Freie Universität Berlin

Scenario interpretation for automated driving at urban intersections

Dissertation zur Erlangung des Grades eines Doktors der Naturwissenschaften (Dr. rer. nat.) am Fachbereich Mathematik und Informatik der Freien Universität Berlin

von

David Perdomo López

Berlin
2018

Erster Gutachter:	Prof. Dr. Raúl Rojas
Zweiter Gutachter:	Prof. Dr. Marco Block-Berlitz
Datum der Disputation:	29.06.2018

Foreword

This thesis was developed during my three years activity at the automated driving department of the Volkswagen Group Research in Wolfsburg, Germany.

The results, opinions and conclusions expressed in this thesis are not necessarily those of Volkswagen AG.

Kurzfassung

Seit den 1980er Jahren ist das Forschungsinteresse am Feld des automatisierten Fahrens schnell gewachsen und wächst auch heute noch exponentiell weiter. Damit einhergehend sind zahlreiche neue Ansätze, Techniken und Lösungen entstanden. Trotzdem ist weitere Forschungsarbeit erforderlich, um die Vision des automatisierten Fahrens in allen denkbaren Szenarien verwirklichen zu können. Insbesondere urbane Szenarien stellen in Bezug auf automatisiertes Fahren nach wie vor eine komplexe wissenschaftliche Herausforderung dar.

Die vorliegende Arbeit beschäftigt sich mit der Entwicklung eines Szenariointerpretationskonzeptes für automatisiertes Fahren an urbanen Kreuzungen. Der Fokus liegt dabei auf der Interpretation der zu Verfügung stehenden Daten des Unfallwahrnehmungsmoduls, mit dem Ziel, das Ego-Fahrzeug ein gewünschtes Fahrmanöver ausführen zu lassen.

Der erste Beitrag dieser Arbeit behandelt die Problemadressierung und analyse und nimmt eine Klassifikation aller möglichen Szenarien in Bezug auf potentielle Konflikte mit anderen Verkehrsteilnehmern vor. Diese Klassifikation veranschaulicht den Bedarf nach einer Lösung zur Interpretation solcher Szenarien und Umsetzung entsprechender Fahrmanöver. Der entwickelte Lösungsansatz basiert auf der Berechnung einer diskreten Wahrscheinlichkeitsverteilung, welche die verschiedenen Zustände der Passierbarkeit von Kreuzungen unter Berücksichtigung aller möglichen Ampelschaltungen und Verkehrszeichen für Ego-Fahrzeuge abbildet. Das vorliegende Konzept beachtet dabei sowohl Input-Unsicherheiten, als auch Schwankungen über die Zeit.

Die Beiträge dieser Arbeit stützen sich auf das Kernkonzept der Elementaren Situationen. Die Idee dahinter ist das Zerlegen eines gewünschten Fahrmanövers in einzelne Sequenzen. Diese Vereinfachung ermöglicht eine Einschätzung der potentiellen Bewegungen anderer Verkehrsteilnehmer aus denen anschlieend die Belegwahrscheinlichkeiten für die einzelnen elementaren Situationen berechnet werden können.

Ein weiterer relevanter Beitrag beschreibt die Entwicklung einer taktischen Manöverplanung, welche nicht sichtbare Bereiche, beziehungsweise fehlende Informationen innerhalb von Kreuzungen, zum Gegenstand hat.

Die prototypische Umsetzung der Ansätze ermöglicht die Erfassung der Ergebnisse als *proof of concept* mit realen Daten. Weiterhin kann anhand der erzielten Ergebnisse angenommen werden, dass die entwickelte Lösung für die Szenariointerpretation an urbanen Kreuzungen geeignet ist.

Abstract

The research interest in the field of automated driving has been growing rapidly from the first milestones in the 1980s decade to our days. Actually, this interest keeps growing exponentially. The increasing research effort implies a large number of novel approaches, techniques and solutions. Nevertheless, further research is needed to achieve automated driving systems in all different kind of scenarios. Particularly, automated driving at urban scenarios still represents a very complex challenge for the research community.

This thesis deals with the development of a scenario interpretation module for automated driving at urban intersection. The main objective is to interpret the information from the perception module and guide the ego vehicle along the desired path to complete the maneuver.

The first contribution of this work is focused on the analysis of the problem, which yields to a classification of all scenarios considering the potential conflicts with other road users. Based on this statement, the need of a proper pass permission interpretation becomes crucial. The proposed solution in this thesis consist on calculating the probability that every pass permission state is valid for the ego vehicle. This is done by representing all possible traffic lights and traffic signs in two different probability mass functions, which are consequently combined. The developed approach deals not only with the uncertainty of the received inputs, but also with the fluctuations over time.

The concept of primary situations is presented as the core approach for further contributions. The key idea is to interpret the scenario by breaking down the whole desired maneuver into a sequence of primary situations. Using this concept, the object prediction module is significantly simplified. The movement of other involved road users is estimated, and this estimation is used to calculate the probability that every primary situation is occupied. In this thesis, the concept is presented for vehicles and pedestrians.

Another relevant contribution of this thesis is the use of the generated primary situations to perform the subsequent tactical decision making based on a state machine algorithm. The proposed solution enables to handle occlusions in a simpler manner imitating the human reaction.

The implementation of the developed approaches in a demonstrator vehicle and the consequent evaluation confirms that the presented solution is suitable for interpreting the scenario at urban intersections.

Contents

1. Introduction

1.1. Motivation

Autonomous driving has become a very active research field with an increasing interest in the automobile industry and research community in the recent years. Consequently, the development of different approaches, technologies and concepts are growing rapidly, bringing established auto-makers, new start-ups and investors together in research projects.

Autonomous driving has its origin during the decade of the 1980's, pioneered by Dickmanns et. al.[1]. They developed an automated vehicle based on lane detection using a single camera. Interest in the field of autonomous driving was then followed by several projects and competitions that caused an expansion of techniques and approaches. For example, the Eureka Program for a European Traffic of Highest Efficiency and Unprecedented Safety (PROMETHEUS) Project aimed to develop new concepts and solutions for a more efficient and fluid traffic flow [2] [3]. Subsequent research projects such as Mobilität und Transport im intermodalen Verkehr (MoTiV) [4] or Intelligenter Verkehr und nutzergerechte Technik (INVENT) [5] continued attracting interest and pushing the development in this research area.

Other milestones for autonomous driving are the Defense Advanced Research Projects Agency (DARPA) Grand Challenges, which had the purpose to leverage American ingenuity to accelerate the development of autonomous vehicle technologies that can be applied to military requirements. The concrete goal of the first challenge in 2004 was to drive autonomously an entire 241.4 kilometers route through the Mojave desert to Primm (Nevada) without participant interventions [6]. Although none of the participating teams achieved the goal, the event prompt important technological contributions. Other DARPA challenges followed in 2005 [7] and 2007 [8]. The DARPA Urban Challenge in 2007 is considered one of the most important events, bringing several key technical developments and approaches and further research efforts. This challenge was not only focused on the environment perception and trajectory planning, but also on the reaction of the driver-less vehicle within its surroundings, in which other road users were involved to simulate simple urban scenarios.

In the last 10 years several projects have continued to develop concepts and solutions to enable autonomous or driving assistance systems. For example, SAFESPOT was an integrated research project co-funded by the European Commission Information Society Technologies [9] that aimed to develop key enabling technologies such as dynamic cooperative networks, where the vehicles and the road infrastructure communicate to share information gathered on board and at the roadside to enhance the drivers' perception of the vehicle surroundings. Also using communication between road users, the Grand Cooperative Driving Challenge (GCDC) was an open competition between research groups on the topic of cooperative and autonomous driving [10] [11]. In this challenge, held in the Netherlands, nine international teams aimed to develop a system that performs longitudinal control of the vehicle in a platooning setup, commonly known as Cooperative Adaptive Cruise Control (CACC).

This kind of engagement between the international research community and the automobile industry continued growing resulting in an increasing number of projects and challenges such as the Hyundai Autonomous Challenge [12], Cooperative Autonomous Car Train (CoAct) [13], VisLab International Autonomous Challenge [14], Functional Safety and Envolvable architectures for autonomy (FUSE) [15] or GCDC 2016 [16].

Although these projects did not have identical scopes (i.e. different environments, requirements, level of automations...), they produced important progress in all the areas of automated or assisted driving: sensor technologies, standardization of communication protocols, environmental modeling, control, trajectory planning, etc. The complexity of the challenge depends on the level of automation (see the classification of different levels based on [17]), the majority of approaches and techniques are virtually identical across levels, so that both technical advances and shortages at one level are also present on the remaining levels. On the other hand, the environment on which the system is meant to operate has a big impact on the complexity of the problem and the lack of research solutions. In particular, self driving systems at urban intersections can be still considered the bottle neck problem of this area.

Nevertheless, driver-less cars are still not a reality for all different scenarios. Further research efforts are still needed to enable safety self-driving vehicles in daily life, especially on environments for which there is a lack of research solutions. In particular, self driving systems at urban intersections can be considered the bottle neck problem of this area:

1. They present a large number of possible conflicts with other road users.

2. They present a complex infrastructure, where the traffic flow has to be controlled by traffic signs/lights.

3. The interpretation of the scenario requires sophisticated solutions to reduce the

complexity of the problem.

As a result, even though the priority is regulated with specific rules, the scenario interpretation of urban intersections is a highly complex challenge, not only for self driving systems but also for humans.

1.2. Hypothesis

Throughout this thesis, I will provide evidence supporting the following hypothesis:

> Is it feasible to design and build a system capable of interpreting the surroundings of the ego vehicle to enable automated driving at urban intersections?

The statement *interpreting the surroundings of the ego vehicle* refers to the ability to understand the required information that enables automated driving.

The statement *driving at urban intersections* should be interpreted in the full sense, i.e. driving forward, turning left, turning right and making a U-turn.

The architecture of the system should be such that facilitates proper and systematic planning of each of the maneuvers according to the German traffic rules.

1.3. Objectives

The present thesis, in pursuing the extraction of evidences supporting the hypothesis above, has the following objectives:

1. To describe the problem of automated driving at urban intersections.

2. To design and develop an algorithm for the scenario interpretation at urban intersections.

3. To design and develop an algorithm to interpret the pass permission at intersections.

4. To identify the object motion prediction method compatible with the proposed approach.

5. To design and develop an algorithm to handle the problem of occlusions.

6. To design and develop an algorithm for planning and executing all the different maneuvers.

7. To achieve *proof of concept* state of the developed contributions, implementing them in a demonstrator vehicle and allowing their coherent testing and evaluation in different scenarios.

As a starting point, a complete system designed for autonomous driving at highway scenarios is provided. The contributions of this thesis are joint with the development of the given baseline system and its concepts.

In addition, the following issues, covered by such baseline system, are out of the scope of this work: traffic lights/signs detection, road lanes estimation, object detection and recognition, localization and pose estimation, sensors data fusion, trajectory planning and control tasks.

1.4. Structure of this thesis

This thesis is structured in 6 chapters.

Related work: (Chapter 2) I introduce the relevant work in the field of autonomous driving at intersections. The definition of the term *scenario interpretation* is described and some important methods are reviewed. Considering the current state of the art, some relevant techniques in the field of intention and motion prediction are introduced. Then, a review on decision making approaches used in the literature is given. Motivated from the problem stated problem, this chapter ends with a quick discussion that bases the proposed solution.

Overview of the baseline system: (Chapter 3) I provide an overview of the baseline system, which corresponds to the starting point of this work. After giving a simplification of the conceptual architecture, the most relevant modules are briefly explained: perception, driving function, planning and control.

Contributions of this thesis: (Chapter 4) Once the baseline is described, I explain in detail all the contributions of this thesis. First, I analyze the problem in such a way that the most relevant challenges are identified and the complexities of the main issues are addressed. Then, I tackle the interpretation of the pass permission at urban intersection, describing in detail the proposed solution to understand how the ego vehicle should pass the intersection. I follow by introducing our approach based on primary situations, which corresponds to the key idea for the further contributions. Hereafter, the proposed motion prediction of objects is explained, which consists on estimating the occupancy probability of the primary situations due to the possible conflict of the ego vehicles path with other road users. The tactical decision making is then depicted. Finally, I describe the proposed solution to handle occlusions.

Evaluation: (Chapter 5) To give the reader an accurate view of how the contributions have been tested, I describe the evaluation methodology in an objective manner. Hereafter, a quick interpretation of the results states the most relevant outcomes of the achieved evaluation.

Conclusion: (Chapter 6) I summarize and discuss the findings of this thesis and provide an outlook to further research questions. Moreover, potential improvements of the developed approaches are presented, based on the identified limitations and drawbacks.

2. Related work

This chapter introduces some related work in autonomous driving systems at intersections. First, different definitions for the term *scenario interpretation* in the current literature are introduced. In order to place the reader in front of the most relevant issues involved in this thesis, the problem is described from a general point of view. Then, some approaches related to the scenario interpretation at intersections are given. In Section 2.4, relevant techniques used for intention and motion prediction are reviewed. Then, Section 2.5 gives a quick overview of the decision making techniques used in the literature. Finally, a quick discussion is provided to stand out the outcomes that yield to the proposed contributions.

2.1. Definition of scenario interpretation

The concept of *scenario interpretation* is quite widespread in the autonomous driving research community. Nevertheless, it is possible to find different meanings in the current literature. This section starts reviewing the term *scenario.* After this general definition is given, some differences in the most common system architectures are also introduced in order to consider the dependencies between the system architecture and the usage of the term *scenario interpretation.*

The authors in [18] propose a definition for relevant terms in the autonomous driving content (situation, scene, scenario...). The scenery is defined as the combination of all possible single static elements (e.g. road network, number of lanes, crosswalks, position of traffic lights, speed limits...). As it can be seen in Fig. 2.1, the scene contains the scenery and the information of all dynamic objects with their corresponding states. The situation consists of the scene and optional ego vehicle. It describes the current state, and therefore, it could persist several seconds conditioned by a defined criteria. On the other hand, the scenario describes different states over time, so that the it contains at least one situation.

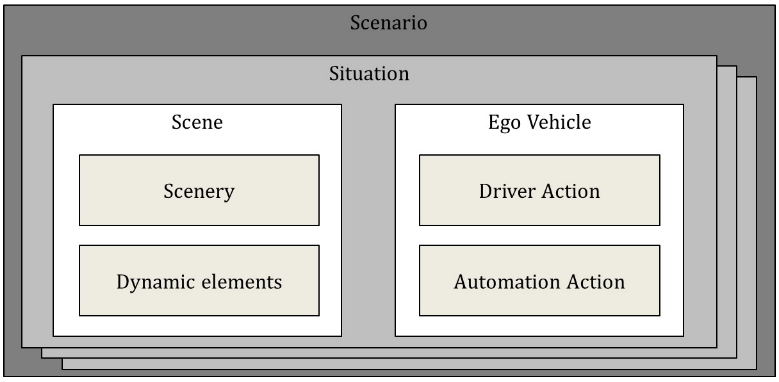


Figure 2.1.: *Contextualization of the terms scenario, situation, scene, scenery, dynamic elements, ego vehicle, driver action and automation action according to [18].*

Nevertheless, other definitions are also possible. For example, the authors in [19] present a coherent review and comparison of these terms and propose their own definitions. But even considering these meanings, definition discrepancies about the term *interpretation* are usually derived from the different structures of the system. To contextualize the different meanings and simplify the system architecture, the Fig. 2.2 shows a simplified conceptual flowchart of the whole autonomous driving process in 4 steps: (1) environment perception, (2) interpretation, (3) planning and (4) control.

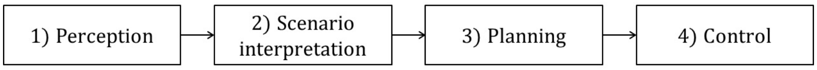


Figure 2.2.: *Basic conceptual flowchart for autonomous driving based on 4 main steps: 1) perception, 2) scenario interpretation, 3) planning and 4) control.*

According to the simplification presented in Fig. 2.2, the environment perception (1) represents the low level processing of sensors and a priori data (e.g. image processing, object recognition and tracking, localization and mapping, etc.). The scenario interpretation (2) corresponds with the understanding of the processed environment data. Finally, the planning (3) makes the proper decisions and delivers them to the control module (4), which represents the proper transformation into acceleration and steering terms.

In this basic representation, the perception module provides the description of the surrounding world to the interpretation module, which achieves the comprehension of the relevant information for the following planning and control. However, other architecture concepts are possible. In [20] the authors categorize the autonomous driving systems in 2 approaches: behavior reflex and mediated perception. The behavior reflex approach achieves a mapping directly from the sensory input to the driving action mediated perception approach, which is the most common one. It is based on several subcomponents from the sensor data acquisition to the steering and acceleration control. That is to say, the whole system is divided into several subcomponents that deal with different task separately. In fact, the authors in [20] introduce a direct perception paradigm, which falls between the both commented approaches, so that the task of interpreting the scenario is involved in this process.

Considering an architecture based on components with different tasks, Behere et. al. [21] propose a functional architecture for autonomous driving divided in 3 components (see Fig. 2.3): perception, decision and control, and vehicle platform manipulation. Here, the perception includes 5 sub-tasks: sensing, sensor fusion, localization, semantic understanding and word model. In particular, the semantic understanding can involve annotation of objects with prediction, detection of ground planes, road geometries and representation of drivable areas. In specific cases, it may also use the ego vehicle data to continuously parameterize a model of the ego vehicle for purposes of motion control, error detection and potential degradation of functionality.

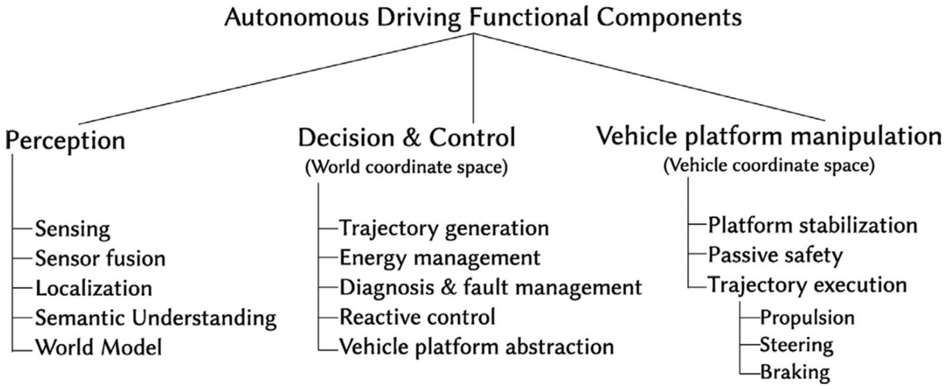


Figure 2.3.: *Structure of the functional components for autonomous driving. These are divided into: perception, decision and control, and vehicle platform manipulation (see [21]).*

On the contrary, the American National Standard Institute (ANSI) presents a general reference architecture for intelligent systems (see [22]), where the world model component corresponds to the core of the system. This simplification is illustrated in Fig. 2.4.

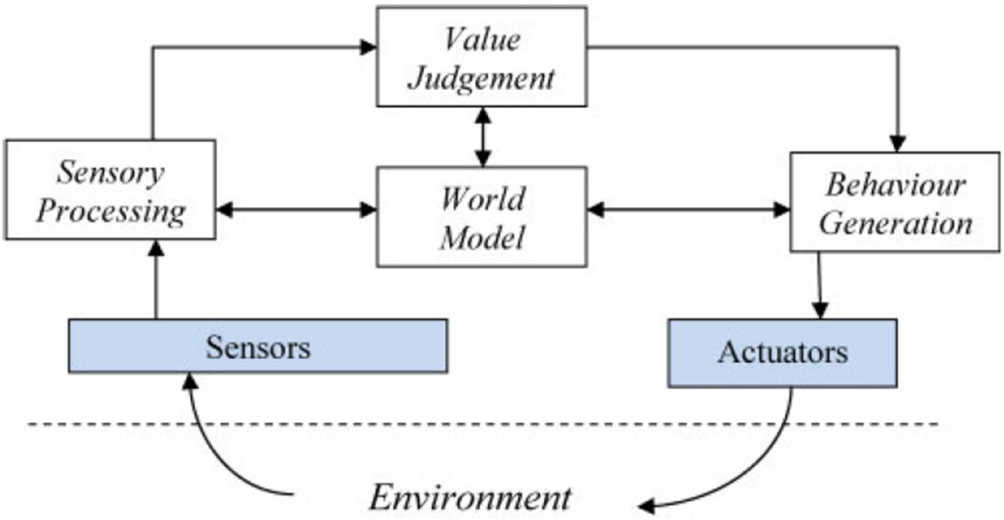


Figure 2.4.: *Reference architecture of ANSI of intelligent systems in the most generic form (see [23]). Here, the world model represents the core module between sensory processing, value judgment and behaviour generation.*

In this generic form the *scenario understanding* tasks are integrated in the world model, which represents the core module of the architecture.

2.2. Problem description

The main problem is focused on understanding the perceived information around the ego vehicle. This interpretation should enable to plan the proper vehicle motion at urban intersections. In other words, the surrounding of the ego vehicle has to be described, and accordingly, the scenario interpretation gives a proper meaning to this description. In fact, the scenario interpretation at intersection involves, among others, the following tasks:

- Filtering relevant information.
- Using the information of the road network with corresponding logical correspondences.
- Predicting the intention of other vehicles.
- Handling occlusions.
- Achieving risk assessment.

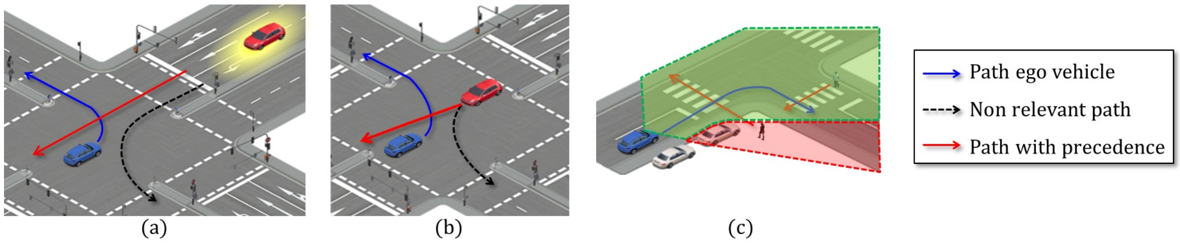
- Considering logical traffic rules.
- Handling the right of way.
- Handling localization uncertainty.

In order to address the problem clearly, the following examples help to understand the goal of the interpretation module (see Fig. 2.5):

(a) Intention prediction of an oncoming vehicle with inaccurate position: the ego vehicle (colored in blue) is turning to the left (blue path) while an oncoming vehicle (colored in red) is approaching the intersection. The uncertainty of its measured position is represented by a yellow blob surrounding the vehicle. If the oncoming vehicle is driving forward (red path), both paths intersect. Otherwise (black dotted path), there is no collision between both driving corridors.

(b) Intention prediction of an oncoming vehicle with accurate position: the ego vehicle is turning left and the other car (which is already in the intersection and its position is accurate enough) could drive forward or turn left.

(c) Handling occlusion while approaching an intersection: the ego vehicle aims to turn to the right. Due to an obstacle (e.g. another vehicle), the occlusion hinders to detect a crossing pedestrian at the right side of a zebra crossing. The green and red colored regions indicate the *perceptible* and *non-perceptible areas*, respectively



(a) (b) (c)

Figure 2.5.: *3 examples of scenarios at intersections based on [24]: (a) Interaction with an oncoming vehicle, which its position is inaccurate. (b) Interaction with an oncoming vehicle, which its position is accurate. (c) Handling occlusion while approaching an intersection.*

In the first illustration of Fig. 2.5 (a) the ego vehicle (colored in blue) is turning to the left and another vehicle (colored in red) is approaching the intersection. It becomes obvious that it is crucial to know the **position** of the other vehicle, and consequently on which lane the other car is driving, to determine a possible collision: if the red car is driving on its most left lane, it is only allowed to turn to the left, so that a collision with the ego vehicle is not expected. Alternatively, if the other car is not driving on its most left lane, its path has a conflict with the ego driving corridor. Thus, if the position of other vehicles (or ego vehicle) is not accurate enough (e.g. due to localization uncertainty), the scenario interpretation module has to manage the **uncertainty of the information** in order to understand how critical the situation is.

A proper **intention prediction** is crucial (depending on the road network and its turning possibilities) even considering a perfect accuracy of the position of both vehicles. As shown in Fig. 2.5 (b), the ego vehicle is turning left and the other car could perform 2 maneuvers: driving forward or turning left. In this case, the accuracy of the state of the other car (e.g. yaw, velocity, etc.) is relevant to achieve a proper intention prediction.

Finally, **handling occlusions** is an important task of the scenario interpretation module. It is not only crucial to understand the provided information, but also to take into account which information is missing. For example, in Fig. 2.5 (c) the ego vehicle (blue) is approaching the intersection and an obstacle (a parked car colored in white) hinders to detect a possible pedestrian. For this given scenario, the first pedestrian (behind the obstacle) is not detected due to the occlusion, but a proper scenario interpretation should be able to interpret the occlusion as a critical missing information. Consequently, it is not clear if more pedestrians are approaching the crosswalk.

Another way to describe an intersection is to consider its **topology** in detail (i.e. number of roads, lanes, etc). Other authors [25] have analyzed in detail the most common topologies to determine the relations between different topologies and traffic accidents. Nevertheless, in this thesis it is assumed that the large number of possible different topologies yields to the following statement: a scenario interpretation based on specific topologies is not appropriate. Therefore, the proposed solution aims to achieve the maneuver independently on the intersection topology.

2.3. Scene interpretation at intersections

This section reviews relevant publications that handle the issue *scene interpretation.*

Due to the complexity of the scenario at urban intersections, its interpretation requires a high level representation that enables further maneuver planning. For this purpose, some authors ([26], [27]) use Description Logic (DL), which enables representing and reasoning complex knowledge in a simple manner.

Hülsen et. al. [26] use DL to describe an ontology that represents the road networks, objects, their relations and traffic rules (see Fig. 2.6).

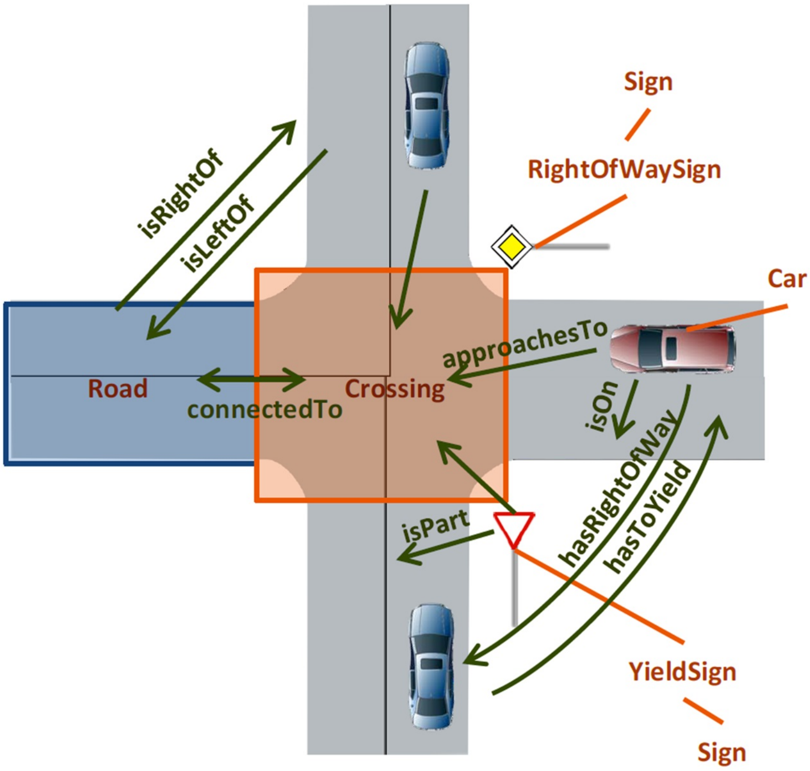


Figure 2.6.: *Simple intersection ontology. The relation between elements are represented by arrows. See [26].*

Roads are joint to the concept *crossing* with the relation *connectedTo* and to other roads with *isRightOf* / *isLeftOf*. Signs are assigned to roads by *isPart* and the cars by *isOn*. Consequently the corresponding reasoning of the resulting situation is achieved. The goal is to reason the relations *hasRightOfWay* and *hasToYield*. The authors used a simple and a complex intersection as example. Since the reasoning process takes between 1 and 3 seconds, a real usage of this approach is not suitable.

On the other hand, the authors in [27] introduce a case- and rule-based approach for situation interpretation using Web Ontology Language Description Logic (WOL-DL) to achieve the high level representation. First, the data is collected from a central database from different sensors and processing units. Then, the data is mapped using description logic to apply rules of the background knowledge. And finally, the current traffic situation is interpreted by applying the case-based reasoning paradigm. In fact, the current situation is described by a case, which consist of the road network of the local scene, all objects, the mission goal of own and other vehicles and its relations. The interpretation is done using the case-based reasoning, where a case corresponds to a situation interpretation. The construction of the case-base is basically the indexing of the cases to ease the search between cases. As it can be seen in Fig. 2.7, the cases are classified hierarchically.

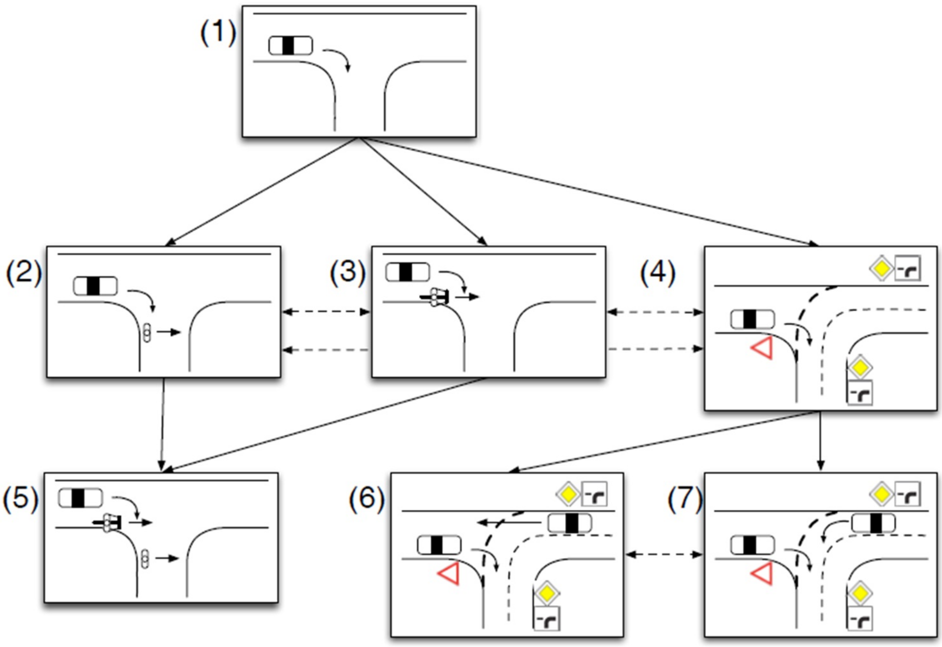


Figure 2.7.: *Example of simple intersection ontology. The connections between cases (arrows) represent the hierarchical structure (see [27]).*

Although the authors proved that the concept provides coherent results, DL ontology is generally not compatible with real time computations due to the high computational cost for complex reasoning process.

Zhao et. al. [28] present a machine understandable ontology-based knowledge base, which contains maps and traffic regulations. A time-stamp based Resource Description Framework (RDF) is used to represent the data from sensors. To express the right-of-way rule, the Semantic Web Rule Language (SWRL) is used. As it can be seen in Fig. 2.8, the scenario interpretation involves the representation of the sensor data and the decision making module.

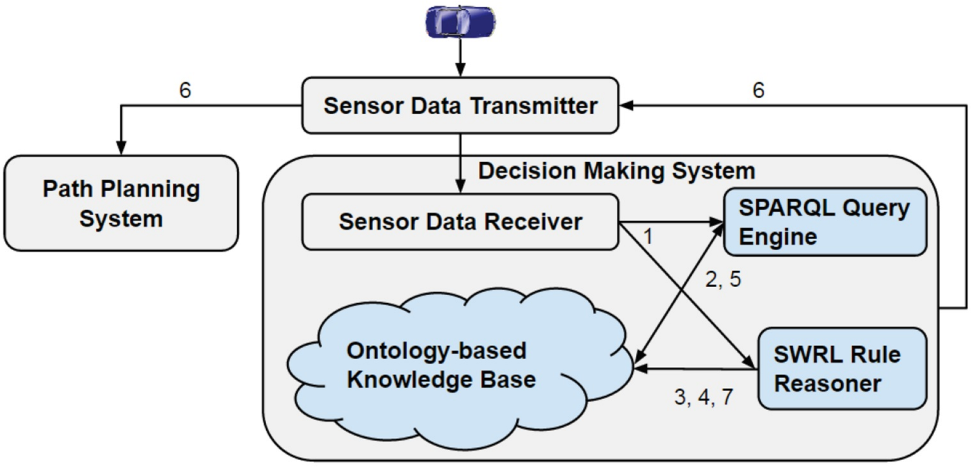


Figure 2.8.: *Structure proposed by Zhao et. al. (see [28]). The decision making system represents the main module that involves the scenario interpretation. .*

2.4. Intention and motion prediction

Undoubtedly, the field of intention or motion prediction plays an important role for a comprehensive scenario interpretation at urban intersections. Therefore, this section aims to review some relevant approaches that handle this issue.

A comprehensive survey can be found in [29], where the motion prediction approaches are classified in 3 models: physics-based, maneuver-based or interaction-aware models. A simplified classification is illustrated in Fig. 2.9

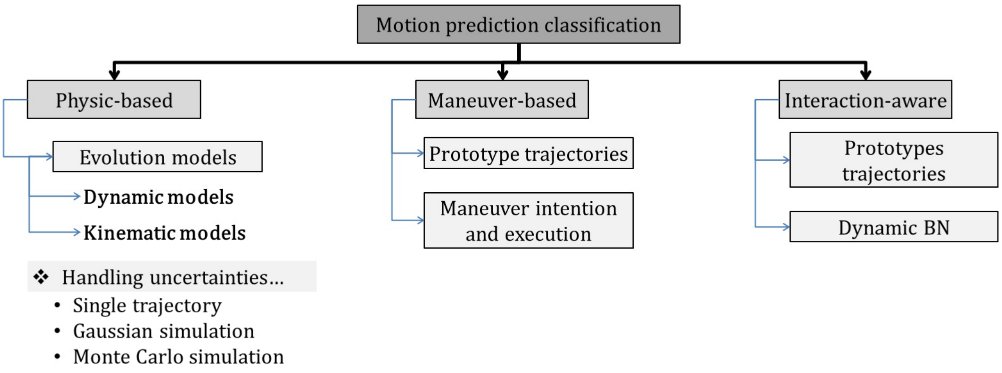


Figure 2.9.: *Simplified classification of motion prediction based on [29]). The approaches are classified in physic-based, maneuver-based and interaction-aware.*

The physic based approaches consider the laws of the physics. Although they are the most simple models, they are limited to short-term prediction (i.e. the prediction is normally calculated for less than 1 second). The dynamic models are described based on Lagrange equations (steering, acceleration, car weights, road friction, etc.). The kinematic models are based on position, velocity and acceleration, so that no forces are involved. Another classification is possible depending on how the uncertainty is handled. Using single trajectories, which is computationally simple. Using Gaussian simulation correspond to another used solution, which could be appropriate only if the motion evolution and sensor data are linear. The Monte-Carlo simulation is also a well-known approach. It is based on sampling randomly the input variables to generate potential future trajectories (e.g. penalizing with weights those ones that do not respect the road constrains).

Other approaches reviewed in [29] are the maneuver-based. The main idea is to achieve a long-term prediction based on the prediction of the maneuver intended by the driver. The trajectories can be learned from samples, that is, previously observed characteristics (e.g. mean and standard deviation), or from a digital map with traffic rules information. Then, the partial trajectories (executed so far) are compared with learned patterns. In case of using Gaussian Process (GP) for that purpose, the distance is computed as the probability that the partial trajectory corresponds to the GP. In case of using finite set of prototypes, the similarity is measured, for example, using metrics such as the average of Euclidean distance between trajectory points, the modified Hausdorff, the Longest Common Subsequence (LCS), etc. Nevertheless, the main limitations of using prototype trajectories are the hard adaptation to road topologies and the complexity of handling variations of velocity.

Furthermore, different machine learning approaches to estimate the maneuver intention are reviewed. Some of the most relevant techniques are the Multi-Layer Perceptron (MLP), Logistic Regression (LR), Relevance Vector Machines (RVM), Support Vector Machines (SVM), etc. However, the authors assume the strong inter-vehicle dependencies as the main limitation of the maneuver-based approach. The 3$^{\text{rd}}$ type of motion prediction is based on considering the influence of other vehicles for a better understanding of the situation (see Interaction-aware in Fig. 2.9). The main drawback is the complexity of relations between road users. Since there are too many motion patterns to be learned, the inter-vehicle influences cannot be taken into account during the learning phase, but only during matching phase. For this reason, pairwise dependencies are commonly used. These dependencies can be modeled e.g. with Coupled Hidden Markov Models (CHMM). However, it becomes very quickly a very complex problem due to the number of objects. Therefore, the authors identify its computational cost as its main drawback, and consequently, the incompatibility with real-time requirements.

Then the authors in [29] divide the risk assessment in colliding future trajectories and unexpected behavior. Lefevre et. al. [30] address this issue by modeling the vehicle motions with a Dynamic Bayesian Network (DBN). The inference is performed to estimate the intention of the driver and the expectation (i.e. what a driver is expected to do). These 2 estimations are then compared to detect dangerous situations. In contrast, a combination of context, kinematic knowledge and sensor information with Bayesian filtering is used in [31] for predicting the motion of vehicles at intersections.

Another way to model the traffic situation is used in [32]. The authors use a Gaussian Process Regression (GPR) and the trajectories are predicted incorporating an Unscented Kalman Filter (UKF). The system tries to recognize the intention of vehicles approaching the intersection and to predict the long-term trajectory of detected vehicles. The motion flow is modeled with a 2-dimensional Gaussian process regression that describes the spatial-temporal characteristic. The regression models are learned from training instances. Once the situation is recognized, the trajectory is predicted using its motion model with a unscented Kalman filter. An examples is illustrated in Fig. 2.10.

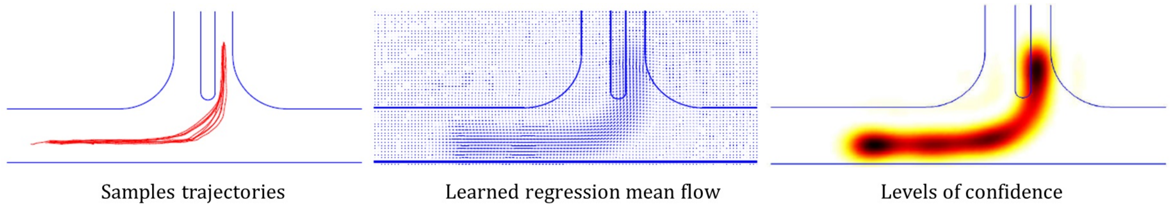


Figure 2.10.: *Example of GPR for motion flow. Left: sample trajectories of one situation. Left: regression mean flow learned from data. Right: corresponding levels of confidence (see [32]).*

In [33], a graph-based road network in is used to model the corresponding prediction. The vehicles are matched to the edges, so that in case this matching is ambiguous, the model generates hypotheses for every relevant edge. The intention of the ego vehicle driver is predicted using a Bayesian network (see Fig. 2.11 (a)).

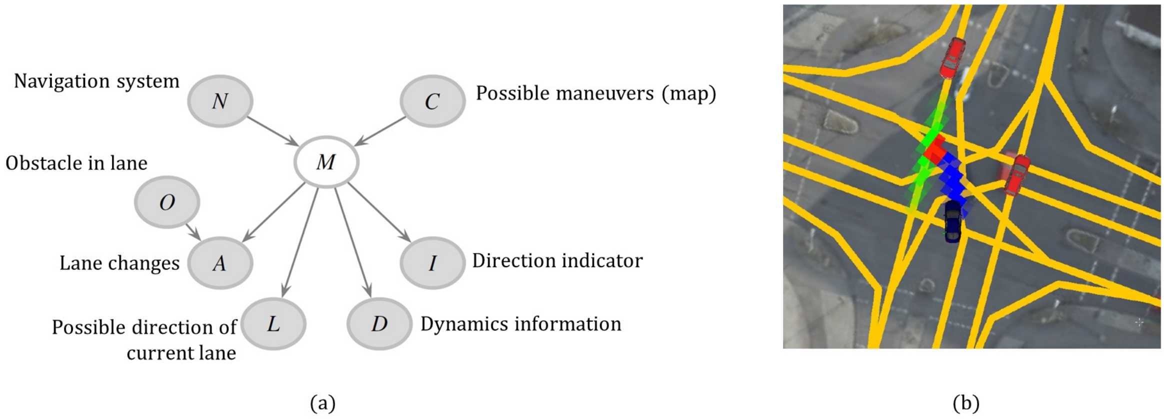


Figure 2.11.: *(a) Bayesian network for driver intent prediction and (b) probabilistic reachable sets at an urban intersection according to [33]).*

For other vehicles, the position prediction is achieved considering all the possible reachable position (slowing down, constant driving, expediting, turning left or driving forward.) with its corresponding probabilities. Using this possible reachable position, a probability function is modeled along the edges using linear approximation. A Gaussian distribution is assumed depending on the lane width (see Fig. 2.11 (b)). In case of no lanes (e.g. middle of an intersection) the variance of the function is larger in order to consider different turning trajectories.

2.5. Decision making

Nevertheless, the scenario interpretation for autonomous driving involves in most cases not only the knowledge representation and motion prediction of objects, but also the decision making process. Therefore it is also crucial to define the terms path, maneuver and trajectory. Based on [34], which presents a comprehensive survey on motion planning methods, these concepts are simplified as (see Fig. 2.12):

Path: Geometric trace that the vehicle should follow in order to reach its destination without colliding with obstacles.

Maneuver: High-level characterization of the vehicle motion, regarding the position and speed of the vehicle on the road (e.g. going straight, turning, overtaking, etc).

Trajectory: It represents the different states (including the velocity) of the vehicle over time. Usually, it is referred to as motion planning.

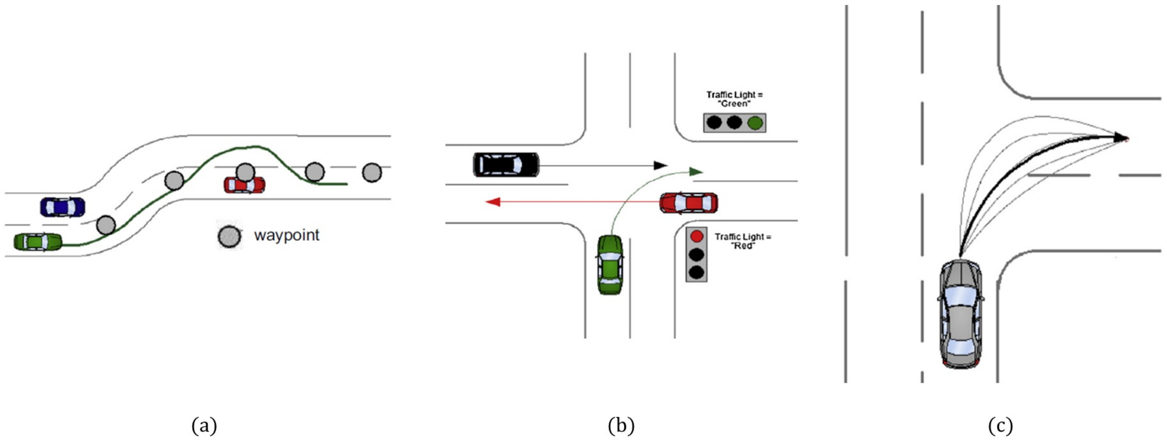


Figure 2.12.: *Graphical definition of the terms: (a) path, (b) maneuver and (c) trajectory planning according to [34].*

Therefore, these authors divide the planning module into 3 levels (finding the best geometric path, maneuver and trajectory). A fully classification is illustrated in Fig. 2.13.

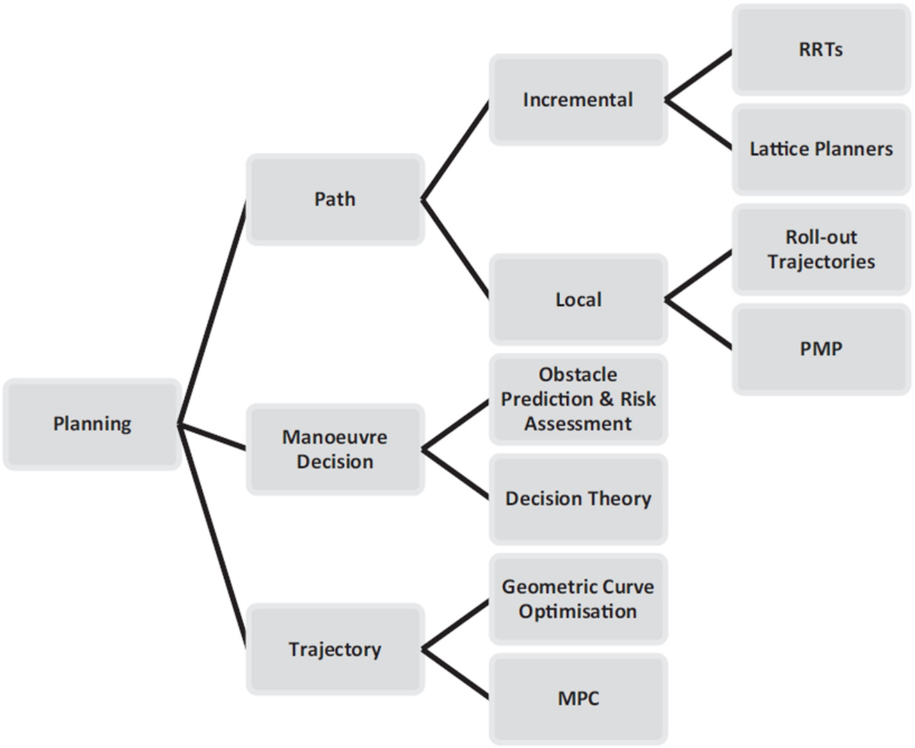


Figure 2.13.: *Classification of planning approaches into path calculation, maneuver decision or trajectory calculation (see[34]).*

Another consistent review of motion planning and control techniques for self-driving urban vehicles can be found in [35]. The authors decompose the decision making module in 4 components: routing planning, behavior layer, motion planning and local feedback control (see Fig. 2.14).

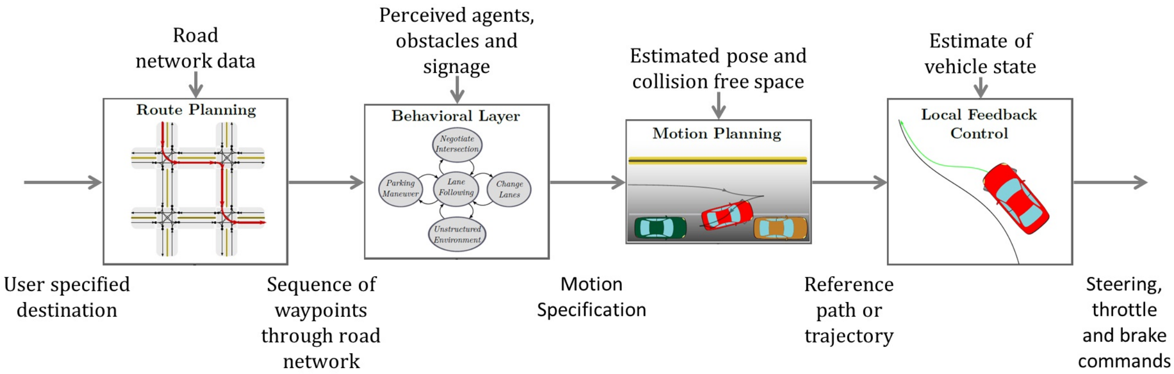

Road network data
Perceived agents, obstacles and signage
Estimated pose and collision free space
Estimate of vehicle state
Route Planning
Behavioral Layer
Negotiate Intersection
Parking Maneuver
Lane Following
Change Lanes
Unstructured Environment
Motion Planning
Local Feedback Control
User specified destination
Sequence of waypoints through road network
Motion Specification
Reference path or trajectory
Steering, throttle and brake commands

Figure 2.14.: *Hierarchy of decision-making processes based on: route planning, behavioral layer, motion planning and local feedback control (see[35]).*

The route planning, which represents the highest level, plans a route through the road network to reach a given destination. The behavioral layer decides a local driving task that guides the vehicle towards the destination following the rules of the road (e.g. stop, turn right, pass the intersection, etc.). Then, the motion planning translates the desired maneuver to a path and trajectory. And finally, the local feedback control reactively corrects errors in the execution of the planed motion.

In particular, the behavior layer should select the appropriate driving maneuver based on other objects, traffic rules, conditions, etc. A common approach to achieve this goal is to model every behavior as a state in a finite state machine, in which the transitions are controlled based on the perceived driving context, such as relative position with respect to the planned route and nearby vehicles. For example, in [36] Gindele et. al. present a hierarchical state machine based on 3 basic models: *Pause*, *Active* or *Error* (see Fig. 2.15).

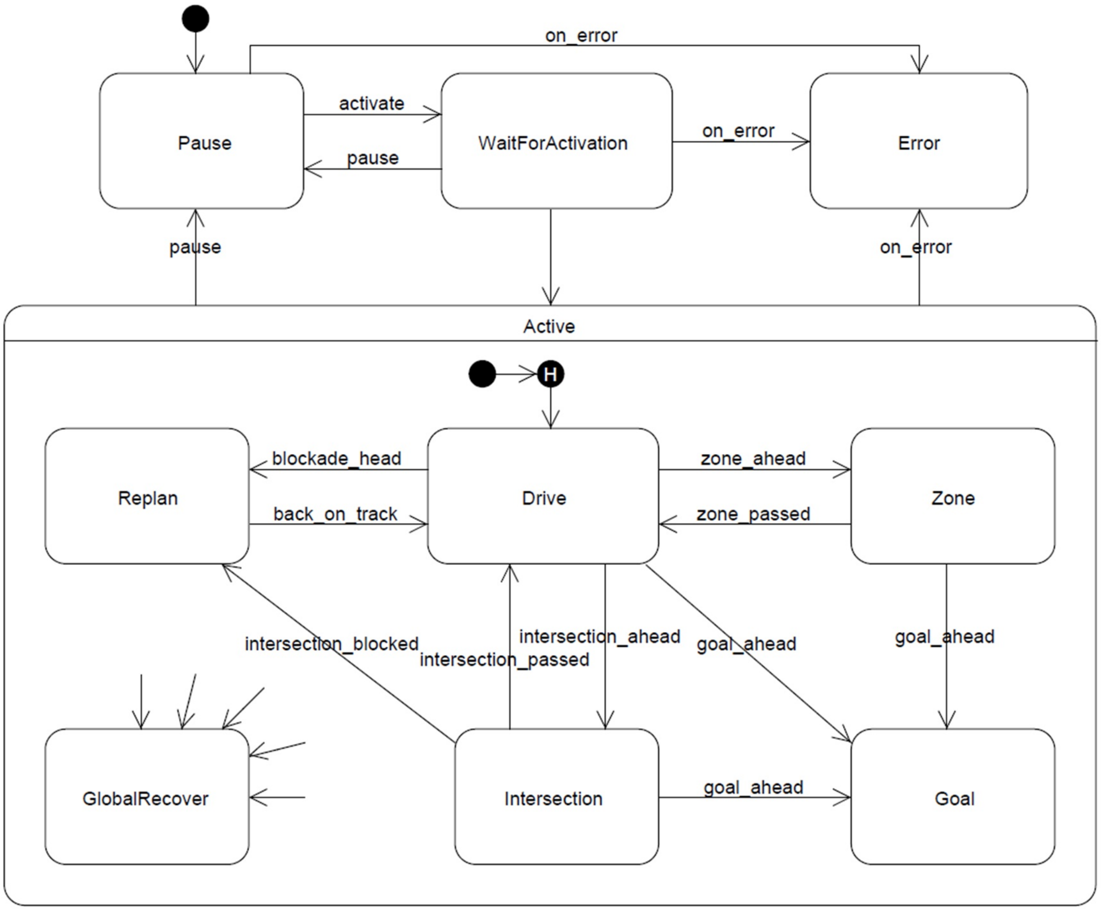


Figure 2.15.: *Diagram of the main driving states proposed by Gindele et. al. [36]. The state machine is based on three basic modes of operation: pause, active and error.*

Its design divides the movement in 3 different types: normal driving on streets, intersections and unstructured environments (see *Drive*, *Intersection* and *Zone* in Fig. 2.15). For handling intersections, 4 states are used:

IntersectionStop: if ego vehicle is in a yield road.

IntersectionPrioDriveInside: if it crosses the intersection on a priority road.

IntersectionPrioStop: if stop and wait is needed.

IntersectionRecover: if the ego vehicle did not make progress for a determinate amount of time.

Another similar approach using states machines is presented in [37]. Alonso et. al. introduce 2 methods for priority conflict resolution (priority charts and priority levels) using a Vehicle-to-Vehicle (V2V) communication system. The first method uses some vectors that describe the turning possibilities of all vehicles and their corresponding priority signs. Then, an auxiliary table containing all possible vectors associated with a Boolean values is used to indicate if the ego vehicle has to move or stop. This table contains 111 different cases without considering the traffic signs combinations (3 for 1 vehicle, 27 for 2 vehicles, and 81 for 3 vehicles). On the other hand, the second proposed method aims to determine whether ego vehicle can continue or must wait by interpreting the different priority levels (using an auxiliary truth table to detect potential conflicts with other vehicles). These 2 proposed methods depends on a determined topology (in this case a 2 road intersection). Moreover, V2V communication is required. The flowchart proposed by the authors to handle the right of way problem is shown in 2.16.

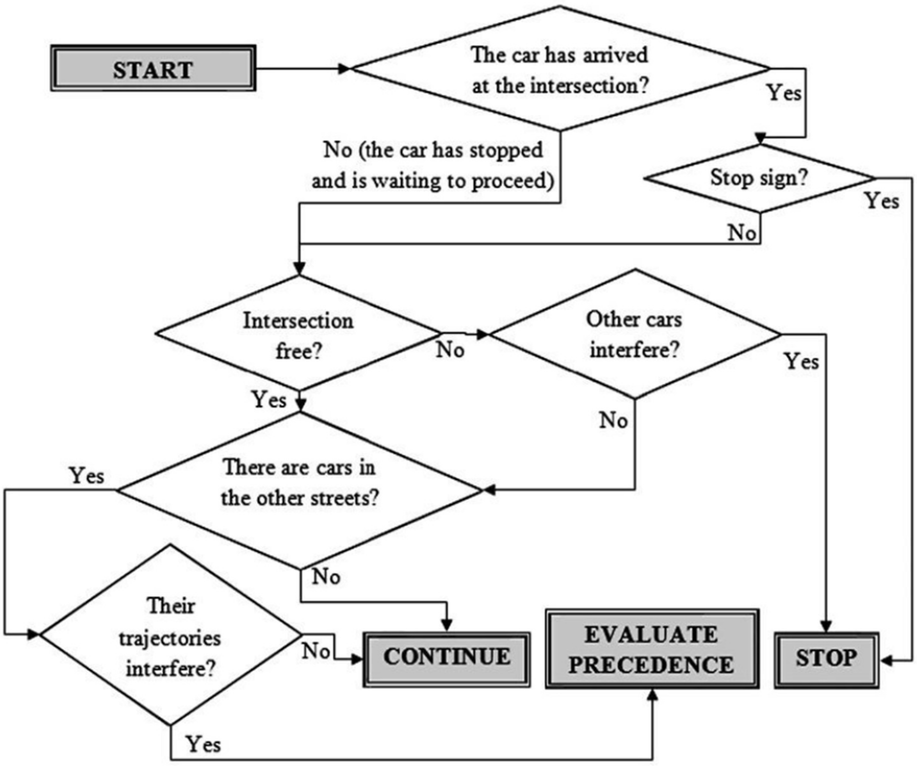
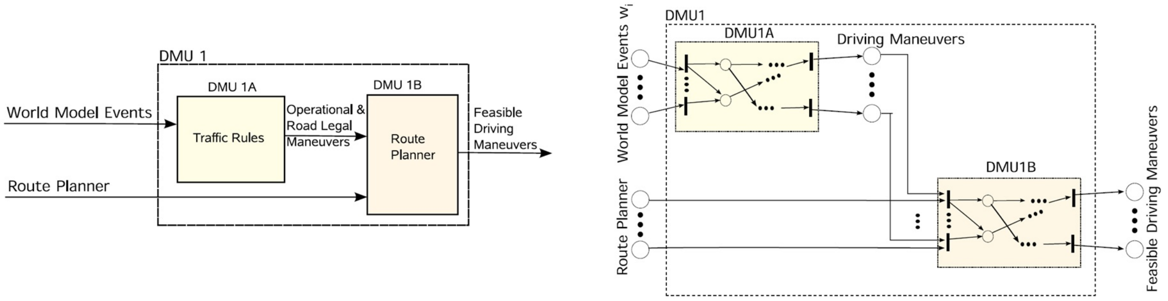

START
The car has arrived at the intersection?
Yes
No (the car has stopped and is waiting to proceed)
Stop sign?
Yes
No
Intersection free?
No
Other cars interfere?
Yes
Yes
No
Yes
There are cars in the other streets?
No
Their trajectories interfere?
No
CONTINUE
EVALUATE PRECEDENCE
STOP
Yes

Figure 2.16.: *Decision making flowchart at intersections proposed by [37])*

In a similar way, the authors in [38] model the operational behaviors as a Deterministic Finite Automaton (DFA). The decision making module is decomposed in 2 stages. First, the set of feasible maneuvers are chosen using nested Petri nets (see the concept in Fig. 2.17 (a) and the subnets representation in (b)). The given inputs are the tuple of events from the world model and the route indications.



(a)

(b)

Figure 2.17.: *Decision making using Petri nets (see [38]). The decision making unit is divided into the traffic rules reasoning and route planner (a), which are modeled by the subnets DMU1A and DMU1B (b).*

The output of the petri net represents the driving maneuvers that could be executed considering the traffic rules and comfort constrains. Then, the most appropriate maneuver is selected using Multiple Criteria Decision Making (MCDM).

Another interesting approach to deal with the decision making issue is presented in [39]. Here, the authors propose a maneuver-based planning for automated vehicles. Based on the desired maneuver (or set of maneuvers over the time) the proposed system plans the proper lane change in urban environments. The approach was tested in a multi-lane road network without other road users. Nevertheless, turning at intersections with different intersection states is not the focus of this work.

However, Partially Observable Markov Decision Process (POMDP) can also considered a relevant technique in the decision making module. For example, Brechtel et. al. [40] use a continuous POMDP that can be automatically optimized for different scenarios, so that it is not necessary to discretize the state space a priori. The idea is that the solver learns a good representation of the specific situation.

2.6. Discussion

At this point, it is clear that the problem of interpreting the scenario at urban intersections represents a complex research challenge. The idea is to understand the surrounding of the ego vehicle in order to enable a proper maneuver execution according to the traffic rules. For this purpose, the problem involves different tasks such as understanding the pass permission of the intersection from an ego perspective, predicting the intention and motion of other road users, deciding the resulting maneuver, etc.

From general to more specific, the scenario interpretation receives the data from the perception module and provides a logical high level representation that makes possible the further planning to execute the desired maneuver. In this context, complex reasoning calculations involving e.g. description logic seems a proper solution for complex scenarios. Nevertheless, high computational costs and dependencies with fixed topologies represent important drawbacks of this kind of approaches.

It has also been identified, that the complexity of understanding the preprocessed data depends on its quality. Namely, the more inaccurate the perception is, the more difficult is the interpretation of the provided data. But even if the perception provides accurate information about the surrounding of the ego vehicle, the problem is not simple.

The large number of possible collisions with other road users at urban intersections makes the problem a very complex challenge. This yields to the following statement: a requirement for a suitable solution is to analyze the problem in such a way that simplifies the problem.

The key of the problem is to determine which information is important to describe the scenario, or preferably, how to classify all the possible situations depending on the available information. Therefore, a feasible solution should aim to extract (only) the most relevant information, in order to achieve a simplification of the problem, and consequently, a successfully interpretation that enables further execution of the desired maneuver at intersections.

3. Overview of the baseline system

The contributions of this thesis are integrated in an existing system, which is considered as the baseline to a further development of the objective described in 1.3. In other words, this system represents the starting point for a further research work. Therefore, this system is briefly introduced within their modules, so that only the essential concepts are explained for a further understanding of the approaches proposed in this thesis.

First, a quick overview of the baseline system and its architecture is given. Then, the second section briefly explains the perception module. The driving function module used for autonomous driving at highway scenarios is explained in the third section. And finally, the key concepts of planning and control are introduced in the fourth section.

3.1. System architecture

The baseline system is essentially designed for automated driving at highways or rural scenarios. In particular, the system aims to achieve a conditional automation level (level 3 based on [17]). The system controls the acceleration, braking and steering, in order to reach a given target position on a map. Consequently, it is able to keep in the lane, adapt the velocity considering curvatures, follow other vehicles whilst keeping the proper distance, making lane changes to pass other vehicles if necessary, react to traffic lights, speed limits, etc. In a nutshell, the baseline system is able to make required comfort maneuvers. On the other hand, other maneuvers such as emergency braking, parking, reaction to Vulnerable Road Users (VRU), driving at urban scenarios or turning at intersections are not considered.

For this purpose, the system architecture can be simplified in 4 main modules: perception, function, planning and control. This simplification is illustrated in Fig. 3.1 within the most relevant modules.

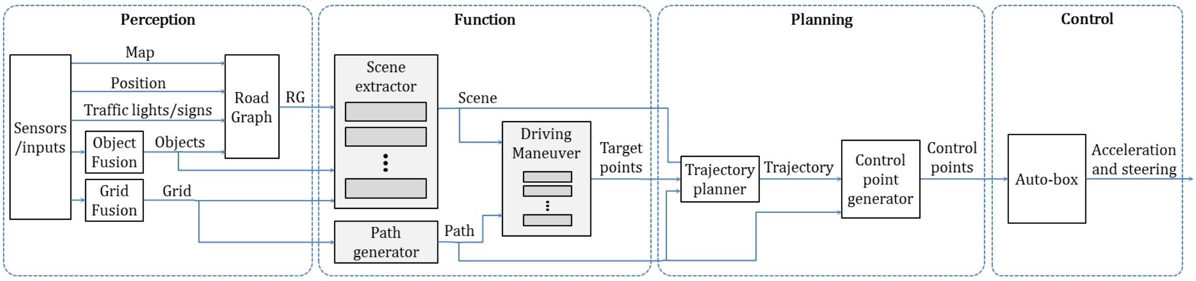


Figure 3.1.: *Simplification of the baseline system architecture based on four main modules: perception, function, planning and control.*

The low level processing of sensors and a priori data (e.g. image processing, object recognition and tracking, localization and mapping, etc.) is represented by the perception module. On the other hand, the function module involves understanding the described environment and making a tactical decision, so that it provides a calculated path and target points to the next module. Then, the planning module calculates the most comfortable trajectory as a velocity profile along the expected driving corridor to set the so-called control points. And finally, these control points are converted to provide the adequate signals in terms of steering and acceleration.

3.2. Perception

In the introduced architecture simplification, the perception module integrates the sensors/inputs and its low level preprocessing in order to describe the surrounding of the ego vehicle.

The first relevant aspect of the system is the sensor setup, which represents the first step of the acquisition process. This sensor setup consists of the following sensors (see Fig. 3.2):

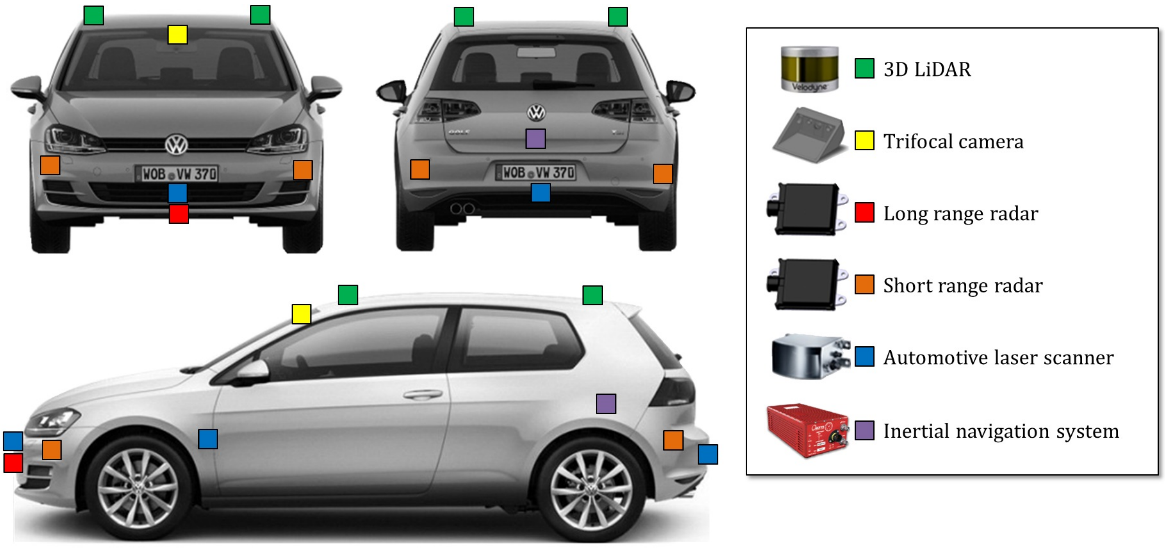


Figure 3.2.: *Overview of the baseline system sensor setup. The demonstrator vehicle is shown from 3 different perspectives (front, rear and lateral) with the used sensors represented by colored rectangles. A legend illustrates the meaning of the colored rectangles.*

Inertial and Global Positioning System (GPS) measurement system (x1) This device is an instrument for making precision measurements of motions in real-time. An inertial sensor block with 3 accelerometers and 3 gyros (angular rate sensors) is used to compute all the outputs. A World Geodetic System 1984 (WGS84) modeled strap-down navigator algorithm compensates for earth curvature, rotation and Coriolis accelerations.

3D LIDAR (x5) The Light Detection and Ranging (LIDAR) sensors are mounted on the top of the car (see Fig. 3.2 and 3.3). Their 16 scan lines have a range up to 100 m with a vertical and horizontal field of view of ±15 ° and 360 °, respectively.

Long range radar (x1) The long range radar is mounted at the front of the vehicle. Its approximately 250 m range makes this radar suitable for detecting objects at highway scenarios.

Short range radars (x6) These short rage radars achieve the detection of objects around the vehicle. As it can be seen in 3.2, 2 of them are mounted at the front and the other 4 at the rear side of the vehicle.

Trifocal camera (x1) Every objective has different field of view (34 °, 46 ° and 150 °). These cameras are controlled with a control device, which involves its own low level data processing.

Automotive laser scanners (x4) The automotive laser scanners are mounted around the vehicle (see Fig. 3.2). These have a range of approximately 80 m with 3 scan lines that alternate to provide 4 different levels.

This configuration has been engineered for autonomous driving purposes without considering design or marketing aspects. As it can be seen in Fig. 3.3, the different sensors have been mounted in a Volkswagen E-Golf 7.

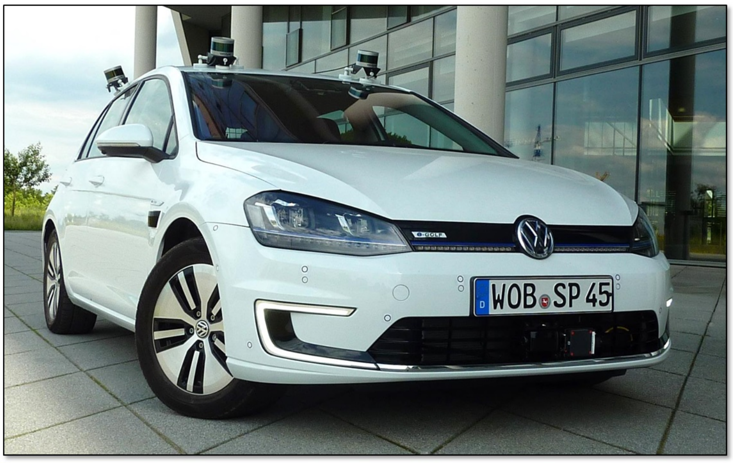


Figure 3.3.: *Mounted sensor setup in the demonstrator vehicle (Volkswagen E-Golf 7).*

In order to briefly introduce the most relevant modules of the perception, an example scenario at the highway is given in Fig. 3.4. Here, different views for the same time stamp are illustrated: the camera image (a), the Road-Graph (RG) with the ego position (b), the detected objects (c) and the grid (d). As it can be seen in the camera image, the road way contains 2 lanes and the ego vehicle is driving at the right lane. The most relevant objects are the ones driving in the same lane in front of the ego vehicle and the other 2 objects driving on the left lane.

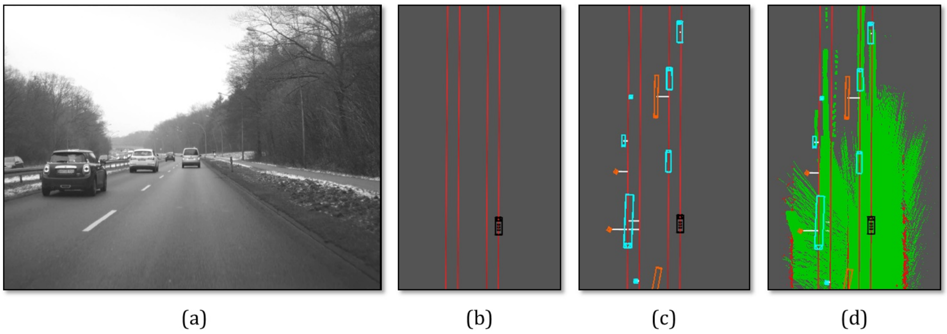


Figure 3.4.: *Visualization of the perception modules for a given example at the highway. From lefto to right: (a) front camera view, (b) RG with the position of the ego vehicle represented by a black rectangle, RG with the matched objects as a result of the object fusion module and (c) RG with matched object and grid (d).*

The **RG** is probably the most important interface between the perception and the function modules. It represents the a priori road network information, in which every lane corresponds to a graph. These graphs can be attributed with different information such as the position of the ego vehicle, speed limits, traffic signs, dynamic objects, etc. (see [41]).

The **object fusion** module (see 3.1) receives the acquired information from the sensors and fuses it to estimate the position and state of all detected objects in the surrounding of the ego vehicle. The objects are provided as a list of detected objects with their corresponding estimated states (i.e. velocity, position, etc.). These provided objects are not only sent to the function module, but also to the RG. Consequently, they are matched to the RG as attributes at their corresponding position. For the given example in Fig. 3.4 the resulting matched attributes are illustrated in (c).

The **grid fusion** processes the inputs from the sensors in order to provide an estimation of the surface around the ego vehicle. The provided grid consist of a matrix, in which every element corresponds to a cell of the grid. These cells are labeled with the states *free* if the ground surface has been detected by the corresponding sensor(s). In case there is a static object in this cell, it is labeled with the state *occupied*. If the cells are not inside the field of view of the sensors (or enough information is missing), they are marked as *unknown*. The provided result of the grid fusion at the given example is illustrated in Fig. 3.4 (d). Here, the cells with the states *free*, *occupied* or *unknown* are colored in green, red and black, respectively.

In this manner, the perception processes the sensor data to provide a description of the surrounding of the ego vehicle to the function module.

3.3. Driving function

Using the inputs from the perception module, the goals of the driving function can be summarized in 3 sub-tasks (see the *Scene extractor*, *Path generator* and *Driving manager* in Fig. 3.1). These are briefly described in the following paragraphs, using the same example as in the previous section (see the example in Fig. 3.4).

The purpose of the **path generator** is to optimize the path considering the a priori map information and the grid from the perception module. In this context, the term *path* refers to the 2-dimensional geometric trace that the vehicle should follow. It optimizes the path not only in order to avoid objects, but also to minimize the lateral acceleration along the calculated driving corridor. This is illustrated in Fig. 3.5 (a) and (b). The first illustration (a) shows the input from the perception (i.e. the Road-Graph with the ego position, the matched objects from the object fusion and the result from the grid fusion). Due to the occupied cells in the grid, the path generator modifies the desired driving corridor. This is clearly illustrated in Fig. 3.5, where the red solid line represents the driving corridor extracted from the a priori road network and the black dotted line corresponds to the result of the path generator.

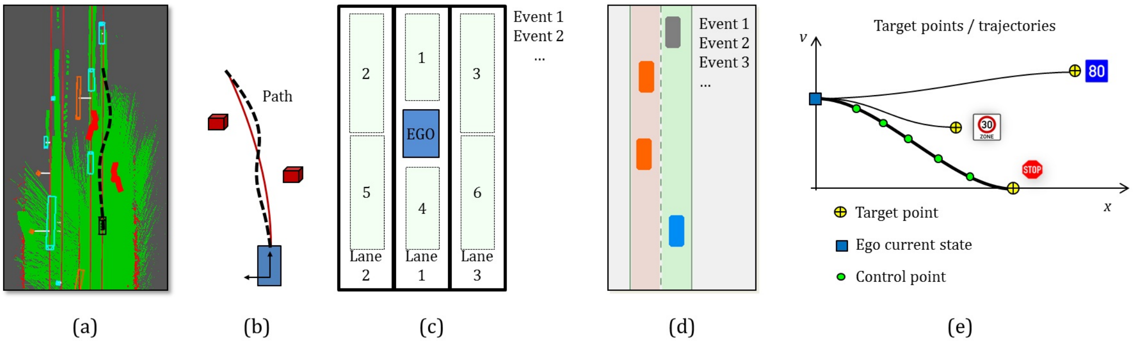


Figure 3.5.: *Example: driving function at the highway. (a) Perception visualization for the given example. (b) Calculated path considering the objects of the grid. (c) General representation of the scene. (d) Representation of the scene for the given example. (e) Simple example with 3 target points, their corresponding trajectories, control points and the selected trajectory.*

The **scene extractor** module consist of a set of filters. Every filter aims to obtain the relevant information required for executing all possible comfort maneuvers at highway scenarios (i.e. lane changes, pass other vehicles, follow objects, adapt the velocity to curvatures, etc.). This process can be described as the generation of a high level world representation that contains the needed information about the surroundings of the ego vehicle in Frenet coordinates along the desired path. In contrast to Cartesian coordinate system, the used Frenet coordinates system describes the surrounding of the ego vehicle using the distance from the rear axis along the driving corridor and the lateral offset at that distance. A general simplified representation of the *scene* is illustrated in Fig. 3.5 (c). Here, 3 lanes (ego, left and right), the detected vehicles around ego and the set of extracted objects are shown. Moreover, the illustration (d) shows the corresponding extracted *scene* for the given example at the highway. In other words, every scene extractor is in charge of extracting the relevant information for executing the corresponding maneuvers at the highway.

Then, this information in the *scene* is used as input for the **driving maneuver** module (see Fig. 3.1), which also consists of different filters. Every *maneuver manager* filter generates a target point considering the information of the *scene*. For example, in case that 3 events have been stored in the *scene* in the following order: a 30 Km/h speed limit, a stop sign and a advisory speed of 80 Km/h, every corresponding maneuver manager would generate a target point at every desired distance with its corresponding target velocity. This example is illustrated in Fig 3.5 (e), where the target point is drawn as a yellow crossed circle in a plot. This plot illustrates the desired target velocity v over the distance x along the calculated path.

The calculated scene, the optimized path and the provided target points correspond to the output of the driving function module (see main architecture in Fig. 3.1).

3.4. Planning and control

The **trajectory planer** aims to calculate a trajectory for every received target point, select the most appropriate trajectory and provide its control points. This control points are the input of the **control module**, which converts them into acceleration, braking and steering terms. This is graphically explained in the system architecture overview in Fig. 3.1.

In other words, the set of target points is provided to the **trajectory planer**, which calculates the corresponding trajectory for every target point. In this context, the term *trajectory* refers to the velocity profile over the distance along the calculated path. This velocity profile is illustrated for the given example in Fig. 3.5 (e) with a black curve from the ego state (blue rectangle at $x = 0$) to every target point. This curve is calculated by solving a 5^{th} order polygon with certain constrains (see [42] for more details).

Once all trajectories have been calculated, the most conservative one is selected as the desired trajectory. This is converted into control points, so that the **control module** transforms these into the corresponding signals to execute the desired acceleration, braking and/or steering. For the given example, the selected trajectory is drawn as a bold curve and the corresponding control points as little green circles along the velocity profile (see Fig. 3.5).

4. Contributions of this thesis

According to the previously mentioned objectives of this thesis, this chapter aims to structure and describe the relevant contributions of this work. The developed approaches are organized in 6 categories. First, the problem of scenario understanding for automated driving at urban intersections is addressed from a general point of view. After this quick analysis of the problem, the concept *pass permission* is introduced and the proposed approach for its interpretation is described in detail, which represents the first contribution of this thesis. Furthermore, the second contribution consists of a concept to interpret the scenario at intersections based on 4 primary situations. Based on this concept, the next approach tries to achieve a systematic motion prediction of pedestrians and crossing vehicles. Afterward, the fifth section of this chapter explains how the decision making is done. Finally, the last approach describes how the system handles occlusions taking advantages of the primary situations concept.

4.1. Problem analysis

The way the problem is addressed in this thesis is based on the following statement: the most coherent way to address the problem is to describe an intersection in a comprehensive manner and classify the possible scenarios that could occur. Therefore, a conceptual description of the scenario answering 3 questions is the first step to address the problem. This description based on these 3 questions is illustrated in Fig. 4.1.

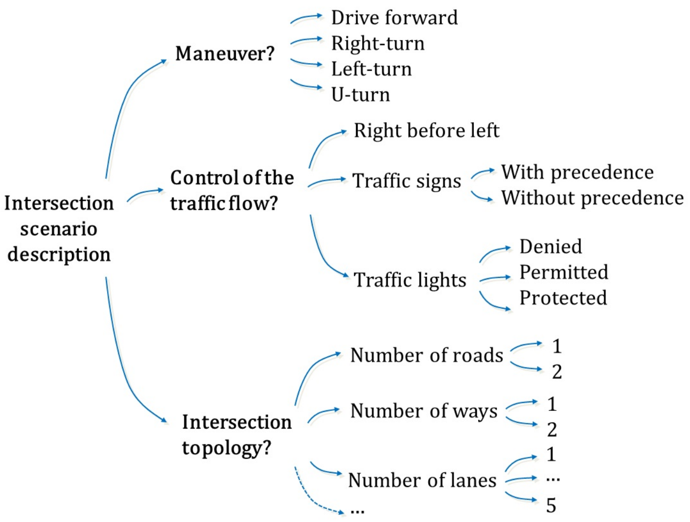


Figure 4.1.: *Description of the intersection scenario based on 3 questions. (1) Which maneuver is desired? (2) How is the traffic flow controlled? and (3) How is the topology of the intersection?*

For automated driving systems, it can be taken for granted that the route of the ego vehicle is well known, and consequently, the desired path and its maneuver, too. However, this is not the only issue. Understanding the current pass permission when approaching an intersection is a very important information to take into account. But unfortunately there is no standard regulation that controls the traffic flow at intersections in a unique manner for all the possible scenarios all over the world. Therefore, the presented work in this thesis considers the regulation described in the Vienna Convention on Road Signs and Signals [43] and the German regulation in particular [44]. But in order to simplify the problem, it is assumed that the traffic flow at intersections can be controlled in 3 different ways: by the right of way rule, with traffic signs or traffic lights. In this context, the presented approach does not consider other inputs such as special vehicles, police officer indications, constructions, etc.

These considerations facilitate a coherent classification, in which analyzing the possible conflicts with other road users is feasible. In other words, the ego vehicle intention and the control of the traffic flow yield different scenarios and potential conflicts with other vehicles or VRUs. Fig. 4.2 illustrates the different possible scenarios considering a common intersection topology. This classification is an improved version of the method proposed by Fastenmeier [45].

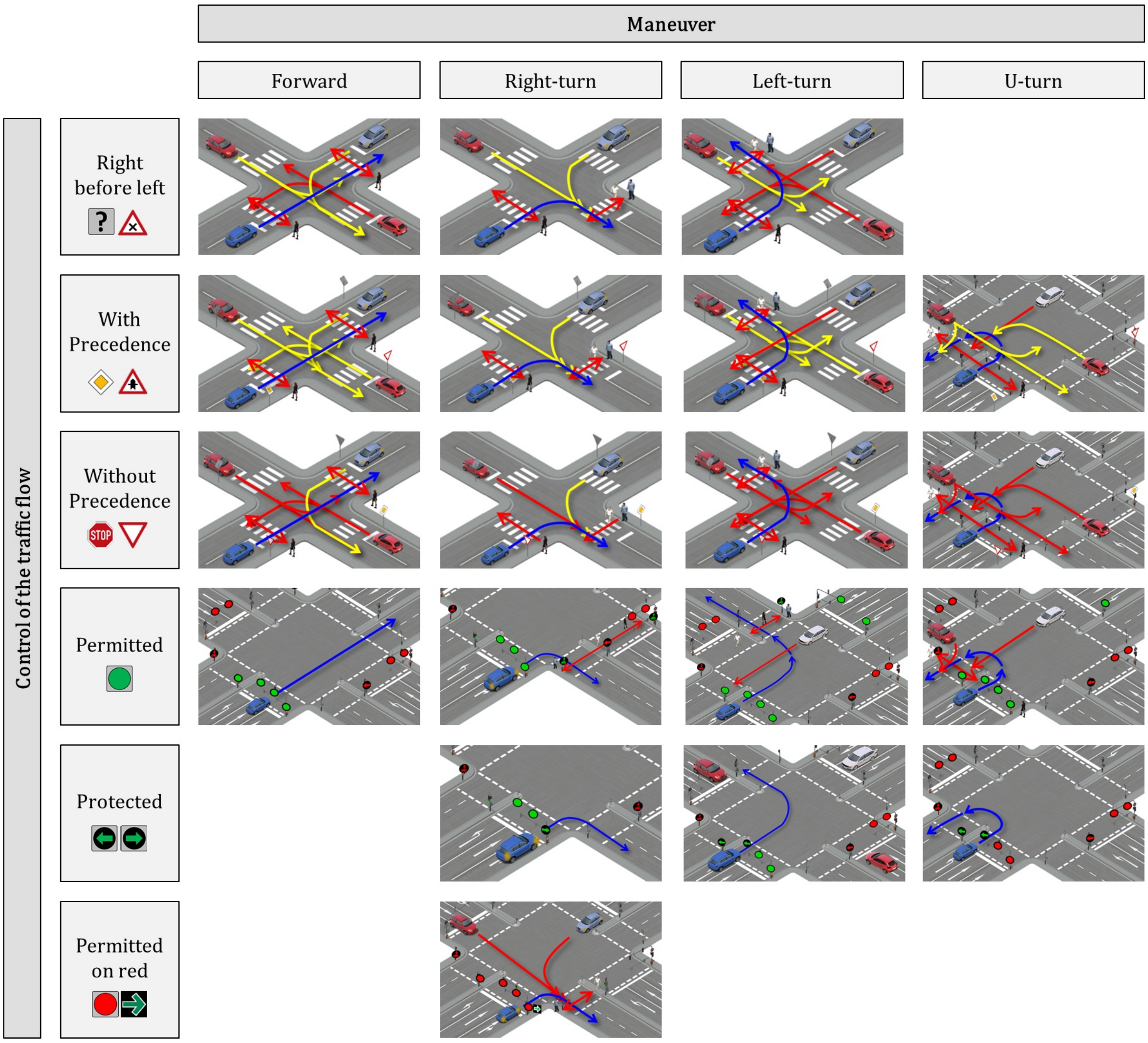


Figure 4.2.: *Classification of possible scenarios at a common intersection topology considering the desired maneuver and how the traffic flow is controlled (based on [24]). Every row corresponds to a pass permission and every column to a possible maneuver at a generic intersection. The path of the ego vehicle is indicated by a blue arrow. The path of other road users with and without the right of way (with respect to the ego vehicle) are represented by a red or yellow arrow, respectively.*

Every row represents a different maneuver for the ego vehicle with its path (blue). Every column corresponds to a different manner to control the traffic flow. All possible paths of other vehicles with a potential collision with ego vehicle are colored depending on its priority. Other vehicles (or VRU) with a red path have priority with respect to ego vehicle. Other vehicles with yellow paths are required to give way to ego.

4.2. Pass permission interpretation at urban intersections

The first matter of a human driver approaching an intersection is to determine how the traffic flow is controlled. Here, this issue is called: *pass permission interpretation*, which is not only about estimating if one is allowed to pass the intersection or not [46] [47]. In fact, understanding how one should pass the intersection and under which conditions is crucial for a further decision making. The core of the problem is to interpret the different recognized traffic lights and signs in a proper manner, so that the system can plan and execute the desired maneuver considering the traffic rules. This task is not only a challenge for a self-driving car, but also for human drivers [48] [49]. According to this, an autonomous system should first perceive the surrounding of the ego vehicle, then interpret it and finally set the appropriate driving strategy.

First, how to handle the uncertainty of the inputs that influence the pass permission is explained in the following subsection. The aim is to define the probability of a certain assignment of a traffic light or sign into a lane. The next subsection describes the concept based on modeling the pass permission as a probability mass function of discrete states. Subsequently, the third subsection explains how the fluctuations of the probability mass functions are interpreted over time and depending on the distance to the intersection.

4.2.1. Handling uncertainty of pass permission inputs

Among other problems, the inaccuracy of the perceived inputs and their time fluctuations increase the difficulty of understanding the current situation. This inaccuracy can be caused by different reasons. For example, in case that the localization is not accurate enough because of poor GPS signal, the traffic lights are erroneously detected or there are discrepancies between the a priori map and the real road network. The probability that a particular traffic light is valid for the ego vehicle depends on the accuracy of previous modules. Therefore, to handle the uncertainty considering other modules in the main system architecture is crucial. In Fig. 4.3 a simplified architecture structured in 5 main modules is illustrated: (1) sensors/input, (2) perception, (3) scenario interpretation, (4) planning and (5) control.

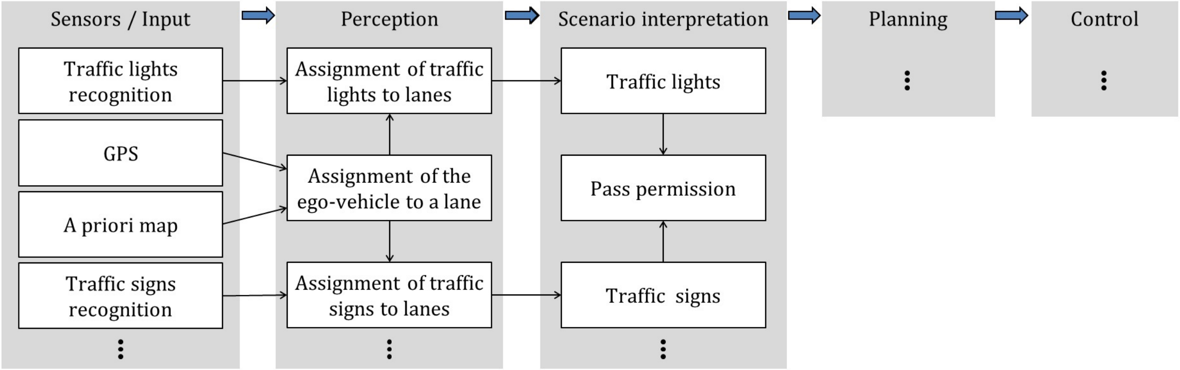

Sensors / Input
Traffic lights recognition
GPS
A priori map
Traffic signs recognition
Perception
Assignment of traffic lights to lanes
Assignment of the ego-vehicle to a lane
Assignment of traffic signs to lanes
Scenario interpretation
Traffic lights
Pass permission
Traffic signs
Planning
Control

Figure 4.3.: *Simplification of the main system architecture with the relevant submodules for the pass permission interpretation: sensors/input, perception, scenario interpretation, planning and control.*

The perception module receives the detected traffic signs and traffic lights (e.g. from the camera or Vehicle-2-X (V2X) communication), the position of the ego vehicle (from the GPS or other localization approaches) and an a priori map that describes the road network. The assignment of the ego vehicle into a lane is done using its position and the a priori map as input. Once it is estimated on which lane the ego vehicle is, this information is used to achieve the assignment of the detected traffic lights (and traffic signs) into lanes. In other words, the association between every traffic light (and traffic sign) to every lane is estimated and passed to the scenario interpretation module. This information is used to understand how the behavior of the ego vehicle at the intersection should be (namely, the pass permission from an ego perspective). Consequently, this state suggests the behavior of the ego vehicle, so that the proper maneuver can be provided to the planning module. Finally, the controller converts its input (a suggested trajectory) into acceleration and steering values.

The scenario interpretation module deals with the *Bayesian probability* of its input. This probability does not indicate how frequently an event occurs, but how certain a given hypothesis is. For example, if the location is very inaccurate (and hence it is improbable that the ego vehicle is assigned correctly to its lane), the hypothesis of achieving a perfect association of several detected traffic lights into lanes is very unlikely.

The basic idea is to calculate the Bayesian probability $P(TL_k)$ of every traffic light state k considering the probability of previous modules as evidences. Here, the key of the problem is to analyze where does the uncertainty come from to achieve a suitable calculation of the assignment of traffic lights and traffic signs to the ego lane. The way how the propagation of uncertainty is modeled is illustrated in Fig. 4.4.

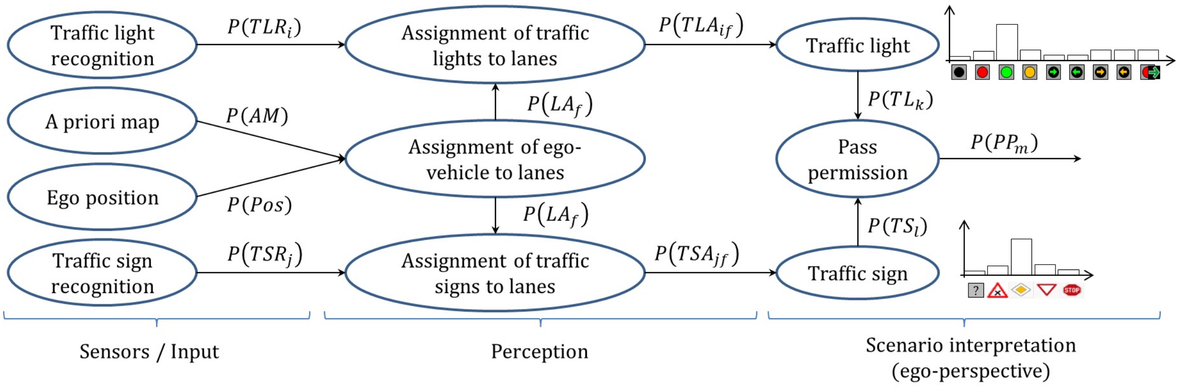


Figure 4.4.: *Uncertainty propagation for the modules involved in the pass permission interpretation: sensors/input, perception and scenario interpretation. Every relevant element (represented by an ellipse) provides an output with its corresponding probability.*

The process is divided into 3 relevant components: sensors/input, perception and scenario interpretation (compare Fig. 4.4 and 4.3). The involved probabilities are defined as:

$P(Pos)$ Probability of the correct localization of the ego vehicle.

$P(AM)$ Probability that the a priori map is correct.

$P(TLR_i)$ Probability of the proper recognition of a detected traffic light i, where i is an index from 1 to the number of detected traffic lights ($i = \{1, 2, ..., I\}$).

$P(TSR_j)$ Probability of the proper recognition of a detected traffic sign j, where j is an index from 1 to the number of detected traffic signs ($j = \{1, 2, ..., J\}$).

$P(LA_f)$ Probability that the assignment of the ego vehicle to the lane f is correct.

$P(TLA_{if})$ Probability of the assignment of a detected traffic light phase i into a lane f.

$P(TSA_{jf})$ Probability of the assignment of a detected traffic sign j into a lane f.

$P(TL_k)$ Probability that every possible traffic light phase k is valid for the ego vehicle (see Fig. 4.6).

$P(TS_l)$ Probability that every possible traffic sign l is valid for the ego vehicle.

$P(PP_m)$ Probability that the pass permission m is valid for the ego vehicle.

Because there is an independent probability for every association of every traffic light ($i = \{1, ..., I\}$) to every lane ($f = \{1, ..., F\}$), the resulting assignment probabilities can be expressed as a (IxF)-matrix in which every element represents an independent probability [1]:

$$P(TLA_{if}) = \begin{bmatrix} P(TLA_{1,1}) & P(TLA_{1,2}) & \cdots & P(TLA_{1,F}) \\ P(TLA_{2,1}) & P(TLA_{2,2}) & \cdots & P(TLA_{2,F}) \\ \vdots & \vdots & \ddots & \vdots \\ P(TLA_{I,1}) & P(TLA_{I,2}) & \cdots & P(TLA_{I,F}) \end{bmatrix} \tag{4.1}$$

An example with 5 detected traffic lights and 3 lanes is illustrated in Fig. 4.5. The first 4 traffic lights ($i = \{1, 2, 3, 4\}$) are valid for the 3 lanes ($f = \{1, 2, 3\}$) and the 5th traffic light is valid just for the third lane.

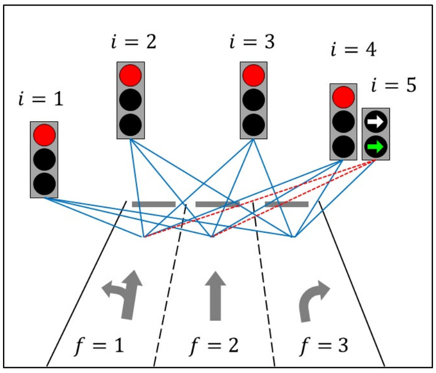


Figure 4.5.: *Example of the assignment of 5 detected traffic lights ($I = 5$) into 3 lanes ($F = 3$). The blue and red lines represent a correct and erroneous assignment of every detected traffic light i into every lane f, respectively.*

According to this concept, an ideal assignment should provide the following probabilities:

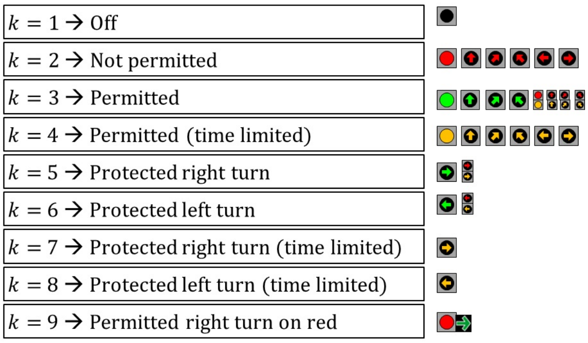
$$P(TLA_{if}) = \begin{bmatrix} P(TLA_{1,1}) & P(TLA_{1,2}) & P(TLA_{1,3}) \\ P(TLA_{2,1}) & P(TLA_{2,2}) & P(TLA_{2,3}) \\ P(TLA_{3,1}) & P(TLA_{3,2}) & P(TLA_{3,3}) \\ P(TLA_{4,1}) & P(TLA_{4,2}) & P(TLA_{4,3}) \\ P(TLA_{5,1}) & P(TLA_{5,2}) & P(TLA_{5,3}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.2}$$

[1] Note that this matrix should not be misunderstood with a stochastic matrix. Every element of the matrix represents an independent hypothesis.

In the same manner, the probability of the assignment of traffic signs into lanes $P(TSA_{jf})$ can be also expressed in a general form as the following matrix:

$$P(TSA_{if}) = \begin{bmatrix} P(TSA_{1,1}) & P(TSA_{1,2}) & \cdots & P(TSA_{1,F}) \\ P(TSA_{2,1}) & P(TSA_{2,2}) & \cdots & P(TSA_{2,F}) \\ \vdots & \vdots & \ddots & \vdots \\ P(TSA_{I,1}) & P(TSA_{I,2}) & \cdots & P(TSA_{I,F}) \end{bmatrix} \tag{4.3}$$

As it can be seen in Fig. 4.4, the variables i and j represent the index of the detected traffic lights and signs respectively. In contrast, the variable k represents the index of every possible state of a traffic light and l indicates every type of traffic sign. In the case of traffic lights, the different colors (red, amber, green), forms (normal, left arrow, right arrow...) result in multiple combinations. The combinations that indicate the same pass permission are grouped together into 9 different states ($k = \{1, 2, ..., 9\}$), so that the term *state* corresponds to a group (i.e. every group indicates a different behavior for the ego vehicle). This is shown in Fig. 4.6.



k = 1 → Off

k = 2 → Not permitted

k = 3 → Permitted

k = 4 → Permitted (time limited)

k = 5 → Protected right turn

k = 6 → Protected left turn

k = 7 → Protected right turn (time limited)

k = 8 → Protected left turn (time limited)

k = 9 → Permitted right turn on red

Figure 4.6.: *Set of different possible states k of the traffic light. Every different state involves a different reaction (or pass permission) for the ego vehicle.*

Finally, $P(TL_k)$ corresponds with the probability that every possible state k is valid for the ego vehicle. Using the assignments of the ego vehicle to lanes and traffic lights to lanes, the conditional probability that every traffic light state k is valid for the ego vehicle ($P(TL_k)$) is calculated:

$$\begin{aligned} P(TL_k) = & P(TL_k|TLA_{if}) \cdot P(TLA_{if}|LA_f, TLR_i) \cdot P(TLR_i) \cdot \\ & P(LA_f|AM, Pos) \cdot P(Pos) \cdot P(AM) \end{aligned} \tag{4.4}$$

Similarly, the probability that every vertical traffic sign state l is valid for the ego vehicle results:

$$\begin{aligned}P(TS_l) =&P(TS_l|TSA_{jf}) \cdot P(TSA_{jf}|LA_f, TSR_j) \cdot P(TSR_j)\cdot \\ &P(LA_f|AM, Pos) \cdot P(Pos) \cdot P(AM)\end{aligned} \tag{4.5}$$

In a nutshell, to handle the inaccuracy of the input, the probability of previous modules is considered by modeling the uncertainty propagation of the involved modules. Finally, the conditional probability that every possible traffic light phase k is valid for the ego vehicle (i.e. $P(TL_k)$) is calculated. In the same way, the key idea is also used to traffic signs (i.e. $P(TS_l)$).

4.2.2. Modeling the pass permission as a probability mass function

The objective is to help the system to set the proper maneuver automatically according to the European traffic rules. This is achieved generating a probability mass function in which every discrete state corresponds to a possible pass permission from the ego perspective. Every resulting state indicates a certain behavior of the ego vehicle with respect to the intersection. Nevertheless, the approach has been developed only considering the traffic lights and signs as input, but not special situations with indications such as emergency vehicles, police officer, temporal constructions sites, etc. Consequently, a probability mass function is generated for the traffic lights and another one for the traffic signs. Then, these are combined to set the corresponding pass permission.

The set of states $k = \{1, 2, ..., 9\}$ is expressed as a discrete random variable of a probability distribution. This resulting mass function indicates the probability that every state of the traffic light k is valid for the ego vehicle. Since every state is interpreted as a dependent hypothesis, the sum of the probabilities of every state is 1:

$$\sum_{k=1}^{K} P(TL_k) = 1 \tag{4.6}$$

Likewise, a probability mass function is calculated for the different traffic signs (l):

$$\sum_{l=1}^{L} P(TS_l) = 1 \tag{4.7}$$

Since the main idea is to estimate the current pass permission state based on the traffic signs and lights, both functions are combined to calculate the probability of every pass permission state $P(PP_m)$:

$$\sum_{m=1}^{M} P(PP_m) = 1 \tag{4.8}$$

The pass permission from an ego vehicle perspective is simplified in states $(m = 1, 2, ..., M)$ that indicate different behaviors of the ego vehicle at the intersection:

(m = 1) Not permitted The ego vehicle shall not enter the intersection (for example, if a red light is valid for its lane).

(m = 2) Permitted Passing is allowed if the driving corridor of ego vehicle is not congested. However, it shall yield the right of way to oncoming vehicles or vulnerable road users at parallel crosswalks/bike-lanes.

(m = 3) Permitted (time limited) The ego vehicle shall stop before the intersection unless the stopping cannot be made safely. Otherwise, the permission is interpreted as permitted.

(m = 4) Protected While turning, the ego vehicle is protected from oncoming vehicles and crossing bikes/pedestrians, which shall not be permitted to enter the intersection.

(m = 5) Protected (time limited) Ego vehicle shall stop before the intersection unless the stopping cannot be made in safety. Otherwise, the permission is interpreted as protected.

(m = 6) Permitted turn on red The ego vehicle is allowed to turn right just if the way is clear and the maneuver is safe from a collision with other road users.

(m = 7) right-before-left The ego vehicle shall yield the right of way to the vehicles crossing from the right.

(m = 8) With precedence Other crossing vehicles shall give way to the ego vehicle.

(m = 9) Yield Passing is allowed, but the ego vehicle has to yield the right of way to other vehicles.

(m = 10) Stop The ego vehicle shall stop before entering the intersection and then give way to other possible crossing vehicles.

The key question is to combine the probability mass functions for traffic lights and traffic signs in a coherent way, to finally compute the probability mass function that indicates the pass permission. This idea is illustrated in Fig. 4.7.

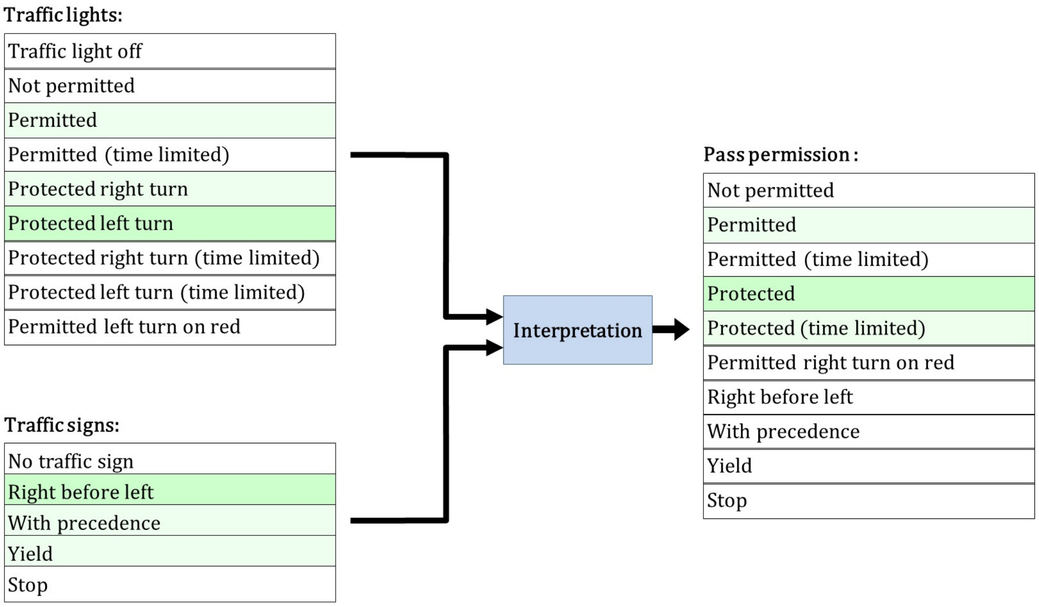


Figure 4.7.: *Combination of the traffic light (top-left) and traffic signs (bottom-left) probability mass functions ($P(TL_k)$ and $P(TS_l)$) to calculate the pass permission probability mass function ($P(PP_m)$).*

Considering the regulations indicated in [43], the traffic flow at intersections is first controlled by traffic lights, then traffic signs (if there are no traffic light) or by the *right-before-left* rule. Taking this into account, the combination is done using the mode of the probability mass functions, so that the most probable state of $P(TL_k)$ is considered to update the states of $P(PP_m)$ if the value of the mode is large enough. That is:

$$P(PP_m) \begin{cases} h(P(TL_k)) & \text{if } E_{TL} >= Th \\ h(P(TS_l)) & \text{else if } E_{TS} < Th \\ default & \text{otherwise} \end{cases} \tag{4.9}$$

where $h(\cdot)$ is the mapping function that indicates which probability mass function is used to update the pass permission ($P(PP_m)$). The terms E_{TL} and E_{TS} represent the difference between the mode and the mean value of the traffic lights and traffic sign function, respectively:

$$E_{TL} = mode(P(TL_k)) - \sum_{k=1}^{K} \frac{P(TL_k)}{K} \tag{4.10}$$

and

$$E_{TS} = mode(P(TS_l)) - \sum_{l=1}^{L} \frac{P(TS_l)}{L} \tag{4.11}$$

Here, Th corresponds to a threshold set empirically to 0.2. Therefore, the pass permission is mapped considering the traffic light states if the value with the largest probability is at least 20% over the mean. Else, the traffic sign states are considered. Moreover, the default state is set to *right-before-left* if there is not traffic sign.

4.2.3. Probability mass function over time

A big bottleneck of the interpretation module is to handle uncertainty and fluctuations of the perceived inputs over time. In other words, the objective is to interpret temporal changes of the calculated probabilities in a proper manner. This section explains how the interpretation over time is done using the traffic light probability mass function $P(TL_k)$ as an example.

In real situations, the traffic lights are often erroneously (or not) detected, so that the resulting probability varies over time in an illogical way (e.g. from *not permitted* to *permitted* suddenly, and back to *off*). These errors are typically due to false detections of the camera, wrong assignment to lanes, occlusions, etc. An example of these typical fluctuations of the traffic light states over time ($P(TL_k, t)$) is shown in Fig. 4.8.

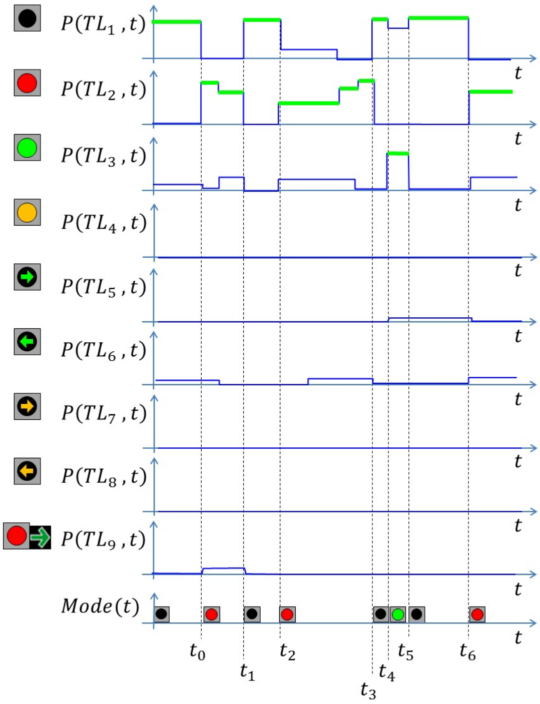


Figure 4.8.: *Example of a probability mass function for traffic lights over time $P(TL_k, t)$. The probability of every state k is plotted from the top ($k = 1$) to the bottom ($k = 9$). The mode of the function is remarked in green in every single plot and separately plotted at the bottom (mode(t)).*

Thanks to this example, it is possible to note visually how often the mode value ($mode(t)$) changes over time. The mode is *off* (i.e. $k = 1$) until t_0, then it changes to *not permitted* until t_2, and so on... Consequently, it becomes obvious the need of mitigating suddenly changes over time (i.e. to smooth changes over time). At first, in order to keep this filtering uncomplicated, the function is smoothed over time using an exponential moving average approach:

$$P^*(TL_k, t) = P(TL_k, t)(1 - \alpha) + P(TL_k, t - 1)\alpha, \tag{4.12}$$

where $P^*(TL_k, t)$ represents the smoothed function and the factor $\alpha = \{0, 1\}$ indicates how effective the smoothing is over time. Nevertheless, observing the values of the traffic light state *not permitted* (i.e $P(TL_2, t)$) is easy to notice if a constant value of α is appropriate or not: an increasing and decreasing probability (see the fluctuation at $P(TL_2, t_0)$ and $P(TL_2, t_4)$) would smooth $P(TL_2, t)$ with the same value of α. From a logical point of view, a decreasing probability of *not permitted* requires a smaller value of α than an increasing one (since it could be unsafe to smooth the probability of a properly recognized red traffic light). Therefore, a conditional exponential moving average with different α values is used to increase and decrease the probabilities (α_{in} and α_{de}):

$$\alpha = \begin{cases} \alpha_{in} & \text{if } P(TL_k, t) > P(TL_k, t-1) \\ \alpha_{de} & \text{otherwise} \end{cases} \tag{4.13}$$

Thanks to this concept, the terms *conservative* and *non-conservative* smoothing can be introduced. Accordingly, the smoothing is conservative when $\alpha_{in} < \alpha_{de}$. The concept of a conservative or non-conservative smoothing is graphically explained in Fig. 4.9 with an example.

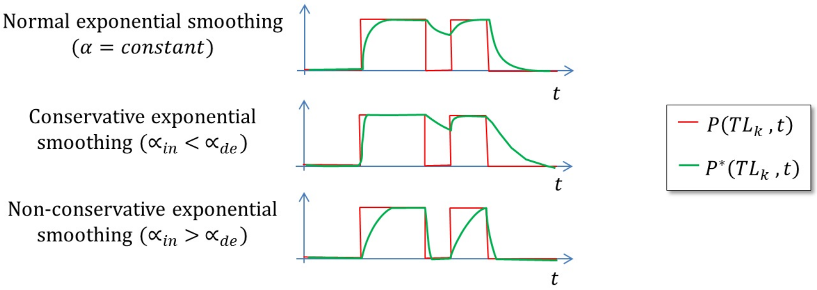


Figure 4.9.: *Examples of different exponential smoothing over time: normal (top), conservative (center) and non-conservative (bottom).*

Nevertheless, the position of the ego vehicle with respect to the intersection, has to be taken into account in order to update the changes of the probability mass function over time. For example, lets say one is approaching the intersection with the intention of turning left. Firstly, the focus of a human driver approaching the intersection is to interpret how to pass it. Once the ego vehicle is inside the intersection, the perceived pass permission remains fixed until one has complete the left turn maneuver. In other words, a human driver pays special attention to the possible changes over time when approaching the intersection. Then, once inside it (i.e. the traffic light is behind the ego vehicle), the last interpreted pass permission keeps being valid until the end of the whole maneuver. In order to imitate this behavior in the proposed approach, the factor $\delta(d)$ is introduced in the equation 4.12, so that the resulting smoothed function $P^*(TL_k, t)$ also depends on the distance d from the rear axis of the ego vehicle to the start of the intersection:

$$P^*(TL_k, t) = P(TL_k, t)(1 - \alpha \cdot \delta(d)) + P(TL_k, t-1) \cdot \alpha \cdot \delta(d), \tag{4.14}$$

with

$$\delta = \begin{cases} \delta_{min} & \text{if } C = 0 \\ (\frac{1-\delta_{min}}{d_{fov}-d_{bumper}})(d - d_{bumper}) + \delta_{min} & \text{if } C = 1 \text{ or } d < d_{fov} \\ 1 & \text{else,} \end{cases} \tag{4.15}$$

where the variable $C = \{0, 1, 2\}$ represents the crossing state of the ego vehicle with respect to the intersection (i.e. *crossing*, *approaching* and *unknown* respectively). As it can be seen in Fig. 4.10, the value δ_{min} ensures that even when the ego vehicle is already inside the intersection, the smoothing is always active (the closest $\delta(d)$ to 0.0 is, the slower is the update of the states over time). Based on experimental results, the value of δ_{min} was set to 0.01. On the other hand, variables d_{bumper} and d_{fov} indicate the distance from the rear axis to the front bump of the ego vehicle and the optimal field of view distance to detect traffic lights, respectively.

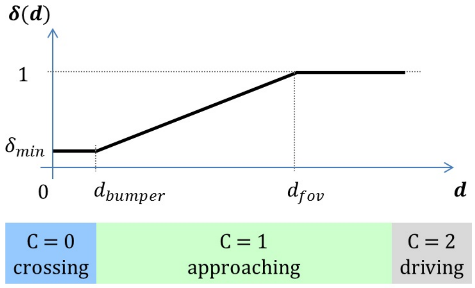


Figure 4.10.: *Value of δ depending on the distance to the start of the intersection d and the crossing state C. The crossing states are colored in blue (crossing), light-green (approaching) and gray (normal driving).*

The Table 4.1 shows the selected values of α_{de} and α_{in} for every state. These values have been selected and optimized empirically based on experimental results.

4.3. Scenario interpretation based on primary situations

The proposed approach aims to make the interpretation of the scenario (and further planning) easier by breaking it down into primary situations [24] [50] [51]. In order to give an overview of the concept, the relevant sub-modules in the main system flowchart are introduced. The next section describes how a scenario based on primary situations is defined. And finally, how the ego vehicle is guided to complete the desired maneuver using this concept is explained.

Table 4.1.: *Selected values of α_{de} and α_{in}.*

	α_{de}	α_{in}
Traffic lights		
Unknown	0.5	0.01
Off	0.5	0.01
Not permitted	0.1	0.5
Permitted	0.1	0.3
Permitted (limited)	0.1	0.3
Protected right turn	0.1	0.3
Protected left turn	0.1	0.3
Protected right (limited)	0.1	0.3
Protected left (limited)	0.1	0.3
Permitted right on red	0.1	0.3
Traffic signs		
No traffic sign	0.5	0.5
right-before-left	0.5	0.5
With precedence	0.5	0.5
Yield	0.5	0.5
Stop	0.5	0.5
Pass permission		
Unknown	0.9	0.5
Not permitted	0.5	0.9
Permitted	0.8	0.8
Permitted (limited)	0.8	0.8
Protected	0.8	0.8
Protected (limited)	0.8	0.8
Permitted right turn on red	0.5	0.5
right-before-left	0.5	0.5
With precedence	0.5	0.5
Yield	0.5	0.5
Stop	0.5	0.5

4.3.1. Overview

From a general point of view, the basic conceptual flowchart of a self driving system can be simplified in 4 submodules: perception, scenario interpretation, planning and control. This is illustrated in Fig. 2.2.

The perception module represents the low level processing of sensors and a priori data (e.g. image processing, object recognition and tracking, localization and mapping, etc.). The scenario interpretation, which is the focus of this thesis, corresponds with the understanding of the processed data. Then, the planning calculates the proper trajectory and delivers it to the control module, which finally provides the adequate signals in terms of steering and acceleration.

The basic idea, which is based on [24], consists in achieving a scenario representation using only the relevant information provided by the perception module. This interpretation should contain the essential information in order to make the proper decisions to guide the ego vehicle along the desired driving corridor. In this context, a scenario consists of mainly 3 important components: the current pass permission, which indicates how the ego vehicle should pass the intersection and under which conditions (see section 4.2); the intention of the ego vehicle, and accordingly its maneuver; and a set of primary situations linked along the driving corridor. Furthermore, it is assumed that some basic information, such as the ego motion, is well known.

The key issue is to use the classification shown in Fig. 4.2 to define a set of primary situations based on the possible conflicts between the ego vehicle and other road users. For this reason the concept on 4 different primary situations (and combinations of them) is used, so that the main advantage is that the whole maneuver can be broken down into a set of expected primary situations (see Fig. 4.11):

A: There is a potential conflict with a perpendicular with VRU lane (e.g. a crosswalk, zebra crossing or bike lane) in front of the ego vehicle.

B: The driving corridor of the ego vehicle intersects a left-cross lane (e.g. at a T-form intersection without right-crossing lanes). $B1$ is not considered a primary situation on its own, but a mirrored version of B, in which the cross lane comes from the right side. In addition, $B2$ corresponds to a combination of B and $B1$ (e.g. at a X-form intersection).

C: The ego vehicle has a conflict with a parallel crosswalk, zebra crossing or bike lane. Perpendicular and parallel conflicts with VRUs by turning at intersections have to be handled in a different manner compared to situation A. For example, at an intersection controlled with traffic lights, when the state is permitted, the ego vehicle has precedence with respect to the VRUs crossing a perpendicular crosswalk.

On the contrary, the ego vehicle has no precedence with respect to VRUs crossing a parallel crosswalk (this is explained graphically in Fig. 4.2).

D: The ego vehicle has a conflict with an oncoming vehicle.

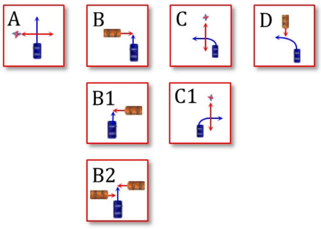


Figure 4.11.: *Set of primary situations (A, B, C and D) and the resulting combination of them (B1, B2, and C1).*

Every situation should contain at least the following information (these terms are described in detail hereinafter):

Observation area. It consists of a geometric area (as a 2 dimensional polygon) that has to be observed for every primary situation. It represents the area where relevant objects are expected. Namely, if an object is detected inside this area, it should be considered to predict a potential conflict with it.

Occupancy probability. This is a discrete function indicating the probability over time, that the primary situation is occupied. There are 2 types of occupancies: real and virtual. These are calculated considering real detected objects or virtual expected objects, respectively. The concept of virtual objects is explained in detail in the following paragraphs.

Critical area. This represents the area that is used for calculating the occupancy over time. In other words, the occupancy represents the probability that the critical area is occupied by other road users over time.

Distance to situation. It corresponds to the distance along the desired driving corridor between the front bumper of the ego vehicle and the start of the primary situation.

Type. This indicates the type of primary situation (see Fig. 4.11).

Angle. It indicates the angle between the driving corridor of the ego vehicle and the intersecting lane at the point where both intersect.

The process of extracting all this information from the perception module and creating every primary situation corresponds to the first relevant step of this approach. An overview of this process is described graphically in Fig. 4.12 as a simplified flowchart.

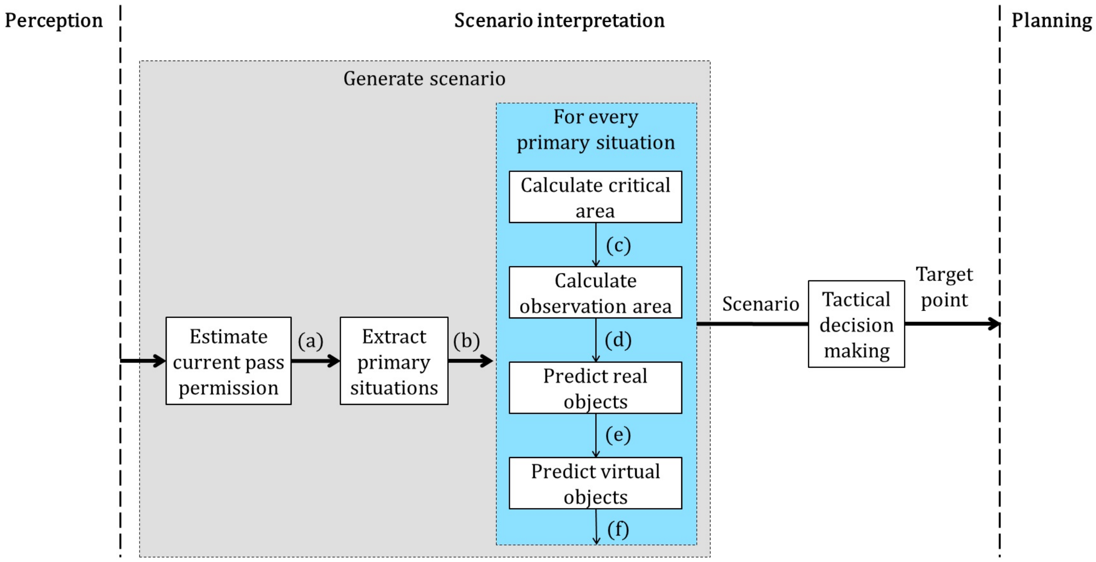


Figure 4.12.: *Simplified flowchart representing the most relevant steps of the proposed interpretation process. The scenario interpretation receives the input from the perception and provides its results in form of set of target points to the planning module. The generation of the scenario based on primary situations is inside a gray rectangle. The light-blue rectangle represents the steps done for every primary situation.*

In a nutshell, the outputs provided by the perception module are used to extract the relevant information and generate the scenario, which represents the input of the tactical decision making. This provides a target point as output indicating a desired position and velocity along the driving corridor of the ego vehicle. Namely, a target point represents *where* and *how fast* the ego vehicle should drive to complete the maneuver at the intersection, and correspondingly, is the result of the interpretation module. This is then used by the trajectory planner to calculate the optimal velocity profile to reach this target point.

4.3.2. Generating the scenario based on primary situations

The result of estimating the pass permission (see (a) in Fig. 4.12) indicates which pass permission is currently valid for the ego vehicle, i.e. under which conditions the ego vehicle is allowed to pass the intersection (denied, permitted, protected, etc.). Once this is estimated, the next relevant step of the proposed approach is to extract the set of primary situations along the ego driving corridor.

Extracting the primary situations consists of calculating the possible conflicts between the ego vehicle and other road users. This can be automatically calculated considering the road network information (i.e. assessing the intersection points between the path of the ego vehicle and other intersecting paths). Then, every primary situation is linked representing the order in which consecutive single primary situations are expected, so that a scenario S denotes the connections of primary situations PS_i:

$$S = \{PS_1, PS_2, ..., PS_M\}, \tag{4.16}$$

where M is the number of different situations. In other words, M represents the total number of primary situations along the desired path. This is graphically described in Fig. 4.13 using a simple example.

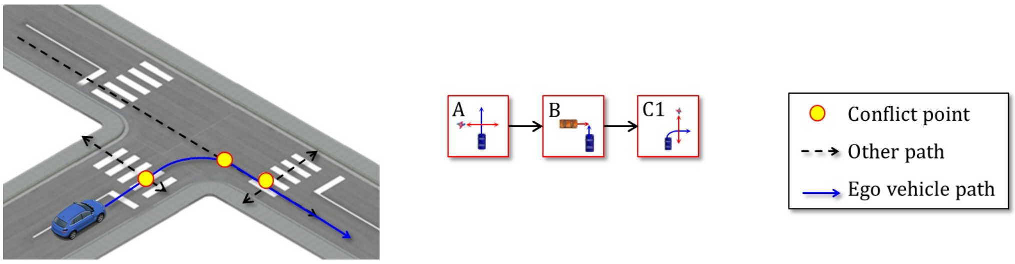


Figure 4.13.: *Simple example of the extraction of 3 primary situations (A, B, C1). The ego vehicle approaches an T-intersection. Its desired path by turning to the right intersects with a zebra crossing (A), a left cross lane (B) and a parallel zebra crossing (C1). The conflict points between the path of the ego vehicle (solid blue arrow) and the path of other road users (dotted black arrows) are indicated with a small yellow rectangle.*

In the given example, the ego vehicle is turning to the right. Using the road network information, the conflicting points with the path of other road users are calculated. In fact, the only necessary information is the geometry of the paths and the type of intersecting lane (crossing vehicle lane or vulnerable road user such as crosswalk, bike lanes, etc.). In short, the distance from the bumper of the ego vehicle to the situation, the angle between the conflicting path and the type of situation is extracted from the a-priori road network. This information corresponds to (b) in Fig. 4.12.

Once the set of primary situations is extracted, the next step of the flowchart is done for every situation (see the blue rectangle in Fig. 4.12). First, the critical area is calculated based on static geometric information of the a priori road network. This area is in fact divided into 3 sub-areas (S_1, S_2 and S_3)) and every form depends on the type of the intersecting lane.

For vehicles, the first critical sub-area is a polygon of P points $S_1 = \{\overrightarrow{s_{11}}, ..., \overrightarrow{s_{1P}}\}$ that represents the overlapping area (i.e. where the lanes of ego and the other vehicles overlap) along the intersecting lane with the length d_1. This is illustrated in Fig. 4.14 with a red rectangle. The second critical sub-area S_2 indicates the area from the overlapping area to the start of the intersecting lane (see the intersecting lane colored in blue in Fig. 4.14 with the length d_2). The third critical sub-area S_3 is calculated with an empirical length d_3 and indicates the area before the other objects drive into the intersection.

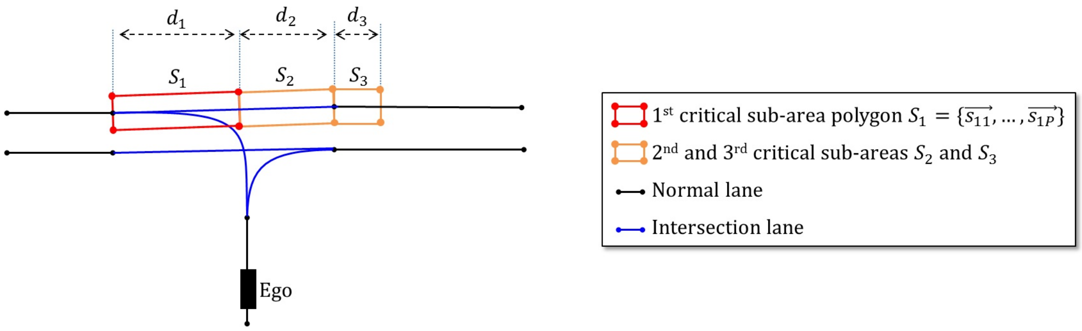

d_1
d_2
d_3
S_1
S_2
S_3
Ego
1st critical sub-area polygon $S_1 = \{\overrightarrow{s_{11}}, \dots, \overrightarrow{s_{1P}}\}$
2nd and 3rd critical sub-areas S_2 and S_3
Normal lane
Intersection lane

Figure 4.14.: *Graphical explanation of the critical sub-areas for vehicles based on an example. The ego vehicle (black rectangle) intends to turn to the left, so that its path intersects with a crossing lane from the right.*

In case of an intersecting lane for pedestrians (i.e. crosswalk or zebra crossing), the first critical sub-area polygon $S_1 = \{\overrightarrow{s_{11}}, ..., \overrightarrow{s_{1P}}\}$ represents the area that the ego vehicle would drive over the pedestrian lane (see red polygon in Fig. 4.15).

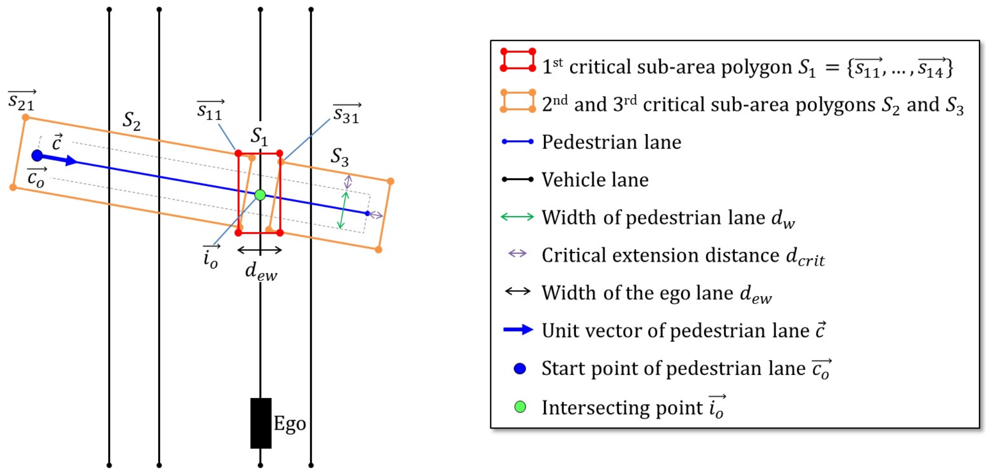

$\overrightarrow{s_{21}}$
S_2
$\overrightarrow{s_{11}}$
S_1
$\overrightarrow{s_{31}}$
S_3
$\vec{c}$
$\overrightarrow{c_o}$
$\overrightarrow{i_o}$
d_{ew}
Ego
1st critical sub-area polygon $S_1 = \{\overrightarrow{s_{11}}, \dots, \overrightarrow{s_{14}}\}$
2nd and 3rd critical sub-area polygons S_2 and S_3
Pedestrian lane
Vehicle lane
Width of pedestrian lane d_w
Critical extension distance d_{crit}
Width of the ego lane d_{ew}
Unit vector of pedestrian lane $\vec{c}$
Start point of pedestrian lane $\overrightarrow{c_o}$
Intersecting point $\overrightarrow{i_o}$

Figure 4.15.: *Example of a primary situation for pedestrians (crosswalks or zebra crossing) with a graphical explanation of how the critical sub-areas are generated. The first point of every polygon is marked at the upper-left corner. The legend at the right side defines all used symbols in the illustration.*

Since it is assumed that every pedestrian lane is straight, i.e. has no curvature, these critical sub-areas for pedestrians can be described as rectangles ($P = 4$). The second and third critical sub-areas ($S_2 = \{\overrightarrow{s_{21}}, ..., \overrightarrow{s_{2P}}\}$ and $S_3 = \{\overrightarrow{s_{31}}, ..., \overrightarrow{s_{3P}}\}$) are calculated using the distance d_{crit}. This distance, which is set empirically, represents a constant extension of the areas:

$$\begin{aligned}
\overrightarrow{s_{21}} &= \overrightarrow{c_o} - d_{crit} \cdot \overrightarrow{c} + (0.5 \cdot d_w + d_{crit}) \cdot \overrightarrow{c_\perp} \\
\overrightarrow{s_{22}} &= \overrightarrow{i_o} - (0.5 \cdot d_{ew}) \cdot \overrightarrow{c} + (0.5 \cdot d_w + d_{crit}) \cdot \overrightarrow{c_\perp} \\
\overrightarrow{s_{23}} &= \overrightarrow{s_{22}} - (d_w + 2 \cdot d_{crit}) \cdot \overrightarrow{c_\perp} \\
\overrightarrow{s_{24}} &= \overrightarrow{s_{21}} - (d_w + 2 \cdot d_{crit}) \cdot \overrightarrow{c_\perp}
\end{aligned} \tag{4.17}$$

and

$$\begin{aligned}
\overrightarrow{s_{31}} &= \overrightarrow{i_o} + 0.5 \cdot d_{ew} \cdot \overrightarrow{c} + (0.5 \cdot d_w + d_{crit}) \cdot \overrightarrow{c_\perp} \\
\overrightarrow{s_{32}} &= \overrightarrow{c_o} + (d_l + d_{crit}) \cdot \overrightarrow{c} + (0.5 \cdot d_w + d_{crit}) \cdot \overrightarrow{c_\perp} \\
\overrightarrow{s_{33}} &= \overrightarrow{s_{32}} - (d_w + 2 \cdot d_{crit}) \cdot \overrightarrow{c_\perp} \\
\overrightarrow{s_{34}} &= \overrightarrow{s_{31}} - (d_w + 2 \cdot d_{crit}) \cdot \overrightarrow{c_\perp}
\end{aligned} \tag{4.18}$$

where the variables d_w and d_l indicate the width and length of the pedestrian lane, respectively. The distance d_{ew} corresponds to the width of the ego lane. In other words, the first critical sub-area represents the overlapping area along the ego lane, while the second and third one represent the pedestrian lane (extended with the distance d_{crit}) at the left and right side of the ego vehicle, respectively.

On the other hand, another important concept of the primary situation is the observation area, which is a polygon with Q points describing the area where relevant objects could be ($O = \{\overrightarrow{o_1}, ..., \overrightarrow{o_Q}\}$). This is calculated depending on the type of the intersecting lane and the expected time that the ego vehicle needs to reach the situation t_{area}.

In the case of a primary situation for vehicles, the area is calculated along the path of other objects: between the end of the intersection lane and the calculated distance d_{obs} (see the green polygon in Fig. 4.16):

$$d_{obs} = \begin{cases} d_{min} & \text{if } v_{obj} \cdot t_{area} < d_{min} \\ d_{max} & \text{if } v_{obj} \cdot t_{area} > d_{max} \\ v_{obj} \cdot t_{area} & \text{else} \end{cases} \tag{4.19}$$

where d_{min} and d_{max} are constrains set empirically that indicate the minimal and maximal distance of the observation area, respectively. v_{obj} represents the maximal expected velocity of a possible object and t_{area} is the time that the ego vehicle needs to reach the situation:

$$t_{area} = \frac{-v_{ego}\sqrt{v_{ego}^2 + a_{ego} \cdot d}}{a_{ego}}. \tag{4.20}$$

To calculate t_{area} the distance from the front bumper of the ego vehicle to the situation d, the current ego velocity v_{ego} and acceleration a_{ego} are used. This results in a polygon of Q points describing the observation area ($O = \{\overrightarrow{o_1}, ..., \overrightarrow{o_Q}\}$), where the width corresponds with the width of the lane in the driving corridor of the other vehicles.

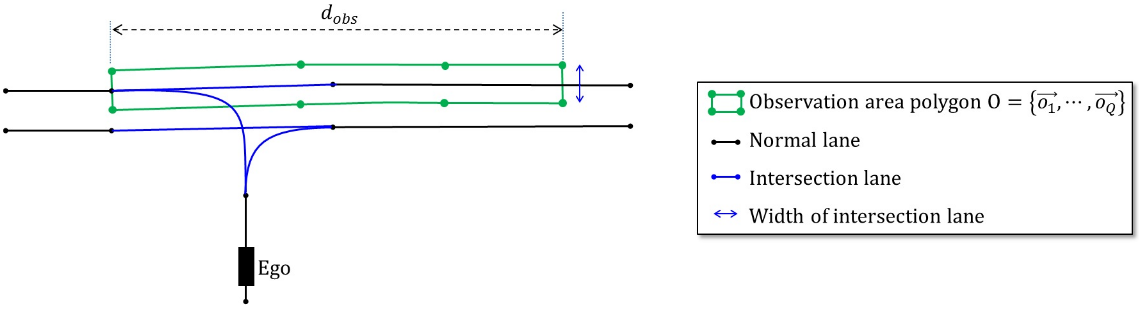

d_{obs}
Ego
Observation area polygon $O = \{\overrightarrow{o_1}, \cdots, \overrightarrow{o_Q}\}$
Normal lane
Intersection lane
Width of intersection lane

Figure 4.16.: *Example of an observation area for a primary situation of crossing vehicles. The ego vehicle (black rectangle) intends to turn to the left at an T-intersection. Its path intersects with a right cross lane, so that the observation area is generated.*

In contrast to the the situations for vehicles, the observation area for pedestrians is calculated based on the basic form described in Fig. 4.17.

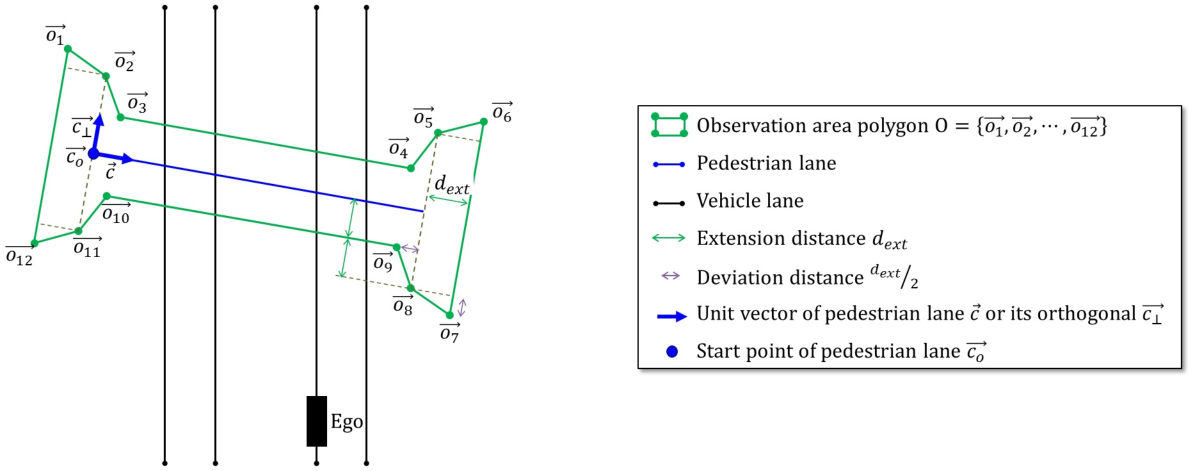

$\overrightarrow{o_1}$
$\overrightarrow{o_2}$
$\overrightarrow{o_3}$
$\overrightarrow{c_\perp}$
$\overrightarrow{c_o}$
$\vec{c}$
$\overrightarrow{o_{10}}$
$\overrightarrow{o_{11}}$
$\overrightarrow{o_{12}}$
$\overrightarrow{o_4}$
$\overrightarrow{o_5}$
$\overrightarrow{o_6}$
d_{ext}
$\overrightarrow{o_9}$
$\overrightarrow{o_8}$
$\overrightarrow{o_7}$
Ego
Observation area polygon $O = \{\overrightarrow{o_1}, \overrightarrow{o_2}, \cdots, \overrightarrow{o_{12}}\}$
Pedestrian lane
Vehicle lane
Extension distance d_{ext}
Deviation distance $^{d_{ext}}/_2$
Unit vector of pedestrian lane $\vec{c}$ or its orthogonal $\overrightarrow{c_\perp}$
Start point of pedestrian lane $\overrightarrow{c_o}$

Figure 4.17.: *Example of an observation area for VRUs. The ego vehicle (black rectangle) approaches a primary situation for VRUs (crosswalk or zebra crossing represented by a blue line), so that the corresponding observation is generated.*

This geometry aims to cover the area where pedestrians could be relevant for the situation. For this reason, the observation area is expressed as a polygon of twelve points $O = \{\overrightarrow{o_1}, \overrightarrow{o_2}, ..., \overrightarrow{o_{12}}\}$ calculated depending on the unit vector of the pedestrian lane $\overrightarrow{c}$, its origin $\overrightarrow{c_o}$, its length d_{length} and the extension distance d_{ext}:

$$
\begin{aligned}
\overrightarrow{o_1} &= \overrightarrow{c_o} - d_{ext} \cdot \overrightarrow{c} + 2.5 \cdot d_{ext} \cdot \overrightarrow{c_\perp} \\
\overrightarrow{o_2} &= \overrightarrow{o_1} + d_{ext} \cdot \overrightarrow{c} - 0.5 \cdot d_{ext} \cdot \overrightarrow{c_\perp} \\
\overrightarrow{o_3} &= \overrightarrow{o_2} - d_{ext} \cdot \overrightarrow{c_\perp} + 0.5 \cdot d_{ext} \cdot \overrightarrow{c} \\
\overrightarrow{o_4} &= \overrightarrow{o_3} + d_{length} \cdot \overrightarrow{c} - d_{ext} \cdot \overrightarrow{c} \\
\overrightarrow{o_5} &= \overrightarrow{o_4} + d_{ext} \cdot \overrightarrow{c_\perp} + 0.5 \cdot d_{ext} \cdot \overrightarrow{c} \\
\overrightarrow{o_6} &= \overrightarrow{o_5} + d_{ext} \cdot \overrightarrow{c} + 0.5 \cdot d_{ext} \cdot \overrightarrow{c_\perp} \\
\overrightarrow{o_7} &= \overrightarrow{o_6} - 5 \cdot d_{ext} \cdot \overrightarrow{c_\perp} \\
\overrightarrow{o_8} &= \overrightarrow{o_7} - d_{ext} \cdot \overrightarrow{c} + 0.5 \cdot d_{ext} \cdot \overrightarrow{c_\perp} \\
\overrightarrow{o_9} &= \overrightarrow{o_8} + d_{ext} \cdot \overrightarrow{c_\perp} - 0.5 \cdot d_{ext} \cdot \overrightarrow{c} \\
\overrightarrow{o_{10}} &= \overrightarrow{o_9} + (d_{ext} - d_{length}) \cdot \overrightarrow{c} \\
\overrightarrow{o_{11}} &= \overrightarrow{o_{10}} - d_{ext} \cdot \overrightarrow{c_\perp} - 0.5 \cdot d_{ext} \cdot \overrightarrow{c} \\
\overrightarrow{o_{12}} &= \overrightarrow{o_{11}} - d_{ext} \cdot \overrightarrow{c} - 0.5 \cdot d_{ext} \cdot \overrightarrow{c_\perp}
\end{aligned}
\tag{4.21}
$$

where $\overrightarrow{c}$ indicates the unit vector of the pedestrian lane and $\overrightarrow{c_\perp}$ its orthogonal vector. The larger the time is that ego needs to reach the situation, the larger should the extension of the observation area be. In order to consider this dependency, the distance d_{ext} depends on t_{area}:

$$
d_{ext} = \begin{cases} d_{maxExt} & \text{if } t > t_{area} \\ t_{area} \cdot \frac{d_{maxExt} - d_{minExt}}{t_{max}} + d_{minExt} & \text{if } 0 < t < t_{area} \\ d_{minExt} & \text{if } t < 0 \end{cases}
\tag{4.22}
$$

The variables d_{minExt} and d_{maxExt} indicate the minimal and maximal extension distance, respectively. t_{max} represents the maximal considered time. This dependency is illustrated in Fig. 4.18.

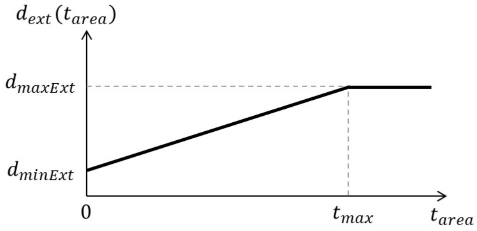


Figure 4.18.: *The extension distance d_{ext} depending on the time that ego needs to reach the area t_{area}. The value of d_{ext} increases linearly from $t_{area} = 0$ to t_{max} between d_{minExt} and d_{maxExt}.*

The scenario with the extracted critical and observation area corresponds to (c) and (d) in the flowchart of Fig. 4.12, respectively. Using this information, the next step is the motion prediction of the objects detected inside the observation area. This estimation enables the calculation of the probability that the situation is occupied over time (e).

Once this information is extracted for every situation (see blue colored rectangle in Fig. 4.12) the scenario generated completely, so that the next step is to plan the maneuver by generating the corresponding target points.

4.4. Objects motion prediction

This section aims to describe the proposed object motion prediction used in this thesis. According to the concept explained in 4.3, the motion prediction of objects consists of estimating the probability that every primary situation is occupied due to relevant objects. That is to say, this information is part of every primary situation. Therefore, the main goal is to consider every relevant object inside the corresponding observation area (see Fig. 4.16 and 4.17) and calculate the probability that the situation is going to be occupied. This is called the *occupancy probability*, which corresponds to (e) in Fig. 4.12. In other words, the output of the object motion prediction for every situation is a discrete function representing how occupied the situation could be over time.

Although the format of the delivered information of this module is the same, the way it is calculated depends on the type of situation (and thereby on the type of object). In this thesis, only 2 types of motion prediction have been achieved as a proof of concept. First, the motion prediction approach for vehicles is explained in detail. Then, the next subsection explains how the occupancy probability is calculated for pedestrians based on a feasible motion prediction approach.

4.4.1. Vehicles motion prediction

In case that a crossing vehicle is detected, the main objective is to estimate where it could be over time. The key question is: if the object is driving with constant velocity, when would the object be inside the critical area? Or to put it another way, how probable is it, that the object is inside the critical area for every discrete time t? This has to be done for every object in the observation area. To achieve this, the position and dynamic of every detected object i is considered to assess the occupancy probability for every discrete time t (i.e. the function depends on the time and the object index: $P_{occ}(t,i)$). Then, for I objects, the maximal occupancy probability value of every object i is considered, so that the occupancy probability is finally represented as a function over time t. Namely, the occupancy probability for the time t corresponds to the maximal probability of all objects detected inside the observation area at this time:

$$P_{occ}(t) = \max_{i=\{1,\cdots,I\}} (P_{occ}(t,i)). \tag{4.23}$$

The Fig. 4.19 illustrates an example with a primary situation in which a crossing vehicle has been detected inside the observation area.

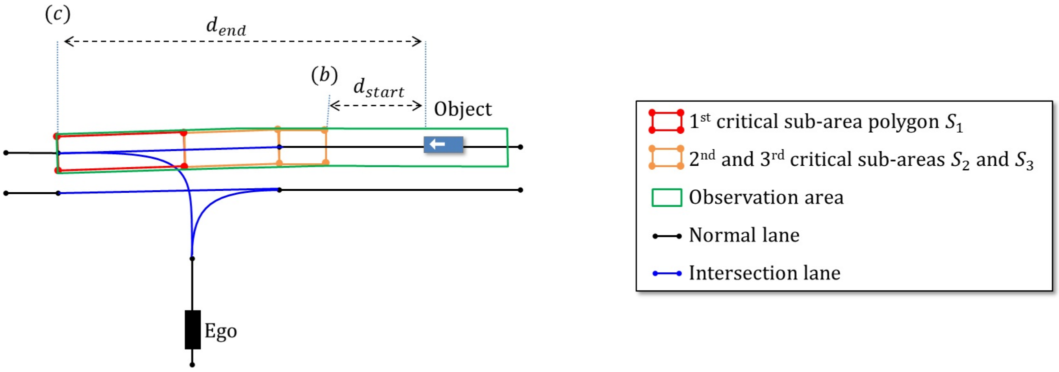

(c)
d_{end}
(b)
d_{start}
Object
Ego
1st critical sub-area polygon S_1
2nd and 3rd critical sub-areas S_2 and S_3
Observation area
Normal lane
Intersection lane

Figure 4.19.: *Graphical explanation of the calculation of the occupancy probability $P_{occ}(t,i)$ for a given example. The ego vehicle (black rectangle) intends to turn left at a T-intersection and a cross vehicle i (blue rectangle with white arrow indicating the driving direction) is detected inside the observation area (colored in green).*

Fig. 4.20 shows a plot of an occupancy probability for a single object.

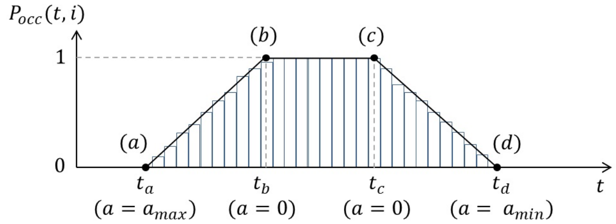


Figure 4.20.: *Example of a plotted occupancy probability over time $P_{occ}(t, i)$ for a given vehicle i. The indicated point (b) and (c) are also illustrated in Fig. 4.19.*

Since the intention of every object is unknown (i.e. it is not known if the object is about to brake, accelerate...), some coherent assumptions are considered:

- The occupancy probability for the object i at the time that the object would reach the critical area driving with constant velocity is 1 (i.e. $P_{occ}(t_b, i) = 1$). See the marked point (b) in Fig. 4.20.

- The occupancy probability should be 1 if it is predicted that any object is inside the critical area. This corresponds to plotted function between the points (b) and (c) in Fig. 4.19 and 4.20.

- Since it is unlikely that any object accelerates with a given acceleration a_{max}, the occupancy probability is minimal for the time that the object needs to reach the critical area with a positive acceleration a_{max} (i.e. $P_{occ}(t_a, i) = 0.0$). This is the marked point (a).

- The difference (in time) between the marked point (a) and (b) is used to set the time $t_d = t_c + |t_b - t_a|$ (i.e. the object would brake and reach the critical area later than driving with constant velocity). At this time it is considered unlikely that the object reaches the critical area ($P_{occ} = 0.0$).

In a nutshell, it is considered very probable that the current vehicle i does not modify its velocity and improbable that it accelerates with an acceleration a_{max}. Accordingly, the occupancy probability changes linearly between these assumed values (see Fig. 4.20):

$$P_{occ}(t,i) = \begin{cases} 0 & \text{if } t < t_a \\ \frac{-t_a}{t_b - t_a} \cdot t & \text{if } t_a < t < t_b \\ 1 & \text{if } t_b < t < t_c \\ \frac{-t_c}{t_d - t_c} \cdot t + 1 & \text{if } t_c < t < t_d \\ 0 & \text{if } t > t_d. \end{cases} \tag{4.24}$$

Where t_a represents the time that the object would need to reach the critical area (i.e. to drive the distance d_{start}) with its current velocity v_{obj} and an improbable acceleration a_{max}:

$$t_a = \frac{-v_{obj} \pm \sqrt{v_{obj}^2 \cdot 2 \cdot a_{max} \cdot d_{start}}}{a_{max}}, \tag{4.25}$$

t_b indicates the time that the object would need to drive the distance to critical area d_{start} with constant velocity v_{obj}:

$$t_b = \frac{d_{start}}{v_{obj}}, \tag{4.26}$$

t_c is the the time that the object would need to drive the distance to the end of the critical area d_{end} with constant velocity v_{obj}:

$$t_c = \frac{d_{end}}{v_{obj}}, \tag{4.27}$$

and t_d corresponds to:

$$t_d = t_c + |t_b - t_a|. \tag{4.28}$$

In this manner the probability over time is calculated for every object i inside the observation area. Then, the maximal probability is considered. This is illustrated with an example in Fig. 4.21 and 4.22. In this given example, 2 objects have been detected inside the observation area (i.e. $I = 2$), so that an occupancy probability for every object is calculated ($P_{occ}(t, i = 1)$ and $P_{occ}(t, i = 2)$). Then, the resulting occupancy probability $P_{occ}(t)$ is calculated for the current primary situation as the maximal value of every object over time.

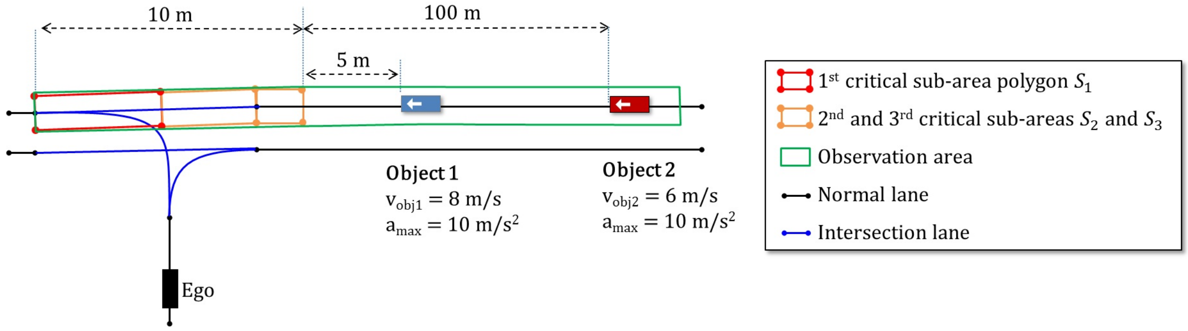


Figure 4.21.: *Example explaining the calculation of the occupancy probability for 2 detected crossing vehicles (blue and red rectangle). The ego vehicle intends to turn left at a T-intersection.*

The first vehicle, colored in blue, is approaching the intersection with a current velocity of 8 m/s and the distance to the critical area is 5 m. The second vehicle, colored in red, drives with 6 m/s and its distance to the critical area is 100 m. The maximal acceleration a_{max} used to calculate the occupancy probability is 10 m/s^2 (see equation 4.25). The resulting $P_{occ}(t)$ is plotted in Fig. 4.22.

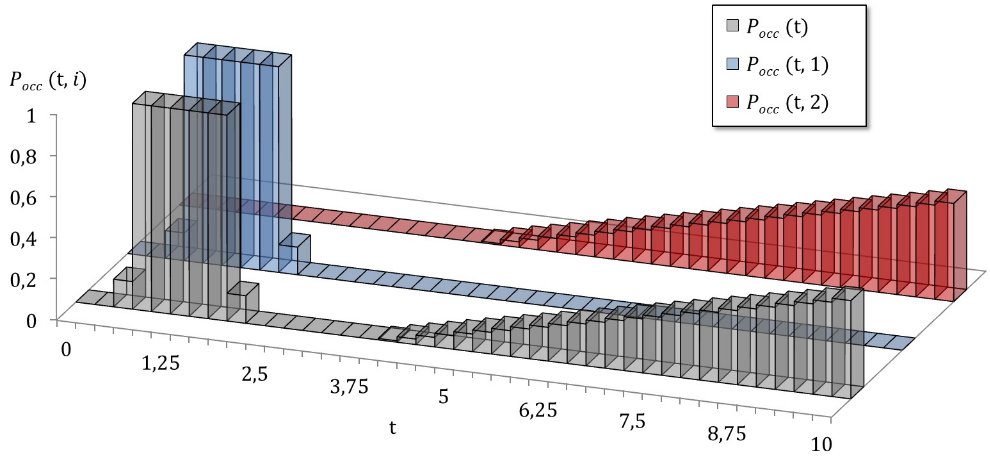


Figure 4.22.: *Plot of the occupancy probability for the given example in Fig. 4.21. The occupancy probability for both detected objects $P_{occ}(t, 1)$ and $P_{occ}(t, 2)$ result in the corresponding occupancy probability $P_{occ}(t)$ (gray plot).*

The calculated $P_{occ}(t)$ for the given situation is colored in gray. The occupancy probability for the first and second vehicle is colored in blue and red, respectively.

4.4.2. Pedestrians motion prediction

The objective is the same as for crossing vehicles: to estimate how probable it is, that the primary situation is occupied over time due to objects inside the observation area.

Although the geometric description of the pedestrian lane is known from the a priori road network information, pedestrians do not walk strictly over a given lane. Therefore, their motion prediction requires a different approach than assuming a fixed driving corridor, such as for crossing vehicles. Namely, it is necessary to estimate the area that could be occupied by every pedestrian and then compare it with the corresponding critical area.

To achieve the estimation of the future occupied area of a given object (i.e. where the pedestrian could be over time), the proposed approach in this thesis simplifies other related works such us [52] and [53] by using a polygon in form of a trapezoid (see [52]). The Fig. 4.23 illustrates this polygon.

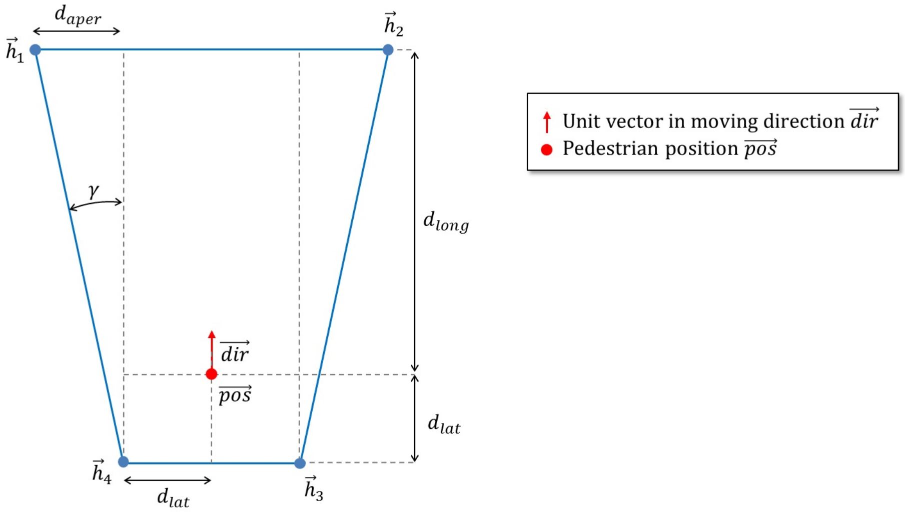


Figure 4.23.: *Polygon representing the possible future occupied area by a given pedestrian. The position of the pedestrian and its moving direction is marked with a red point and arrow, respectively.*

The form of the polygon H is easily calculated using 3 variables: the longitudinal extension d_{long}, the lateral extension d_{lat} and the aperture angle γ:

$$H = \{\overrightarrow{h_1}, \overrightarrow{h_2}, \overrightarrow{h_3}, \overrightarrow{h_4}\}, \tag{4.29}$$

with:

$$
\begin{aligned}
\overrightarrow{h_1} &= \overrightarrow{pos} + (d_{lat} + d_{aper}) \cdot \overrightarrow{dir_\perp} + d_{long} \cdot \overrightarrow{dir} \\
\overrightarrow{h_2} &= \overrightarrow{pos} - (d_{lat} + d_{aper}) \cdot \overrightarrow{dir_\perp} + d_{long} \cdot \overrightarrow{dir} \\
\overrightarrow{h_3} &= \overrightarrow{pos} - d_{lat} \cdot \overrightarrow{dir_\perp} - d_{lat} \cdot \overrightarrow{dir} \\
\overrightarrow{h_4} &= \overrightarrow{pos} + d_{lat} \cdot \overrightarrow{dir_\perp} - d_{lat} \cdot \overrightarrow{dir}.
\end{aligned} \tag{4.30}
$$

Obviously, the form of the polygon is not constant. The basic idea is to generate this polygon depending on the pedestrian velocity v_{obj} and the time t. For this purpose, it is assumed that the faster the pedestrian walks, the larger should the longitudinal extension be. Besides, the larger the considered time is, the larger should the area that the pedestrian could occupy be. Therefore, the longitudinal extension is calculated as follows:

$$
d_{long} = \begin{cases} d_{longMin} & \text{if } t \cdot v_{obj} \cdot K_{long} < d_{longMin} \\ d_{longMax} & \text{if } t \cdot v_{obj} \cdot K_{long} > d_{longMax} \\ t \cdot v_{obj} \cdot K_{long} & \text{else} \end{cases}, \tag{4.31}
$$

where $d_{longMin}$ and $d_{longMax}$ indicate the minimal and maximal longitudinal extension and the constant K_{long} is a value that adapts the dependency of d_{long} on the considered time t and pedestrian velocity v_{obj}. This is graphically described in Fig. 4.24.

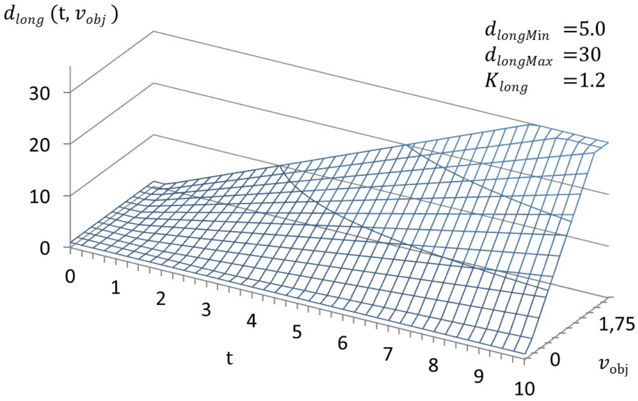


Figure 4.24.: *Plot of the longitudinal extension d_{long} depending on the pedestrian velocity v_{obj} and time t.*

In a similar way, the variable d_{lat} describes the lateral extension of the polygon depending on the pedestrian velocity v_{obj}, the time t and the constant K_{lat} (see Fig. 4.25):

$$d_{lat} = \begin{cases} d_{latMin} & \text{if } t \cdot v_{obj} \cdot K_{lat} < d_{latMin} \\ d_{latMax} & \text{if } t \cdot v_{obj} \cdot K_{lat} > d_{latMax} \\ t \cdot v_{obj} \cdot K_{lat} & \text{else} \end{cases} . \tag{4.32}$$

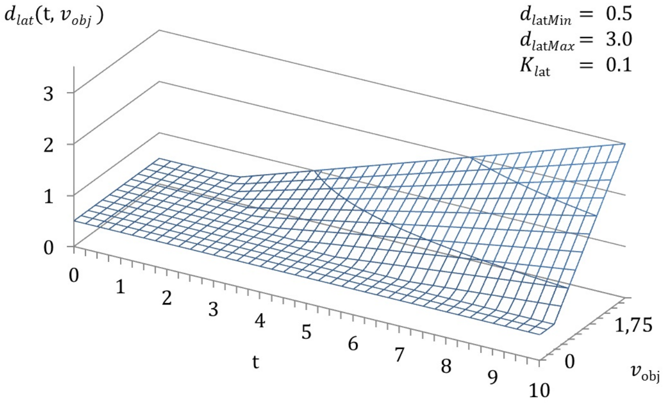


Figure 4.25.: *Plot of the lateral extension d_{lat} depending on the pedestrian velocity v_{obj} and time t.*

In a nutshell, the longitudinal and lateral extension of the polygon H are directly proportional to v_{obj} and t with the constant slope factors K_{long} and K_{lat}, respectively (see Eq. 4.31 and Eq. 4.32).

Furthermore, the faster the pedestrian moves, the more unlikely it is that the object changes its moving direction. In other words, if the pedestrian is walking very fast, it is assumed that sudden changes of its moving direction are less probable than in the case that the pedestrian walks slowly. Based on this assumptions, the aperture extension of the polygon d_{aper} is calculated as follows (see Fig. 4.26):

$$d_{aper} = (d_{long} + d_{lat}) \cdot tan(\gamma), \tag{4.33}$$

where γ is indirectly proportional to the velocity of the pedestrian v_{obj} and limited by a minimal and maximal aperture angle (γ_{min} and γ_{max}) (see Fig. 4.27):

$$\gamma = \begin{cases} \gamma_{min} & \text{if } t \cdot v_{obj} \cdot K_{lat} < d_{latMin} \\ \gamma_{max} & \text{if } t \cdot v_{obj} \cdot K_{lat} > d_{latMax} \\ \frac{(\gamma_{min} - \gamma max)}{v_{max}} \cdot v_{obj} + \gamma_{max} & \text{else} \end{cases} \tag{4.34}$$

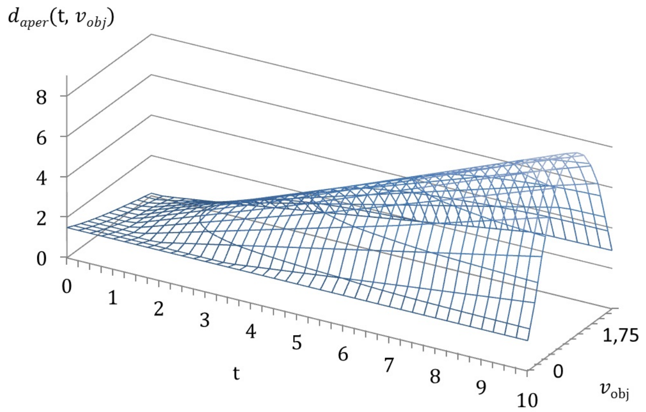


Figure 4.26.: *Plot of the aperture extension d_{aper} depending on the pedestrian velocity v_{obj} and time t.*

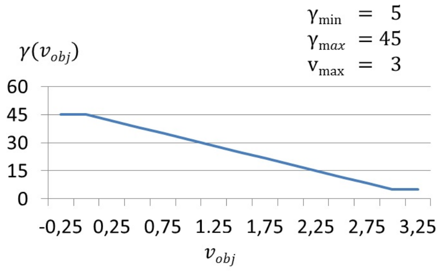


Figure 4.27.: *Plot of the aperture angle γ depending on the pedestrian velocity v_{obj}.*

In this manner, the trapezoid aims to represent the probable occupied area, so that the aperture angle γ and its corresponding extension d_{aper} consider typical changes on the walking direction. Nevertheless, it is necessary to consider also non-typical movements such as sudden changes of the walking direction. For this purpose, not only 1 polygon is calculated using the walking direction, but a second polygon representing the worst-case direction, too. In fact, the worst case polygon W is a copy of H but rotated to the intersecting point of the ego driving corridor and the pedestrian lane. This is graphically explained in Fig. 4.28.

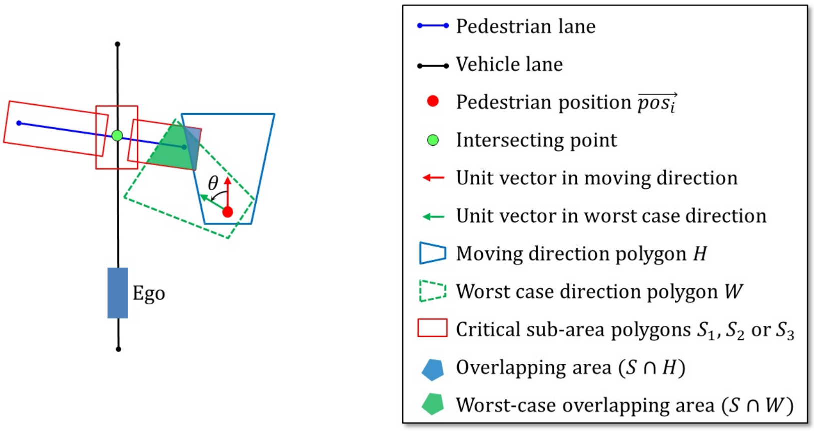

θ
Ego
Pedestrian lane
Vehicle lane
Pedestrian position $\overrightarrow{pos_i}$
Intersecting point
Unit vector in moving direction
Unit vector in worst case direction
Moving direction polygon H
Worst case direction polygon W
Critical sub-area polygons S_1, S_2 or S_3
Overlapping area $(S \cap H)$
Worst-case overlapping area $(S \cap W)$

Figure 4.28.: *Example of overlapping areas between pedestrian polygons (H and W) and critical subareas (S). The ego vehicle (blue rectangle) approaches a crosswalk of zebra crossing and a detected VRU (red point) walks at the right side of it. The walking and worst case directions are represented with a red and green arrow, respectively.*

Since the desired occupancy probability is represented as a function over discrete time (t), both polygons H and W are calculated for every pedestrian inside the observation area and for every time (t). Then, the overlapping areas between the critical sub-areas (S) and both calculated pedestrian polygons (H and W) are used to assess the occupancy probability of the pedestrian i:

$$P_{occ}(t,i) = \frac{\alpha \cdot (S \cap H) + \beta \cdot (S \cap W)}{S} \tag{4.35}$$

where α and β are weightings factors of the normal and worst-case polygons, correspondingly:

$$\begin{aligned} \alpha &= 1 - \beta \\ \beta &= |1 - \frac{|\theta|}{180}|. \end{aligned} \tag{4.36}$$

The influence of both overlapping areas ($S \cap H$ and $S \cap W$) are weighted depending on θ, which is the angle difference between the pedestrian moving and worst-case direction (see the red arrow, the green arrow and the angle θ in Fig. 4.28). In other words, if the pedestrian is walking straight in the opposite direction to the intersecting point $\overrightarrow{i_o}$, i.e. $\theta = 180\,°$, the weighting factor β is 0.0, so that the worst case polygon is not considered. In case that the difference angle θ is $90\,°$, both weighting factors α and β are 0.5.

So far, the calculation of the occupancy probability has been detailed only for 1 critical sub-area. But in fact, every elementary situation for pedestrians contains 3 critical sub-areas, are used to set the occupancy to 1.0 if the object is inside these areas in a logical manner: considering the pedestrian position and moving direction. This dependency is based on the following logical principles:

- If the pedestrian is inside the first critical sub-area S_1 (i.e. the area that the ego vehicle would overdrive), the occupancy probability is set automatically to 1.0.

- If the pedestrian is inside S_2 or S_3 (i.e. at the left or right side of the ego vehicle, respectively) and the pedestrian is getting closer to the intersecting point i_o, the occupancy probability is set to 1.0).

In this regard, the occupancy probability for every time (t) and every pedestrian (i) is calculated as follows:

$$P_{occ}(t,i) = \begin{cases} 1 & \text{if } pos \text{ is inside } S_1 \\ 1 & \text{else if } \overrightarrow{pos_i} \text{ is inside } S_2 \wedge_i \phi < 0 \\ 1 & \text{else if } \overrightarrow{pos_i} \text{ is inside } S_3 \wedge_i \phi > 0 \\ \frac{\alpha \cdot (S \cap H) + \beta \cdot (S \cap W)}{S} & \text{else,} \end{cases} \tag{4.37}$$

where ϕ_i is the difference of the angle between the ego lane vector and the pedestrian direction:

$$\phi_i = \angle(\overrightarrow{e}, \overrightarrow{pos_i}). \tag{4.38}$$

To clarify this concept graphically, the Fig. 4.29 illustrates an example of a pedestrian crossing from left to right ($\phi_i < 0.0$). As it can be seen in (1), the pedestrian is outside any critical sub-area. In (2) the pedestrian is inside the sub-area S_2 and getting closer to the intersecting point, so that the occupancy probability is set to 1.0. In (3) the occupancy probability is set to 1.0 independently to the moving direction because *pos* is inside S_1. On the contrary, although the pedestrian in (4) is inside the area S_3, the occupancy is not set to 1.0 because the pedestrian is getting away from the intersecting point.

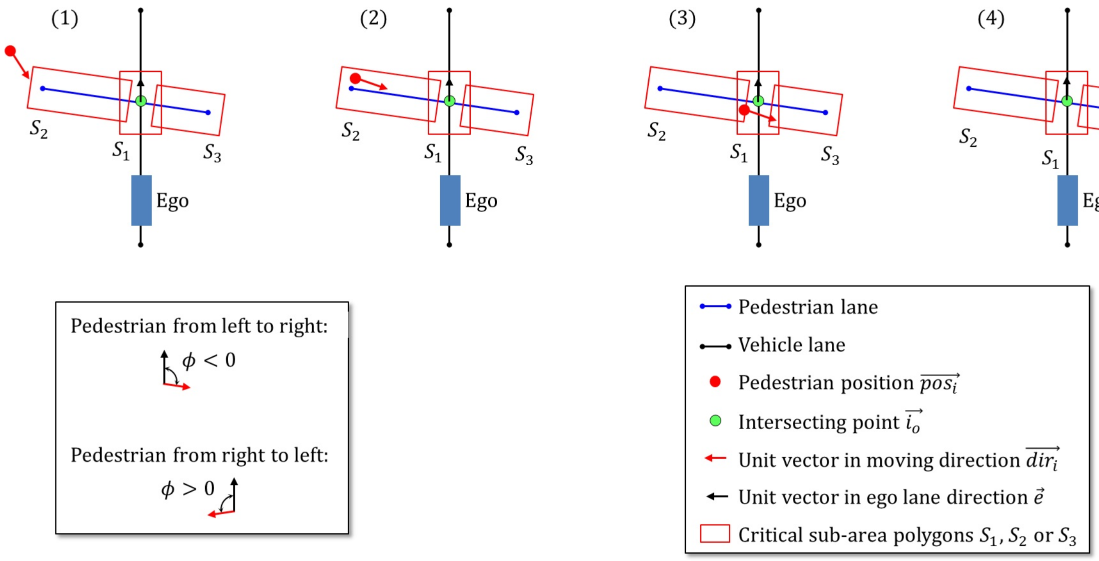

(1)
(2)
(3)
(4)
S_2
S_1
S_3
Ego
Pedestrian from left to right:
$\phi < 0$
Pedestrian from right to left:
$\phi > 0$
Pedestrian lane
Vehicle lane
Pedestrian position $\overrightarrow{pos_i}$
Intersecting point $\vec{i_o}$
Unit vector in moving direction $\overrightarrow{dir_i}$
Unit vector in ego lane direction $\vec{e}$
Critical sub-area polygons S_1, S_2 or S_3

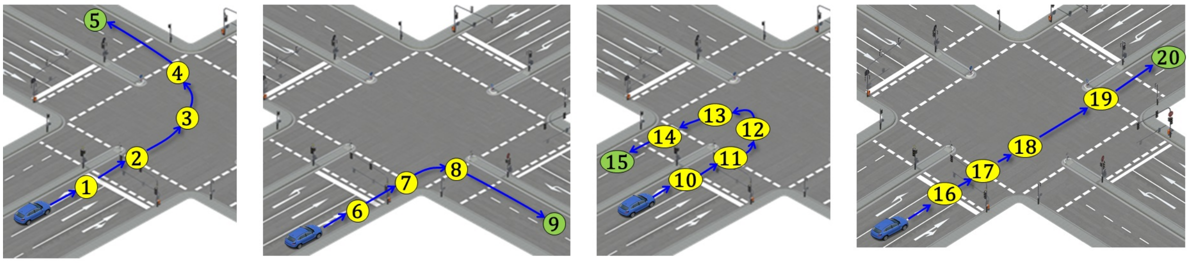
Figure 4.29.: *Example of a pedestrian (red point) crossing a crosswalk (blue line) from left to right. (1), (2), (3) and (4) represent the scenario in consecutive times.*

Similar to the occupancy probability calculation of crossing vehicles, the $P_{occ}(t, i)$ is calculated for every pedestrian i inside the observation area, and consequently, the total occupancy probability corresponds to the maximum of every pedestrian (see to Eq. 4.23 and the graphical explanation in Fig. 4.22 for the case of crossing vehicles).

4.5. Tactical decision making

Once the scenario is completely generated and the required information is extracted, the next step in the scenario interpretation process is to set the appropriate target points (see Fig. 4.12), which corresponds to the tactical decision making of the proposed approach. In other words, according to the available information (i.e. to the generated scenario as successive primary situations), the ego vehicle has to be guided along the desired path in order to execute the whole maneuver properly.

An important advantage of the proposed approach is that the required information for every primary situation is extracted and suitable for making a tactical decision in a systematic manner. For this reason the tactical decision making is designed as a state machine algorithm. To explain this concept, let's assume a non-specific intersection example, in which the position of the target points corresponds with the beginning of every primary situation. These positions are illustrated in Fig. 4.30 for different maneuvers.



(a) (b) (c) (d)

Figure 4.30.: *Example: Generic position of target points for different maneuvers: (a) left-turn, (b) right-turn, (c) U-turn and (d) forward. The targets point are illustrated with a yellow circle and its corresponding index. On the contrary, the final target point, which represents the end of the desired maneuver is colored in green.*

Although the tactical decision making for automated driving at intersections seems to be complicated, the used approach based on primary situations makes this task easier. In fact, it is possible to plan the maneuver in a logical order of primary situations using a systematic algorithm. This is graphically explained in Fig. 4.31, 4.32, 4.33 and 4.34 for turning left, right, making a U-turn and driving forward, respectively.

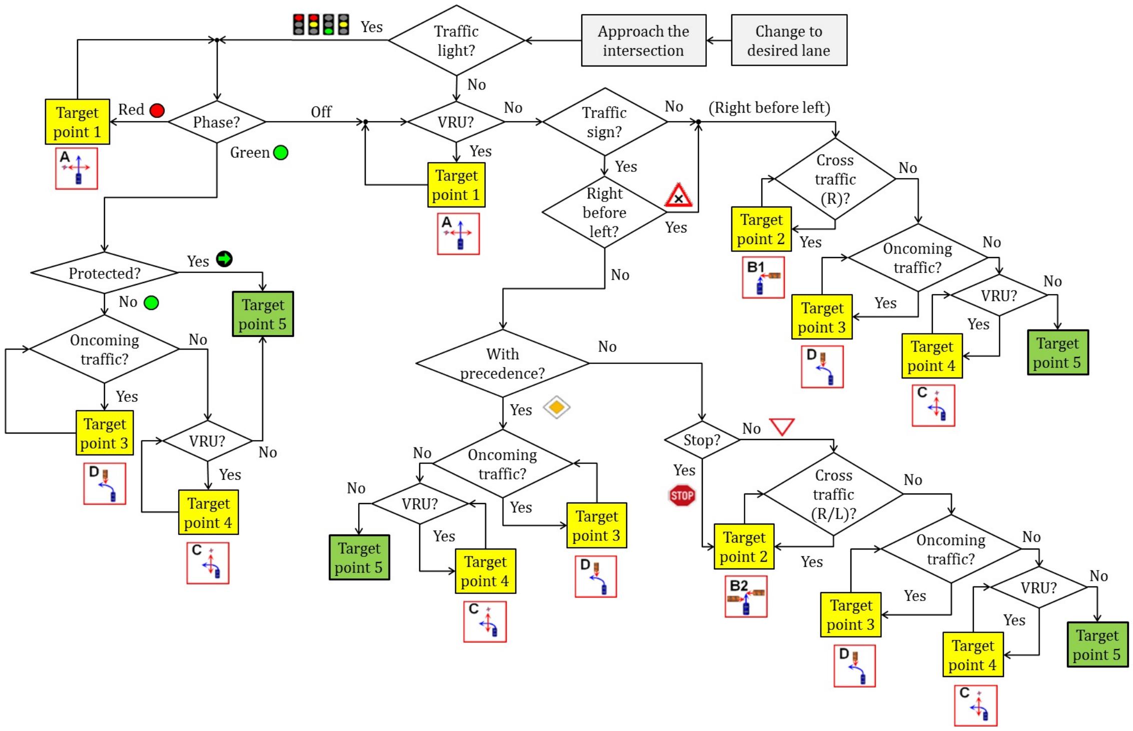


Figure 4.31.: *Decision making flowchart based on primary situations for turning left at a generic X-intersection. The maneuver ends when the ego vehicle reaches the target point 5 (green rectangle).*

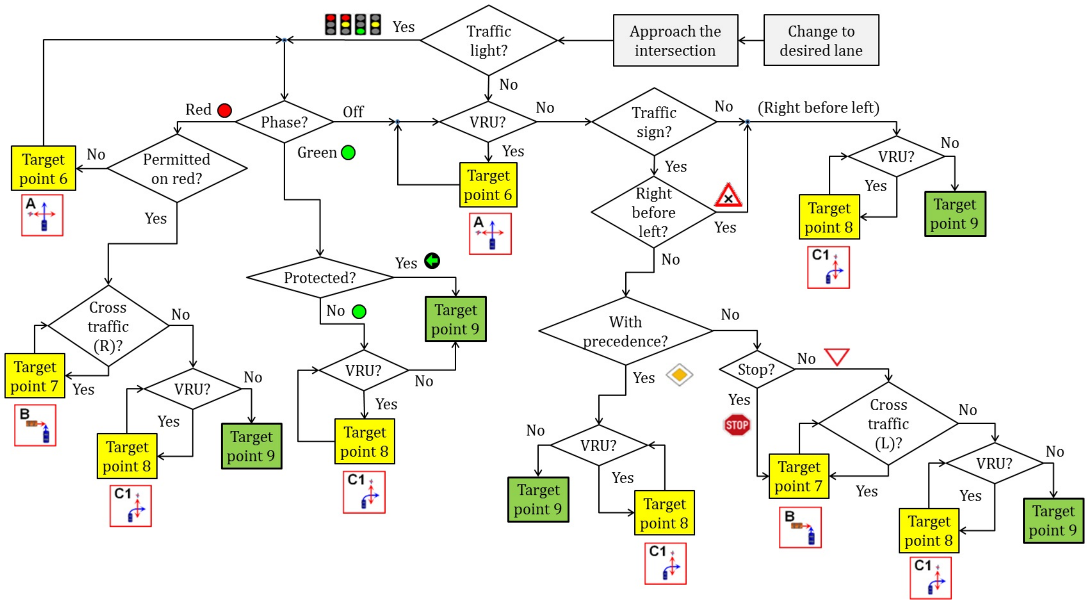


Figure 4.32.: *Decision making flowchart based on primary situations for turning right at a generic X-intersection. The maneuver ends when the ego vehicle reaches the target point 9 (green rectangle).*

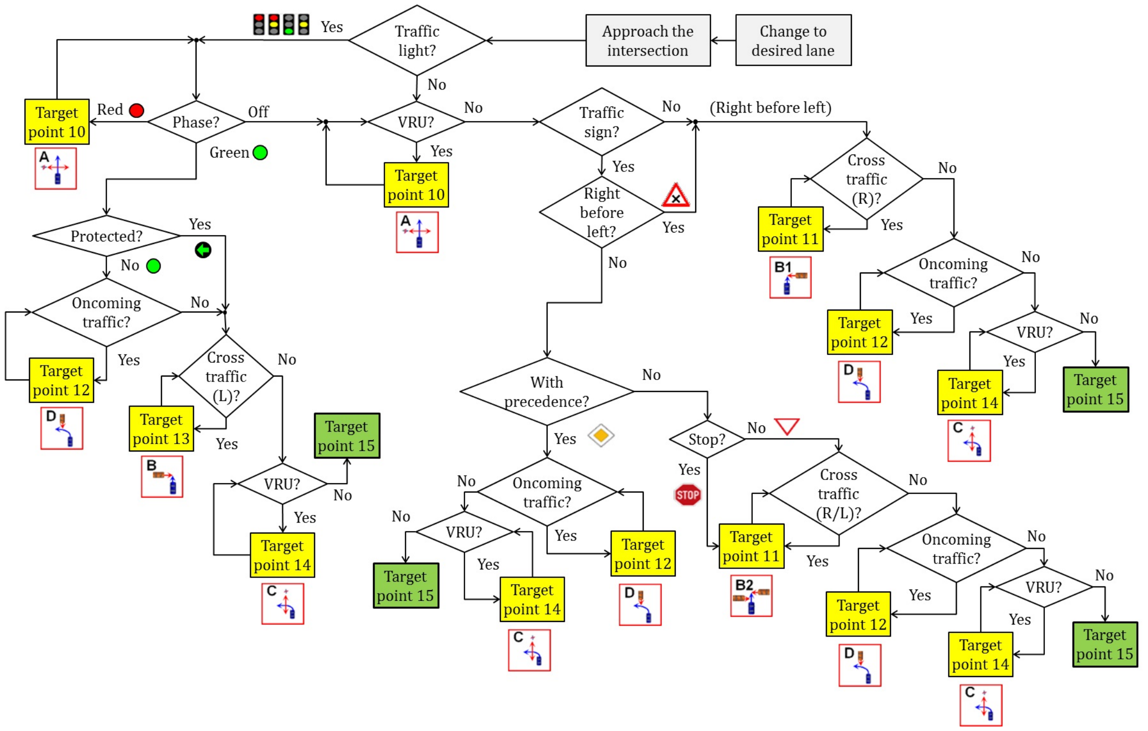


Figure 4.33.: *Decision making flowchart based on primary situations for making a U-turn at a generic X-intersection. The maneuver ends when the ego vehicle reaches the target point 15 (green rectangle).*

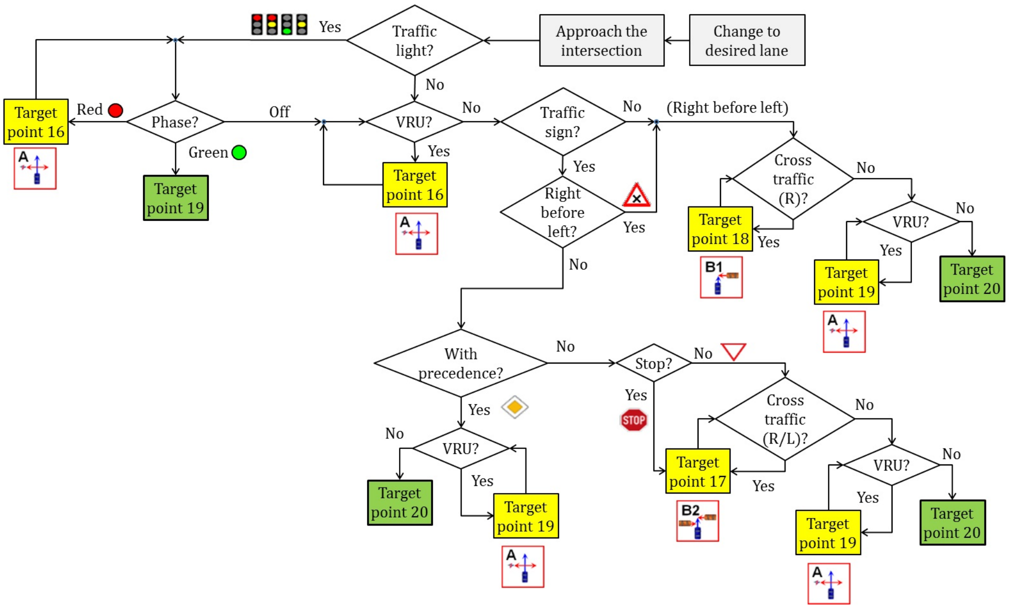


Figure 4.34.: *Decision making flowchart based on primary situations for driving forward at a generic X-intersection. The maneuver ends when the ego vehicle reaches the target point 20 (green rectangle).*

These flowcharts explain which information is needed to set the position of the target points for every primary situation. For the sake of clarity, these diagrams have been kept simple by considering only a basic topology. Namely, the consideration of different topologies (e.g. difference between handling a T- or X-form intersections) is omitted to ease its representation and understanding. Furthermore, the pass permission *permitted on red* is completely omitted.

As an example (see Fig. 4.30 and 4.31), let's assume that the ego vehicle aims to turn to the left at a usual intersection without precedence (e.g. because a yield sign gives the right of way to other road users). In this given example, once the ego vehicle approaches the intersection and has done the proper lane change(s), the first required information of the scenario is how the traffic flow is controlled, namely by traffic lights, traffic signs or the right-before-left rule. This determines the first main branching of the flowchart. Then, a primary situation A (with its corresponding target point 1 illustrated in Fig. 4.30 (a)) is expected depending on the existence of a perpendicular conflict with VRUs. Since in the given example a yield sign was detected, and the ego vehicle intends to turn left, both possible left and right crossing vehicles have the right of way (primary situation $B2$). Consequently, if a collision with crossing vehicles from both sides inside the corresponding observation areas is predicted, the target point 2 forces the ego vehicle to stop in front of the first critical sub-area as long as no collision is expected. Then, the next primary situation D implies setting the target point 3 to avoid a collision with oncoming vehicles. But in case this situation is free (e.g. because there are no oncoming vehicles in the corresponding observation area or the occupancy probability for the current time to area is 0.0), the next target point 4 (primary situation C) is set. Finally, the left turn maneuver is completed with the target point position 5 (colored in green) if the last situation is passable.

However, this does not solve the problem of setting the proper velocity. The main idea is to set a velocity for every situation depending on the pass permission and/or the occupancy probability. For example, if the current pass permission is *not permitted*, the target velocity should be 0 (i.e. $v_{target} = 0.0$). On the other hand, the target velocity is set depending on the occupancy probability. In fact, the target velocity v_{target} decreases exponentially in function of P_{occ} (see Fig. 4.35):

$$v_{target}(P_{occ}) = K(P_{occ}) \cdot v_{max}, \tag{4.39}$$

with

$$K(P_{occ}) = (1 - P_{occ}) \cdot (e^{-\alpha_r \cdot P_{occ}}), \tag{4.40}$$

where v_{max} indicates the maximal desired velocity, P_{occ} is the occupancy probability at the time that is expected until the ego vehicle reaches the primary situation and α_r represents the exponential relation between the target velocity and the occupancy. Therefore, a large value of α_r involves a conservative reaction to the predicted occupancy probability (see the different values of α_r in Fig. 4.35). On the contrary, the smaller the value of α_r, the more conservative is set the target velocity.

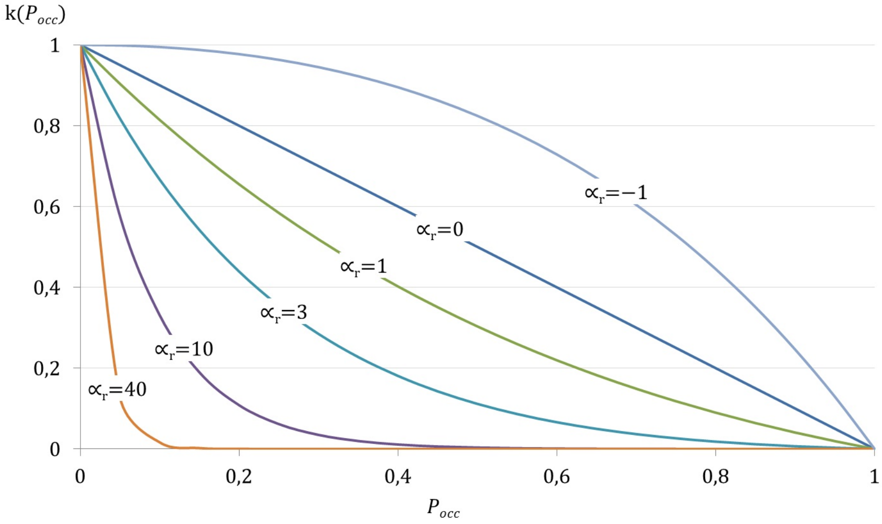


Figure 4.35.: *Exponential relation between the target velocity (i.e. $K(P_{occ})$) and the occupancy probability P_{occ}. The larger the value of α_r is, the more conservative is set the velocity.*

In this manner the tactical decision making module sets a target point with its calculated target velocity considering the extracted information in previous modules. So far it is important to note that the reaction to the occupancy (i.e. how large is the target velocity) depends on the exponential relation plotted in Fig. 4.35, which is given by the value of α_r. This way of adapting the reaction of the ego vehicle enables to set an appropriate value of this parameter empirically.

4.6. Handling occlusions

As a human driver, one react not only to the perceived information, but also to the lack of it. For example approaching an intersection without precedence, if one is not able to assure that no crossing vehicles are approaching, one would drive carefully (e.g. adapting the velocity) to prevent a potential collision with a possible object that cannot be seen. Providing a self driving system with this kind of cognitive human reactions represents one of the most difficult bottle necks in the automated driving field.

A very important advantage of the proposed approach in this thesis is the usage of observation areas within the primary situations. Due to this concept, the system knows which regions should be considered for every primary situation. This is, the system calculates automatically those regions that have to be perceived by the sensors depending on the situation, desired maneuver and the velocity of ego (see the generation of observation areas in Subsection 4.3.2). If some region is not perceived completely by the sensors (e.g. due to occlusions), the system should adapt its velocity to prevent a collision with other possible existing objects.

The main idea is to imitate the cognitive human reaction to occlusions in a ingenious way: setting a virtual object and reacting to it. This object is set in the worst case position inside the *non-perceptible area(s)* of every primary situation. In other words, handling occlusions consists of considering the lack of information and generating a virtual object for every situation if required. Then, in a similar way as for real detected objects, the occupancy probability function is calculated for these generated virtual objects (see (f) in Fig. 4.12 to situate this step within the scenario interpretation flowchart).

To clarify this concept, Fig. 4.36 illustrates an usual example in 3 different timestamps (represented as t_0, t_1 and t_2). Here, the ego vehicle (blue) is making a right turn maneuver (black path) and the other vehicle (white) hinders to perceive the observation area (green polygon) completely. The *perceptible* and *non-perceptible areas* are colored in yellow and red respectively. Here is to note that the term *non-perceptible area* refers to the area(s) inside the observation area that cannot be perceived by the sensors. The illustrated pedestrian represents a generated virtual object (with its path marked with a dotted red arrow). The ego vehicle is approaching a zebra-crossing before turning to the right (primary situation A). At the right side of the road, a parked vehicle causes an occlusion (i.e. a *non-perceptible area*) at the observation area. Therefore, the virtual pedestrian is generated and the ego vehicle slows down (see time stamp t_0). The closer the vehicle gets, the smaller is the *non-perceptible area* (see the red polygon over time t_0, t_1 and t_2).

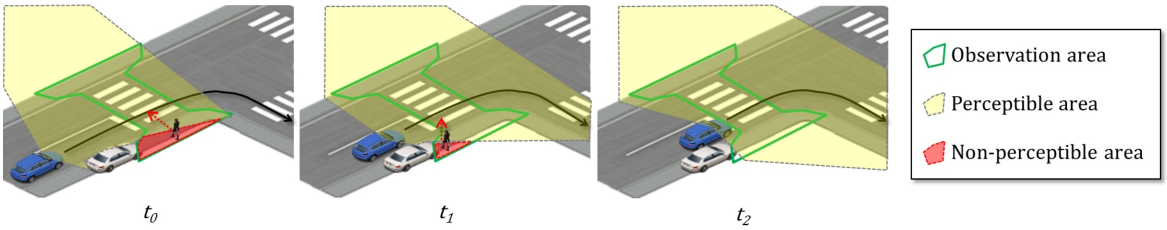


Figure 4.36.: *Occlusion example illustrated over 3 time stamps (t_0, t_1 and t_2). The ego vehicle (blue) approaches a zebra crossing but a parked vehicle (white) impedes to perceive the observation area (green) completely.*

The ego vehicle should react in a different manner to real and virtual objects. Otherwise the target velocity could be set to 0 for a very long time (or even forever) if the position of the virtual object does not change over time. Therefore, the main objective is to adapt the target velocity, so that the ego vehicle drives carefully to avoid a potential collision with a virtual object. The idea is to change the exponential relation between the target velocity and the occupancy probability over time (see the different values of α_r in Fig. 4.35). Therefore, another variable (α_v) is used to set the target velocity v_{target} to adapt the reaction to virtual objects. The key question is to find the most appropriate value of α_v between -1 and a fixed maximal value (i.e. $\alpha_v = (-1, \alpha_{max})$). This is achieved by considering 3 factors, which are expressed with a value between 0.0 and 1.0 (a, b and c):

Stop time (a): This factor considers the time that the ego vehicle has been stopped (or driving very slow). From a logical point of view, once the ego vehicle is stopped, the value of α_v should decrease over time, so that the target velocity increases:

$$a = \frac{t_{stop}}{t_{forget}}, \tag{4.41}$$

where the variable t_{stop} counts the time that the ego vehicle has been stopped and t_{forget} is a constant value set empirically. After this time ($t > t_{forget}$) the reaction should not be conservative any more, i.e. the value of α_v should decrease. This is illustrated in Fig. 4.37.

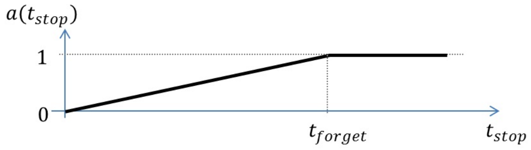


Figure 4.37.: *Plot of factor a: how the reaction to virtual objects depends on the time that ego has been stopped.*

Ego velocity (b): The current velocity of the ego vehicle. The faster the ego vehicle drives, the more conservative should the target velocity be. On the contrary, if the ego vehicle drives too slow, the value of α_v should be increased to avoid that the ego vehicle drives too slow.

$$b = \frac{v_{ego} - v_{stop}}{v_{leave} - v_{stop}}. \tag{4.42}$$

This value is plotted in Fig. 4.38.

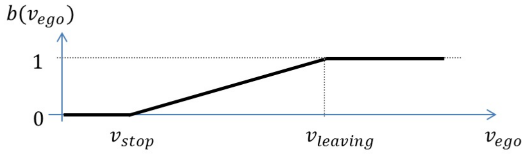


Figure 4.38.: *Plot of factor b: how the reaction to virtual objects depends on the ego velocity.*

Virtual object position (c): This factor is related to the change of the position of the virtual object. If its position changes suddenly, the ego velocity should be adapted in a conservative way. In other words, the smaller is the change of the virtual object position, the smaller should be the value of c. For example, if the worst case position inside a *non-perceptible* area (and therefore the position of the generated virtual object) changes suddenly, this factor increases rapidly, so that the reaction to the phantom object turns more conservative when the worst case position changes:

$$c = \frac{\Delta_d - d_{min}}{d_{max} - \Delta_d}, \tag{4.43}$$

where Δ_d indicates the absolute distance between the position of the current and the last virtual object in focus.

The Fig. 4.39 clarifies the dependency on how the position of the virtual object changes.

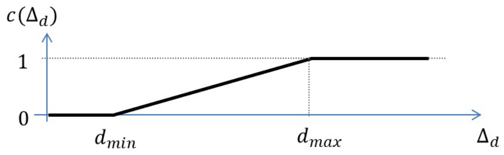


Figure 4.39.: *Plot of factor c: how the reaction to virtual objects depends on the change of the position of the virtual object.*

The variable α_v, which indicates the exponential relationship between the target velocity and the occupancy probability, is calculated using the factor a, b and c as follows:

$$\alpha_v = \frac{(\alpha_{max} + 1) \cdot (a + b + c)}{3}. \tag{4.44}$$

In a nutshell, how conservative is the reaction of the ego vehicle to *non-perceptible areas* depends on these 3 factors. Namely the ego velocity, the time that the ego vehicle has been stopped and how changes the position of the virtual object is consider to adapt how conservative the reaction to the generated virtual object should be. Accordingly, the calculated value of α_v changes over time between -1 and α_{max}), so that the system aims to imitate the cognitive human reaction to occlusions.

Since the different parameters define how the values of a, b and c are calculated, these parameters are crucial to define how the ego vehicle reacts to occlusions. As a proof of concept these values where set empirically as it can be seen in Table 4.2.

Table 4.2.: *Selected values of the parameters for handling occlusions.*

Parameter	**Value**
t_{forget}	1.2 s
v_{stop}	0.2 m/s
$v_{leaving}$	2.0 m/s
d_{min}	0.3 m
d_{max}	1.5 m
α_{max}	0.0

5. Evaluation

The objective of this chapter is to describe the results of testing the proposed approaches. First, the methodology used to evaluate the results is explained in detail. And finally, a consequent interpretation of the results is given in the last section.

5.1. Methodology

Evaluating only the described contributions of this thesis is a very complex task because of 2 reasons: due to the dependencies of other modules and due to the lack of an existing baseline to compare the scenario interpretation at urban intersections.

As it can be seen in the simplified flowchart in Fig. 4.12, the outputs of the perception module are provided to the scenario interpretation, which corresponds to the developed approaches of this thesis. The outputs of this module are provided to the planning. In simple terms, the scenario interpretation module receives the description of the surroundings of the ego vehicle and delivers the corresponding target point along the calculated path to the planning module. The scenario interpretation module aims to understand the perceived information and decide the proper target point considering the corresponding traffic rules. For this reason, the results of the proposed approaches can hardly be evaluated independently. For example, in case that the perception does not detect correctly a crossing pedestrian in front of the ego vehicle, it is obvious that the scenario interpretation module is not able to predict the intention of the pedestrian and plan the corresponding maneuver. Consequently, a coherent evaluation should consider the integration of the scenario interpretation module in the whole architecture. That is, the results of the approaches can be seen as the results of the whole system but considering the dependencies of other modules.

On the other hand, a consistent methodology to evaluate the proposed scenario interpretation could be to compare the developed system with another baseline system. Then, it would be possible to find a set of quantitative or qualitative parameters that describes the effectiveness or improvement of the developed system compared to the baseline. However, the existing baseline (described in chapter 3), is developed for automated driving at highways, so that a direct comparison with the proposed contributions is not viable.

In order to handle the lack of an existing baseline, another possible evaluation method would be to compare the developed system with the way a human driver drives. The idea would be to reproduce a certain scenario several times and then find the proper set of parameters that describe how the system (and a human driver) drives. For example by analyzing the path and trajectory (i.e. velocity profile along the desired path) when a human driver is driving and then compare it to path and trajectory generated by the system for the same situation. Obviously, reproducing exactly the same scenario (e.g. movements of other road users) is not possible, since little deviations of the ego vehicle influence also on what other objects do (or how exactly they react to the ego vehicle). But even taking for granted that this accurate comparison would be plausible, the results of comparing how a human being and an autonomous vehicle drive is not the focus of this thesis. Indeed, big differences between both ways of driving does not necessary correspond to an error of the developed system. For example, in case a pedestrian approaches a crosswalk in front of the ego vehicle, it is coherent to expect that the self-driving system reacts more carefully to the pedestrian than a human driver, since an appropriate autonomous vehicle should not only drive according to the traffic rules, but also transmit confidence to the person inside the vehicle and to the crossing pedestrian. In other words, a self-driving vehicle must not necessary drive exactly like a human being (or even similar). Moreover, this kind of comparison would involve analyzing the psychological influence of how a self-driving system drives, which is clearly out of the scope of this thesis. Therefore, this option to evaluate the developed system is also discarded.

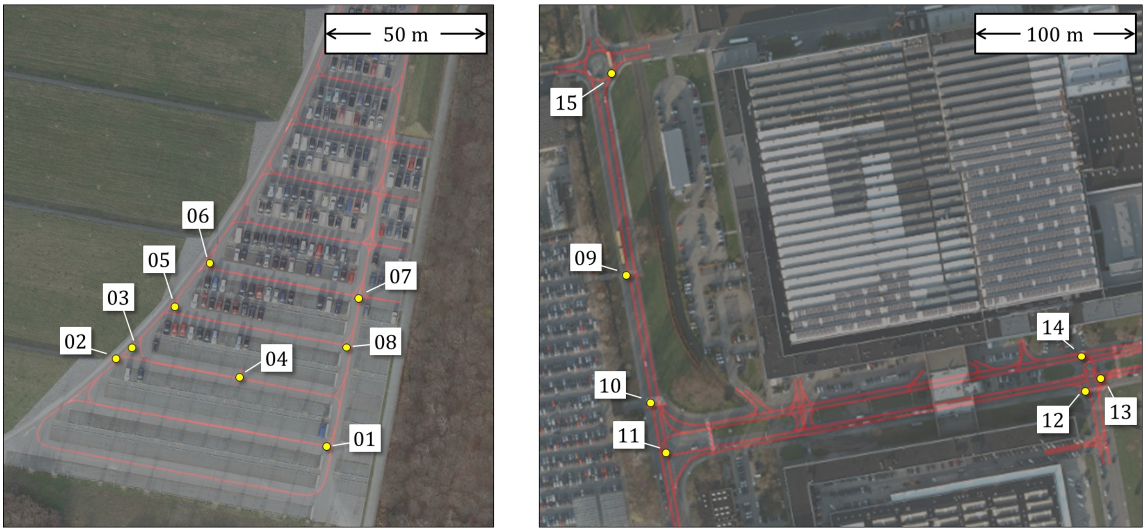
Taking this facts into account, the results of this thesis are evaluated by describing the scenarios used to test the system. The evaluation of the proposed approaches consist in testing the whole system in different locations with different scenarios in order to generate a database. The generated database contains the recorded videos from the interior of the ego vehicle for every evaluated scenario. These videos and the complete database are attached in digital form (see the description of the included data storage device in Appendix A).

In order to describe objectively the used methodology, the locations used to evaluate the system are detailed in the following section. Then, a description of the generated database is given.

5.1.1. Location description

The results evaluated in this thesis has been recorded in 2 different locations in the city of Wolfsburg. In particular, in a parking ground and along a track of approximately 700 m at the Volkswagen factory. The parking ground can be seen in Fig. 5.1 (a). In this Fig.

the road network (colored in red) and an aerial image are superposed. Thanks to this representation, it is possible to get a general idea of the different involved intersections. In order to categorize the results, a reference point is marked with a yellow circle for every exact localization. These reference points are labeled with an index from 01 to 08 in the parking ground. In the same way, the reference points from 09 to 15 are shown in Fig. 5.3 (b). It is to note, that this environment contains different type of intersections, which enlarges the variety of testing possibilities, including 2 intersections, 2 zebra-crossings and 1 roundabout.



(a) (b)

Figure 5.1.: *Reference positions used for the evaluation. 8 positions in the parking ground (a) and 7 along a track inside the Volkswagen factory in Wolfsburg (b). The exact position of the locations are illustrated with a little yellow circle and its corresponding index.*

Moreover, the Universal Transverse Mercator (UTM) (WGS84) coordinates for every reference point is given in Table 5.1.

Table 5.1.: *ID and UTM coordinates (WGS84) of the reference points used for the evaluation.*

ID	**UTM coordinates**	
01	$6^{\circ}1'87.60''E$	$58^{\circ}0'90.99''N$
02	$6^{\circ}1'86.86''E$	$58^{\circ}0'91.22''N$
03	$6^{\circ}1'87.00''E$	$58^{\circ}0'91.28''N$
04	$6^{\circ}1'87.24''E$	$58^{\circ}0'91.22''N$
05	$6^{\circ}1'87.14''E$	$58^{\circ}0'91.43''N$
06	$6^{\circ}1'87.21''E$	$58^{\circ}0'91.62''N$
07	$6^{\circ}1'87.70''E$	$58^{\circ}0'91.45''N$
08	$6^{\circ}1'87.63''E$	$58^{\circ}0'91.32''N$
09	$6^{\circ}1'97.53''E$	$58^{\circ}1'06.89''N$
10	$6^{\circ}1'97.71''E$	$58^{\circ}1'05.92''N$
11	$6^{\circ}1'97.74''E$	$58^{\circ}1'05.75''N$
12	$6^{\circ}2'00.92''E$	$58^{\circ}1'06.02''N$
13	$6^{\circ}2'01.15''E$	$58^{\circ}1'06.09''N$
14	$6^{\circ}2'01.10''E$	$58^{\circ}1'06.25''N$
15	$6^{\circ}1'97.43''E$	$58^{\circ}1'08.49''N$

According to the terminology used in this thesis (see chapter 2 section 2.1), the location where the data was recorded corresponds only to the *scenery*, but not to the occurrences during the recording. Describing the place involves basically indicating how the road network is (i.e. control of the traffic flow, possible conflicts with other road users, etc.). For this reason, a more precise description of every reference point in the parking ground is illustrated in Fig. 5.2.

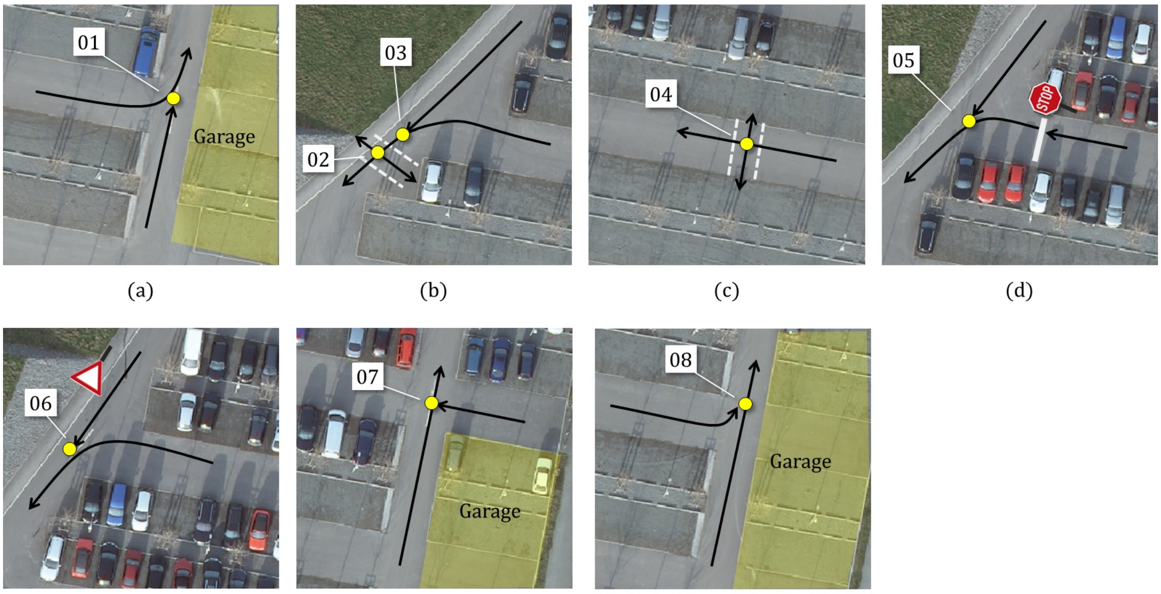
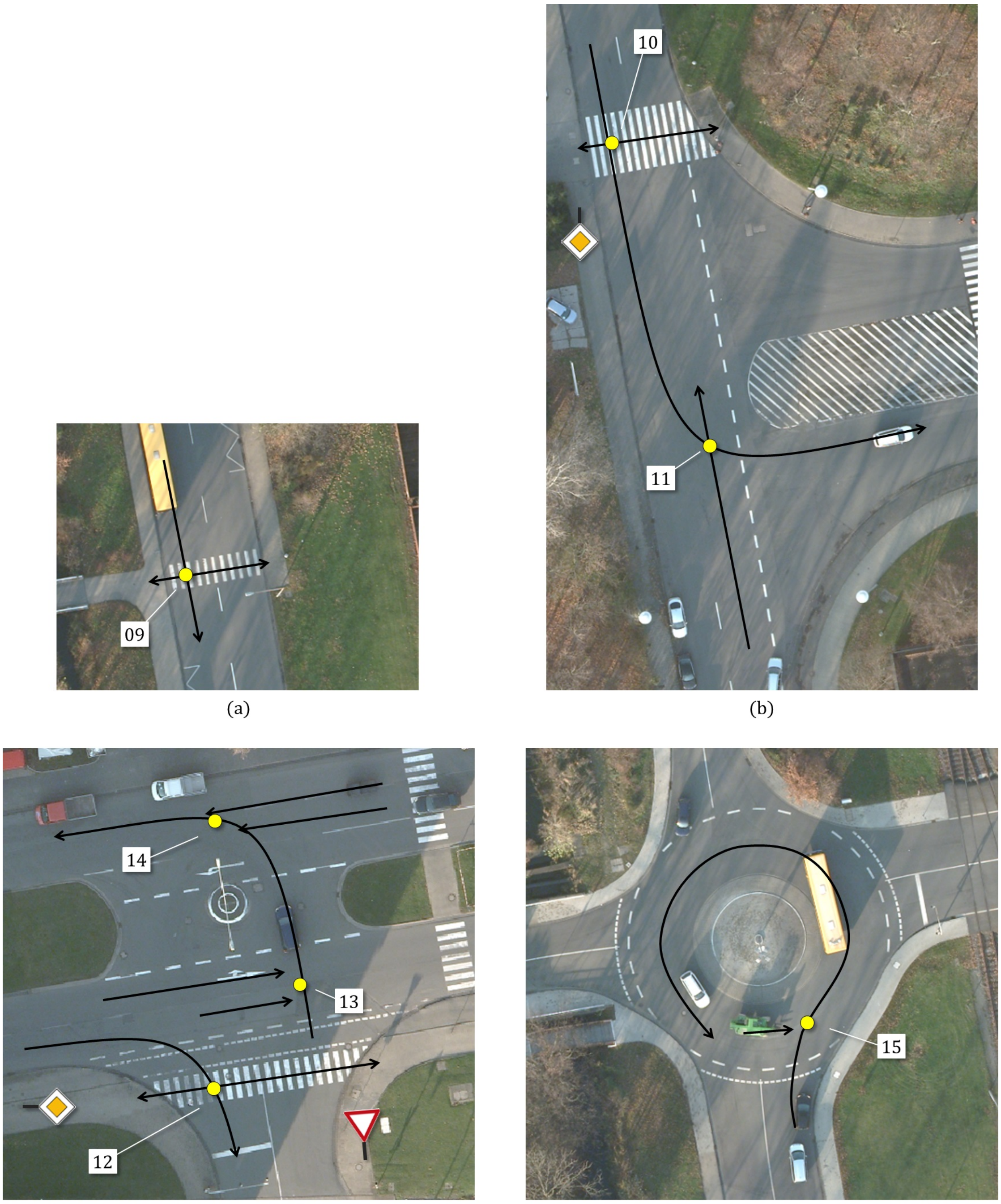


Figure 5.2.: *Detailed description of the reference points illustrated in Fig. 5.1 (a). The yellow circles indicate the reference positions (see UTM coordinates in Table 5.1). The black arrows represent the paths of vehicle or VRUs.*

As it can be seen in Fig. 5.2 (a) and (g), the reference points 01 and 08 mark the position where 2 vehicle paths intersect in a T-form intersection in which the traffic flow is controlled by the right-before-left rule. That is, the vehicle turning left has to give the right of way to the other vehicle driving forward. The reference point 03 is very similar but the angle of the turning left path is larger. Furthermore, this path intersects with a crosswalk, which corresponds to the reference point 02 (see (b)). In contrast, the point with the index 04 (illustrated in (c)) indicates a conflict between a vehicle path driving forward and a perpendicular crosswalk (both with a width of 3 m).

The Fig. 5.2 (d) shows also a T-form intersection but with the traffic flow controlled by a stop sign. The traffic flow of the T-form intersection illustrated in (e) is controlled by a yield sign, so that the vehicle driving forward have to yield the right of way to the other crossing vehicles.

A X-form right-before-left intersection is shown in Fig. 5.2 (f). The special characteristic of this situation, marked with the reference point 07, is the existence of a garage (see the semi-transparent yellow rectangle), which impedes to observe directly a possible vehicle coming from the right side. Since these cars have the right of way, this situation becomes ideal for testing how the proposed approach handles occlusions at intersections.



(a)

(b)

(c)

(d)

Figure 5.3.: *Detailed description of the reference points illustrated in Fig. 5.1 (b). The yellow circles indicate the reference positions (see UTM coordinates in Table 5.1). The black arrows represent the paths of vehicle or VRUs.*

Furthermore, the Fig. 5.3 (a) illustrates the reference point 09, which indicates the intersection between a vehicle path driving forward and a zebra-crossing perpendicular to it with a width of approximately 3 m. This same situation is also marked in the position with the index 10 in (b). Here, a left-turning path with priority and a possible conflict with oncoming vehicles is also graphically described (see reference point 11).

The reference point 12, illustrated in Fig. 5.3 (c), indicates the conflict of a right turning path with precedence and a zebra-crossing. In the same Fig. the reference points 13 and 14 are shown. They correspond to the conflict between crossing vehicles from left and right, respectively, and a left turning path.

Moreover, the driving path at a roundabout is illustrated in Fig. 5.3 (d). The reference point marked with the index 15 indicates the conflict between this path and another path of a vehicle inside the roundabout.

5.1.2. Classification of occurrences

As a result of the evaluation of the contributions of this thesis, a total of 302 video sequences have been recorded and a corresponding database has been generated (see the description of the attached data storage device in Appendix A). Although the places have been described in the previous section, describing the occurrences in the recorded data becomes crucial. It is important to describe not only the location where every sequence was recorded, but also what occurs exactly in every sequence. But describing precisely what and how it occurs is not feasible. Therefore the key question is how to describe what takes place in every sequence in a objective way, so that similar events can be clustered in types of occurrences and then evaluated.

In this sense, an occurrence refers to some concrete happening in the recorded data, but without describing exactly how it happens. For example, in one certain reference point where the path of the ego vehicle intersects a crosswalk, a pedestrian approaches the crosswalk, stays at the sidewalk very close to it, then, while crossing it, stays in the middle of the street and finally crosses the road. In this example sequence, 3 occurrences can be clearly defined: (1) a pedestrian stays at the sidewalk close to crosswalk, (2) it continues walking and stays in the middle of the road, and (3) it crosses. How exactly the pedestrian crosses (i.e. how fast or how the walked path exactly is) is not described in the type of occurrence. Nevertheless, this clustering enables the reaction of the ego vehicle to similar scenarios to be analyzed, and consequently, yields on a descriptive analysis of the scenario interpretation module. In other words, the classification of the recorded sequences in types of occurrences facilitates to describe how the system interprets the different scenarios.

The database contains every recorded sequence, where all relevant occurrences are clustered in 15 types of them. This classification does not consider the exact topology of the situation, but only what occurs. A graphical description of these occurrences is given in Fig. 5.4. Moreover, every type of occurrence is objectively explained hereunder.

(a) Whilst the ego vehicle approaches a crosswalk (see the primary situation A in Fig. 4.11), a pedestrian stays at least for 1 second at the sidewalk (right or left), close to the crosswalk (inside a critical sub-area defined as $\overrightarrow{S_2}$ and $\overrightarrow{S_3}$ and illustrated in Fig. 4.15). This occurrence implies that the ego vehicle aims to stop in front of the crosswalk due to the presence of a VRU inside a critical sub-area. Even if not only 1 pedestrian is involved, but more than 1, only a single occurrence is considered and added to the database. This occurrence is evaluated as successfully if the ego vehicle stops in order to let the pedestrian cross the road. In case the VRU does not move for more than 5 seconds, the system should interpret that the pedestrian does not intend to cross, and therefore, should keep driving.

(b) Similar to (a), a pedestrian (or more than one) stays in front of the ego vehicle, inside the critical sub-area denoted as $\overrightarrow{S_1}$ in Fig. 4.15 (i.e. first critical sub-area, which would be overdriven by the ego vehicle). This occurrence is considered successfully if the ego vehicle stops in front of the crosswalk as long as the pedestrian is inside the first critical sub-area.

(c) This occurrence refers to the fact that a pedestrian (or more than one) crosses the crosswalk. It does not refer to the crossing velocity, exactly walking path, direction, etc. The occurrence is considered successfully if the scenario interpretation module predicts correctly how occupied the primary situation is and reacts to it stopping in front of the crosswalk (and consequently, it enables the pedestrian to cross the road).

(d) While the ego vehicle approaches a crosswalk, a pedestrian (or more than one) walks along the sidewalk beside the crosswalk independently on the walking direction or its position in respect to the road. The reaction of the ego successes if it drives carefully to avoid a collision in case the pedestrian crosses the road suddenly.

(e) When the ego vehicle is approaching a crosswalk, a bicycle crosses in front of the ego vehicle. A successfully reactions implies that the ego vehicle predicts correctly the movement of the bicycle and, consequently, stops to let it cross.

(f) Similarly to (d) but a bicycle is involved (instead of a pedestrian), which goes perpendicular to the crosswalk without crossing the road.

(g) Whilst approaching an intersection, a yield traffic sign gives the ego vehicle no precedence, so that it has to yield the right of way to other crossing vehicle. If this happens, the occurrence is considered successfully.

(h) The ego vehicle approaches an intersection and a stop sign indicates that other vehicles have the right of way. The ego vehicle has to stop and then pass the intersection if no cars are crossing. The type of maneuver done by the ego vehicle is not relevant (i.e. turning right, left...). The occurrence is considered successfully if the ego vehicle reacts properly to the stop sign (independently on the existence of other crossing vehicles).

(i) This occurrence implies that the ego vehicle approaches a right-before-left intersection and another car with precedence crosses the intersection. Consequently, the ego vehicle gives the way to other crossing vehicle(s).

(j) Similarly to (i), the ego vehicle approaches a right-before-left intersection but the crossing object comes from the left, so that the ego vehicle should pass the intersection without yielding the way to the other vehicle.

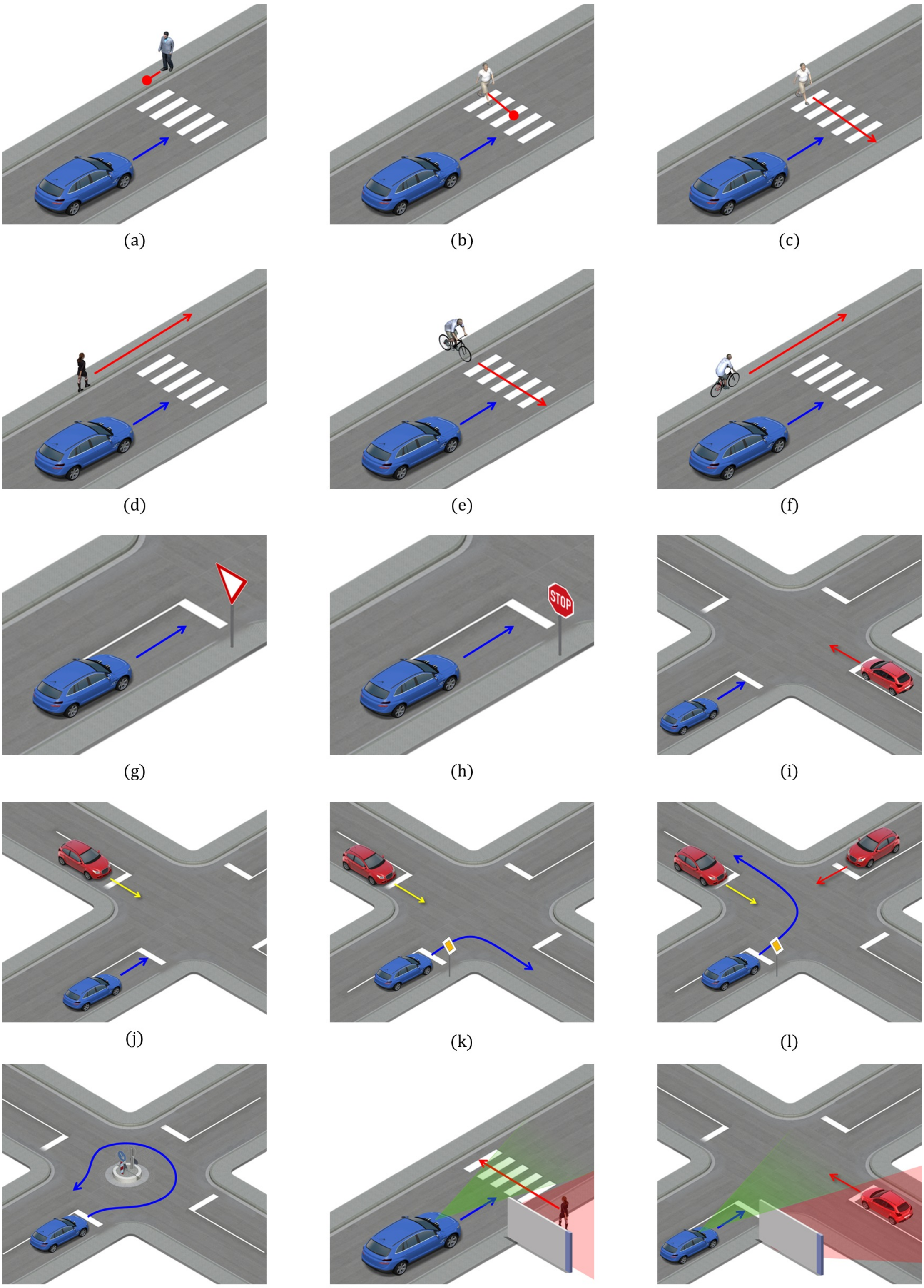
(k) The ego vehicle turns right whilst a priority traffic sign is valid to its lane. This occurrence does not imply the presence of other road users, so that it is considered successfully if the pass permission is interpreted properly (i.e. the ego vehicle has priority but has to consider possible VRU while turning) and the right turn maneuver is achieved correctly.

(l) Similar to (k), a priority sign gives the ego vehicle the right of way while approaching an intersection in which it turns to the left. In case an oncoming car approaches the intersection, the ego vehicle has to yield the right of way. In this case, the occurrence is evaluated as successfully.

(m) The ego vehicle drives into a roundabout independently on the fact that other road users are involved or not. If the ego vehicle drives autonomously inside the roundabout, this occurrence is considered successfully.

(n) While approaching a crosswalk, an occlusion impedes to observe the whole observation area, so that a virtual object is generated and the ego vehicle react carefully to it (see section 4.6).

(o) Due to an occlusion, the system generates a virtual object while the ego vehicle approaches an intersection and reacts to it. This occurrence type does not consider the pass permission.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

Figure 5.4.: *Images representing the catalog of different types of occurrences from (a) to (o).*

In order to describe what happens in the recorded data and then achieve the corresponding evaluation, every relevant happening of the database, i.e. every event to be evaluated, corresponds to an occurrence. Therefore, all occurrences are clustered based on its type (i.e. from (a) to (o)). The resulting classification is illustrated in Fig. 5.5 with a circular chart, which gives an overview of the distribution of the different types of occurrences in the database.

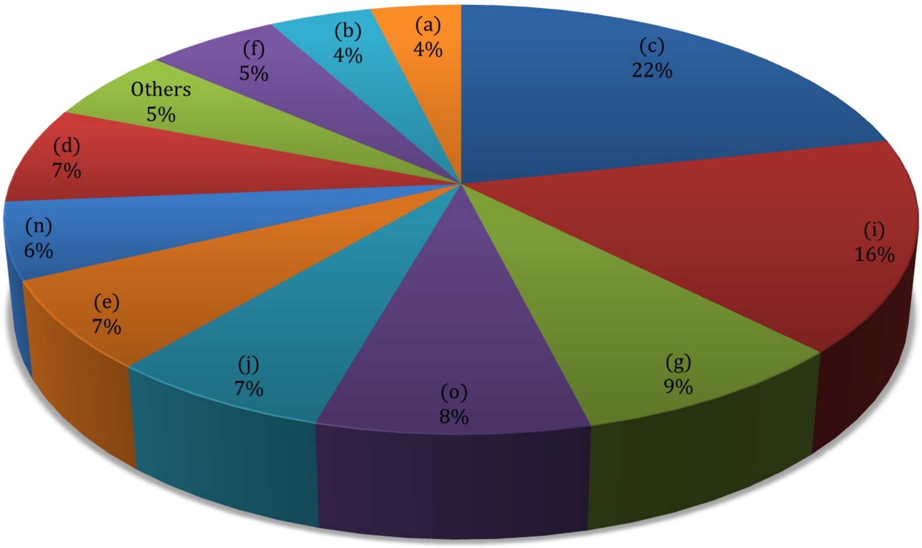


Figure 5.5.: *Percentage of types of occurrences in the database. Every sector of the pie chart represents the frequency of a type of occurrence (other grouped types represent 5% of the database).*

This representation facilitates the interpretation of the database based on what happens in the recorded data.

For example, the most recorded type of occurrences is (c). In detail, in 22 % of the occurrences in the database, a pedestrian crosses a crosswalk in front of the ego vehicle. The second most recorded type of occurrence corresponds to a crossing vehicle that comes from the right at a right-before-left intersection (16 %). Nevertheless, despite some types of occurrences, the Fig. 5.5 shows that the database is almost uniformly distributed, which contributes to a wide spectrum of different types of occurrences.

5.2. Interpretation of the results

This section aims to give a brief interpretation of the results. That is, to summarize the most important aspects of the generated database, which represents the results of this thesis.

As introduced in previous sections, the database consist of 302 video sequences and 399 analyzed occurrences (i.e. one video sequence could contain more than one occurrence). Consequently, every occurrence in the data base is evaluated as successful or not. In this manner, the achieved analysis gives an overview of whether the developed approach works or not, depending on the type of occurrence. The key question is to determine if the recorded data has to be evaluated as successful or not. That is, if some kind of error occurs or the ego vehicle do not react as expected, the evaluated occurrence is marked as not successful. Nevertheless, an important aspect is not only to analyze if the occurrence is successful or not, but also to determine the cause. Therefore, based on the reason why it is not successful, the errors are classified in 4 types:

Object fusion An occurrence is set to unsuccessful because of this type of error if the information from the object fusion causes the error. This happens, for example, if an object is not detected, or if its provided detection is not consistent over time (e.g. if the detection fluctuates or its assigned plausibility is not stable). This type of error can be easily recognized in the videos by observing the existence or lack of detected objects. For a better comprehension of this type of error, an example is illustrated in Fig. 5.6.

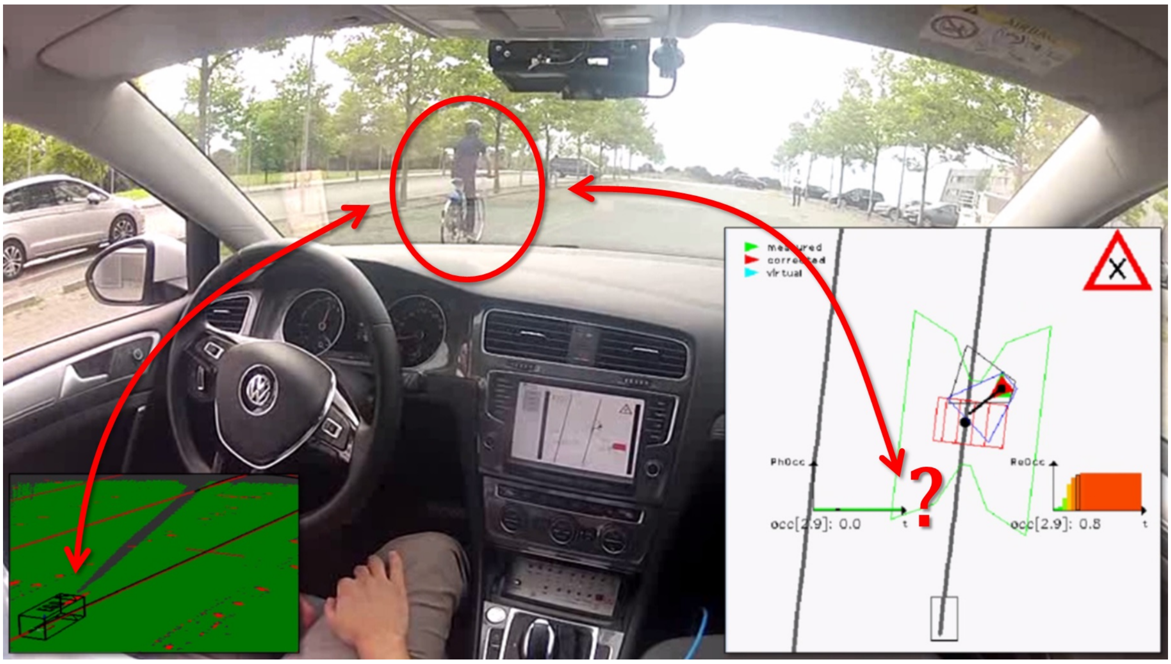


Figure 5.6.: *Example of a not successful occurrence due to an object fusion error. A cyclist (remarked with a red ellipse) approaches a crosswalk but is not detected from the sensor fusion. See Fig. A.6 and A.7 for a detailed description of the meaning of the images.*

Here, it can be seen a frame of the video, in which a bicycle goes parallel to the ego vehicle at the left side. Although the pedestrian at the right side was correctly detected, it is to note in both developer views that the bicycle at the left side was not detected (see appendix A for a better description of the video's content). In this case, the object fusion does not provide a consistent detection, so that a further reaction to it is not possible.

RG object Even if the object fusion module detects and provides correctly the objects, these objects have to be matched to one (or more) lanes of the RG. If the object is not matched properly to the corresponding edge in RG, the interpretation module does not consider the object. In other words, not only the object fusion has to detect and provide the objects, but these objects have to be correctly matched to the edges in the RG, which represents the interface of the perception and the interpretation module. In Fig. 5.7 an example of this error is illustrated.

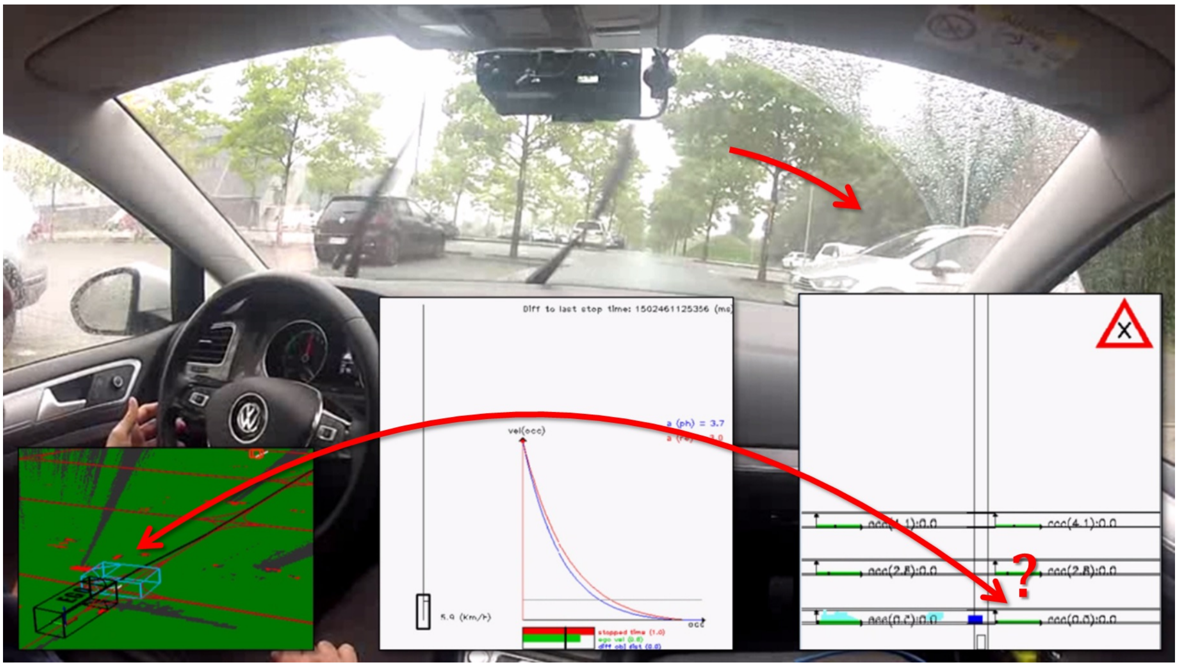


Figure 5.7.: *Example of a not successful occurrence due to a RG object matching error. The vehicle crossing from the right (white car remarked with a red arrow) was detected by the object fusion but was not matched to the RG correctly.*

Fig. 5.7 illustrates a graphical explanation, in which the vehicle crossing from the right side was detected by the object fusion module, but wrongly matched in the RG. This causes that the object is not considered by the developed interpretation module.

Function In spite of coherent results from the other modules, the interpretation module (here denoted as *function*) causes a wrong reaction to the current situation, which corresponds to an error of the developed algorithm.

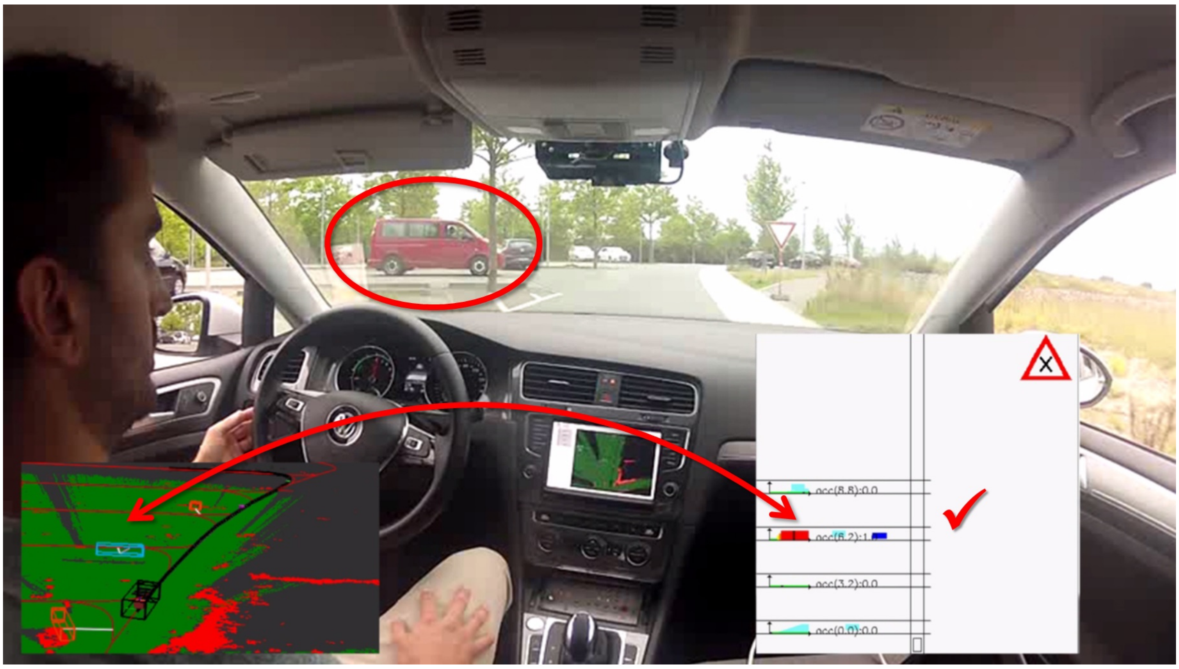


Figure 5.8.: *Example of a not successful occurrence due to a function error. The red vehicle (remarked with a red ellipse) was correctly detected and matched to the RG, but due to a function error, the ego vehicle do not react as expected.*

As it can be seen in Fig. 5.8, the ego vehicle approaches an intersection without the right of way and the vehicle crossing from the left is correctly detected and matched in the RG. In fact, a proper estimation of the occupancy probability was done, but the function did not react properly due to a bug in code by setting the target points. In detail, this error in code was caused because the desired velocity of the target point was erroneously overwritten. However, this code's error was solved after the recordings on 11.08.2017.

Parameters This type of error is not considered an error of the system itself, but a human mistake, which is caused by setting some parameter(s) in a wrong manner. In other words, the different modules work depending on variables, which are set empirically to optimize the results of every module. These parameters can be set while achieving the driving tests. Some of theses parameters are very useful for testing concrete situations, such as activate and deactivate the reaction to non-perceptible areas for different situations. In fact, this type of error occurs 3 times because the reaction to virtual objects was deactivated (see section 4.6).

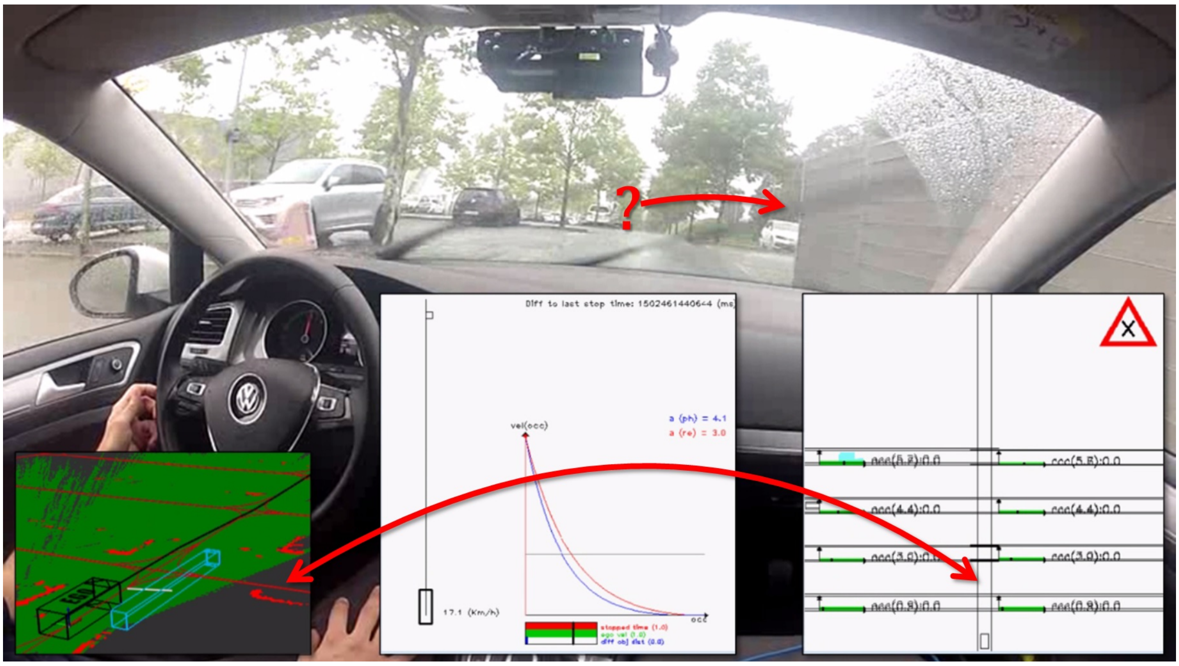


Figure 5.9.: *Example of a not successful occurrence due to a wrong parameter value. The ego vehicle approaches a right-before-left intersection but the fence at the right side impedes to perceive if a cross vehicle comes or not (see red arrow and question mark). Due to an erroneously set parameter, the ego vehicle dot not react as expected.*

For example, as it can be seen in Fig. 5.9, the ego vehicle is approaching a right-before-left intersection and the observation area at the right side is not perceptible from the sensors. This should cause that the interpretation module generates a virtual crossing vehicle. Since the reaction to occlusions was deactivated while recording the results, the occurrence was evaluated as not successful due to wrongly set parameters.

As it is shown in Fig. 5.10, this classification of types of errors enables to have a wide point of view of the different reasons why the whole system didn't work. In fact, it yields to an objective clustering of unsuccessfully occurrences and their causes.

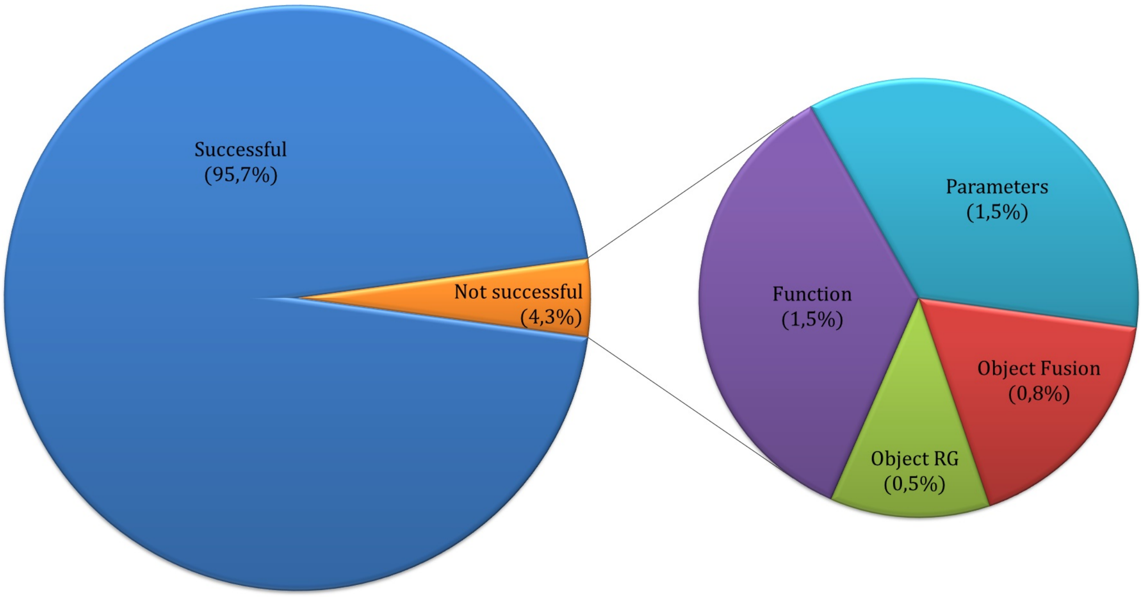


Figure 5.10.: *Success overview: the circle chart at the left side gives an overview in percentage of the successfully recorded occurrences. The circle chart at the right side illustrates the distribution of different causes of unsuccessful occurrences.*

Moreover, a detailed representation of the of the success for every type of occurrence is shown in Table 5.2.

Table 5.2.: *Success overview considering the type of occurrence.*

	Yes	**No - Object fusion**	**No - Function**	**No - Parameters**	**No - RG object**	**Total**
(a)	15	0	0	0	0	**15**
(b)	17	0	0	0	0	**17**
(c)	86	0	0	0	0	**86**
(d)	29	0	0	0	0	**29**
(e)	27	0	0	0	0	**27**
(f)	21	1	0	0	0	**22**
(g)	26	2	6	0	0	**34**
(h)	10	0	0	0	0	**10**
(i)	59	0	0	3	2	**64**
(j)	27	0	0	0	0	**27**
(k)	1	0	0	0	0	**1**
(l)	8	0	0	0	0	**8**
(m)	3	0	0	0	0	**3**
(n)	22	0	0	1	0	**23**
(o)	31	0	0	2	0	**33**
Total	**382**	**3**	**6**	**6**	**2**	**399**

The achieved evaluation shows that the ego vehicle reacted as it was expected according to the German traffic rules in 382 occurrences (95.7%). In contrast, in 17 recorded occurrences (4.3%) the test was unsuccessful. It is also important to note that the type of error *Function* occurs in the occurrence type (g), which represents the 17.6% of the cases. That is, 6 of 34 occurrences clustered as (g) where unsuccessful due to an error classified as type *Function.*

In general, the achieved evaluation aims to represents a coherent analysis of the results considering the types of occurrences and the reason why they where not successfully. This confirms the good performance of the developed approaches in different scenarios.

6. Conclusion

This chapter is structured in 2 sections. First, a quick summary of the achieved work in this thesis is given. And finally, based on the results and its corresponding analysis, some further research work is suggested as a potential improvement of the developed system.

6.1. Summary

In this thesis, I developed a **scenario interpretation concept for automated driving at urban intersections** during my activity at the Automated Driving Department of the Volkswagen Group Research in Wolfsburg, Germany.

The core of the developed approaches is focused on the scenario interpretation module. The main objective of this developed module is to understand the acquired information from the perception. Based on this interpretation of the scenario, the system makes the corresponding decisions to guide the ego vehicle along the calculated driving corridor. In this way the system executes the desired maneuver considering the interpreted traffic rules. In other words, the developed system interprets the surroundings of the ego vehicle and takes the corresponding decision to complete the desired maneuver. This implies understanding the current pass permission of the intersection, handling the possible conflicts with other road users (vehicles and VRUs) and handling occlusions.

After reviewing the state of the art I realized the complexity of the problem and the wide spectrum of methods and possible solutions. In particular for automated driving at urban intersections, the **analysis of the problem** brought out a classification of possible scenarios and conflicts with other road users at urban intersections, which addressed the complexity of the problem from a systematic point of view.

During the process of reading up on the field of automated driving at urban environments, I contributed also to the **second patent application** [54], which described a method for determining a desired trajectory at urban intersections.

In general, the process of analyzing the problem and searching for solutions involved the conception of ideas, suggestions, possible approaches and also thinking digressions. From these reflections came up the **first patent application** [51] regarding the presented thesis, which proposed the core concept of simplifying the scenario interpretation process. In addition, I also described the proposed **concept of primary situations** in [24]. The main idea of this approach consists of considering the possible conflicts with other road users along the driving corridor of the ego vehicle, so that it is possible to brake down the problem into primary situations. In this manner the scenario is interpreted as a logical sequence of situations, in which every one contains information such as which areas are critical or should be observed from the ego vehicle (i.e. *observation areas*). I extended and described this approach in the **second publication** [50] with the focus of the generation of the scenario itself and the relevant information of the primary situations (observation areas, critical areas, etc.). This concept represents the basis for further approaches, and consequently, the further developed contributions are adapted to this concept.

When I started with the implementation of concepts and ideas, a complete system for automated driving at highways scenario was provided. This **baseline system** represented the starting point for a further development of the contributions of this thesis, which should enable the automated driving at urban intersections.

Nevertheless, I realized that the first issue at an intersection is to comprehend how the traffic flow is controlled and under which conditions the intersection should be passed. In this regard I wrote the **third patent application** [47], which was followed by my **third publication** [46]. This approach aims to interpret the current **pass permission** of the intersection based on the acquired information (i.e. road network, traffic lights, traffic signs, etc.). The main idea of this approach is to calculate a discrete probability distribution, in which every state corresponds to a possible pass permission state that defines how the intersection should be passed (e.g. not permitted, permitted, right-before-left, etc.). Moreover, I described in detail how the approach deals with other problems such as the inaccuracy of the detected traffic lights and traffic signs, or the proper interpretation of changes over time. Furthermore, I explained how the developed concept estimates the probability that every possible pass permission is valid for the ego vehicle.

However, even if the pass permission is known and the concept of primary situation works, further development was needed to achieve a complete scenario understanding at intersections. I explained the developed approach to **predict the motion of objects** in section 4.4. It takes advantage of the developed approach to estimate the probability that every primary situation is occupied considering the objects inside its observation area. This approach considers only the relevant objects and assesses the probability that the critical area is occupied over time. First, I described in detail how this is achieved for primary situations where vehicles are involved. Then I described the next approach, which aims to estimate where relevant VRUs could be over the time, and correspondingly, how occupied the primary situation could be.

After the description of how the scenario is interpreted as a sequence of primary situations with its relevant information, I explained how the ego vehicle is guided along its driving corridor to execute the desired maneuver (i.e. driving forward, making an U-turn, turning left or right). I described in detail the **decision making flowchart** that allows to set the corresponding target points that guide the ego vehicle along its driving path considering the traffic rules.

Moreover, I completed the explanation of the contributions of this thesis in section 4.6, which clarifies how the proposed approaches are used to **handle occlusions**. The key idea is to consider the areas that should be observed in every primary situation and the perceptible areas, so that if an observation area cannot be correctly perceived from the sensors, a virtual object is set. Consequently, I described in detail how the developed concept achieves a coherent reaction to non-perceptible areas imitating a human driver reaction.

After I developed these concepts and adapted them to the baseline system, I carried out the corresponding **implementation in a demonstrator vehicle** (see Fig. 3.2 3.3) in order to evaluate the contributions with real scenarios. I depicted in detail the used methodology for the **evaluation** in chapter 5.1. With the purpose of obtaining objective results, I recorded a total of 302 video sequences in 2 different locations in the city of Wolfsburg. Since the exact description of every scenario is not feasible in an objective manner, I clustered the occurrences in 15 different types. Thanks to this classification, I achieved an objective representation of the database and its content. Then I evaluated every recorded occurrence, so that these results represent the generated database attached to this thesis in digital form (see appendix A).

Finally, I gave an **overview of the interpretation of the results** in chapter 5.2 by describing the most relevant obtained outcomes. Here I indicated how the implemented approaches turned out to be appropriate for the formulated challenges. In this regard, I achieved a **proof of concept of the developed contributions** in this thesis in real scenarios.

6.2. Future work

Since the field of autonomous driving is a very active research field, it is expected that increasing efforts bring notable progress in the next years. Especially in the area of interpreting the scenario, the growing interest of the automotive industry in this research area involves the development of new approaches and technologies. To achieve full automated driving in a secure manner, particularly in complex urban environments, a combination of research contributions and further development of the automobile industry becomes crucial.

The achieved evaluation denotes rather successful results for the involved scenarios. Nevertheless there is still a considerable lack of further research work. Potential improvements could be done not only in some specific approaches, but in the whole system. At this point it is important to consider that some of the developed methods were implemented only as a proof of concept, so that a further development is needed for the correct functionality of the whole system with more complex scenarios involving real traffic.

For example, the way the proposed system calculates the occupancy probability over time for crossing vehicles could be combined with more sophisticated artificial intelligence approaches in order to improve the results. The combination of other techniques to consider different maneuvers of other vehicles could bring more robustness to the whole system. The intention of other road users could be predicted using techniques such as deep learning, so that the use of this information could increase the accuracy of the calculated occupancy probability over time.

Another potential improvement that has to be remarked is the motion prediction of VRUs. As it can be seen in section 4.4.2, the proof of concept in this thesis was focused only on pedestrians. Although the evaluation proved reasonable results of the approach also for bicycles, further development to predict the movement of other kind of VRUs has to be done. Future research efforts should consider the motion prediction of all possible road users using their particularities such as separated and parallel bike lanes. Another possible research work in this direction is to consider multiple trajectories considering relevant infrastructure elements such as bus stops, important building entrances, variable traffic congestion or other road network information.

The way how the developed decision making module handles different possible situations could be also improved using machine learning methods such as reinforcement learning. In this way, the design of the decision making flowcharts (see Fig. 4.34, 4.31, 4.32 and 4.33) could be optimized automatically based on wrongly executed maneuvers. In this way, the concept of primary situation could be applied in a more flexible way, so that the system could perform the logical order of primary situations learning from incorrect interpretations. This key concept could also be used for adapting the developed approach to different traffic regulations such as left before right rules.

The proposed system should be tested in different circumstances, traffic scenarios, weather conditions, etc. For example, as it was explained in section 4.2, the developed approach interprets the perceived traffic lights and traffic signs to estimate the probability of every pass permission state from an ego perspective. This module could also consider other inputs such as road markings, instruction of police officers, existence of special emergency vehicles, etc. This could be easily implemented by considering not only 2 probability mass functions, but a function for every input and then achieving the corresponding combination of them.

Either way, the field of scenario interpretation at urban scenarios will definitely keep being an important research focus in the next years.

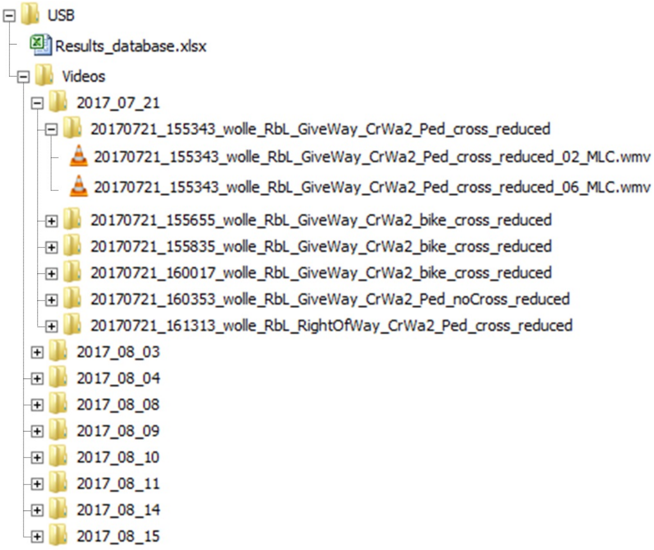
A. Included data storage device

This appendix describes the attached data, which corresponds to the results explained in chapter 5. This are provided in digital form (Universal Serial Bus (USB) storage device) as a crucial part of this thesis.

First, the database and the folder structure of the provided USB storage device is described. And finally, the format of the videos and the used developer views is explained in order to ease the comprehension of its contents.

A.1. Database and folder structure

The provided storage device contains 2 data: a Microsoft Excel datasheet and the videos structured in folders. The contents and the folder structure is illustrated in Fig. A.1.

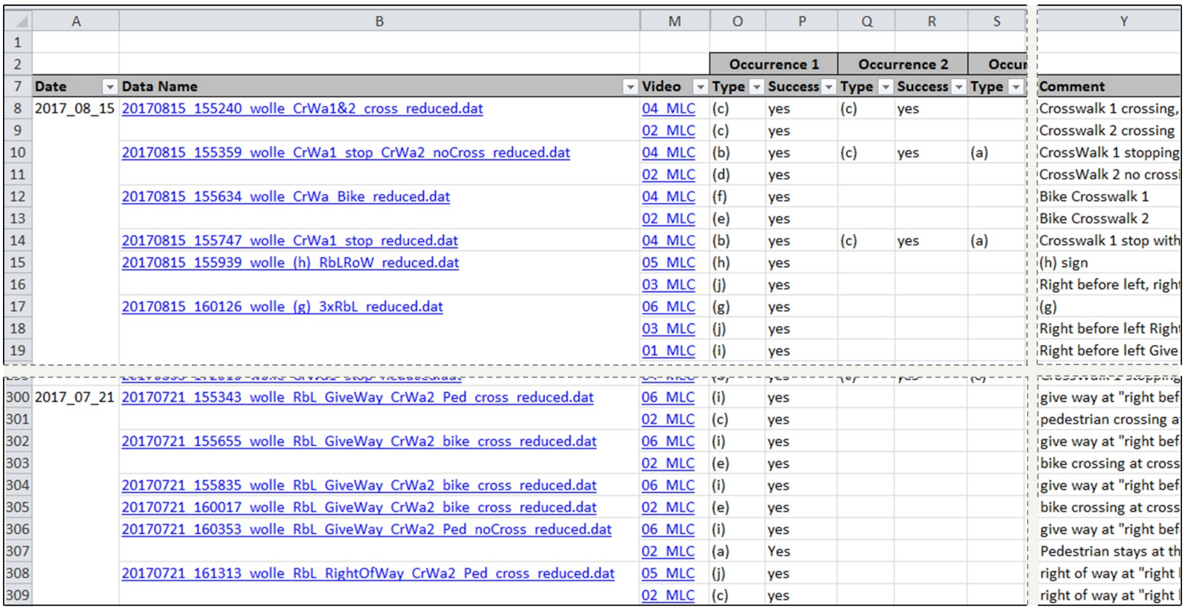


```
⊟ USB
   Results_database.xlsx
  ⊟ Videos
    ⊟ 2017_07_21
      ⊟ 20170721_155343_wolle_RbL_GiveWay_CrWa2_Ped_cross_reduced
          20170721_155343_wolle_RbL_GiveWay_CrWa2_Ped_cross_reduced_02_MLC.wmv
          20170721_155343_wolle_RbL_GiveWay_CrWa2_Ped_cross_reduced_06_MLC.wmv
      ⊞ 20170721_155655_wolle_RbL_GiveWay_CrWa2_bike_cross_reduced
      ⊞ 20170721_155835_wolle_RbL_GiveWay_CrWa2_bike_cross_reduced
      ⊞ 20170721_160017_wolle_RbL_GiveWay_CrWa2_bike_cross_reduced
      ⊞ 20170721_160353_wolle_RbL_GiveWay_CrWa2_Ped_noCross_reduced
      ⊞ 20170721_161313_wolle_RbL_RightOfWay_CrWa2_Ped_cross_reduced
    ⊞ 2017_08_03
    ⊞ 2017_08_04
    ⊞ 2017_08_08
    ⊞ 2017_08_09
    ⊞ 2017_08_10
    ⊞ 2017_08_11
    ⊞ 2017_08_14
    ⊞ 2017_08_15
```

Figure A.1.: *Folder structure of the provided storage device. The results database is given as a Microsoft Excel datasheet. The videos are grouped into dates-folders. Each of them contains a sub-folder for every recorded data, which contains the videos.*

All recorded videos are structured in a logical manner: a main folder for every date with a sub-folder for every data, which contains the videos. For example, as it can be seen in Fig. A.1, the folder with the name *2017_07_21* contains seven different data sub-folders and the first one (named: *20170721_155343_wolle_RbL_GiveWay_CrWa2_Ped_cross_reduced*) contains 2 different videos recorded at the reference position with the index 02 and 06, respectively (see 5.1 (a)).

The provided datasheet represents the overview of the results, in which every row of the datasheet corresponds to a recorded video (from 8th to 309th).



	A	B	M	O	P	Q	R	S	Y
1									
2				Occurrence 1		Occurrence 2		Occur	
7	Date	Data Name	Video	Type	Success	Type	Success	Type	Comment
8	2017_08_15	20170815 155240 wolle CrWa1&2 cross reduced.dat	04 MLC	(c)	yes	(c)	yes		Crosswalk 1 crossing,
9			02 MLC	(c)	yes				Crosswalk 2 crossing
10		20170815 155359 wolle CrWa1 stop CrWa2 noCross reduced.dat	04 MLC	(b)	yes	(c)	yes	(a)	CrossWalk 1 stopping
11			02 MLC	(d)	yes				CrossWalk 2 no crossi
12		20170815 155634 wolle CrWa Bike reduced.dat	04 MLC	(f)	yes				Bike Crosswalk 1
13			02 MLC	(e)	yes				Bike Crosswalk 2
14		20170815 155747 wolle CrWa1 stop reduced.dat	04 MLC	(b)	yes	(c)	yes	(a)	Crosswalk 1 stop with
15		20170815 155939 wolle (h) RbLRoW reduced.dat	05 MLC	(h)	yes				(h) sign
16			03 MLC	(j)	yes				Right before left, righ
17		20170815 160126 wolle (g) 3xRbL reduced.dat	06 MLC	(g)	yes				(g)
18			03 MLC	(j)	yes				Right before left Righ
19			01 MLC	(i)	yes				Right before left Give
300	2017_07_21	20170721 155343 wolle RbL GiveWay CrWa2 Ped cross reduced.dat	06 MLC	(i)	yes				give way at "right bef
301			02 MLC	(c)	yes				pedestrian crossing a
302		20170721 155655 wolle RbL GiveWay CrWa2 bike cross reduced.dat	06 MLC	(i)	yes				give way at "right bef
303			02 MLC	(e)	yes				bike crossing at cross
304		20170721 155835 wolle RbL GiveWay CrWa2 bike cross reduced.dat	06 MLC	(i)	yes				give way at "right bef
305		20170721 160017 wolle RbL GiveWay CrWa2 bike cross reduced.dat	02 MLC	(e)	yes				bike crossing at cross
306		20170721 160353 wolle RbL GiveWay CrWa2 Ped noCross reduced.dat	06 MLC	(i)	yes				give way at "right bef
307			02 MLC	(a)	Yes				Pedestrian stays at th
308		20170721 161313 wolle RbL RightOfWay CrWa2 Ped cross reduced.dat	05 MLC	(j)	yes				right of way at "right
309			02 MLC	(c)	yes				right of way at "right

Figure A.2.: *Screenshot of the provided Microsoft Excel datasheet. Every row from the 8th to the 309th row corresponds to a recorded video. The used columns are from A to Y (Note: only the relevant columns are visible). The screenshot image is divided, so that the start (top-left) and end (bottom-right) of the table is shown (see dotted lines).*

As it can be seen in Fig. A.2, the datasheet is organized using the same structure as the folders: (1) date, (2) data and (3) videos. The relevant columns contains the following information:

Column A Date of the recorded data. The cells corresponding to the same date are grouped together.

Column B Name of the data. Every grouped cell represents to a recorded data. The content of every row has a Hyper-link to the corresponding sub-folder of its data. For example, the reader can directly go to the folder of the data recorded at the

15/08/2017 (i.e. folder named: *20170815_155634_wolle_CrWa1_stop_CrWa2_noCross_reduced.dat* by clicking on the content of the cell *10B* (compare Fig. A.2 and A.1).

Column M Hyper-link to every single video.

Columns from O to X Since a single video could contain up to 5 occurrences, these columns contain the type and success of every occurrence in the video.

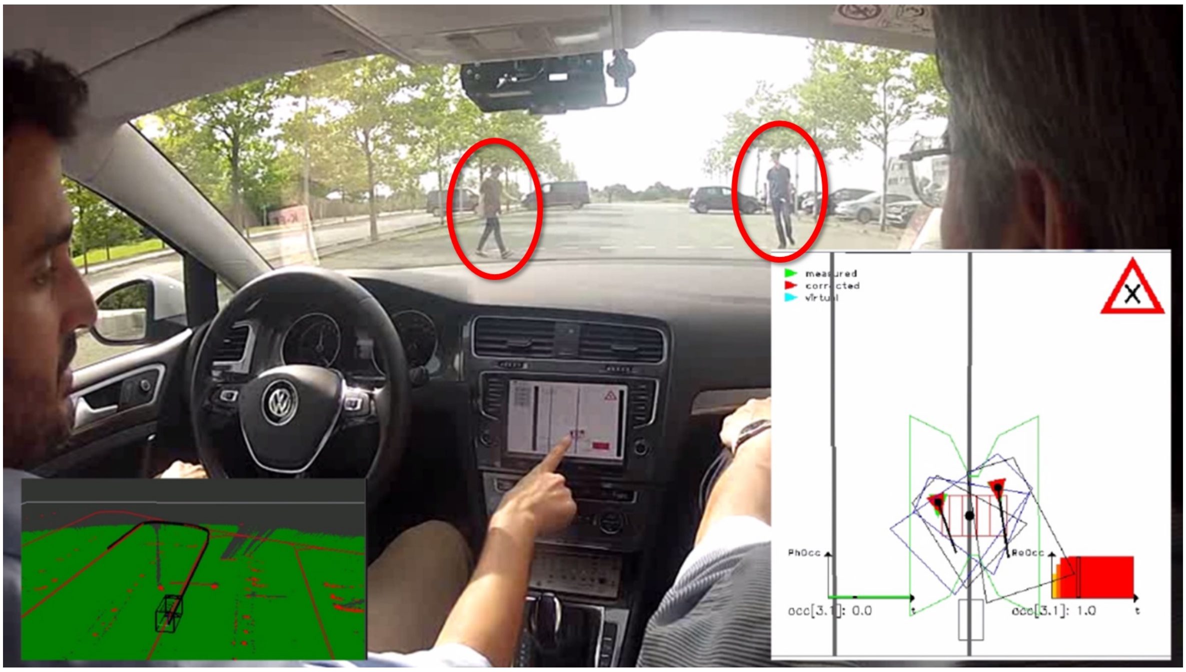


Figure A.3.: *Example of a video that contains 2 occurrences from the data with the name 20170815_155240_CrWa1&2_cross. The ego vehicle is approaching a crosswalk while 2 pedestrians (remarked with red circles) are crossing it.*

Fig. A.3 shows an except of a video with 2 occurrences (see cell *8M* in Fig. A.2). According to the used methodology for the evaluation, 2 occurrences (classified as type (c)) are considered and evaluated separately (see the cells from *8O* to *8R* in Fig. A.2).

Column Y Quick comment about the content of the video.

Thanks to this provided datasheet and the hyper-links, the reader can easily navigate through the folder structure or be linked directly to the desired videos. Furthermore, it is also possible to filter the database (see the little arrow at the right side of every cell of 7th row in Fig. A.2), so that e.g. the reader can filter all occurrence of an specific type or search for particular type of error.

A.2. Description of the videos

The videos aim to illustrate not only what occurred during the evaluation, but also to show how the developed system reacted to the corresponding situations. For this reason, the main videos (recorded from the interior of the ego vehicle), contain the so called *developer views*. These views, which are generated with development purposes, show important information about what the developed system does.

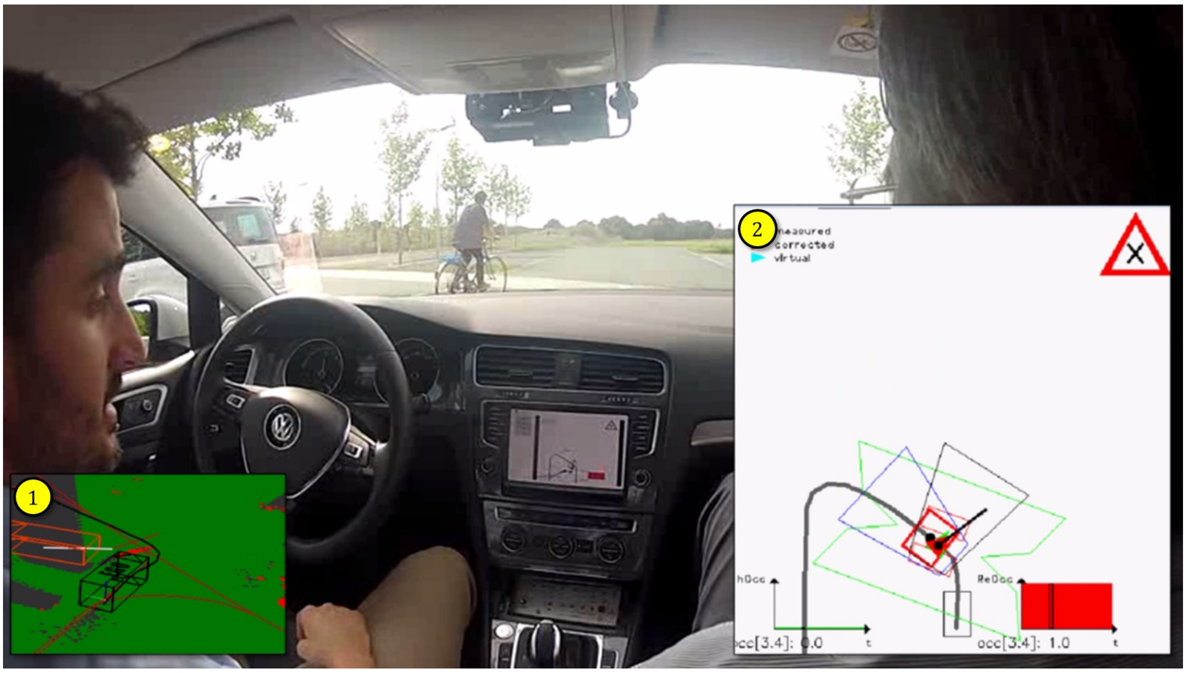


Figure A.4.: *Example of a video with 2 developer views, in which a cyclist is crossing a crosswalk in front of the ego vehicle while turning left (occurrence type (e) according to subsection 5.1.2). The displayed developer views are: (1) RG and (2) pedestrians motion prediction.*

As it can be seen in Fig. A.4, the main video contains 2 developer views, so that it is possible to understand the received inputs and provided outputs of the scenario interpretation module.

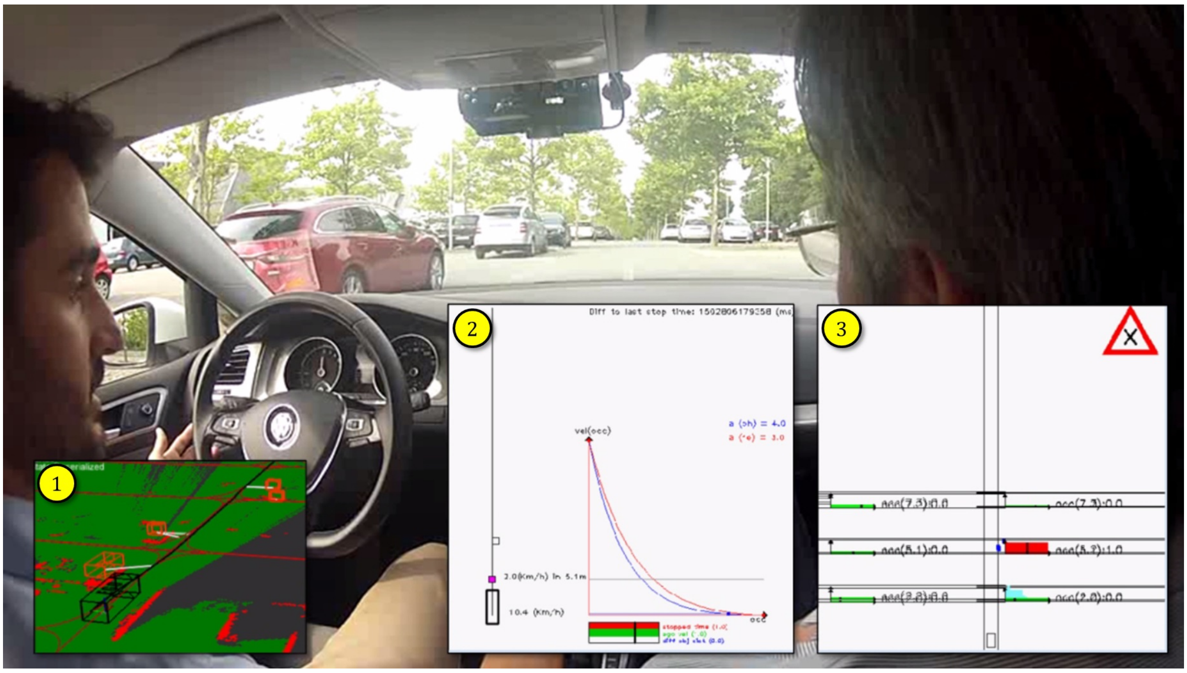


Figure A.5.: *Example of a video with 3 developer views, in which the ego vehicle is approaching a right-before-left intersection while an occlusion impedes to observe if a crossing vehicle is coming from the right (occurrence type (o) according to subsection 5.1.2). The displayed developer views are: (1) RG, (2) occlusion handling and (3) vehicles motion prediction.*

Fig. A.5 illustrates a different example, in which the ego vehicle reacts to the lack of information. Due to an occlusion by approaching a right-before-left intersection, a virtual object should be generated and the ego vehicle should adapt its velocity to be able to brake comfortably in case a crossing vehicle comes. These 3 developer views enable to understand which inputs are received and what the system is doing.

In the provided videos, 4 different developer views are used:

RG This view gives information about the perceived input from the perception module. Here, some crucial information such as the ego vehicle, the road network, the detected objects or the grid are illustrated.

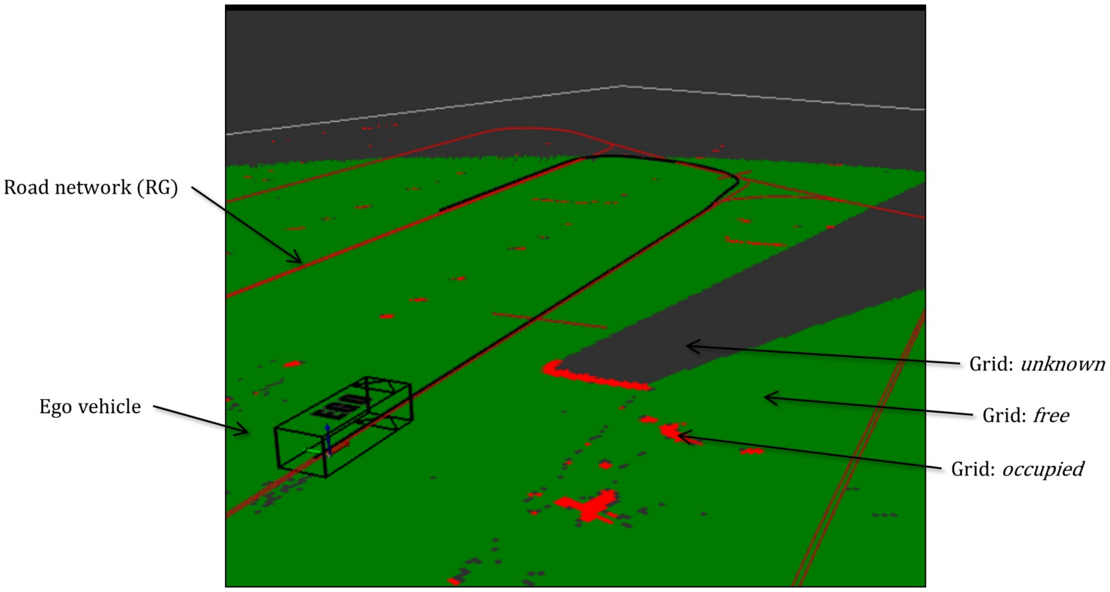


Figure A.6.: *Explanation of the RG developer view. The road network is shown with red lines. The 3-dimensional black square represents the ego vehicle. The different estimated states of the grid are colored in green, red and gray.*

In the given example in Fig. A.6, the ego vehicle is approaching a crosswalk, so that if the corresponding observation area cannot be perceived from the sensors, a virtual pedestrian should be generated. Thanks to the grid visualization, the reader can easily verify which regions are perceived from the sensors (see the displayed gray areas near the crosswalk, which is represented by a red line perpendicular to the path of the ego vehicle).

Pedestrians motion prediction This view aims to display how the system predicts the motion of real and virtual pedestrians (or VRUs). The view contains the ego vehicle, its planed path, the involved crosswalk (or zebra crossing), the critical areas, the observation area, the plot of the calculated occupancy probability over time (real and virtual) and the involved objects.

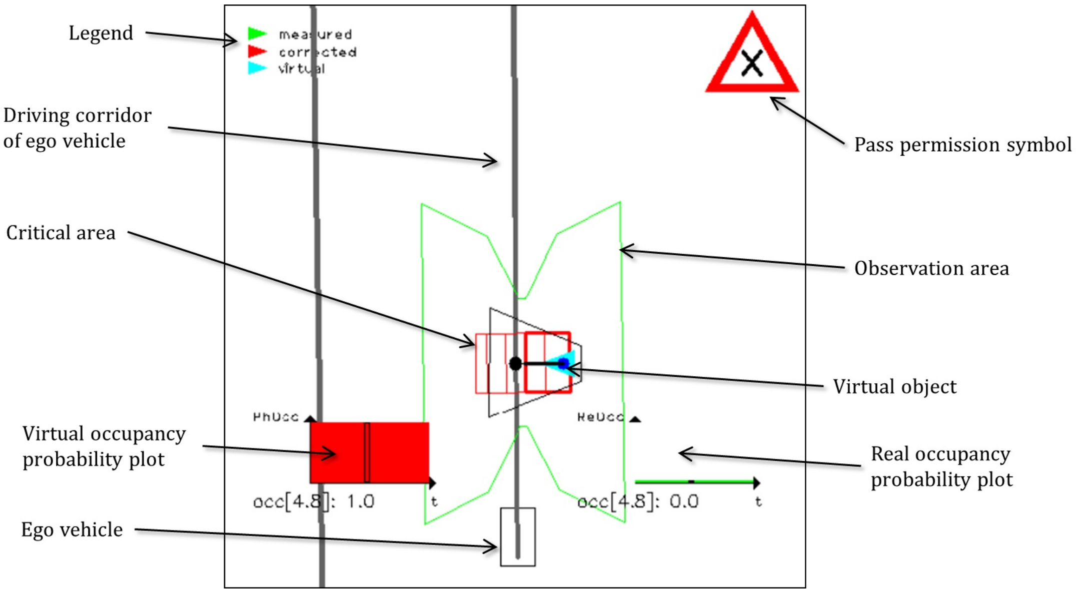


Figure A.7.: *Explanation of the pedestrian motion prediction view. At the top-let, a legend explains the different types of displayed objects. The current pass permission symbol is shown at the upper-right corner. The critical and observation areas are displayed in red and green, respectively. At the right and left side of the crosswalk, the real and virtual calculated occupancy probability over time is plotted. The black rectangle at the bottom represents the ego vehicle with its planed path represented with a gray line.*

Fig. A.7 explains graphically the most relevant information of the pedestrians motion prediction view for the same situation illustrated in Fig. A.6. Since the ego vehicle is approaching a crosswalk and the calculated observation are is not completely perceived from the sensors (see the gray surface of the grid in Fig. A.6), a virtual object is generated. This is remarked with the displayed light-blue triangle and the plot of the calculated occupancy probability for virtual objects (at the left side of the ego vehicle).

Vehicle motion prediction Thanks to this developer view, the calculation of the motion prediction for the relevant vehicles is graphically explained. This view consist of the ego vehicle, its path in Frenet coordinates (therefore it is always straight, even at curves), the crossing (or incoming) lanes that intersect with the ego vehicle's path, the detected objects and the generated virtual objects.

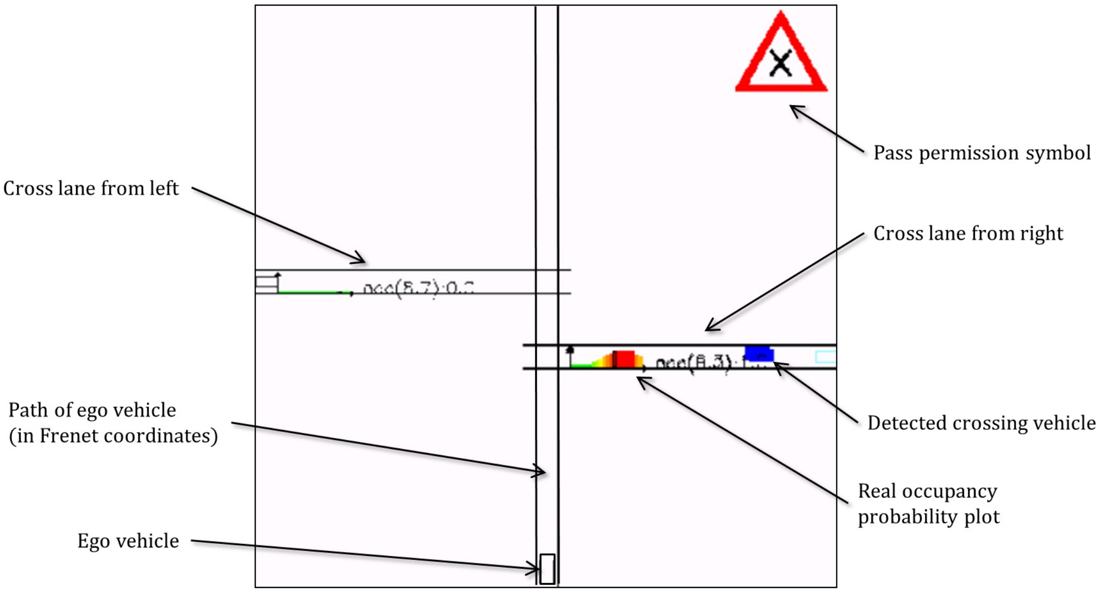


Figure A.8.: *First explanation of the vehicle motion prediction view with an example where a real object is detected. The ego vehicle and its lane are represented by a black rectangle in the bottom and both straight lines, respectively. The other cross lanes, which intersect with the ego's path, correspond the lines perpendicular to the ego's lane. The current pass permission symbol is displayed at the top-right. The calculated occupancy probability over time due to the detected crossing vehicle (blue rectangle) is plotted inside the crossing lane.*

As it is shown in Fig. A.8, the ego vehicle is approaching an intersection, in which the motion of the detected object (see blue rectangle) is estimated. The resulting occupancy probability over time is plotted in a simple axis, so that is it possible to note if the prediction was correctly calculated or not.

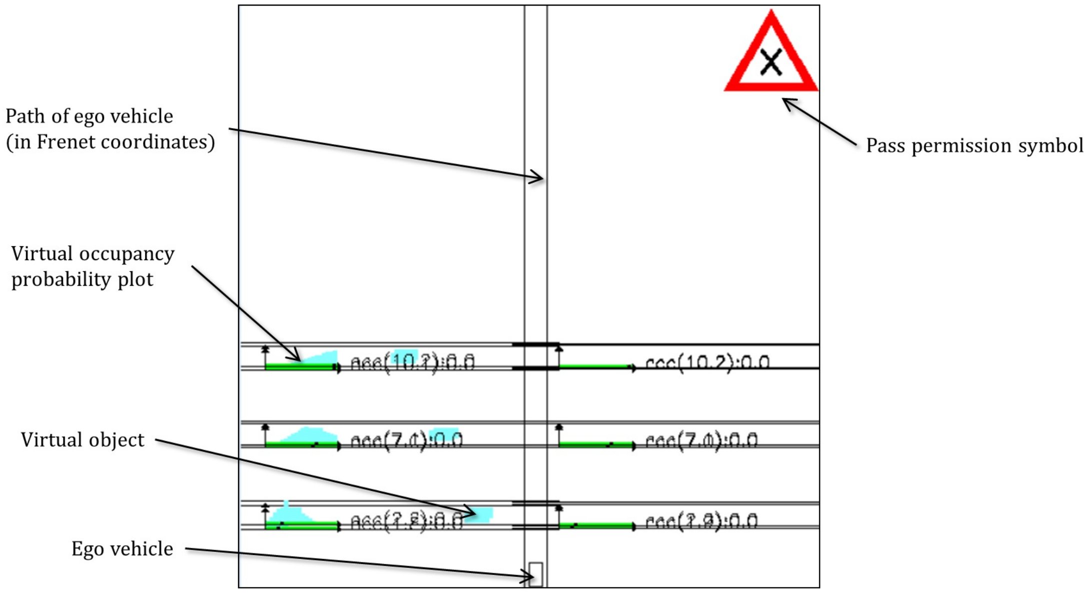


Figure A.9.: *Second explanation of the vehicle motion prediction view with virtual objects. The generated virtual objects and the calculated occupancy probabilities are colored in light-blue.*

In addition, this developer view shows also the generated virtual objects and its plots of the calculated occupancy probability over time (see inside the left cross lanes in Fig. A.9). Since the observation areas of the left cross lanes are not perceived from the sensors, a virtual object for every corresponding situation is generated and displayed as a light-blue rectangle.

Occlusion handling This view explains graphically the reaction of the ego vehicle to non-perceptible areas. It is inserted to the main video (i.e. the video recorded from the interior of the vehicle) when a virtual object is involved, independently on the type of object: i.e. VRUs or vehicles. The view consists of the ego vehicle, its path in Frenet coordinates, the involved virtual object, the plot representing the relation between the velocity and the occupancy probability, and a little representation of the parameters used to handle occlusions (see a, b and c in section 4.6).

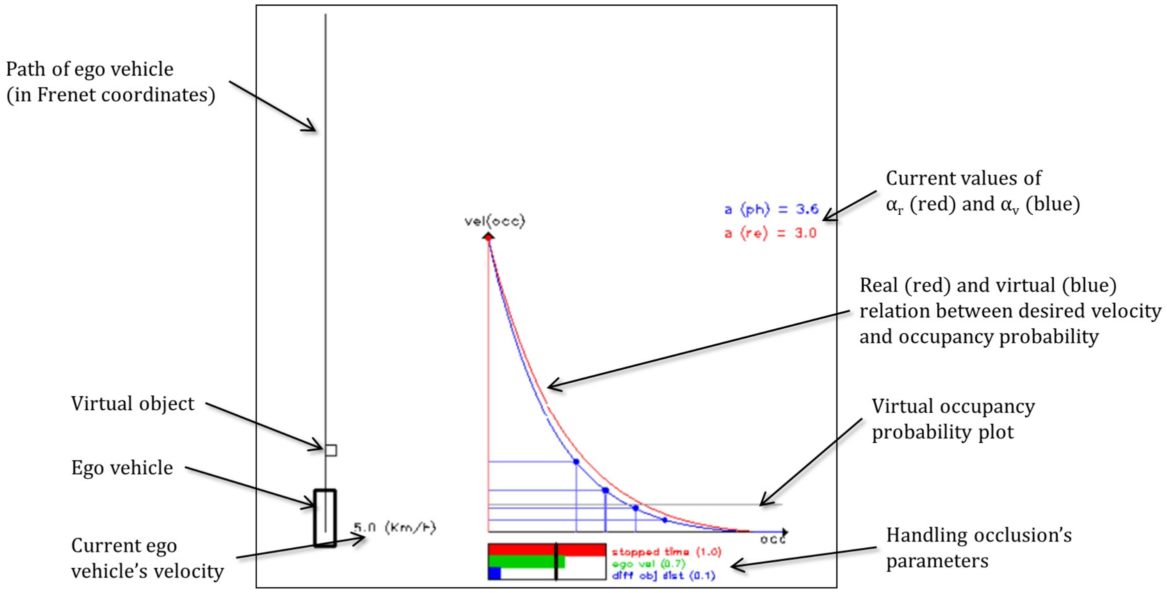


Figure A.10.: *Explanation of the developer view occlusions handling. At the bottom-left, the ego vehicle is represented by a black rectangle and a straight vertical line represents its path in Frenet coordinates. The small black rectangle in front of the ego vehicle represents the generated virtual object. The displayed plot shows the exponential relation between the maximal ego's velocity and the occupancy probability for real and virtual objects (red and blue, respectively). Under the plot, a small bar plot represent the values of the parameters used for the reaction to virtual objects.*

Fig. A.10 illustrates an example in which the ego vehicle is reacting to a virtual object. The current velocity of the ego vehicle is written at its right side and shown in the plot with an horizontal gray line. Thanks to this view, the reader can see how the exponential relation between the maximal velocity and the occupancy probability changes over time (blue curve) depending on the parameters (see small bars plot under the main plot).

List of Figures

List of Tables

List of Acronyms

PROMETHEUS Program for a European Traffic of Highest Efficiency and Unprecedented Safety

MoTiV Mobilität und Transport im intermodalen Verkehr

INVENT Intelligenter Verkehr und nutzergerechte Technik

DARPA Defense Advanced Research Projects Agency

GCDC Grand Cooperative Driving Challenge

CACC Cooperative Adaptive Cruise Control

CoAct Cooperative Autonomous Car Train

FUSE Functional Safety and Envolvable architectures for autonomy

ANSI American National Standard Institute

DL Description Logic

WOL-DL Web Ontology Language Description Logic

RDF Resource Description Framework

SWRL Semantic Web Rule Language

GP Gaussian Process

LCS Longest Common Subsequence

MLP Multi-Layer Perceptron

LR Logistic Regression

RVM Relevance Vector Machines

SVM Support Vector Machines

CHMM Coupled Hidden Markov Models

DBN Dynamic Bayesian Network

GPR Gaussian Process Regression

UKF Unscented Kalman Filter

V2V Vehicle-to-Vehicle

DFA Deterministic Finite Automaton

MCDM Multiple Criteria Decision Making

POMDP Partially Observable Markov Decision Process

VRU Vulnerable Road Users

WGS84 World Geodetic System 1984

LIDAR Light Detection and Ranging

RG Road-Graph

GPS Global Positioning System

V2X Vehicle-2-X

UTM Universal Transverse Mercator

USB Universal Serial Bus

Bibliography

[1] Ernst Dieter Dickmanns and Alfred Zapp. A curvature-based scheme for improving road vehicle guidance by computer vision. In *Mobile Robots I*, volume 727, pages 161–169. International Society for Optics and Photonics, 1987.

[2] M Williams. Prometheus-the european research programme for optimising the road transport system in europe. In *Driver Information, IEE Colloquium on*, pages 1–1. IET, 1988.

[3] MultiMedia LLC. Programme for a european traffic system with highest efficiency and unprecedented safety. http://www.eurekanetwork.org/project/id/45, 1995. accessed December 14, 2017.

[4] GEORG Reichart and WERNER HUBER. Forschungsprogramm motiv. status und weitere perspektiven. *VDI-Berichte*, (1372), 1998.

[5] Eberhard HIPP. Invent-intelligenter verkehr und nutzergerechte technik. *Internationales Verkehrswesen*, 55(9), 2003.

[6] Defense Advanced Research Projects Agency. Darpa grand challenge 2004. http://archive.darpa.mil/grandchallenge04/index.htm, 2004. accessed December 14, 2017.

[7] Defense Advanced Research Projects Agency. Darpa grand challenge 2005. https://www.darpa.mil/news-events/2014-03-13, 2005. accessed December 14, 2017.

[8] Defense Advanced Research Projects Agency. Darpa urban challenge 2007. http://archive.darpa.mil/grandchallenge/, 2007. accessed December 14, 2017.

[9] Institute for Human Factors University of Stuttgart and Technology Management. Safespot. http://www.safespot-eu.org/, 2010. accessed December 14, 2017.

[10] Martin Lauer. Grand cooperative driving challenge 2011 [its events]. *IEEE Intelligent Transportation Systems Magazine*, 3(3):38–40, 2011.

[11] Jeroen Ploeg, Steven Shladover, Henk Nijmeijer, and Nathan van de Wouw. Introduction to the special issue on the 2011 grand cooperative driving challenge. *IEEE Transactions on Intelligent Transportation Systems*, 13(3):989–993, 2012.

[12] Pietro Cerri, Giacomo Soprani, Paolo Zani, Jaewoong Choi, Junyung Lee, Dongwook Kim, Kyongsu Yi, and Alberto Broggi. Computer vision at the hyundai autonomous challenge. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 777–783. IEEE, 2011.

[13] CoAct. Coact - driving the swedish entry in the grand cooperative driving challenge. http://www.coact.se/about/, 2012. accessed December 14, 2017.

[14] Alberto Broggi, Pietro Cerri, Mirko Felisa, Maria Chiara Laghi, Luca Mazzei, and Pier Paolo Porta. The vislab intercontinental autonomous challenge: an extensive test for a platoon of intelligent vehicles. *International Journal of Vehicle Autonomous Systems*, 10(3):147–164, 2012.

[15] Qamcom Semcon SP Comentor, KTH and Volvo. Functional safety and evolvable architectures for autonomy. http://www.fuse-project.se/, 2016. accessed December 14, 2017.

[16] GCDC. Grand cooperative driving challenge. http://www.gcdc.net/en/, 2016. accessed December 14, 2017.

[17] SAE On-Road Automated Vehicle Standards Committee et al. Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems. *SAE Standard J3016*, pages 01–16, 2014.

[18] Sebastian Geyer, Marcel Baltzer, Bernd Franz, Stephan Hakuli, Matthias Kauer, Martin Kienle, Sebastian Meier, Thomas Weißgerber, Klaus Bengler, Ralf Bruder, et al. Concept and development of a unified ontology for generating test and usecase catalogues for assisted and automated vehicle guidance. *Intelligent Transport Systems, IET*, 8(3):183–189, 2014.

[19] Simon Ulbrich, Till Menzel, Andreas Reschka, Fabian Schuldt, and Markus Maurer. Defining and substantiating the terms scene, situation, and scenario for automated driving. In *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, pages 982–988. IEEE, 2015.

[20] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2722–2730, 2015.

[21] Sagar Behere and Martin Törngren. A functional reference architecture for autonomous driving. *Information and Software Technology*, 73:136–150, 2016.

[22] AM Meystel and JS Albus. *Intelligent systems: architecture, design, and control, 2002.* John Wiley & Sons, Inc, 2002.

[23] Sandor M Veres, Levente Molnar, Nick K Lincoln, and Colin P Morice. Autonomous vehicle control systems a review of decision making. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 225(2):155–195, 2011.

[24] David Perdomo López, Rene Waldmann, Christian Joerdens, and Raúl Rojas. Scenario interpretation based on primary situations for automatic turning at urban intersections. In *Proceedings of the 3rd International Conference on Vehicle Technology and Intelligent Transport Systems - Volume 1: VEHITS,*, pages 15–23. INSTICC, ScitePress, 2017.

[25] Marcus Gerstenberger. *Unfallgeschehen an Knotenpunkten: Grundlagenuntersuchung zu Ursachen und Ansatzen zur Verbesserung durch Assistenz*. PhD thesis, München, Technische Universität München, Diss., 2015, 2015.

[26] Michael Hülsen, J Marius Zöllner, and Christian Weiss. Traffic intersection situation description ontology for advanced driver assistance. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 993–999. IEEE, 2011.

[27] Stefan Vacek, Tobias Gindele, J Marius Zollner, and Rudiger Dillmann. Using case-based reasoning for autonomous vehicle guidance. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 4271–4276. IEEE, 2007.

[28] Lihua Zhao, Ryutaro Ichise, Tatsuya Yoshikawa, Takeshi Naito, Toshiaki Kakinami, and Yutaka Sasaki. Ontology-based decision making on uncontrolled intersections and narrow roads. In *Intelligent Vehicles Symposium (IV), 2015 IEEE*, pages 83–88. IEEE, 2015.

[29] Stéphanie Lefèvre, Dizan Vasquez, and Christian Laugier. A survey on motion prediction and risk assessment for intelligent vehicles. *Robomech Journal*, 1(1):1, 2014.

[30] Stéphanie Lefèvre, Christian Laugier, and Javier Ibañez-Guzmán. Risk assessment at road intersections: Comparing intention and expectation. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 165–171. IEEE, 2012.

[31] Andreas Alin, Jannik Fritsch, and Martin V Butz. Improved tracking and behavior anticipation by combining street map information with bayesian-filtering. In *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*, pages 2235–2242. IEEE, 2013.

[32] Quan Tran and Jonas Firl. Modelling of traffic situations at urban intersections with probabilistic non-parametric regression. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 334–339. IEEE, 2013.

[33] Simon Herrmann and Frank Schroven. Situation analysis for driver assistance systems at urban intersections. In *Vehicular Electronics and Safety (ICVES), 2012 IEEE International Conference on*, pages 151–156. IEEE, 2012.

[34] Christos Katrakazas, Mohammed Quddus, Wen-Hua Chen, and Lipika Deka. Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. *Transportation Research Part C: Emerging Technologies*, 60:416–442, 2015.

[35] Brian Paden, Michal Čáp, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on Intelligent Vehicles*, 1(1):33–55, 2016.

[36] Tobias Gindele, Daniel Jagszent, Benjamin Pitzer, and Rudiger Dillmann. Design of the planner of team annieways autonomous vehicle used in the darpa urban challenge 2007. In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 1131–1136. IEEE, 2008.

[37] Javier Alonso, Vicente Milanés, Joshué Pérez, Enrique Onieva, Carlos González, and Teresa De Pedro. Autonomous vehicle control systems for safe crossroads. *Transportation research part C: emerging technologies*, 19(6):1095–1110, 2011.

[38] Andrei Furda and Ljubo Vlacic. Enabling safe autonomous driving in real-world city traffic using multiple criteria decision making. *IEEE Intelligent Transportation Systems Magazine*, 3(1):4–17, 2011.

[39] Felix Lotz and Hermann Winner. Maneuver delegation and planning for automated vehicles at multi-lane road intersections. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pages 1423–1429. IEEE, 2014.

[40] Sebastian Brechtel, Tobias Gindele, and Rüdiger Dillmann. Probabilistic decision-making under uncertainty for autonomous driving using continuous pomdps. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pages 392–399. IEEE, 2014.

[41] Jörn Knaup and Kai Homeier. Roadgraph-graph based environmental modelling and function independent situation analysis for driver assistance systems. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 428–432. IEEE, 2010.

[42] Moritz Werling, Sören Kammel, Julius Ziegler, and Lutz Gröll. Optimal trajectories for time-critical street scenarios using discretized terminal manifolds. *The International Journal of Robotics Research*, 31(3):346–359, 2012.

[43] Economic Commission for Europe-Inland Tansport Committee et al. Convention on road signs and signals. *United Nations Treaty Series*, 1091:3, 1968.

[44] Forschungsgesellschaft für Straßen und Verkehrswesen (FGSV). *Richtlinien für Lichtsignalanlagen (RiLSA)*. 2010.

[45] Wolfgang Fastenmeier et al. *Autofahrer und Verkehrssituation. Neue Wege zur Bewertung von Sicherheit und Zuverlässigkeit moderner Strassenverkehrssysteme*. Number 33. 1995.

[46] David Perdomo López, Rene Waldmann, , and Raúl Rojas. Pass permission interpretation at urban intersections for automated driving systems. *Journal of Traffic and Logistics Engineering*, Forthcoming. Accepted paper.

[47] David Perdomo López and Rene Waldmann. Verfahren zum betreiben eines fahrerassistenzsystems für ein fahrzeug auf einer straße und fahrerassistenzsystem, 2017. Patenanmeldung DE 10 2005 0220 429 A1.

[48] David Noyce and Kent Kacir. Drivers' understanding of protected-permitted left-turn signal displays. *Transportation Research Record: Journal of the Transportation Research Board*, (1754):1–10, 2001.

[49] Tae-Youn Jang and Do-Hyoung Oh. Drivers' satisfaction of protected/permitted left turn (pplt) signal operation. *The Journal of The Korea Institute of Intelligent Transport Systems*, 14(1):46–56, 2015.

[50] David Perdomo López, Rene Waldmann, Christian Joerdens, and Raúl Rojas. Scenario interpretation for automated driving at intersections. In *Communications in Computer and Information Science*. Springer, Submitted paper. Selected paper from VEHITS 2017.

[51] David Perdomo Lopez, Christian Jördens, Lutz Junge, and Michael Darms. Method and apparatus in a motor vehicle for automated driving, November 30 2016. US Patent App. 15/364,441.

[52] Alexander Schumacher. Manöverprädiktion und kritikalitätsbewertung von anderen verkehrsteilnehmern an urbanen kreuzungen, 2016.

[53] M Meinecke, M Roehder, T Nguyen, M Obojski, M Heuer, B Giesler, and B Michaelis. Motion model estimation for pedestrians in street-crossing scenarios. In *Proceedings of the International Workshop on Intelligent Transportation*, volume 7, 2010.

[54] Christian Jördens, David Perdomo Lopez, and Sven Chlosta. Method for determining a desired trajectory for a first traffic user, in particular for a motor vehicle, for a route section, January 27 2017. US Patent App. 15/418,238.

Selbstständigkeitserklärung

Hiermit versichere ich, David Perdomo López, dass ich alle Hilfsmittel und Hilfe angegeben habe und auf dieser Grundlage die Arbeit mit dem Titel Scenario interpretation for automated driving at urban intersections selbstständig verfasst habe. Ich erkläre weiterhin, dass die Arbeit nicht schon einmal in einem früheren Promotionsverfahren eingereicht wurde.

Berlin, 15.02.2018

David Perdomo López