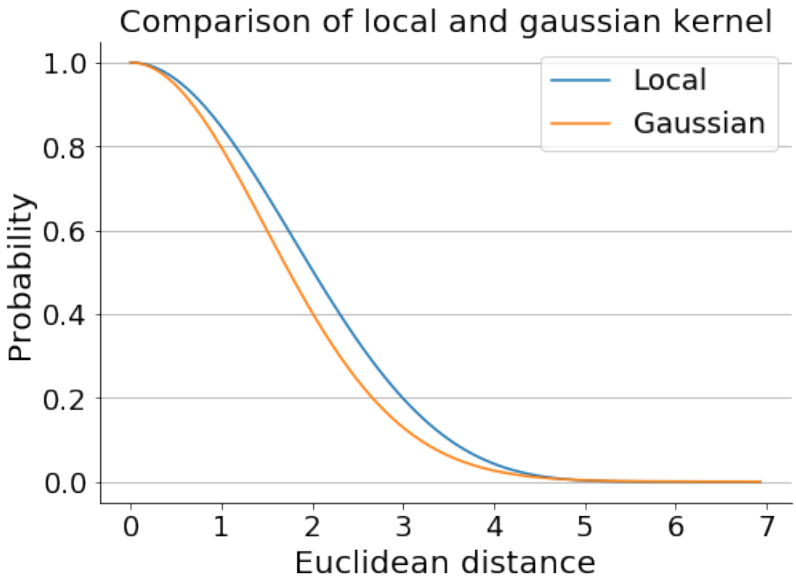


Figure B.1.: Isotropic covariance function with different values of $\beta 5$

B.2. Benefit of truncated correlation

The main advantage in the use of a truncated correlation is the speeding up of the calculation. Indeed this kind of correlation is equal to zero where other relation might assume very small values (often negligible) and a lot of computations become trivial. The final result is much faster implementation although not affecting significantly the final results.

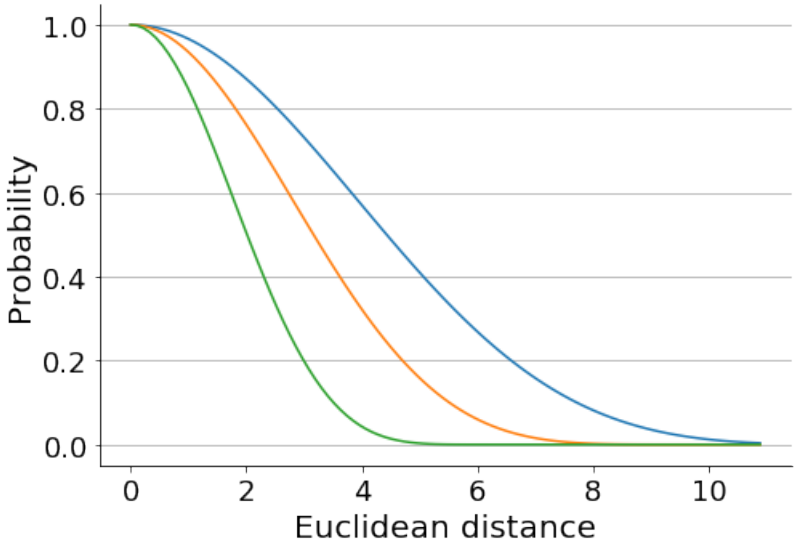


Figure B.2.: Isotropic covariance function with different values of β_corr

C. Convergence of probabilistic approach

Consider the probability $P\left[m(p)|m(p_{j1}),...,m(p_{jn})\right]$ and the function $\gamma\left(m(p_i),m(p_j)\right)$ as defined in subsection 4.2.3; then if $i \in \{j1...jn\}$:

$$\lim_{p\to p_i} P\left[m(p)|m(p_{j1}),...,m(p_{jn})\right] = \lim_{p\to p_i} S\left(\sum_{h=j1}^{jn} L\left(P\left[m(p_i)|m(p_h)\right]\right)\right) = \gamma\left(m(p_i),m(p_j)\right).$$

Proof. Since the Sigmoid function S is continuous in $\mathbb{R}$, the operator $\lim_{p\to p_i}$ can be moved in the argument of S:

$$\lim_{p\to p_i} S\left(\sum_{h=j1}^{jn} L\left(P\left[m(p_i)|m(p_h)\right]\right)\right) = \lim_{p\to p_i} S\left(L\left(P\left[m(p_i)|m(p_i)\right]\right) + \sum_{\substack{h=j1\\h\neq i}}^{jn} L\left(P\left[m(p_i)|m(p_h)\right]\right)\right) =$$

$$= S\left(\lim_{p\to p_i}\left[L\left(P\left[m(p_i)|m(p_i)\right]\right) + \sum_{\substack{h=j1\\h\neq i}}^{jn} L\left(P\left[m(p_i)|m(p_h)\right]\right)\right]\right)$$

Considering $P\left[m(p_i)|m(p_j)\right]$:

$$P\left[m(p_i)|m(p_j)\right] = \begin{cases}\gamma\left(m(p_i),m(p_j)\right) \in \{0,1\} & \text{if } i=j\\ \xi(i,j) & \text{otherwise}\end{cases}$$

where ξ represent a generic value in the interval $(0,1)$, and recalling the behavior of the Logic function L

$$L\left(P\left[m(p_i)|m(p_j)\right]\right) = \begin{cases}\pm\infty & \text{if } i=j\\ finite\ value & \text{otherwise}\end{cases}$$

it results

$$\lim_{p\to p_i}\left(\underbrace{L\left(\gamma\left(m(p_i),m(p_j)\right)\right)}_{+\infty\ or\ -\infty} + \underbrace{\sum_{\substack{h=j1\\h\neq i}}^{jn} L\left(\xi(h,j)\right)}_{finite\ sum}\right) = L\left(\gamma\left(m(p_i),m(p_j)\right)\right)$$

and then

$$\lim_{p \to p_i} S\left(\sum_{h=j1}^{jn} L\left(P\left[m(p_i)|m(p_h)\right]\right)\right) = S\left(L\left(\gamma\left(m(p_i), m(p_j)\right)\right)\right) = \gamma\left(m(p_i), m(p_j)\right)$$

■

D. Conditional probability for Bayesian network

In this section are reported the conditional probabilities used for the example reported in section section 3.3. For simplicity, these values were not extracted from experimental data but were chosen simply based on common sense.

Valid Map	
Value	States
0.7	*Valid*
0.3	*Invalid*

Table D.1.: Probability of *Valid Map*

Enforced traffic Rules	
Value	States
0.8	*Valid*
0.2	*Invalid*

Table D.2.: Probability of *Enforced traffic Rules*

Valid Sensor Measurements	
Value	States
0.95	*Valid*
0.05	*Invalid*

Table D.3.: Probability of *Valid Sensor Measurements*

Value	States	
Valid Traffic Rules	Valid Traffic Rules	Valid Map
0.9	*Valid*	*Valid*
0.1	*Valid*	*Invalid*
0.1	*Invalid*	*Valid*
0.9	*Invalid*	*Invalid*

Table D.4.: Conditional probability of *Valid Traffic Rule*

Value	States	
Valid Traffic Sign	Valid Traffic Sign	Valid Map
0.9	*Valid*	*Valid*
0.1	*Valid*	*Invalid*
0.1	*Invalid*	*Valid*
0.9	*Invalid*	*Invalid*

Table D.5.: Conditional probability of *Valid Traffic Sign*

Value	States	
Valid Geometry	Valid Geometry	Valid Map
0.9	*Valid*	*Valid*
0.1	*Valid*	*Invalid*
0.1	*Invalid*	*Valid*
0.9	*Invalid*	*Invalid*

Table D.6.: Conditional probability of *Valid Geometry*

Value	States			
Good Behavior	Good Behavior	Valid Rules	Enforced Rules	Valid Traffic Sign
0.9	*Valid*	*Valid*	*Valid*	*Valid*
0.1	*Valid*	*Valid*	*Valid*	*Invalid*
0.8	*Valid*	*Valid*	*Invalid*	*Valid*
0.2	*Valid*	*Valid*	*Invalid*	*Invalid*
0.7	*Valid*	*Invalid*	*Valid*	*Valid*
0.3	*Valid*	*Invalid*	*Valid*	*Invalid*
0.7	*Valid*	*Invalid*	*Invalid*	*Valid*
0.3	*Valid*	*Invalid*	*Invalid*	*Invalid*
0.1	*Inalid*	*Valid*	*Valid*	*Valid*
0.9	*Inalid*	*Valid*	*Valid*	*Invalid*
0.2	*Inalid*	*Valid*	*Invalid*	*Valid*
0.8	*Inalid*	*Valid*	*Invalid*	*Invalid*
0.3	*Inalid*	*Invalid*	*Valid*	*Valid*
0.7	*Inalid*	*Invalid*	*Valid*	*Invalid*
0.3	*Inalid*	*Invalid*	*Invalid*	*Valid*
0.7	*Inalid*	*Invalid*	*Invalid*	*Invalid*

Table D.7.: Conditional probability of the *Good People Behavior*

Value	States		
Observed Traffic Sign	Observed Traffic Sign	Valid Traffic Sign	Valid Sensor Measurements
Valid	*Valid*	*Valid*	0.9
Valid	*Valid*	*Invalid*	0.919
Valid	*Invalid*	*Valid*	0.843
Valid	*Invalid*	*Invalid*	0.285
Invalid	*Valid*	*Valid*	0.895
Invalid	*Valid*	*Invalid*	0.468
Invalid	*Invalid*	*Valid*	0.381
Invalid	*Invalid*	*Invalid*	0.056

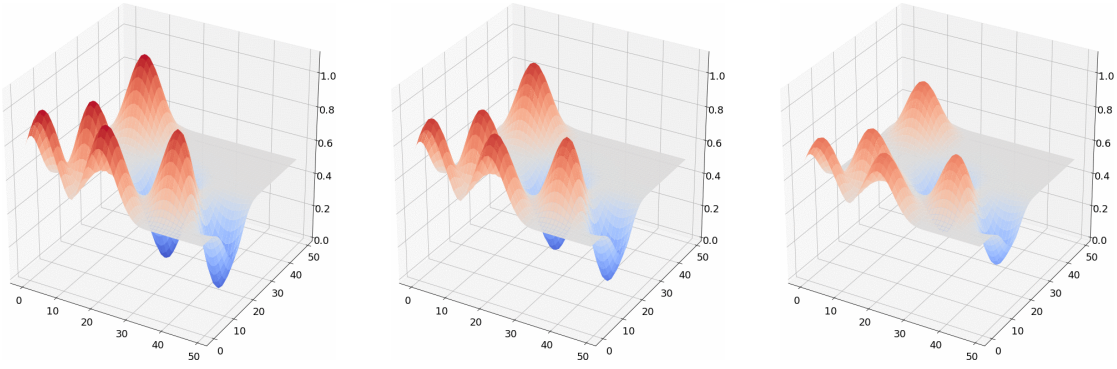
Table D.8.: Conditional probability of the map

E. Time effect in the probabilistic approach

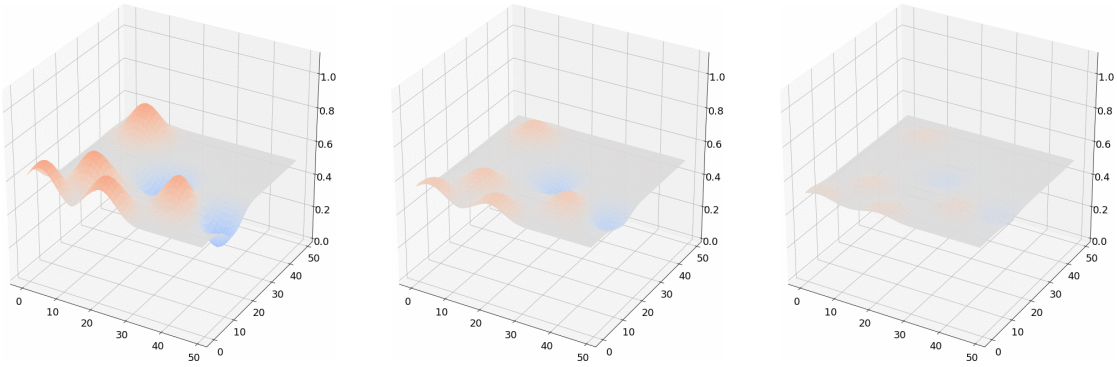
In this appendix we extend the concept of spatial correlation described in section chapter 4 by adding an idea of time correlation. Intuitively, as the measurements become old the information received from the sensors becomes unreliable. So, in this case, if no new update arrives it is reasonable that the map will return to the prior. This idea can be simply taken into account in this framework by adding one dimension to the characteristic of the map points. Each cell will be identified by three variables: x position, y position and time instance t and a new spatial/temporal distance between points can be defined. The latter can be a simple $2-norm$ or, if there is the necessity to weight the spatial and the temporal information in a different way, the following norm can be used:

$$\left\| p_i - p_j \right\|_Q = (p_i - p_j)^T \cdot Q \cdot (p_i - p_j) = \epsilon^T Q \epsilon = \epsilon^T \cdot \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix} \cdot \epsilon \tag{E.1}$$

where the values on the diagonal of Q determinate the effect of each single component of p on the distance function. Using this approach, even if we have a measurement of exactly the point we are trying to update, if the sensor information is *old*, it will be seen as *very far* from the cell to update. The further the point will result, the less its effect will affect the map updating. In Figure E.1 and Figure E.2 are reported the time effect on the example Figure 4.5 and Figure 4.7 respectively.

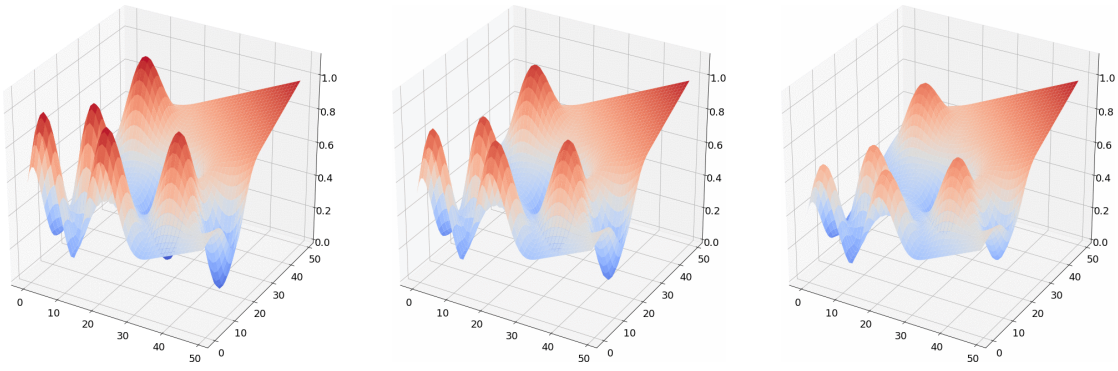


(a) $t = 0$ (b) $t = 1$ (c) $t = 2$

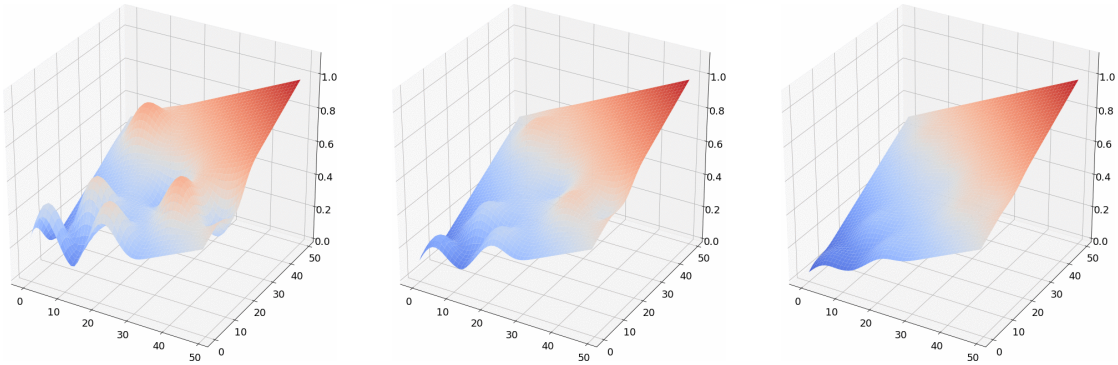


(d) $t = 3$ (e) $t = 4$ (f) $t = 5$

Figure E.1.: Time effect on the first example



(a) $t = 0$ (b) $t = 1$ (c) $t = 2$



(d) $t = 3$ (e) $t = 4$ (f) $t = 5$

Figure E.2.: Time effect on the second example

List of Figures

List of Tables

Bibliography

[1] Keshav Bimbraw. "Autonomous Cars: Past, Present and Future - A Review of the Developments in the Last Century, the Present Scenario and the Expected Future of Autonomous Vehicle Technology". In: *ICINCO 2015 - 12th International Conference on Informatics in Control, Automation and Robotics, Proceedings* 1 (Jan. 2015), pp. 191–198. DOI: 10.5220/0005540501910198.

[2] Todd Litman. "Autonomous Vehicle Implementation Predictions: Implications for Transport Planning". In: *Transportation Research Board 94th Annual Meeting Location: Washington DC, United States* (2015).

[3] Continental Corporate Media Relations. *Where are we heading? Paths to mobility of tomorrow. The 2018 Continental Mobility Study*. Dec. 2018.

[4] Corporate Partneship Board CPB. *International Transport Forum*. https://www.itf-oecd.org/sites/default/files/docs/15cpb_autonomousdriving.pdf. [Online; accessed 9-October-2019]. 2015.

[5] Publitek. *The technological evolution of autonomous vehicles – separating hype from reality.* https://www.publitek.com/wp-content/uploads/2019/01/WhitePaper-Publitek-Autonomous-Driving-RGB-V2FINAL.pdf. [Online; accessed 9-October-2019]. 2018.

[6] L. Fridman et al. "MIT Advanced Vehicle Technology Study: Large-Scale Naturalistic Driving Study of Driver Behavior and Interaction With Automation". In: *IEEE Access* 7 (2019), pp. 102021–102038. DOI: 10.1109/ACCESS.2019.2926040.

[7] Arne Kesting, Martin Treiber, and Dirk Helbing. "Enhanced Intelligent Driver Model to Access the Impact of Driving Strategies on Traffic Capacity". In: *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences* 368 (Oct. 2010), pp. 4585–605. DOI: 10.1098/rsta.2010.0084.

[8] Yougang Bian et al. "An Advanced Lane-Keeping Assistance System With Switchable Assistance Modes". In: *IEEE Transactions on Intelligent Transportation Systems* PP (Feb. 2019), pp. 1–12. DOI: 10.1109/TITS.2019.2892533.

[9] Heiko Seif and Xiaolong Hu. "Autonomous Driving in the iCity—HD Maps as a Key Challenge of the Automotive Industry". In: *Engineering* 2 (June 2016), pp. 159–162. DOI: 10.1016/J.ENG.2016.02.010.

[10] H. Durrant-Whyte and T. Bailey. "Simultaneous localization and mapping: part I". In: *IEEE Robotics Automation Magazine* 13.2 (June 2006), pp. 99–110. DOI: 10.1109/MRA.2006.1638022.

[11] J. K. Wu and Y. F. Wong. "Bayesian Approach for Data Fusion in Sensor Networks". In: *2006 9th International Conference on Information Fusion*. July 2006, pp. 1–5. DOI: 10.1109/ICIF.2006.301810.

[12] Dirk Wollherr Christian Landsiedel. "Road Geometry Estimation for Urban Semantic Maps using Open Data". In: *Advanced Robotics* 0 (Mar. 2007), p. 14.

[13] Georg Tanzmeister, Dirk Wollherr, and Martin Buss. "Grid-Based Multi-Road-Course Estimation Using Motion Planning". In: *IEEE Transactions on Vehicular Technology* 65 (Jan. 2015), pp. 1–1. DOI: 10.1109/TVT.2015.2420752.

[14] G. Tanzmeister et al. "Road course estimation in unknown, structured environments". In: *2013 IEEE Intelligent Vehicles Symposium (IV)*. June 2013, pp. 630–635. DOI: 10.1109/IVS.2013.6629537.

[15] T. Heidenreich, J. Spehr, and C. Stiller. "LaneSLAM – Simultaneous Pose and Lane Estimation Using Maps with Lane-Level Accuracy". In: *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. Sept. 2015, pp. 2512–2517. DOI: 10.1109/ITSC.2015.404.

[16] C. Hasberg and S. Hensel. "Online-estimation of road map elements using spline curves". In: *2008 11th International Conference on Information Fusion*. June 2008, pp. 1–7.

[17] Carlos Fernández et al. "High-Level Interpretation of Urban Road Maps Fusing Deep Learning-Based Pixelwise Scene Segmentation and Digital Navigation Maps". In: *Journal of Advanced Transportation* 2018 (2018), pp. 1–15. DOI: 10.1155/2018/2096970. URL: https://app.dimensions.ai/details/publication/pub.1107560849%20and%20http://downloads.hindawi.com/journals/jat/2018/2096970.pdf.

[18] A. Meyer et al. "Deep Semantic Lane Segmentation for Mapless Driving". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Oct. 2018, pp. 869–875. DOI: 10.1109/IROS.2018.8594450.

[19] Sarang Thombre et al. "GNSS Threat Monitoring and Reporting: Past, Present, and a Proposed Future". In: *Journal of Navigation* 71.3 (2018), pp. 513–529. DOI: 10.1017/S0373463317000911.

[20] Sameer Kumar and Kevin B. Moore. "The Evolution of Global Positioning System (GPS) Technology". In: *Journal of Science Education and Technology* 11.1 (Mar. 2002), pp. 59–80. ISSN: 1573-1839. DOI: 10.1023/A:1013999415003. URL: https://doi.org/10.1023/A:1013999415003.

[21] Xiongwei Zheng et al. "Geometric Accuracy Evaluation of High-Resolution Satellite Images Based on Xianning Test Field". In: *Sensors* 18 (July 2018), p. 2121. DOI: 10.3390/s18072121.

[22] Wei-Chiu Ma et al. *Exploiting Sparse Semantic HD Maps for Self-Driving Vehicle Localization*. Aug. 2019.

[23] Wolfgang Granig, Dirk Hammerschmidt, and Hubert Zangl. "Calculation of Failure Detection Probability on Safety Mechanisms of Correlated Sensor Signals According to ISO 26262". In: *SAE International Journal of Passenger Cars - Electronic and Electrical Systems* 10 (Mar. 2017). DOI: 10.4271/2017-01-0015.

[24] A. Elfes. "Using occupancy grids for mobile robot perception and navigation". In: *Computer* 22.6 (June 1989), pp. 46–57. DOI: 10.1109/2.30720.

[25] Zhao Liu, Daxue Liu, and Tongtong Chen. "Vehicle detection and tracking with 2D laser range finders". In: vol. 2. Dec. 2013, pp. 1006–1013. ISBN: 978-1-4799-2764-7. DOI: 10.1109/CISP.2013.6745203.

[26] H. Chen et al. "Vision-Based Road Bump Detection Using a Front-Mounted Car Camcorder". In: *2014 22nd International Conference on Pattern Recognition*. Aug. 2014, pp. 4537–4542. DOI: 10.1109/ICPR.2014.776.

[27] Mhafuzul Islam et al. *Vision-based Navigation of Autonomous Vehicle in Roadway Environments with Unexpected Hazards (Full paper: https://journals.sagepub.com/doi/10.1177/0361198119855*
Nov. 2018. DOI: 10.13140/RG.2.2.25293.28645.

[28] A. Discant et al. "Sensors for Obstacle Detection - A Survey". In: *2007 30th International Spring Seminar on Electronics Technology (ISSE)*. May 2007, pp. 100–105. DOI: 10.1109/ISSE.2007.4432828.

[29] K. Kaliyaperumal, S. Lakshmanan, and K. Kluge. "An algorithm for detecting roads and obstacles in radar images". In: *IEEE Transactions on Vehicular Technology* 50.1 (Jan. 2001), pp. 170–182. DOI: 10.1109/25.917913.

[30] Z. A. Rahman et al. "Speed trap detection with Doppler effect". In: *Proceedings. Student Conference on Research and Development, 2003. SCORED 2003.* Aug. 2003, pp. 202–206. DOI: 10.1109/SCORED.2003.1459693.

[31] U. Nickel. "Applications of superresolution for radar: Examples, problems and solutions". In: *21st European Signal Processing Conference (EUSIPCO 2013)*. Sept. 2013, pp. 1–5.

[32] Hugh F. Durrant-Whyte. "Sensor Models and Multisensor Integration". In: *The International Journal of Robotics Research* 7.6 (1988), pp. 97–113. DOI: 10.1177/027836498800700608. eprint: https://doi.org/10.1177/027836498800700608. URL: https://doi.org/10.1177/027836498800700608.

[33] Raymond McKendall and Max Mintz. "Robust fusion of location information". In: *Proceedings. 1988 IEEE International Conference on Robotics and Automation* (1988), 1239–1244 vol.2.

[34] A. Visser and F. C. A. Groen. *Organisation and Design of Autonomous Systems*. Textbook, Faculty of Mathematics, ComputerScience, Physics and Astronomy, University of Amsterdam, Kruis-laan 403, NL-1098 SJ Amsterdam, August 1999.

[35] Rudolph Emil Kalman. "A New Approach to Linear Filtering and Prediction Problems". In: *Transactions of the ASME–Journal of Basic Engineering* 82.Series D (1960), pp. 35–45.

[36] F. Gustafsson et al. "Particle Filters for Positioning, Navigation, and Tracking". In: *Trans. Sig. Proc.* 50.2 (Feb. 2002), pp. 425–437. ISSN: 1053-587X. DOI: 10.1109/78.978396. URL: https://doi.org/10.1109/78.978396.

[37] Aakriti Singhal and Christopher Robert Brown. "Dynamic Bayes net approach to multimodal sensor fusion". In: *Other Conferences*. 1997.

[38] C. Coue et al. "Multi-sensor data fusion using Bayesian programming: An automotive application". In: *Intelligent Vehicle Symposium, 2002. IEEE*. Vol. 2. June 2002, 442–447 vol.2. DOI: 10.1109/IVS.2002.1187989.

[39] Anders L. Madsen et al. "Applications of Probabilistic Graphical Models to Diagnosis and Control of Autonomous Vehicles". In: *UAI 2004*. 2004.

[40] Ke Sun et al. "Dense 3-D Mapping with Spatial Correlation via Gaussian Filtering". In: *CoRR* abs/1801.07380 (2018). arXiv: 1801.07380. URL: http://arxiv.org/abs/1801.07380.

[41] Jose Luis Blanco et al. "A Kalman Filter Based Approach to Probabilistic Gas Distribution Mapping". In: *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. SAC '13. Coimbra, Portugal: ACM, 2013, pp. 217–222. ISBN: 978-1-4503-1656-9. DOI: 10.1145/2480362.2480409. URL: http://doi.acm.org/10.1145/2480362.2480409.

[42] S. Bin and Y. Wenlai. "Application of Gaussian Process Regression to prediction of thermal comfort index". In: *2013 IEEE 11th International Conference on Electronic Measurement Instruments*. Vol. 2. Aug. 2013, pp. 958–961. DOI: 10.1109/ICEMI.2013.6743191.

[43] Philippe Smets. "Decision making in the TBM: the necessity of the pignistic transformation". In: *International Journal of Approximate Reasoning* 38.2 (2005), pp. 133–147. ISSN: 0888-613X. DOI: https://doi.org/10.1016/j.ijar.2004.05.003. URL: http://www.sciencedirect.com/science/article/pii/S0888613X04000593.

[44] E. Casapietra et al. "Building a probabilistic grid-based road representation from direct and indirect visual cues". In: *2015 IEEE Intelligent Vehicles Symposium (IV)*. June 2015, pp. 273–279. DOI: 10.1109/IVS.2015.7225698.

[45] A. Lawitzky et al. "Interactive scene prediction for automotive applications". In: *2013 IEEE Intelligent Vehicles Symposium (IV)*. June 2013, pp. 1028–1033. DOI: 10.1109/IVS.2013.6629601.

[46] Jan Vascak. "Navigation of Mobile Robots Using Potential Fields and Computational Intelligence Means". In: *Acta Polytechnica Hungarica* 4 (Mar. 2007).

[47] Finn V. Jensen and Thomas D. Nielsen. *Bayesian Networks and Decision Graphs*. 2nd. Springer Publishing Company, Incorporated, 2007. ISBN: 9780387682815.

[48] Amos James Storkey. *Efficient Covariance Matrix Methods for Bayesian Gaussian Processes and Hopfield Neural Networks*. 1999.