BETWEEN ENVIRONMENTAL PERCEPTION AND DECISION-MAKING: COMPOSITIONAL ENGINEERING OF SAFE AUTOMATED DRIVING SYSTEMS

Dissertation

zur Erlangung des Grades eines

Doktors der Ingenieurwissenschaften

der Technischen Universität Dortmund
an der Fakultät für Informatik

von

Herrn Robin Sören Philipp

Dortmund

4. April 2024

Tag der mündlichen Prüfung:	18.04.2024
Dekan:	Prof. Dr.-Ing. Gernot A. Fink
Erstgutachter:	Prof. Dr. rer. nat. Falk Howar
Zweitgutachter:	Prof. Dr. Jian-Jia Chen

Kurzfassung

Die Entwicklung hochautomatisierter Fahrzeuge stagniert aktuell. Verantwortlich dafür ist die sogenannte Freigabe-Falle: Während die Industrie die technische Umsetzbarkeit hochautomatisierter Fahrzeuge bereits demonstriert, fehlen nach wie vor zuverlässige Methoden, um deren vollumfängliche Sicherheit zu gewährleisten. Es ist allgemein anerkannt, dass ein Nachweis der relativ höheren Sicherheit im Vergleich zum durchschnittlichen menschlichen Fahrer lediglich durch das Akkumulieren von gefahrenen Erprobungskilometern ökonomisch nicht leistbar ist. Alternative Strategien werden benötigt, um die Sicherheit hochautomatisierter Fahrzeuge nachzuweisen. Einen vielversprechenden Ansatz stellt die Dekomposition in mehrere Sicherheitsnachweise einzelner Komponenten dar. Eine Voraussetzung für diese Strategie ist jedoch, dass die erforderliche Funktionalität jeder Komponente spezifiziert und nachgewiesen werden kann. Es ist jedoch nicht trivial festzulegen mit welcher Genauigkeit das Umfeld wahrgenommen werden muss. Ob Ungenauigkeiten der Wahrnehmung, wie z.B. eine falsche Objektklasse oder eine Fehldetektion auch zu gefährlichem Verhalten führen, kann nur bestimmt werden, wenn sowohl die Verarbeitung der restlichen Wirkkette des hochautomatisierten Fahrzeugs als auch die gegenwärtige Betriebssituation berücksichtigt werden. Diese Arbeit schlägt einen formalen Ansatz für die Absicherung von Wahrnehmungskomponenten vor, welche aus drei aufeinander folgenden Schritten besteht: Erstellung einer Taxonomie bezüglich Wahrnehmungsungenauigkeiten, Erhebung von verifizierbaren Anforderungen an die Wahrnehmung bezüglich dieser Ungenauigkeiten und Auswertung der erhobenen Anforderungen. Dazu umreißen wir zuerst die Spezifikation von Wahrnehmungsfehlern und zeigen einen Ansatz, um die Relevanz von umliegenden Verkehrsteilnehmern im urbanen Verkehr zu bestimmen. Als zweites definieren wir konkrete, verifizierbare Anforderungen an ein Objekterkennungsmodul für ein gegebenes Planungsmodul in verschiedenen Situationen durch simulatives, strukturiertes Testen. Abschließend beschäftigen wir uns mit der Evaluation eines Objekterkennungsmoduls. Dazu gehört zum einen ein Ansatz zur Generierung von Referenzdaten für Objektdimensionen und -klassen und eine exemplarische Evaluation eines Objekterkennungsmoduls bezüglich relevanter Fehler und der zuvor definierten Anforderungen. Nach unserer Kenntnis ist dies das erste Mal, dass ein durchgängiger, formaler Ansatz für eine dekomponierte Absicherungsstrategie von Wahrnehmungskomponenten vorgeschlagen und demonstriert wird. Abschließend halten wir fest, dass die Gesamtheit unserer Beiträge bestehend aus Konzepten, Experimenten und kritischen Reflexionen eine neue Perspektive auf die Schnittstelle zwischen Umfeldwahrnehmung und Entscheidungsfindung eröffnet und dadurch auch die Idee einer dekomponierten Absicherung hochautomatisierter Fahrzeuge weiter vorantreibt.

Abstract

Development of autonomous vehicles has hit a slump in the past years. This slump is caused by the so-called approval trap for autonomous vehicles: While the industry has mostly mastered the methods for building autonomous vehicles, reliable mechanisms for ensuring their safety are still missing. It is generally accepted that the brute-force approach of driving enough mileage for documenting the relatively higher safety of autonomous vehicles (compared to human drivers) is not feasible. Since, as of today, no alternative strategies for the safety approval of autonomous vehicles exist. One promising strategy is decomposition of safety validation into many sub-tasks with compositional sub-goals (akin to safety cases but for a vehicles intended functionality) for replacing mileage by combining validation tasks that together document safety. A prerequisite for this strategy is that the required performance of each component can be specified and shown. Specifying how accurate an environmental perception needs to be, however, is a non-trivial task. Whether perceptual inaccuracies, like a wrongly classified or missing object, also lead to hazardous behavior can only be evaluated when considering both the residual processing chain and the operational situation the autonomous vehicle is in. This thesis proposes a formal approach for the validation of perception components consisting of three consecutive steps: creation of a taxonomy regarding perception component inaccuracy, elicitation of verifiable requirements for perception components regarding these inaccuracies and evaluation of the elicited requirements. To that end, we firstly touch on the specification of perception errors and propose an approach to determine relevance of objects in urban areas. Secondly, we elicit verifiable perception requirements subject to a given decision-making module in different scenarios by structured testing in a simulation framework. Finally, we deal with the evaluation of perception components. This includes our approach for the generation of dimension and classification reference values and an exemplary evaluation of an object detection module regarding relevant errors and our previously elicited requirements. To the best of our knowledge, this is the first time that a coherent, formal approach for a decomposed safety validation of perception components is proposed and demonstrated. We conclude, that our contributions provide a novel perspective on the interface between perception and decision-making and thus further support the idea of a decomposed safety validation for automated driving systems.

Disclaimer

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

Die Ergebnisse, Meinungen und Schlüsse dieser Dissertation sind nicht notwendigerweise die der Volkswagen Aktiengesellschaft.

List of Scientific Contributions

While working on the verification and validation of autonomous vehicles, I did not only have the chance to already publish intermediate results of my own research, but also to participate in related research of colleagues. Research is always the product of collaboration. Thus, in the following, an overview about my participation in these scientific contributions is given. The contributions can be separated into contributions for this thesis and other contributions. I am main author to the contributions, which were created in the context of this thesis.

Contributions for this Thesis (Main Author)

I **”Functional Decomposition of Automated Driving Systems for the Classification and Evaluation of Perceptual Threats”** by Robin Philipp, Fabian Schuldt and Falk Howar. In: *13. Uni-DAS e.V. Workshop Fahrerassistenz und automatisiertes Fahren, 2020, pp. 93-105.*

Comment on participation: This paper describes fundamental considerations regarding different types of perceptual threats and their impact on the operation of automated driving systems. I developed the proposed taxonomy regarding the terms fault, error and failure in the context of environmental perception. I derived the different types of perception errors by decomposing automated driving systems into components and the environment into elements. I designed the case example to illustrate the proposed taxonomy. I am author to all sections.

II **”Simulation-based elicitation of accuracy requirements for the environmental perception of autonomous vehicles”** by Robin Philipp, Hedan Qian, Lukas Hartjen, Fabian Schuldt and Falk Howar. In: *International Symposium on Leveraging Applications of Formal Methods. Springer, Cham, 2021, pp. 129-145.*

Comment on participation: This paper proposes a structured testing approach for the simulation-based elicitation of perception requirements for an autonomous vehicle. I am responsible for the overall idea and concept of the approach. I designed and implemented the error models. I designed the structured testing approach and supervised its implementation. I designed the scenarios to be tested, conducted the experiments and evaluated the results. I elicited the accuracy requirements and discussed their validity. I am author to all sections.

III **”Automated 3D Object Reference Generation for the Evaluation of Autonomous Vehicle Perception”** by Robin Philipp, Zhijing Zhu, Julian Fuchs, Lukas Hartjen, Fabian Schuldt and Falk Howar. In: *5th International Conference on System Reliability and Safety (ICSRS), Palermo, Italy, 2021, pp. 312-321.*

Comment on participation: This paper proposes a post-processing method for the revision of perceived object states towards their dimension and classification, which can serve as a basis for subsequent error identification. I am responsible for the overall idea and concept of the approach. I designed both dimension and classification revision and supervised their implementation. This comprises the automatic recognition of situations that are beneficial for estimating vehicle dimensions and a decision tree for classifying objects. I conducted and evaluated the experiments. I am author to all sections.

IV **”Systematization of Relevant Road Users for the Evaluation of Autonomous Vehicle Perception”** by Robin Philipp, Jana Rehbein, Felix Grün, Lukas Hartjen, Zhijing Zhu, Fabian Schuldt and Falk Howar. In: *IEEE International Systems Conference (SysCon), Montreal, Canada, 2022, pp. 1-8.*

Comment on participation: This paper proposes an approach to determine relevant areas and thereby relevant road users for an autonomous vehicle in urban traffic. This approach serves as a first step towards a goal-oriented evaluation of perception components. I am responsible for the overall idea and concept of the approach. I designed the six classes of traffic participants and mapped them to driving maneuvers. I designed and implemented the construction of relevant areas. I conducted and evaluated the experiments. I am author to all sections.

Other Contributions (Co-Author)

■ **”Classification of Driving Maneuvers in Urban Traffic for Parametrization of Test Scenarios”** by Lukas Hartjen, Robin Philipp, Fabian Schuldt, Falk Howar and Bernhard Friedrich. In: *9. Tagung Automatisiertes Fahren, Munich, Germany, 2019.*

Comment on participation: This paper proposes an approach for the classification of driving maneuvers in vehicle movements, which contributes to both the modeling of test scenarios and classification of scenarios. I participated in designing the set of classifiable maneuvers, implementing their recognition in measurement data and analyzing experiments. The maneuvers represent a semantic layer of scenarios which is groundwork to this thesis. I am main author to Section IV and co-author to the whole paper.

■ **”Saturation Effects in Recorded Maneuver Data for the Test of Automated Driving”** by Lukas Hartjen, Robin Philipp, Fabian Schuldt and Bernhard Friedrich. In: *13. Uni-DAS e.V. Workshop Fahrerassistenz und automatisiertes Fahren, 2020, pp. 74-83.*

Comment on participation: This paper deals with saturation effects regarding observed driving maneuvers of vehicles in measurement data of conducted test drives. I was involved in discussions which the proposed approach is built on and supported the analysis of experimental results. I am co-author to the whole paper.

- **”Application of Evolutionary Algorithms and Criticality Metrics for the Verification and Validation of Automated Driving Systems at Urban Intersections”** by Andreas Bussler, Lukas Hartjen, Robin Philipp and Fabian Schuldt. In: *IEEE Intelligent Vehicles Symposium (IV), 2020, pp. 128-135.*

  Comment on participation: This paper addresses the application of evolutionary algorithms for the identification of critical test scenarios. This approach represents a structured testing method which searches for critical concrete scenarios within the parameter space of a logical scenario. I supported the analysis of experimental results. I am co-author to the whole paper.

- **”Systematization and Identification of Triggering Conditions: A Preliminary Step for Efficient Testing of Autonomous Vehicles”** by Zhijing Zhu, Robin Philipp, Constanze Hungar and Falk Howar. In: *IEEE Intelligent Vehicles Symposium (IV), 2022, pp. 798-805.*

  Comment on participation: This paper proposes a method for systematization and identification of triggering conditions for automated driving systems. The term ’triggering conditions’ originates from the ISO 21448 [55]. In the context of this thesis, triggering conditions that correspond to perceptual insufficiencies can be understood as external faults to the perception component. I was involved in discussions which the proposed identification method is built on. I am co-author to the whole paper.

- **”Automatic Disengagement Scenario Reconstruction Based on Urban Test Drives of Automated Vehicles”** by Zhijing Zhu, Robin Philipp, Yongqi Zhao, Constanze Hungar, Jürgen Pannek and Falk Howar. In: *IEEE Intelligent Vehicles Symposium (IV), 2023, pp. 1-8.*

  Comment on participation: This paper proposes a method for the automatic reconstruction of scenarios based on measurement data from urban test drives. This method relies on the classification of driving maneuvers [49] and the revision of object dimension and classification values (Paper III [85]). The subsequent identification of relevant road users builds on the concept of relevant areas of Paper IV [83]. Thus, I was involved in discussions to design and implement the method. I am co-author to the whole paper.

Acknowledgement

The ideas, concepts and experiments of this thesis were developed during my time as a PhD student at Volkswagen Group Research and Volkswagen Commercial Vehicles in collaboration with the research group *Automated Quality Assurance* of TU Dortmund University.

First, I want to thank Prof. Dr. rer. nat. Falk Howar for his excellent supervision over the last five years. His regular feedback, comments and suggestions guided me in the process of conducting my research. At the same time, I was provided enough room to be creative and pursue my own ideas. I could not have imagined a better doctoral supervisor. Additionally, I want to thank Prof. Dr. Jian-Jia Chen for becoming the second examiner of my thesis.

Special thanks go to Dr.-Ing. Fabian Schuldt and Daniel Lipinski. They provided me with the opportunity to work in such a challenging field on a very interesting problem. I am grateful that they gave me a platform to both reach out to and discuss with other engineers and scientists and thus helped me to establish a network. This thesis would not exist without their support and supervision.

I am especially grateful for all the smart colleagues I was surrounded with during the time of working on this thesis. Everyone at the former department of *Automated Driving* at Volkswagen Group Research contributed to a motivating and inspiring working environment. In particular, I want to thank Lukas Hartjen, Andreas Bussler, Zhijing Zhu, Kathrin Symkenberg, Hedan Qian and Jana Rehbein. Thanks for all the intense discussions, hard-earned breakthroughs, last minute fixes, but also tons of fun, joy and laughter we shared together. I really learned a lot. It was a pleasure to work with you.

Last but not least, I want to thank my friends and family for accompanying me along this journey. The emotional support I received helped me to keep going and make continuous progress. Thank you for going with me through the ups and downs.

I am curious about what will come next.

Contents

List of Figures

List of Tables

Part I.

Introduction and Research Goal

1. Introduction[1]

Development of autonomous vehicles has hit a slump in the past years. This slump (or trough of disillusionment in terms of the Gartner hype cycle) is caused by the so-called approval trap for autonomous vehicles: While the industry has shown the technical feasibility of designing and building autonomous vehicles, reliable mechanisms for ensuring their safety are still missing. No autonomous vehicles (SAE level 4 and 5 [1]) have yet been adopted to the market on a wide scale, whereas driving systems with lower degree of automation like e.g., advanced emergency braking systems (AEBS) or adaptive cruise control (ACC) (SAE level 1) are widely distributed in today's cars and have a significant impact on road safety [102, 74]. Additionally, more and more SAE level 2 systems which combine longitudinal and lateral assistance find their way into modern cars [24] and even first traffic jam pilots which correspond to level 3 systems can be activated on public roads, albeit still under very specific environmental circumstances and with limited functionality [3, 8]. The significant difference among the differently automated systems and thus origination of the approval trap is the system fallback level. While level 1, 2 and 3 systems only support the human driver, level 4 and 5 systems take over the driving task completely and must therefore be able to cope with any situation that arises. This also shifts liability for accidents from the human driver to the system and thus in direction of the manufacturers. Naturally, this results in stricter safety requirements for autonomous vehicles since errors cannot be corrected by the human driver and can therefore have fatal consequences.

One established strategy regarding validation of level 1 and 2 systems is a distance-based approach, where test mileage is accumulated to make statistical arguments [32]. When applying this strategy of driving enough mileage for the validation of autonomous vehicles, the approval-trap emerges. Wachenfeld and Winner [120] calculate that it would require 6.62 billion test kilometers to verify with a probability of 50 % that a highway pilot system (SAE level 3) is twice as safe as the average human driver with respect to fatal accidents. Kalra and Paddock [61] calculate that 11 billion test miles ($\approx$ 17.7 billion kilometers) are needed to demonstrate with 95 % confidence and 80 % power that an autonomous vehicle failure rate is 20 % lower than the human driver failure rate. Furthermore, the corresponding test mileage would have to be driven again for every new vehicle or even change in a vehicle's software [120]. Thus, it is generally accepted within the community that the brute-force approach of driving enough mileage for documenting the relatively higher safety of autonomous vehicles (compared to human drivers) is not feasible. Since, as of today, no alternative strategies for the safety approval of autonomous vehicles exist, predictions for the availability of SAE level 5 (fully autonomous) vehicles have changed in the meantime from early 2020s to mid 2030s [109, 60]. The question arises how this approval trap can be overcome and how safety of autonomous vehicles can be ensured.

[1] This chapter contains verbatim content previously published in Paper I [84], II [82], III [85] and IV [83].

A prevalent paradigm in both engineering and computer science is divide-and-conquer. Breaking down a system into subsystems is also an approach that is being followed by the automotive domain, where the OEM and its suppliers work together to design and build cars. Decomposition of safety validation into many sub-tasks with compositional sub-goals (akin to safety cases but for a vehicle's intended functionality) is one promising strategy for replacing mileage by combining validation tasks that together document the safety of an autonomous vehicle. We see that decomposition can be applied in several dimensions:

Scenarios. The task of autonomous driving can be decomposed into a sequence of different driving scenarios, varying widely in complexity. Driving on a straight, empty, well-marked section of a motorway in the middle of the day is relatively easy and has a low risk of endangering passengers or other traffic participants. Detection of lanes and following these is sufficient for driving safely. System failure can be mitigated by slowing down and moving over to the shoulder. Making an unprotected left turn across a crowded inner-city intersection at night in the presence of pedestrians and cyclists, on the contrary, is quite complex and has a comparatively high risk of endangering passengers and other traffic participants. It seems only natural that more effort shall be spent on validating the safety of an autonomous system in complex high-risk scenarios. The systematization and identification of test scenarios as well as the definition of corresponding pass/fail criteria has become an active and established field of research in recent years. Meanwhile, test scenarios are featured within technical standards, regulations and laws: The norm ISO 21448 [55] (also called SOTIF - safety of the intended functionality) follows a scenario-based safety validation for autonomous vehicles. The under development series ISO 3450x provides various standardization efforts corresponding to the definition and evaluation of test scenarios. The UN regulation UNECE R157 [117] lays out concrete scenarios in which an automated lane keeping system has to work safely. In 2022, the European Union has published an implementing regulation [30] regarding type-approval rules for autonomous vehicles which also covers behavioral requirements for various scenarios like e.g., crossing and turning at intersections. Another example is the recently published German regulation [33] for the approval and operation of autonomous vehicles which requires manufacturers and mobility providers of autonomous vehicles to create a test scenario catalog (cf. AFGBV [33, § 12]). While challenges still exist regarding this dimension of decomposition and a release of safe commercial autonomous vehicles based on scenarios is yet to be shown, the aforementioned examples clearly indicate the acceptance of a scenario-based validation approach.

Validation Methods. Another dimension of decomposition are the methods used for validation: instead of purely driving on the road, a hierarchy of validation approaches (ranging from road tests, to proving ground tests, to vehicle-in-the-loop, to hardware-in-the-loop and to software-in-the-loop) can be combined to reduce test efforts on the road. This means, that — dependent on the method — different components of the autonomous vehicle and parts of the environment are virtualized and simulated [107, 80]. Including virtual environments and components into the testing process confers both advantages and disadvantages. Simulation-based testing can potentially be faster than real time, evaluating electronic control devices on test benches saves assembly and calibration effort, and allowing the autonomous vehicle to respond to virtual objects or dummies in a controlled, real-world environment reduces the risk of injury and damage - to both the vehicle and people who may be involved in the test [80].

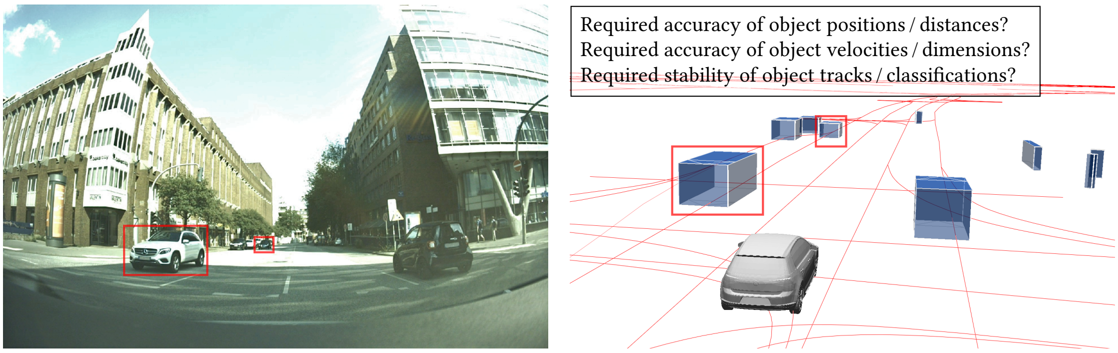
Furthermore, reproducibility of test cases and results is significantly increased when more components and the environment are virtualized during testing [80]. Utilizing all these different test methods shows the potential to make the validation more manageable and flexible while also saving time and costs. However, all of these methods come with their own challenges which impact the test validity. A simulation framework and its models need to be validated, which is particularly difficult regarding physical effects of sensors and vehicle dynamics [91, p. 33]. Computer screens that are recorded by a video sensor in the context of hardware-in-the-loop tests can exhibit delays compared to the real word, which might then lead to inaccuracies along the whole processing chain of the autonomous vehicle and ultimately result in implausible behavior. Modern test targets that are used on proving grounds (e.g., for EURO NCAP tests [31, Chapter 5]) differ significantly from real pedestrians regarding their looks and behavior (e.g., gaze, arm motion). Thus, next to the advantages that all these test methods offer, the non-negligible downside of any other method except road testing is the actual validity of the test results. Ultimately, the validation must show that the autonomous vehicle behaves safely in the real world where it shall interact with real traffic participants. X-in-the-loop methods, simulation frameworks, virtual environments as well as the validation of all of them constitute an active line of engineering and research [92, 27, 103, 73]. Regulations like the AFGBV [33, Annex 1 - 11.], the implementing regulation EU 2022/1426 [30, Annex III - Part 2 - 4.2.] or the UNECE R157 [117, Annex 4 - 4.2.] already allow computer simulations, e.g., to demonstrate compliance with specified requirements or to assess scenarios that are difficult to test in real driving conditions — under the requirement, that the validity of the simulation shall be demonstrated by the manufacturer. The AFGBV [33, Annex 1 - 11.] states further that any tests carried out in the simulation may need to be revisited in the real world when demanded by the technical service. While the benefit of the aforementioned methods for accelerating development processes is undeniable and their prospective use case of virtual homologation is already recognized by law, there is still a gap which needs to be closed in order to leverage these test methods to actual validation methods.

System Architecture. The software and system architecture of autonomous driving systems lends itself to an assume/guarantee-style decomposition of safety validation. Systems consist of three major components: *Sense*, *Plan*, and *Act* [5, p. 41 ff.] (cf. Section 2). These three components are based on different methods and principles and typically share well-defined interfaces: the *Sense* component relies on sensors scanning the surrounding area and corresponding methods to detect and track features in the environment (e.g., deep neural networks for object tracking). The *Plan* component works on this discrete representation of the environment, interprets and predicts development of the scene and then generates an optimal future trajectory. The *Act* component is responsible for executing this trajectory based on principles and techniques from the field of control theory. The components *Sense*, *Plan* and *Act* rely on each other. Insufficiencies in these components or inaccuracies in their provided output can materially affect the functionality of the other components. This can cause errors to propagate along the whole processing chain of the automated driving system and ultimately lead to hazardous behavior. Exemplarily, if a vehicle in front is not detected, a corresponding reaction, such as a braking maneuver, cannot be planned and subsequently not carried out. Therefore, to be able to validate individual components, their actual accuracy and timing requirements in the system context must be elicited and meeting these must be verified – in other words, components must be designed and verified to guarantee behavior within certain bounds under certain assumptions. Assume/guarantee-style reasoning uses pairs of assumptions and guarantees on components for proving a property P on a system S of sequentially composed components $C_1, \ldots, C_n$ by proving guarantee G_i for component C_i under assumption A_i. Now, if additionally every guarantee G_i implies assumption A_{i+1} and G_n implies P, the sequence of proofs on the components establishes that the system S satisfies the property P under the assumption A_1. To the best of our knowledge there are just few works touching on specification of assume-guarantee contracts in context of safety validation regarding *Sense*, *Plan* and *Act*, which are only kept simplistic and theoretical [79, 17]. While (as of today) we cannot apply the above pattern in a rigid formal approach, we can still use the pattern in manually constructed and validated sequences of arguments.

Effective decomposition strategies of the validation task in all three of these dimensions (scenarios, virtualization, architecture) have to be the topic of future research and standardization efforts. Being based on different principles, different methods are required for verifying correctness and validating the safety of *Sense*, *Plan*, and *Act* components. Validation methods for *Act* components already exist: Vehicles already ship with steer-by-wire and brake-by-wire functionality. Safety of these functions is ensured through functional safety approaches (fail operational modes, FMEA[2], FTA[3]). Correctness and performance of the *Plan* component can be validated through scenario-based testing and evaluation of corresponding behavioral requirements (cf. EU 2022/1426 [30, Annex III - Part 1 - 1.3.3]). However, validating the performance of *Sense* components is a particular challenge since severity of perception errors is often not diagnosable without considering situational context and error compensation capabilities of the subsequent *Plan* component. This makes it difficult to specify general quality criteria. Especially in the industrial context, where systems often consist of various components coming from different suppliers, standardized interfaces and verifiable quality criteria will be indispensable for assembling safe, autonomous vehicles. Here, the question arises how requirements for *Sense* components can be formulated. Simply requiring, that, e.g., an object detection module shall always track all surrounding traffic participants perfectly accurate would be an unrealistic requirement and actually not really needed. A human driver for instance has the ability to safely navigate through traffic by focusing on relevant traffic participants, which can even be occluded, and roughly estimating their velocities and intentions. As long as these intentions of relevant traffic participants are correctly predicted and considered for the driver's own actions, the risk of self-inflicted collisions and resulting harm is acceptably low (or else humans would not be allowed to drive). This indicates, that there must be a minimum required performance for the task of perceiving the surrounding environment. Figure 1.1 shows an exemplary situation in urban traffic for an automated driving system. In order to safely turn left, the system needs to predict that the oncoming vehicles will cross the junction and then decide to give way and wait at a safe position. The basis for that is sufficiently accurate detection and tracking of these oncoming vehicles. It is not yet known how accuracy requirements related to the minimum performance of a *Sense* component can be formulated or elicited. Additionally, as long as

[2] Failure Mode Effect Analysis

[3] Fault Tree Analysis


Required accuracy of object positions / distances?
Required accuracy of object velocities / dimensions?
Required stability of object tracks / classifications?

(a) An unprotected left turn scenario with oncoming vehicles **(b)** Surrounding vehicles perceived by the autonomous vehicle

Figure 1.1.: Example of object detection in urban areas: How accurate and stable does the autonomous vehicle need to detect and track the oncoming vehicles?

the existence of such requirements is not shown, the performance of *Sense* components cannot be optimized towards quality criteria, which are actually needed for safe planning. This is the main issue we will address in this thesis.

While autonomous vehicles as a technology and the corresponding quest for their safety clearly bring up novel challenges, the problem of specifying and eliciting quantifiable design criteria for a system is not new to the engineering community. A popular example of such a design criteria problem was faced by avionics engineers throughout the 1920s and 1930s. Back then, they were facing the challenge to find out which engineering requirements to design for, so that aircraft obtain flying qualities which are satisfactory to pilots [118, Chapter 3]. As we know from reliability of today's aircraft, this challenge has been long overcome. Design requirements for aircraft have changed from being qualitative and ill-defined to being quantitative and well-defined — translated by a community of engineers [118, p. 98].

In this thesis, we propose a formal approach for the validation of *Sense* components consisting of three consecutive steps:

1. Creation of a taxonomy regarding *Sense* component inaccuracy
2. Elicitation of verifiable *Sense* component requirements regarding these inaccuracies
3. Evaluation of *Sense* components towards the elicited requirements

In the following, we provide different concepts and methods for each of these steps and illustrate the applicability of our approach with experiments on a real, prototypical autonomous driving stack for urban areas. We show that safety-critical requirements for the *Sense* component of our prototypical autonomous driving stack exist, that they can be identified and how they can be evaluated. To the best of our knowledge, this is the first time that a coherent, formal approach for a decomposed safety validation of *Sense* components is proposed and demonstrated. The next section provides an overview of tackled research questions.

1.1. Research Questions

As motivated previously, we investigate the decomposition of an automated driving system in the context of safety validation. In doing so, we especially focus on the perception module (*Sense*), the subsequent prediction & planning module (*Plan*) as well as the interface in between. A reliable perception constitutes the basis for safe predicting & planning and thus safe operation of automated driving systems. This implies, that a prediction & planning module demands a perception module to deliver a sufficiently accurate conceptualization of the surrounding environment in order to not make hazardous decisions which can potentially lead to harm of traffic participants. However, due to automated driving systems arising as a new technology, there exist no standardized design patterns for both perception and prediction & planning modules yet. Thus, both their implementation and the interface in between can be diverse. Perception modules can e.g., be based on either a single sensor technology or a multitude of different sensors, computer vision algorithms relying on neural networks or rule-based classifiers. Prediction & planning modules can make decisions based on an object list and an HD map or only on a segmented camera image. To better cope with the implementation variety and the complexity of automated driving systems in general, we firstly establish a comprehensive taxonomy regarding dependability threats in the scope of Research Question 1.

Research Question 1

How can dependability threats to automated driving systems pertaining to perception components be characterized?

For the establishment of a comprehensive taxonomy regarding dependability threats to and from perception components, we functionally decompose an automated driving system into components with well-defined tasks to receive precise interfaces. For that matter we extend existing approaches to functional decomposition (cf. [6], [95]) by respecting the individual steps of the perceptual processing chain of automated driving systems (cf. [98, p.47]). Moreover, we adapt the taxonomy for dependability threats by Avižienis et al. [9] (fault, error, failure) for the perception component of automated driving systems. Since we investigate the effect of flawed perception performance on the subsequent prediction & planning module, we deduct what types of perception errors exist in the scope of Research Question 2.

Research Question 2

What types of perception errors do exist and how can they be classified?

When considering the task of perceiving the environment and processing different sensor data, there are several possibilities for the occurrence of errors ranging from the raw scan of the environment up to the generated environmental model. Based on the functional decomposition, which is part of Research Question 1, and by considering common approaches to perceive and describe the environment, we derive possible errors of perception components. However, not every error or inaccuracy has to be relevant for the automated driving system to safely perform its driving task. Thus, we work on the systematization of relevant road users (cf. Research Question 3) in urban areas to facilitate the specification of goal-oriented quality criteria for perception components which have the task of detecting surrounding traffic participants.

Research Question 3

Which perception errors are of relevance for a prediction & planning module?

Furthermore, sensor measurements and features interpreted from these like e.g., object hypotheses or estimated lane boundaries can be subject to uncertainties. For instance, an object's estimated velocity can be inaccurate, or a lane marking segment can be entirely missed. This can e.g., be the effect of environmental influences, sensor noise or functional insufficiencies of components or algorithms. Therefore, the prediction & planning module is required to compensate minor inaccuracies and errors coming from the perception module. Specifying the required performance by knowing which inaccuracies and errors are safety-critical is a non-trivial task, since this relies heavily on both error compensation capabilities of the implemented prediction & planning module and the operational situation the automated driving system is in. Elicitation of the actually needed perception performance is a prerequisite for testing and verifying perception modules and is thus essential for a decomposed validation concept for automated driving systems. Research Question 4 addresses this challenge.

Research Question 4

How can requirements for the perception component be elicited?

While the previous research questions tackle the challenge of specifying sufficiently safe perception performance, the following question investigates the corresponding verification which equals the proof that all specified quality criteria are met by the perception component. From this follows, that the specified requirements and the perception component meeting these must be verifiable. For instance, verifying that a perception component achieves a detection of surrounding traffic participants with centimeter-level accuracy requires that true positions and dimensions of these objects are known (so-called ground truth). While these references are easily ascertainable when running tests in a simulation engine (which then creates the new challenge of simulation validity), they are costly to acquire in the real world. Deploying highly precise reference sensor systems and manually annotating sensor data by humans, e.g., labeling object bounding boxes in camera images, are followed strategies for the generation of real world reference data. However, reference sensor systems can also be adversely affected by environmental influences and manually labeled data is often expensive, unsustainable and can exhibit an unsteady degree of quality. Research Question 5 further discusses this issue.

Research Question 5

How can the costly generation of reference data, which is needed for the evaluation of perception modules, be approached?

1.2. Research Approach

This thesis and its contributions came to life during my time at Volkswagen Group Research and Volkswagen Commercial Vehicles. The different modules which have been used as systems under test, like the prediction & planning module in Chapter 4 or the object detection & tracking module in Chapters 3 and 5 were part of autonomous prototypes on the basis of Volkswagen e-Golfs. The whole autonomous driving stack of these prototypes, i.e. sensor setups, compute and networking hardware and software for localization, object detection, tracking & prediction and behavioral decisions, was designed and created both under the assignment of and directly by Volkswagen Group Research. These prototypes were able to conduct autonomous test drives and were mainly tested in Wolfsburg - both inside and outside the Volkswagen Plant - and in Hamburg, Germany. One of the goals of this thesis and the corresponding research was to investigate, assess and test modules of a real, working autonomous prototype instead of individual, openly available components to demonstrate applicability of the proposed methods and concepts in an industrial context. Still, autonomous driving in general is a new technology and successful introduction of a fully automated system or service to the market on a bigger scale is yet to be shown. Thus, the concepts and methods proposed in the scope of this thesis should not and cannot be seen as validated methods to assess and prove correctness of autonomous driving stack modules. However, we are of the opinion that our concepts, methods and experiments provide answers to existing research questions, create new research questions and show the potential to play a role in the future validation of safe autonomous vehicles.

1.3. Related Work

Working on answers for the different research questions requires an understanding about the state of the art of their associated fields of research. This section covers related work corresponding to the aforementioned parts of our proposed approach for the validation of *Sense* components. This comprises decomposition and classification of perceptual inaccuracy (Research Question 1, 2 & 3), elicitation of accuracy requirements for the perception (Research Question 4) and automatic generation of reference data for perception modules (Research Question 5).

1.3.1. Decomposition and Classification of Perceptual Inaccuracy

In this subsection, we discuss related work regarding the decomposition of automated driving systems, classification of perceptual inaccuracy and consideration of error relevance when assessing *Sense* components.

Functional Decomposition

Behere and Törngren [10] propose to split components of autonomous driving systems into three main categories: *Perception*, *Decisions & Control* and *Vehicle platform manipulation*. They further map components to these different categories (e.g., *Sensing* as part of *Perception*) and compare their architecture to the Observe-Orient-Decide-Act (OODA) model [16], which can be applied to discretize a human driver. Serban et al. [100] provide another functional software architecture for autonomous vehicles. They cluster different multiple functional components to the classes *Sensors Abstraction*, *Data Management*, *Actuators Interface*, *Sensor Fusion*, *World Model*, *Behavior Generation*, *Planning*, *Vehicle Control* and *System and Safety Management*. While both functional architectures provided by Behere and Törngren [10] and Serban et al. [100] already exhibit a high degree of detail and address several tasks that need to be carried out by an autonomous driving system, they do not define explicit interfaces between these components. Another approach is given by Amersbach and Winner [6]. They functionally decompose automated driving systems into six layers based on the human driving task for the definition of particular test cases with well-defined interfaces. The decomposition layers are information access, information reception, information processing, situational understanding, behavioral decision, and action. Their proposed decomposition is not further distinguished into more layers to be applicable for various automated driving systems. However, to define requirements for the perception component there is a need for the definition of dependability threats based on a more specific decomposition of the environmental perception and the subsequent processing into an environmental model. Therefore, in Chapter 2 we build on the decomposition by Amersbach and Winner [6] and focus on the information processing layer by decomposing it further and identifying corresponding dependability threats. While the decomposition by Amersbach and Winner [6] does not go into more detail regarding the task of perceiving the environment, Rosenberger et al. [95] take a closer look into the information processing layer and functionally decompose a lidar sensor system. They define differently abstract interfaces along the lidar data processing chain: the raw scan of the lidar sensor, the resulting point cloud and an object list which contains geometric and physical attributes. These interfaces are then used for a more detailed comparison of real and synthetically generated lidar measurement data using different metrics for different interfaces. Holder et al. [52] show a typical, decomposed signal processing chain for radar sensors. They define raw sensor data to be the lossless representation of radar sensor readings after spectral analysis and prior to thresholding. Both contributions by Rosenberger et al. [95] and Holder et al. [52] are considered by us when talking about the differently abstract representations of sensor data and corresponding errors in Chapter 2.

Classification of Perceptual Inaccuracy

A contribution that deals with the identification of perceptual inaccuracy is provided by Hanke et al. [46]. They examine the construction of a statistical sensor model for the virtual test of automated driving systems. To provide more realistic testing conditions they investigate the integration of lossy perception process characteristics into sensor models. To do so, they define the output interface of the model to consist of several model units, where each of these units deals with one specific perception error. However, their work primarily focuses on objects and does not distinguish between different processing steps of sensor data. A contribution for the classification of perceptual uncertainty is made by Dietmayer [26]. He describes the task of machine perception for automated driving and distinguishes its uncertainty into three uncertainty domains: state uncertainty, existence uncertainty and class uncertainty. State uncertainty deals with uncertainty regarding estimation of state variables such as position, kinematic or size of detected objects. Existence uncertainty refers to the uncertainty whether an object that was perceived actually exists. Class uncertainty describes the uncertainty concerning the semantic classification of detected objects. In Chapter 2, we combine the classification of perception threats with where they can occur along the processing chain by considering differently abstract representations of sensor data. However, due to the different components processing the sensor data and therefore several potential causes for dependability threats arising, there is a need to differentiate these threats. A general approach to classify dependability threats is conducted by Avižienis et al. [9]. They establish basic concepts for the dependability of computing and communicating systems and distinguish threats to dependability into faults, errors and failures and define them subsequently. While faults are causes to errors, errors can propagate and eventually lead to a failure of a subsystem. Moreover, the characteristics of faults, errors and failures are discussed and different measurements to handle dependability threats are addressed. We adapt the definitions of Avižienis et al. [9] to the perception component of automated driving systems in Chapter 2.

Object Relevance

A straight-forward way to assess relevance of surrounding objects is by simply considering attributes like distance to the object [67, 76, 123, 124, 14, 72] or the object's heading [108]. However, urban traffic is complex, distances and headings of objects are dependent on the actual infrastructure and thus object relevance can not be simply broken down to one object attribute [78]. To that end, Volk et al. [119] propose a novel safety metric which identifies zones relevant to potential collisions and rates non-detected vehicles in these areas as safety-critical. Their metric is based on the CLEAR MOT metrics [11] and the *Responsibility Sensitive Safety* (RSS) [101]. Another novel metric is proposed by Wolf et al. [121]. They extend the mean Average Precision metric by not only considering distance to a detected pedestrian but also the corresponding time-to-collision. Philion et al. [81] measure importance of a perceived object by removing it from their scene and then assessing whether the decision of a subsequent planning component significantly changes. Moreover, their corresponding metric can also consider the effect of synthetically added phantom objects. Ivanovic and Pavone [56] develop an exemplary planning-aware prediction metric while also calling for additional research in the field of task-oriented perception evaluation. Chu et al. [23] define a minimum required perception area consisting of a longitudinal and lateral component for the use case of forward obstacle detection. Their calculation method is comparable to the RSS [101] and thus to our basic area corresponding to *Lead and Neighboring Traffic Participants* (cf. Section 3.1.1). The two contributions closest in spirit to our work in Chapter 3 are given by Schönemann et al. [99] and Topan et al. [111]. Schönemann et al. [99] define a safety zone based on maneuvers for the use case of automated valet parking. A superposition of areas derived from six scenarios and five maneuvers defines a required perception range. Shortly after the publication of our relevant area approach [83], Topan et al. [111] have published a different method following the same idea. Their approach also features the construction of relevant areas, which they call safety zones. The corresponding method then builds on Hamilton-Jacobi reachability, a set of possible control actions the objects can execute and modeling vehicle dynamics. This step is contrary to our approach [83] (cf. Chapter 3), since we are mainly considering the road network, identifying which lanes are relevant based on traffic rules and finally following longitudinal safety distances along these lanes. Our focus lies more on the systematization, which addresses the dependencies between different maneuvers and traffic participant classes. While there exist differences between the construction method for relevant areas proposed by Topan et al. [111] and the one proposed by us, the main idea is the same. Future works should therefore build upon both approaches. While several contributions in the context of task-oriented perception evaluation have been made recently, no gold standard has been set yet. In Chapter 3, we aim at contributing to this line of research by providing a method for the construction of relevant areas based on the executed maneuver and a given map of the urban environment.

1.3.2. Elicitation of Accuracy Requirements for the Perception

We first discuss the assessment of *Sense* components in the context of safe *Plan* components, then mention decomposed and structured testing approaches for cyber-physical systems, and lastly touch on the topic of requirement mining.

Sense Assessment for Safe Planning

Stellet et al. [106] point out existing safety validations approaches for automated driving systems which also include a decomposition strategy of combining statistically validated sensing and formally safe planning. They discuss the need to validate sensing towards situations being erroneously considered unsafe and erroneously considered safe, while also stressing that not every perception error must lead to a failure of the overall system. Stahl et al. [105, 104] propose an online verification concept for a *Plan* component. Their concept requires that all objects in the scene have to be detected and perceived properly (without any further specification) in order to assure a safe trajectory. Klamann et al. [64] further emphasize the difficulty of defining pass-/fail criteria on component level. Schönemann et al. [99] propose a fault tree-based definition of general safety requirements for cooperative valet parking following the sense-plan-act paradigm. Among other safety requirements, they derive an allowed object position inaccuracy of 7.5 cm. Requirements for object position accuracy are also investigated and quantified in Chapter 4, using simulation tests instead of a mathematical derivation. While an accurate perception is a prerequisite for safe planning and a safe overall system, general quality criteria which can be assessed to validate a *Sense* component are yet to be defined. In Chapter 4, we are not only defining but also quantifying exemplary requirements regarding acceptable inaccuracies of the *Sense* component for a given *Plan* component.

Decomposed & Structured Testing

The increasing complexity of cyber-physical systems as well as enormous parameter spaces for possible test inputs emphasize the need for novel testing methods. Systematic analysis of input stimuli and compositional falsification are recent approaches to meet the challenge of increasing complexity. Rao et al. [89] discuss the possibility of fault injection at the different interfaces of automated driving systems. However, their shown experiment deals with fault injection within *Sense* components (e.g., adding noise to camera images which are then considered by an object detection module) and not with the consequences of *Sense* errors on a *Plan* component. Fremont et al. [41] perform structured testing to identify scenarios that lead to a failure of a neural network-based aircraft taxiing system by Boeing and subsequently retrain the system to achieve a better performance. Dreossi et al. [28] conduct a compositional falsification of a machine learning-based perception component and an advanced emergency brake system to identify potentially relevant misdetections. While the main focus of their approach lies on a decomposition into machine learning component and the remainder of the system, it corresponds to a split at the interface of *Sense* and *Plan* for their investigated system. However, they do not formulate specific requirements for the *Sense* component based on their experimental results. Tuncali et al. [113, 114] present a framework for test case generation which they utilize to test both a machine learning-based perception component and a collision avoidance controller. They further emphasize the need to not only evaluate *Sense* components isolated but to consider closed-loop behavior of the whole system. While testing strategies of the listed contributions share similarities with our test design, we specifically focus on the performance of a *Plan* component under the influence of synthetically generated *Sense* component errors. The contribution that is the closest in spirit to our research is given by Piazzoni et al. [86, 87]. While also utilizing simulation and handcrafted perception error models, they propose two test cases incorporating different *Sense* errors, i.e. non-detections, tracking loss and position inaccuracies of perceived objects. However, test case results are not aggregated to elicit acceptable errors or requirements. All the error types considered by Piazzoni et al. [86, 87] and mentioned above are analyzed in Chapter 4 and corresponding requirements are subsequently elicited.

Requirement Mining

Another direction of research we want to mention is requirement mining. Both Hoxha et al. [53] and Jin et al. [58] explore properties of a given automatic transmission model that is expressed by a set of ordinary differential equations. By utilizing falsification, temporal logic and optimization methods, they elicit quantified requirements regarding the different input signals. Subsequently, requirement mining is approached with a Pareto optimization and an optimization-based search algorithm respectively and results in a set of quantified requirements. Both contributions are exemplary contributions in the field of requirement mining for given systems. Another mathematical approach within the context of autonomous driving is given by Henze et al. [51]. Henze et al. [51] perform a sensitivity analysis of the *Intelligent Driver Model* by Treiber et al. [112]. The model by Treiber et al. [112] is a car-following model which takes the ego velocity, the velocity of a leading vehicle and the distance to this leading vehicle in account. In return, it calculates an acceleration value for the ego. Henze et al. [51] add stochastic noise to the input signals and formulate a corresponding optimization problem. They later investigate a car following scenario for a convergence analysis and extend the driver model to also examine a roundabout scenario, which they then use for the discussion of possibilities and limitations of their approach. Finally, Henze et al. [51] derive admissible standard deviations for input parameters like ego velocity and distances to other vehicles. The approach by Henze et al. [51] seems fruitful to assess the impact of inaccurate signals on mathematically describable parts of *Plan* components. It would be of interest how their approach can be extended so that not only the impact of noise (and thus *True Positive Inaccuracy*) but also of entirely missing signals (e.g., during *False Negative* errors) can be considered. Our requirement elicitation approach proposed in Chapter 4 relies on structured testing of a black box rather than mathematical optimization, since the prototypical *Plan* component under test cannot be easily described as a mathematical function. We believe, that future contributions should build on the experiences of both the approach by Henze et al. [51] and our structured testing approach.

1.3.3. Automatic Generation of Reference Data for Perception Modules

This subsection gives an overview regarding temporal 3D object detection and the (semi)-automatic generation of ground truth data. All of these topics are related to Chapter 5, in which we propose a method for the automatic generation of dimension and classification values for perceived objects.

Temporal 3D Object Detection

The detection of surrounding traffic participants has been widely researched in the past [40, 7]. However, most approaches focus on single frame inputs while only fewer take also data of previous frames into consideration and thus are also looking back in time. Several contributions in the field of online 3D object detection that also exploit temporal information show the feasibility of considering previous detections to achieve a better performance. Luo et al. [71] demonstrate a fully convolutional end-to-end approach for simultaneous 3D detection, tracking and prediction. Taking previous tracking and prediction information into consideration offers the potential of reducing false negatives. Yin et al. [125] propose a framework for 3D object detection which includes a graph-based network to extract spatial features of individual point cloud frames and an aggregation component to fuse spatio-temporal information. They show the ability to detect objects whose point clouds are sparse due to occlusion. Huang et al. [54] introduce an LSTM-based multi-frame 3D object detection algorithm which assists detection of objects for a given frame by taking previous frames into consideration. While the use-case of online object detection (or detection of other elements) only allows to look at the current and previous frames, an offline process can consider both past and future measurements. This is especially of interest for the generation of ground truth data which is needed as a reference for the evaluation of online perception systems.

Ground Truth Generation with Reference Sensor Systems

One approach to obtain more accurate measurements (like positions, dimensions or velocities of objects) is to utilize reference sensor systems of higher fidelity either equipped on the automated driving vehicle or directly on perceived traffic participants. While Hajri and Rahal [45] mount kinematic sensors to the ego and perceived vehicles, Scheiner et al. [97] utilize portable high accuracy GNSS to obtain accurate position measurements of perceived vulnerable road users. Another approach is demonstrated by Krajewski et al. [65] who use a drone to generate reference values for object poses, velocities and dimensions. Subsequently, they identify perception errors by comparing the reference objects generated from drone data to object hypotheses generated by a vehicle-based perception component.

Semi-Automatic Ground Truth Generation

Furthermore, there exist semi-automatic approaches which assist human annotators and thus accelerate the labeling task. Borkar et al. [15] propose a method where the annotator only needs to select the center of lane markings to start a process which automatically interpolates lane markings. Lee et al. [66] suggest a similar method that relies on manually selecting object centers in lidar point clouds and considers these to automatically generate 3D bounding boxes. Several other works akin to the approach by Lee et al. [66] also demonstrate how to assist human annotators with the task of object labeling [21, 2, 69, 39].

Automatic Ground Truth Generation

An approach that falls into the category of automatic ground truth generation is proposed by Das et al. [25]. They demonstrate an automatic generation method of ground truth lane markings based on camera images, which can be utilized to evaluate lane detection algorithms. Rathore et al. [90] propose a deep-learning-based algorithm that iterates over all 2D bounding boxes of an object in a video and automatically sharpens and thus improves them. There exist few recent contributions in the field of automatic ground truth generation for 3D objects. Zakharov et al. [126] present an automatic labeling pipeline. They predict surface coordinates and vehicle shapes based on 2D objects coming from camera-based off-the-shelf detectors and subsequently fit the coordinates to sparse lidar data in the camera frustum to generate a 3D bounding box. However, their approach only considers single frames and does not take previous or future information into consideration. While they show that their approach is able to recover a substantial amount of vehicle bounding boxes, the accuracy of bounding boxes show potential for improvement. Qi et al. [88] propose an automatic pipeline to generate ground truth 3D object boxes both for static and moving objects. Both processes rely on deep network models to output accurate bounding boxes. Moreover, they emphasize the advantage of object-centric approaches over frame-centric approaches for automated ground truth generation. Yang et al. [122] propose an automatic process to generate object trajectories in 3D space from lidar point clouds. Their approach is decomposed into estimating object dimensions and corresponding motion paths. Yang et al. also rely on improving online measurements, consider that rigid objects (i.e. motorized vehicles) have fixed dimensions through time and thus re-estimate object dimensions. The contributions by Qi et al. and Yang et al. are closest in spirit to our research (cf. Chapter 5). However, while both pipelines consider neural networks to predict object sizes, our approach utilizes concrete situations that are assessed as favorable for estimating object dimensions in a rule-based manner. Additionally, our pipeline includes a reclassification considering a diverse set of traffic participant classes.

1.4. Thesis Structure

Next to the introductory Part I and the concluding Part V at the end, this thesis consists of three main parts corresponding to the three steps of our approach. Part II deals with a taxonomy of perception accuracy. This includes the definition of the terms fault, error and failure in the context of automated driving perception and a taxonomy of perceptual threats, which addresses both Research Questions 1 and 2. In the scope of Research Question 3, we propose a systematic method for the differentiation into potentially relevant and irrelevant traffic participants in urban areas, which can be considered to identify relevant perception errors for a goal-oriented evaluation of an object detection component. Part III contributes answers to Research Question 4 and deals with the elicitation of concrete, verifiable perception requirements via simulation-in-the-loop testing of a given *Plan* component in two different urban scenarios. To that end, we propose a method for the adaptive injection of perception errors with varying severity to assess robustness of a *Plan* component under test. Part IV demonstrates and discusses the evaluation of *Sense* components regarding perception requirements as specified in this thesis. This also includes a reflection on the challenge of reference data generation. Furthermore, we contribute to this challenge by proposing a systematic method to construct dimension and classification labels for perceived traffic participants based on an object list which has beforehand been generated by a real-time object detection pipeline. Thus, this part lies in the scope of Research Question 5.

Table 1.1.: Thesis Structure: Main Body

Part	Chapter	Research Question	Main Contributions
Part II	Chapter 2	Research Question 1	Taxonomy of perceptual threats
		Research Question 2	List of perception errors
	Chapter 3	Research Question 3	Method for construction of relevant areas
Part III	Chapter 4	Research Question 4	Parameterizable perception error models Elicitation of perception requirements by testing
Part IV	Chapter 5	Research Question 5	Revision process for object dimensions and classes
	Chapter 6	-	Evaluation of perception requirements

Part II.

Taxonomy of Perception Accuracy

Decomposing Environmental Perception and its Area of Application

An approach that is already present in other domains and is currently researched for the validation of automated driving systems is the method of functional decomposition [68]. Instead of verifying the complex system as a whole, the verification of less complex single components is examined. Shifting from vehicle level verification to component level verification offers the advantage to apply more specific verification methods for different components. The verification process therefore gains more manageability and flexibility. However, a downside to a decomposition-based verification is that threats have to be accounted for separately, which are not safety-relevant for a single component, but can become safety-critical when propagating along the following components. Regarding an automated driving system, the verification of the perception component is challenging, since standardized, verifiable quality requirements are yet to be defined. The main task of a *Sense* component is to guarantee the detection of all relevant road users [26]. These need to be perceived with a certain quality and in a fixed time interval to ensure safe behavior of the self-driving vehicle in every possible scenario. This requirement, however, is still too vague to be tested and verified considering the ambiguity of the terms *relevant road users*, *certain quality* and *fixed time interval.* Pivotal are hazards on vehicle level, which can mostly only be identified when a *Sense* component is tested in conjunction with the rest of the automated driving stack (*Plan* and *Act*). While a missed object directly in front of the vehicle is very likely to be a safety-critical threat to the system, a missed object in the distance does not necessarily need to be. How accurately (*certain quality*) and quickly (*fixed time interval*) surrounding road users need to be reliably detected is dependent on the performance of both *Plan* and *Act* components and the operational situation the self-driving vehicle is in. In order to formulate meaningful requirements for the environmental perception of an automated driving system, it is essential to identify possible threats for a perception component, where they can potentially originate from and what influence they can have on the whole system performance. Moreover, the identification of possible threats can enable a better understanding of dependability threats a perception component should handle itself, which threats should be handled by subsequent processing and which threats should not occur. Chapter 2 functionally decomposes the *Sense* component and establishes a taxonomy of dependability threats to and coming from the *Sense* component. Chapter 3 shows how the term *relevant road users* can be further refined for urban domains and corresponding operational situations.

2. Functional Decomposition of Automated Driving Systems for the Classification and Evaluation of Perceptual Threats[1]

In this chapter, we establish a taxonomy of perceptual threats regarding automated driving systems. We characterize perceptual threats by functionally decomposing environmental perception components into their constituent processing parts. The resulting interfaces of the decomposed parts can then be used to derive potential dependability threats.

2.1. Perceptual Threats

Robot systems are often distinguished into *Sense*, *Plan* and *Act* components. Adapted to a self-driving vehicle, *Sense* includes the task of perceiving the surroundings and generating a model of the environment. *Plan* subsumes interpreting and predicting of future behavior of surrounding traffic participants based on the environmental model and then choosing a trajectory to be driven. *Act* stands for executing the planned trajectory by steering and accelerating or braking while also performing actions like indicating lane changes. This cycle is repeated for every scene[2]. A more detailed decomposition of automated driving systems is conducted by Amersbach and Winner [6]. Figure 2.1 shows the decomposition layers of Amersbach and Winner [6] mapped onto *Sense*, *Plan* and *Act* components.

Due to automated driving functions being highly complex systems consisting of various components, it is essential to identify the factors that can lead to safety-relevant system failures. In this section, we propose a taxonomy for the classification of dependability threats to automated driving systems while focusing on the perception component. For that, we stick closely to the concept of faults, errors and failures introduced by Avižienis et al. [9] while also considering the differently abstract levels of sensor data representing the environment.

Avižienis et al. [9] define a fault as cause of an error. They distinguish between internal and external faults of a system. When a fault causes an error, it is active, otherwise it is dormant. An error is part of the total state of the system. When one or multiple errors cause the delivered service of the system to deviate from correct service, a failure occurs.

We assume that errors can occur in every step of processing environmental sensor data. Therefore, we have to look at the data each component provides to the following component. The raw scan of the surrounding environment is processed into a model of the surrounding environment and therefore exists in differently abstract levels during the processing. Considering

[1] This chapter is based on Paper I [84] and therefore contains verbatim content previously published.

[2] We adopt the definitions of scene and scenario by Ulbrich et al. [115]

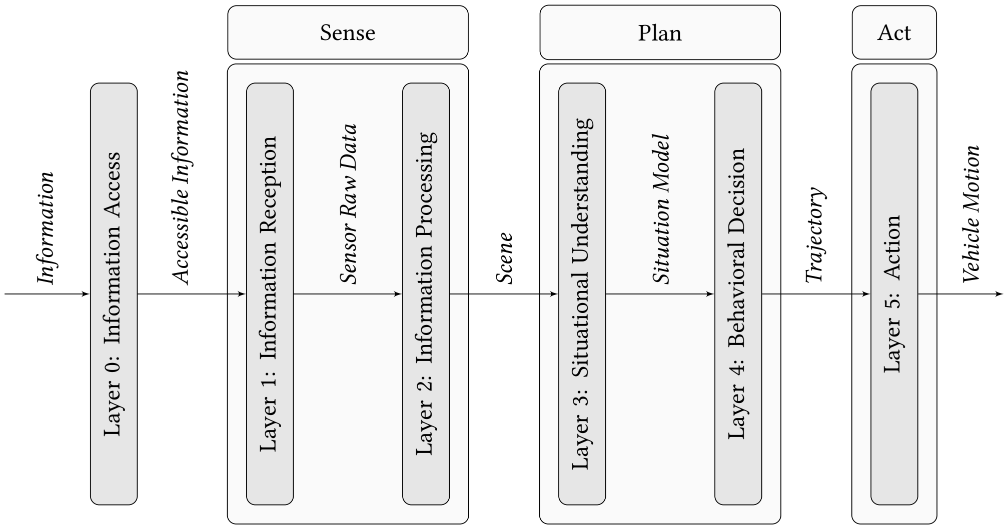

Sense
Plan
Act
Information
Layer 0: Information Access
Accessible Information
Layer 1: Information Reception
Sensor Raw Data
Layer 2: Information Processing
Scene
Layer 3: Situational Understanding
Situation Model
Layer 4: Behavioral Decision
Trajectory
Layer 5: Action
Vehicle Motion

Figure 2.1.: Functional decomposition by Amersbach and Winner [6] mapped onto the *Sense-Plan-Act*-Paradigm

the functional system architecture of automated driving functions [98, p.47] on the lowest level, there is a raw scan of the environment consisting of the data generated by the different sensors. Based on that different features like objects, traffic signs or road markings are detected. On the highest level all features are merged into a scene - a representation model of the environment. Figure 2.2 illustrates the processing of environmental sensor data and summarizes where the dependability threats, which are introduced in the following, can occur.

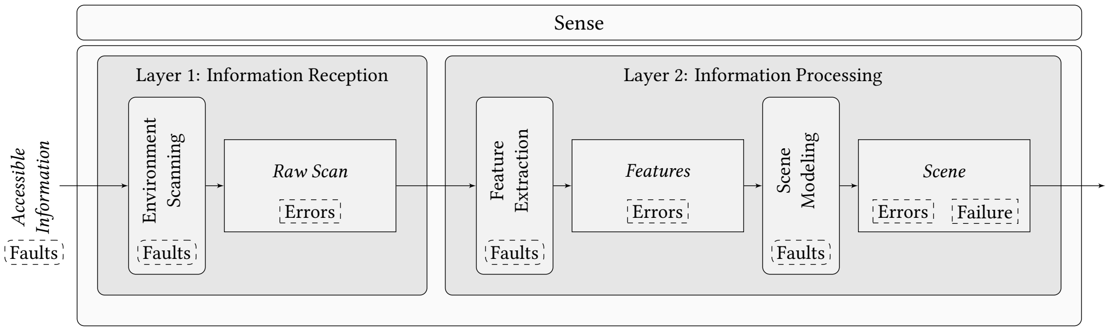


Figure 2.2.: Processing chain of the *Sense* component and potential occurrences of dependability threats relating to the *Sense* component as a system

2.1.1. Fault

Referring to our taxonomy, a fault is the cause of a perception error. Considering that there are different types of perception errors, there are also different types of faults to the perception component subsequently. On the one hand, errors that are propagating along the processing chain can be seen as faults to the resulting errors. On the other hand, each processing step of sensor data can contain its own faults (cf. Figure 2.2). When creating a raw scan of the environment, there are two types of faults: external faults and internal faults. External faults are disturbance variables like environmental conditions which can obscure the accessible information. Internal faults are either linked to the hardware, e.g., a systematic measurement error of a sensor, or are anchored in the software, e.g., a flawed point cloud generation out of received lidar beams. Faults to the processing of the raw scan into features are e.g., bugs in the object segmentation based on point clouds or images. When generating a scene, faults are either errors on feature level or present because of flaws in the scene modeling. An exemplary fault on this level is e.g., an incorrect lane matching algorithm for perceived vehicles. Following the terms and definitions of the ISO 21448 [55], external faults can be seen as triggering conditions for the *Sense* component and internal faults (which are not related to functional safety of E/E systems) can be seen as functional insufficiencies of the *Sense* component [128].

2.1.2. Error

Each of the different representations of the environment can be inaccurate and therefore be subject to errors (cf. Figure 2.2). Examples for errors in these differently abstract representations are e.g., a blurred camera image on raw scan level, an object that is seen which is not existent on feature level and a correctly perceived traffic light that is, however, linked to an incorrect lane on scene level. According to Avižienis et al. [9], many errors do not affect the system's external state.

2.1.3. Failure

According to Avižienis et al. [9] a system failure occurs when the delivered service deviates from correct service. In terms of the environmental perception, the question arises what correct service of the perception component of an automated driving system comprises. Considering the task of perceiving surrounding road users, correct service is delivered by a perception component when all relevant road users are detected [26] with a certain quality within a fixed time interval. Moreover, the road users have to be correctly matched to the traffic infrastructure. Hence, the delivered service deviates from correct service when either not all relevant road users are accurately enough seen or when there is a relevant mismatch in the modeled scene. In this case, the automated driving system would not be able to evaluate the situation appropriately anymore and would therefore not be capable of performing its driving task safely enough.

2.2. Classification of Perception Error Types

In the following, both errors on raw scan level and on feature level are examined. To that end, raw data errors for the sensor technologies camera, lidar and radar are briefly discussed. Consecutively, we will derive errors on feature level by individually considering the single parts that make up the environment. While doing so, we are also referring to commonly used approaches on how this accessible information is included into the scene modeling.

2.2.1. Raw Scan

Errors on raw scan level are anchored in the raw data [52] generated by the deployed sensors. Due to the fact that different types of sensors generate different kinds of raw data, it is not possible to define common errors on this level of environmental representation which are applicable for every type of sensor. Instead, the raw data of the different sensor types has to be looked at separately. Raw data generated by a camera are in general images consisting of pixels. Image noise due to the level of illumination or image distortions caused by effects like rolling shutter are therefore examples for camera raw data errors, as well as whole missing image sections (e.g., missing traffic signs due to flickering when capturing a variable-message sign over time). A lidar sensor emits laser beams into the environment and measures their echoes. For each laser beam, a measured distance is recorded and, depending on the sensor implementation, other values like intensity or echo-pulse-width are also obtained. Therefore, the raw data of a lidar consists of tuples of measured values. [95] Uncertainties in these measurement tuples due to noise, non-measured echoes or broken down channels can be considered as lidar raw data errors. According to Holder et al. [52], raw data of a radar is defined as the range-doppler-beam spectrum at the interface after the spectral analysis of the sensor readings and before the subsequent post-processing, which typically starts with a thresholding. Common distortions that occur in these raw data are defined as artifacts by Holder et al. [52]. Since these artifacts obscure the accessible information, they can be seen as errors. Causes and thus faults of such artifacts are e.g., mirror reflections, aliasing or electronic noise inside the radar [52].

2.2.2. Features

Errors on feature level are dependent on the different features that are considered for the scene modeling. For the definition of errors on this level it does not matter which kind of raw data was considered to extract the feature. Errors regarding features can be derived by looking at the elements which the environment consists of and how these are typically modeled. According to Ulbrich et al. [115], the environment consists of movable objects and the scenery. The scenery is then split up into the lane network, vertical elevation, stationary elements and environment conditions. Lanes and conflict areas belong to the lane network. Stationary elements are e.g., obstacles, curbs and traffic signs/lights. Figure 2.3 illustrates the decomposed environment elements.

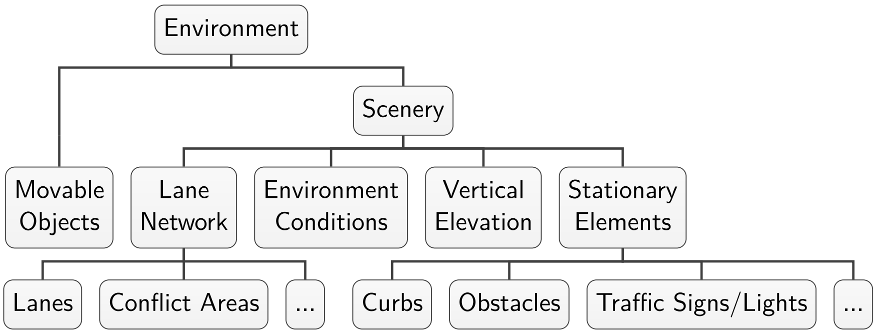


Figure 2.3.: Elements of the environment according to Ulbrich et al. [115]

One part of the environmental perception is to detect existing movable objects. Whenever an object is not detected, an object is missed by the environmental perception. A non-existing movable object, that is detected, is called a phantom object. Both of these cases can increase the risk during automated driving. But even when an existing object is perceived, there is an inaccuracy that comes with every measurement. Ideally a movable object is represented by one bounding box instead of multiple ones. Regarding static non-continuous attributes of movable objects, like the classification, it is trivial to define that any deviation from the real classification is an error. However, concerning attributes that are continuous (e.g., dimensions) and attributes that are additionally dynamic and therefore can change over time (e.g., position and kinematics), it is not obvious when an inaccuracy could propagate into a safety relevant error. This depends on the relevance of the perceived objects to the driving task as well as the robustness of the *Plan* component. Possible errors regarding movable objects are summarized in Figure 2.4.

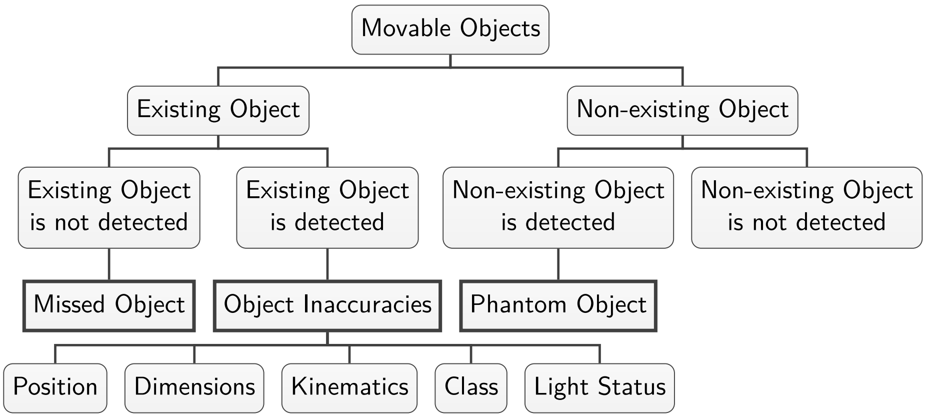


Figure 2.4.: Errors regarding movable objects

Traffic signs and lights are mandatory for managing traffic flow. For an automated driving system to abide by the road traffic regulations, traffic signs and lights need to be correctly captured, matched to their corresponding lanes and considered for predicting and path planning. Regarding the definition of perceptual errors related to traffic signs, we differentiate between missed traffic signs, phantom traffic signs and correctly perceived traffic signs, which are, however, afflicted with inaccuracies. Because traffic signs are static (unlike movable objects), it is easier to define when an inaccuracy might propagate into a safety relevant error. The position of the traffic sign needs to be captured accurately enough to be correctly matched to its corresponding lane. For the interpretation of the traffic sign, both the class (e.g., a speed limit) and the value (e.g., $80\,\mathrm{km\,h^{-1}}$) have to be recorded correctly. While the value of most traffic signs does not change over time, traffic lights and variable-message signs are dynamic elements and therefore do not exclude changes regarding their value (e.g., a traffic light changing from green to yellow). Figure 2.5 summarizes the introduced errors.

Lanes are defined by lane markings and curbs which imply the lane boundaries. Multiple lane marking segments form a continuous lane marking. For the automated driving system to estimate lane boundaries, the lane marking segments need to be captured by the environmental

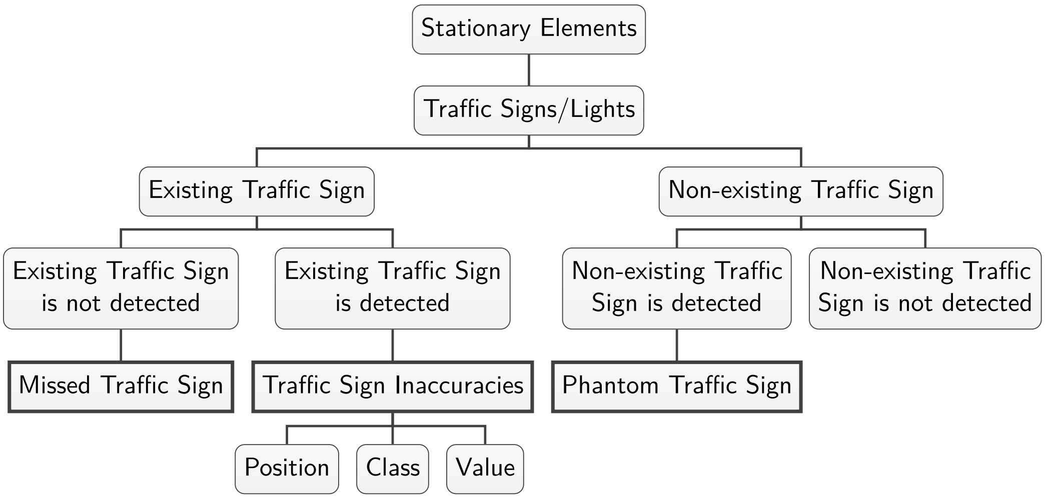

Stationary Elements
Traffic Signs/Lights
Existing Traffic Sign
Non-existing Traffic Sign
Existing Traffic Sign is not detected
Existing Traffic Sign is detected
Non-existing Traffic Sign is detected
Non-existing Traffic Sign is not detected
Missed Traffic Sign
Traffic Sign Inaccuracies
Phantom Traffic Sign
Position
Class
Value

Figure 2.5.: Errors regarding traffic signs

perception. Moreover, overlapping lanes form conflict areas. We define overlooked lane marking segments as missed lane marking segments and detections of non-existing lane marking segments as phantom lane marking segments. Detected lane marking segments can be inaccurate in regard to their exact position and characteristics (e.g., curvature) and their class (e.g., solid, dashed, curbs), which also includes color (usually white or yellow). The class attribute is mandatory to know whether a lane boundary can legally be crossed. Any deviation from the real class can subsequently be considered as an error. Position and characteristics of lane marking segments are continuous values and need to be accurate enough to create a precise lane network. Errors regarding lane marking segments are summarized in Figure 2.6.

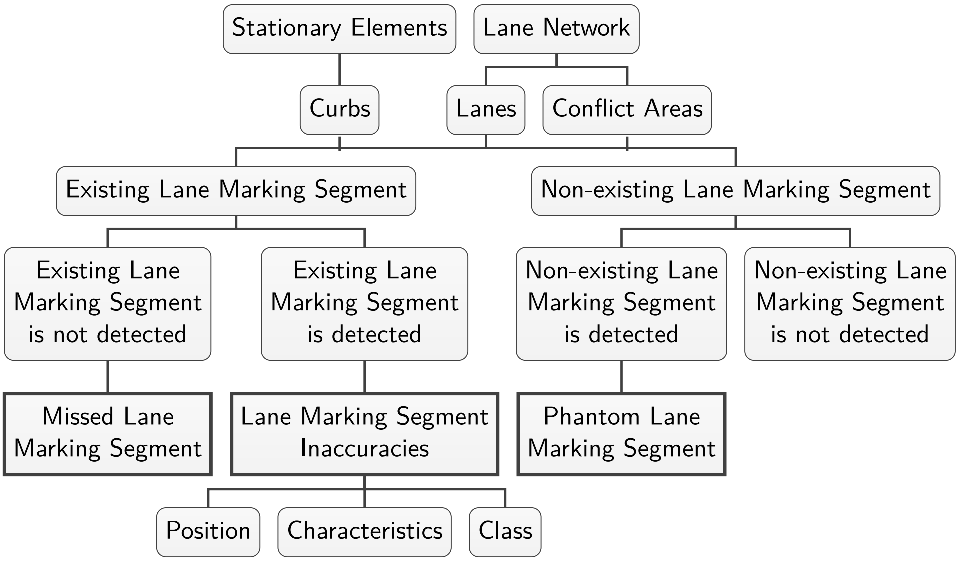

Stationary Elements
Lane Network
Curbs
Lanes
Conflict Areas
Existing Lane Marking Segment
Non-existing Lane Marking Segment
Existing Lane Marking Segment is not detected
Existing Lane Marking Segment is detected
Non-existing Lane Marking Segment is detected
Non-existing Lane Marking Segment is not detected
Missed Lane Marking Segment
Lane Marking Segment Inaccuracies
Phantom Lane Marking Segment
Position
Characteristics
Class

Figure 2.6.: Errors regarding the lane network

One approach to capture vertical elevation is by estimating the ground plane. This information is not only important for path planning, but can also be used to improve quality of object detection [93]. Regarding a point in the environment, it either belongs to the ground plane or not. Subsequently, errors regarding ground mark classification are either overlooked ground marks or misleadingly classified ground marks (cf. Figure 2.7).

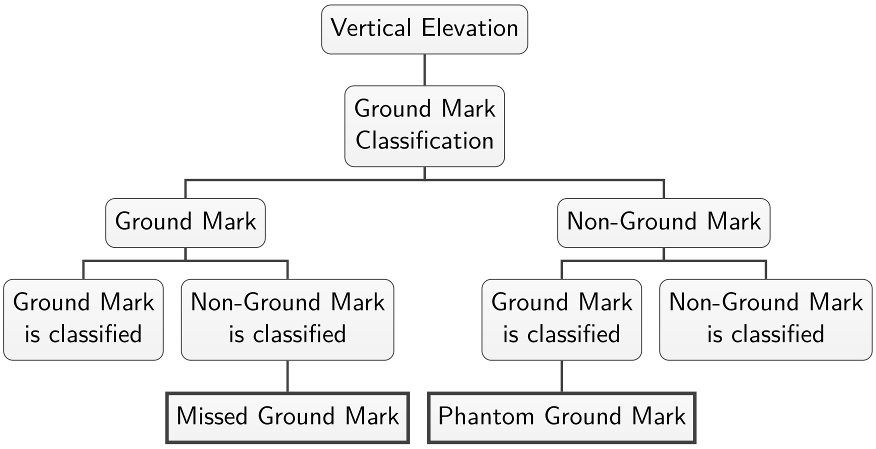

Vertical Elevation
Ground Mark Classification
Ground Mark
Non-Ground Mark
Ground Mark is classified
Non-Ground Mark is classified
Ground Mark is classified
Non-Ground Mark is classified
Missed Ground Mark
Phantom Ground Mark

Figure 2.7.: Errors regarding ground mark classification

The integration of surrounding obstacles and not accessible areas into the path planning of a robot system is often implemented by creating an occupancy grid. For the creation of an occupancy grid, the environment is divided into grid cells. Afterwards, for each grid cell it is determined whether the cell is occupied or not. Hence, possible errors regarding the occupancy grid are either occupied cells which are classified as not occupied (overlooked obstacle) or not-occupied cells which are misleadingly classified as occupied (not-existing obstacle) (cf. Figure 2.8).

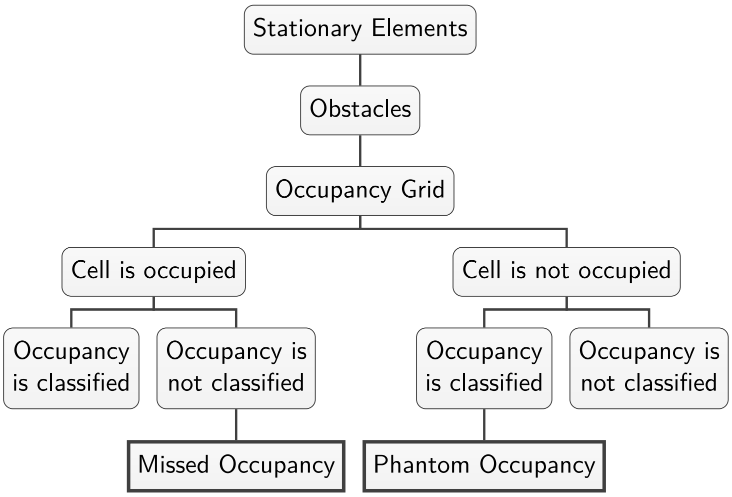


Figure 2.8.: Errors regarding occupancy

2.3. Case Example: Lane Keeping Assistance System

To conclude this chapter and show the applicability of our presented taxonomy, we consider a lane keeping assistance system as case example and its handling of exemplary dependability threats in a hypothetical scenario. Task of the considered assistance system is to detect lane marking segments in a camera image, model them to lanes and subsequently assist the driver with lateral control of the vehicle to keep the lane. Figure 2.9 shows the functional architecture of the *Sense* component of the exemplary system and one possible hazard, which is analyzed in the following.

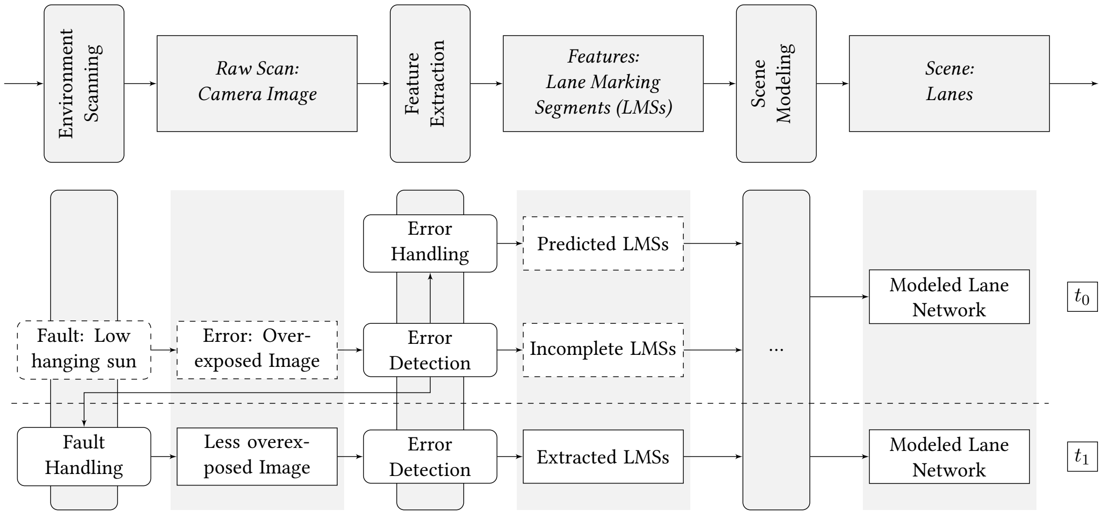


Figure 2.9.: Case Example: Occurence of exemplary dependability threats for a Lane Keeping Assistance System

We now consider for the system to run into a scenario where the correct service cannot be maintained without making adjustments. While the camera captures lane marking segments, we assume a low hanging sun to blind the camera for a short time and therefore cause overexposed images. That results in *Errors* in the raw data because the image misses parts of the environment and therefore does not represent all the accessible information. Extraction of lane marking segments based on these images leads to *Missed Lane Marking Segments* and therefore an incomplete set of lane marking segments (*Errors* on feature level). This would correspond to a *Failure* of the scene modeling since a lane network cannot be accurately modeled anymore and lane keeping cannot be guaranteed. However, our hypothetical system features compensation mechanisms. To deal with these *False Negative Errors*, the hypothetical system contains a component for *Error Detection*, which can trigger a *Fault Handling* mechanism of the environment scanning to avoid *Errors* in the images of upcoming iterations and trigger *Error Handling* to cope with defective images for the current iteration. According to Avižienis et al. [9], the combination of *Fault Handling* and *Error Handling* form *System Recovery*.

In this case example, *Error Handling* is implemented by *Compensation* (cf. [9]). The compensation comprises relying on predicted lane marking segments that were generated during feature extraction of earlier iterations (e.g., by using a Kalman-Filter). Both predicted and the set of incomplete lane marking segments are then provided to the subsequent lane modeling, which is able to generate a sufficiently accurate model of the lane. Simultaneously to *Error Handling*, *Fault Handling* in the environment scanning is triggered. To cope with the low hanging sun and to avoid *Errors* in the camera images, camera settings are reconfigured (cf. highlight compensation (HLC)). Therefore, according to Avižienis et al. [9] *Fault Handling* in this case means *Reconfiguration*. This results in less overexposed camera images for upcoming iterations. Based on the executed *System Recovery*, lane marking segment extraction and subsequent lane modeling can then be sufficiently accurate again for the *Sense* component to deliver correct service without considering predicted lane marking segments of an earlier iteration.

3. Systematization of Relevant Road Users for the Evaluation of Autonomous Vehicle Perception[1]

In this chapter, we address the challenge of specifying relevant road users by proposing a method for the systematic definition of relevant areas in urban traffic situations. When evaluating perception components, traffic participants inside these areas can be considered relevant to enable a task-oriented perception evaluation. Our approach covers the construction of six basic areas which are linked to specific maneuvers in urban traffic. The areas are shaped by the surrounding infrastructure defined by a given map and worst-case assumptions about the potential behavior of traffic participants. In the following, we will discuss construction of each basic area and show validity by analyzing exemplarily situations from urban traffic and their respective, automatically generated, aggregated relevant areas.

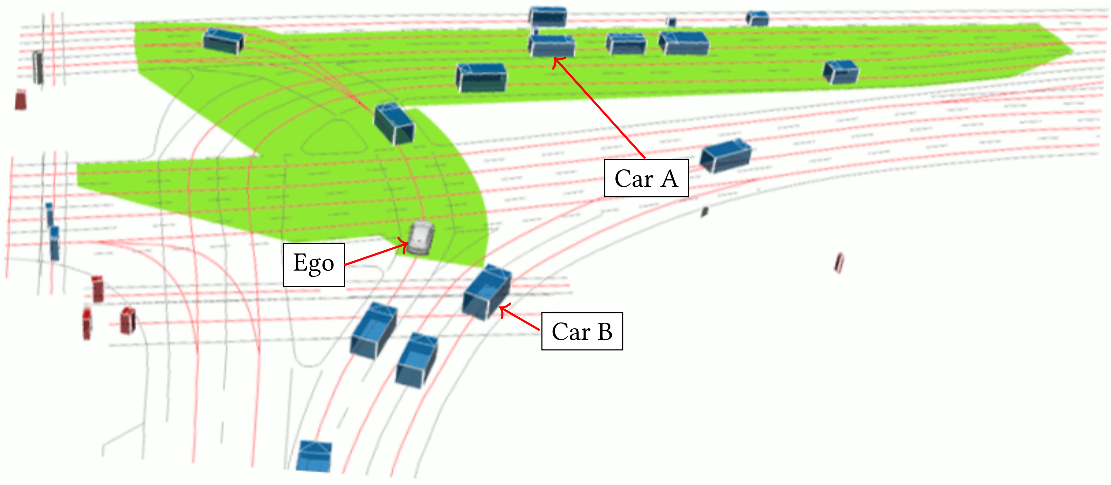


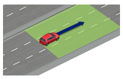
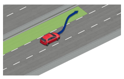
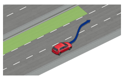
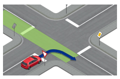
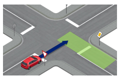
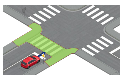
Figure 3.1.: Example of object relevancy in urban traffic: The ego vehicle is performing a left turn maneuver. Areas of main interest (green) are therefore the left turn lanes as well as crossing and merging lanes coming from left and right. Note, that although *Car B* is closer to the ego vehicle, *Car A* is objectively of higher relevance in this situation. (©2022 IEEE)

[1] This chapter is based on Paper IV [83] and therefore contains verbatim content previously published (©2022 IEEE).

3.1. Relevant Areas

For the systematic identification of relevant basic areas which need to be considered by a vehicle operating in urban traffic, we firstly decompose a vehicle's driving behavior into basic maneuvers. Our concept builds on the set of basic infrastructure-related maneuvers for urban traffic proposed by Hartjen et al. [49]. Considered infrastructure elements and corresponding maneuver sets defined by Hartjen et al. [49] are vehicle driving lanes (*Follow Lane*, *Lane Change*), junctions (*Approach Junction*, *Cross Junction*, *Turn Left*, *Turn Right*, *U-Turn*) and crosswalks for vulnerable road users (*Approach Crosswalk*, *Cross Crosswalk*). Executing these maneuvers safely requires monitoring of different surrounding areas on and off the road which can, e.g., comprise driving lanes, crossings or sidewalks. We identify groups of traffic participants which correspond to different maneuvers and specific surrounding areas. Afterwards, we construct basic areas based on the concrete road network, traffic regulations and worst-case assumptions about the movement of hypothetical traffic participants. The proposed basic areas are named after the group of traffic participants they are addressing. A mapping between our basic areas and driving maneuvers shows which areas are deemed to be relevant in which situation. The overview and mapping of our basic areas can be seen in Table 3.1. In the following, we will further describe the construction of each area. Distances which stretch the areas are always measured between vehicle bodies.

Table 3.1.: Mapping of proposed basic areas and infrastructure maneuvers (©2022 IEEE)

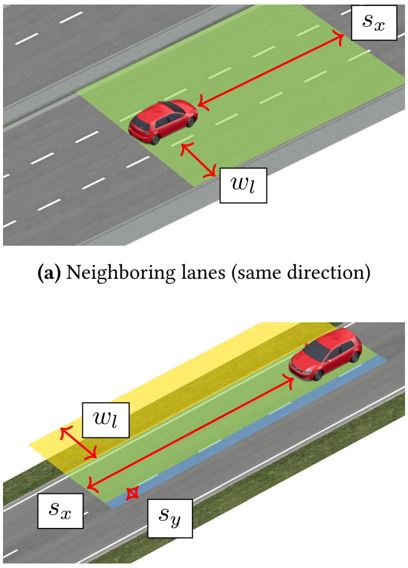
Road User Groups	Lead and Neighboring Traffic Participants	Approaching Vehicles in Lane Change Lane	Approaching Vehicles next to Lane Change Lane	Vehicles Coming from Merging Lanes	Vehicles in Crossing Lanes	VRUs on Crosswalks
Infrastructure Maneuver						
Follow Lane	✓	–	–	–	–	–
Lane Change	✓	✓	✓	–	–	–
Approach Junction	✓	–	–	✓	✓	–
Cross Junction	✓	–	–	✓	✓	–
Turn Left	✓	–	–	✓	✓	✓
Turn Right	✓	–	–	✓	–	✓
U-Turn	✓	–	–	✓	–	✓
Approach Crosswalk	✓	–	–	–	–	✓
Cross Crosswalk	✓	–	–	–	–	✓

3.1.1. Lead and Neighboring Traffic Participants

The decision-making process of an automated vehicle operating in urban traffic is primarily influenced by adjacent traffic participants. Therefore, we deem lead and neighboring vehicles to always be relevant to the driving task, which means that they are relevant to any driving maneuver (cf. Table 3.1). The corresponding basic area comprises a longitudinal (leading vehicles) and a lateral component (neighboring vehicles).

Longitudinal expansion

The longitudinal component of the basic area is determined by estimating the braking distance that the ego vehicle requires until coming to a full stop. This safe longitudinal distance is calculated similarly to the one proposed by Shalev-Shwartz et al. [101]. We also consider a worst-case scenario in which the ego vehicle with current velocity v_0 and possible maximum acceleration $\widehat{a}_x$ realizes after a reaction time t_e that it needs to come to a full stop and thus brakes with at least a minimum deceleration of $\breve{a}_x$. The resulting distance consists of three addends: the distance covered during the reaction time due to acceleration, the distance covered during the reaction time due to the current velocity and the actual distance needed to come to a full stop while decelerating (cf. Equation (3.1)). The relevant area is longitudinally stretched by following the geometry of the vehicle's driving lane for the calculated distance (also considering curvature).



(a) Neighboring lanes (same direction)

(b) Sidewalk and oncoming lane

	Distances
s_x	Long. distance
s_y	Lat. distance
w_l	Lane width

	Parameters
t_e	Ego reaction time
v_0	Ego velocity
$\widehat{a}_x$	Max. acceleration
$\breve{a}_x$	Min. deceleration
$\widehat{a}_y$	Max. acceleration
$\breve{a}_y$	Min. deceleration
t_o	Obj. reaction time

(c) Variables

$$s_x = \frac{\widehat{a}_x \cdot t_e^2}{2} + v_0 \cdot t_e + \frac{\left(\widehat{a}_x \cdot t_e + v_0\right)^2}{2 \cdot \breve{a}_x} \tag{3.1}$$

$$s_y = \frac{\widehat{a}_y \cdot t_e^2}{2} + \frac{\left(\widehat{a}_y \cdot t_e + v_0\right)^2}{2 \cdot \breve{a}_y} + \frac{\widehat{a}_y \cdot t_o^2}{2} + \frac{\left(\widehat{a}_y \cdot t_o + v_0\right)^2}{2 \cdot \breve{a}_y} \tag{3.2}$$

(d) Equations for longitudinal and lateral safe distance

Figure 3.2.: Construction of area for *Lead and Neighboring Traffic Participants* (©2022 IEEE)

Lateral expansion

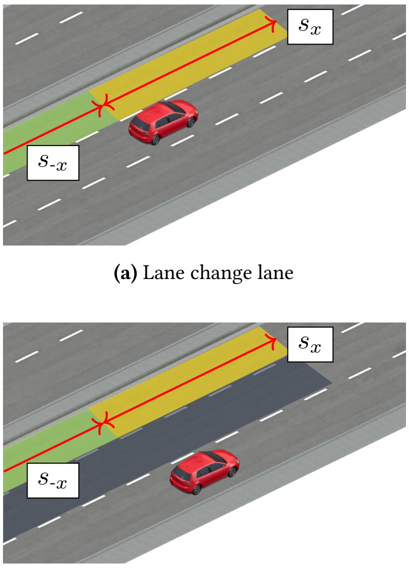
Regarding the basic areas lateral expansion, different cases need to be considered. If there is a neighboring lane with the same driving direction, then it should also be fully observed (lane width) due to vehicles potentially cutting into the lane (cf. Figure 3.2a). If there is no neighboring lane with the same driving direction, either there is the roadside (hard shoulder, sidewalk) or there is a driving lane with opposite driving direction (cf. Figure 3.2b). For the first case we define to extend the area by a lane width to include parallel moving VRUs like cyclists, pedestrians or other objects being close to the road. For the second case we determine a lateral safety distance s_y which is based on the one given by Shalev-Shwartz et al. [101]. We consider that an oncoming vehicle may accelerate in lateral direction towards the ego vehicle with $\widehat{a}_y$ for a duration t_o and then brake laterally with a deceleration of at least $\breve{a}_y$ until there is no lateral velocity (cf. addend 3 and 4 of Equation (3.2)). The same lateral acceleration profile is also assumed for the ego vehicle ($\widehat{a}_y$, t_e, $\breve{a}_y$) (cf. addend 1 and 2 of Equation (3.2)). Both resulting distances are added up (s_y). A summary of the area is given in Figure 3.2.

3.1.2. Approaching Vehicles in the Lane Change Lane

When changing to a parallel lane (*Lane Change*), faster vehicles coming from behind need to be considered. Since incoming vehicles have right of way, the ego vehicle needs to yield for them. Therefore, the ego vehicle should not cause incoming vehicles to slow down by cutting into their longitudinal safety distance. The safe distance of a vehicle with an assumed maximum velocity $\widehat{v}$ (e.g., the current speed limit) and a minimum braking force $\breve{a}_x$ consists of two addends (cf. Equation (3.4)). The first addend describes the distance an incoming vehicle covers before it realizes after a reaction time t_o, that it needs to brake. The second addend describes the braking distance of the oncoming vehicle. Both addends add up to a rearward distance $s_{\text{-}x}$. Additionally, other vehicles or obstacles in front of the ego vehicle in the lane change lane need to be taken into account, thus resulting in the same longitudinal safety distance s_x already known from the previous area. Hence, the resulting basic area for *Approaching Vehicles in the Lane Change Lane* includes the neighboring lane in its entire width for the aforementioned distances. In case that the neighboring lane behind the ego vehicle splits into two ore more lanes before the map is traversed backwards for the whole distance, all of the lanes are included in the basic area since an oncoming vehicle may approach from either one. A visualization of the area can be seen in Figure 3.3a.

3.1.3. Approaching Vehicles Next to the Lane Change Lane

Urban scenarios in larger cities can feature road networks which consist of more than two parallel lanes with the same driving direction. In that case, whenever a lane change is to be attempted in such a situation, not only incoming vehicles or vehicles running ahead in the lane to be changed into are relevant, but also vehicles potentially changing into the same lane from the other side need to be considered. The basic area for *Approaching Vehicles Next to the Lane Change Lane* is constructed similarly to the previous area for *Approaching Vehicles in the Lane Change Lane* (see Figure 3.3b). The area is visualized and summarized in Figure 3.3b.



(a) Lane change lane

(b) Lane next to lane change lane

	Distances
s_x	Long. distance
$s_{\text{-}x}$	Rearward distance
w_l	Lane width

	Parameters
t_e	Ego reaction time
v_0	Ego velocity
$\hat{a}_x$	Max. acceleration
$\breve{a}_x$	Min. deceleration
t_o	Obj. reaction time
$\hat{v}$	Max. obj. velocity

(c) Variables

$$s_x = \frac{\hat{a}_x \cdot t_e^2}{2} + v_0 \cdot t_e + \frac{(\hat{a}_x \cdot t_e + v_0)^2}{2 \cdot \breve{a}_x} \tag{3.3}$$

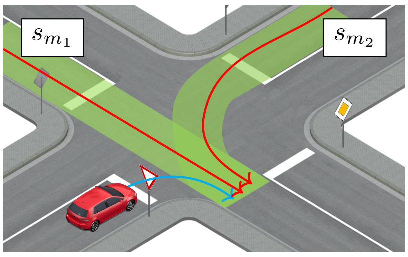
$$s_{\text{-}x} = \hat{v} \cdot t_o + \frac{\hat{v}^2}{2 \cdot \breve{a}_x} \tag{3.4}$$

(d) Equations for safe distances when changing lane

Figure 3.3.: Construction of areas for *Approaching Vehicles in the Lane Change Lane* and *Approaching Vehicles Next to the Lane Change Lane* (©2022 IEEE)

3.1.4. Vehicles Coming from Merging Lanes

Intersection scenarios can be challenging due to a variety of lanes crossing or merging into each other. As other traffic participants may enter the intended path of the ego vehicle, these must be taken into consideration when performing intersection maneuvers (*Approach Junction*, *Cross Junction*, *Turn Left*, *Turn Right*, *U-Turn*). To that end, all relevant lanes merging into the ego vehicle's path are identified. This is done by considering the intended lanes being taken by the ego vehicle and traversing back and forth through the road network. Subsequently, a single relevant area is constructed for each identified merging lane. For each identified lane, we consider the following: The ego vehicle should not impede or endanger other traffic participants when merging into a shared lane. Hence, the time $t_{e,i}$ which is needed for the ego vehicle to reach the shared lane as well as the distance another traffic participant may cover in this time need to be estimated. A merging distance s_m can be calculated by taking both values into consideration and additionally respecting a safety distance which should be upheld without forcing the other traffic participant to brake. By following the entire lane backwards for the distance from the point of merging, an area is constructed (cf. Figure 3.4). The distance is calculated similar to the rearward distance $s_{\text{-}x}$ from the previous areas, but increased by the distance the other traffic participant travels for the time $t_{e,i}$. Hence, this distance decreases and the corresponding area shrinks the closer the ego vehicle is to the point where both lanes merge. There are several ways to approach the estimation of time $t_{e,i}$. When constructing the areas in a post-processing step, the real time the ego vehicle needs to reach the point of merging is known. While this time can be considered for most situations where the point of merging is reached without any delay or disturbance, the rearward distance becomes unrealistically large when the ego vehicle takes an unusual long time (e.g., because of traffic congestion). To cope with this issue as well as addressing the online use case of being able to construct this basic area during operation, a time $t_{e,i}$ can be estimated by considering an acceleration profile and the remaining distance to the point of merging. This acceleration profile can be complex (e.g., taking road curvatures into account). Howerver, for the exemplary application of our concept, we assume that the ego vehicle accelerates with a constant acceleration until the speed limit is reached to forecast its movement.



(a) Merging lanes

	Distances
s_m	Merging distance

	Parameters
$\breve{a}_x$	Min. deceleration
t_o	Obj. reaction time
$\widehat{v}$	Max. obj. velocity
$t_{e,i}$	Ego merging time

(b) Variables

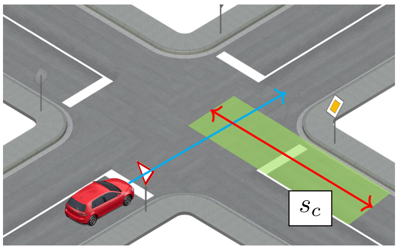
$$s_m = \widehat{v} \cdot t_o + \frac{\widehat{v}^2}{2 \cdot \breve{a}_x} + t_{e,i} \cdot \widehat{v} \tag{3.5}$$

(c) Equation for merging safe distance

Figure 3.4.: Construction of area for *Vehicles Coming from Merging Lanes* (©2022 IEEE)

3.1.5. Vehicles Coming from Crossing Lanes

A counterpart to merging lanes in intersections are crossing lanes. Traffic participants that are not merging into the intended ego lane, but are crossing its path for a short time, are also relevant to the driving task. For right-handed traffic, lanes with a different driving direction are usually crossed when turning left (cross and oncoming traffic) or crossing a junction (cross traffic). If one of these two maneuvers is to be executed, crossing lanes also become already relevant when approaching the junction. Corresponding maneuvers are therefore *Approach Junction*, *Cross Junction* and *Turn Left*. As a first step, relevant crossing lanes are identified. For each of these crossing lanes, an area is constructed. Similarly to the construction of the area for *Vehicles Coming from Merging Lanes*, we estimate a time $t_{e,i}$ that the ego vehicle needs to reach the point of intersection and consider a maximum velocity $\widehat{v}$ (e.g., speed limit) to calculate the distance a crossing vehicle might cover. We are not taking a safety distance between the traffic participant and ego vehicle into account since the ego vehicle does not remain in the crossing lane. For a more conservative specification, either a safety distance could be added (cf. EU 2022/1426 [30, Annex III - Part 1 - 1.3.3.]) or the ego reaching the exit of the junction could be considered (instead of the point of intersection). Again, each identified crossing lane is considered in its whole width and traversed back from the point of intersection for the corresponding calculated distance s_c to construct a basic area (cf. Figure 3.5).



(a) Crossing

	Distances
s_c	Crossing distance

	Parameters
$\widehat{v}$	Max. obj. velocity
$t_{e,i}$	Ego crossing time

(b) Variables

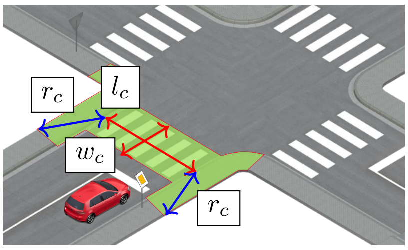
$$s_c = t_{e,i} \cdot \widehat{v} \tag{3.6}$$

(c) Equation for crossing distance

Figure 3.5.: Construction of area for *Vehicles Coming from Crossing Lanes* (©2022 IEEE)

3.1.6. Vulnerable Road Users on Crosswalks

The last basic area addresses vulnerable road users (VRUs) crossing the road via designated crossing aids such as crosswalks, traffic islands or bike lanes. While there exist traffic lights for some of these crossing aids to protect vulnerable road users, other elements like crosswalks in lower speed zones always grant pedestrians right of way. Therefore, not only VRUs who are crossing the street must be considered, but also those close to the crossing aid or moving towards it, as they may cross the road in the near future. Corresponding maneuvers are therefore mainly *Approach Crosswalk* and *Cross Crosswalk* as well as *Turn Left*, *Turn Right* and *U-Turn*. Since these three junction maneuvers result in a change of direction for the ego vehicle, crossing VRUs need to be taken care of since these might have right of way due to a green traffic light. All other crosswalks which appear while the ego vehicle sticks to its driving direction are covered by *Approach Crosswalk* and *Cross Crosswalk*. Exemplary situations are crossing pedestrians at the end of an unprotected *Left Turn* maneuver or crossing cyclists during a *Turn Right* maneuver which were initially moving on the sidewalk or bikelane in parallel to the ego vehicle. The basic area for *Vulnerable Road Users on Crosswalks* consists of two subareas. Firstly, the whole crossing aid needs to be considered to include VRUs already crossing the road. This is specified by the size (length l_c, width w_c) of the crossing aid itself. Secondly, VRUs potentially intending to cross the road need to be taken into account. To that end, we estimate the time $t_{e,c}$ the ego vehicle takes to cross the crosswalk. Again, we consider an acceleration profile of the ego vehicle to determine $t_{e,c}$ for each timestamp. Subsequently, a distance r_c can be calculated by assuming a maximum velocity $\widehat{v}_v$ for VRUs approaching the crosswalk (e.g., sprinting speed of a pedestrian for crosswalks or cyclists speeds for crossing cyclists). The closer the ego vehicle gets to the crosswalk, the smaller is $t_{e,c}$ and therefore r_c. In a last step, the subareas are joined and the drivable and non-walkable parts are cut off. Figure 3.6 summarizes and depicts the construction of the basic area for *Vulnerable Road Users on Crosswalks*.



(a) Crossing VRUs

Distances	
l_c	Crosswalk length
w_c	Crosswalk width
r_c	Approaching radius

Parameters	
$t_{e,c}$	Ego crosswalk time
$\widehat{v}_v$	Max. VRU velocity

(b) Variables

$$r_c = \widehat{v}_v \cdot t_{e,c} \tag{3.7}$$

(c) Equation for crosswalk area

Figure 3.6.: Construction of area for *Vulnerable Road Users on Crosswalks* (©2022 IEEE)

3.2. Case Example

In this section, we will introduce an exemplary implementation of our proposed basic areas. The first part comprises a discussion about our considered parameter values for the construction of the relevant areas. Secondly, we evaluate our concept by discussing different exemplary urban situations and their corresponding estimated relevant areas.

3.2.1. Implementation

Selected parameter values have a significant impact on the size of the relevant areas. Therefore, they must be calibrated thoroughly. For instance, underestimating the reaction time of other traffic participants will result in unreasonably short distances and thus smaller areas. On the other hand, too conservative parameter values will result in oversized relevant areas which might lead to possibly incorporating irrelevant objects. Parameter values used in our initial implementation are shown in Table 3.2. Values for t_e, $\widehat{a}_x$, $\check{a}_x$, $\widehat{a}_y$, $\check{a}_y$ and t_o follow the starting point set of the RSS implementation by Gassmann et al. [43] and can be found in a separate

Table 3.2.: Parameter values of relevant areas (©2022 IEEE)

Parameter		Description
$\widehat{a}_x$	3.5 m s^{-2}	Max. acceleration
$\widecheck{a}_x$	4.0 m s^{-2}	Min. deceleration
$\widehat{a}_y$	0.2 m s^{-2}	Max. acceleration
$\widecheck{a}_y$	0.8 m s^{-2}	Min. deceleration
t_e	1.0 s	Ego reaction time
t_o	2.0 s	Obj. reaction time
$\widehat{v}$	$v_s + 10$ km h^{-1}	Max. obj. velocity
$\widehat{v}_v$	4.6 m s^{-1}	Max. ped. velocity

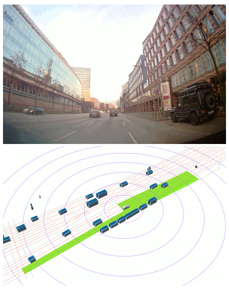
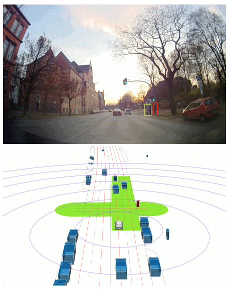
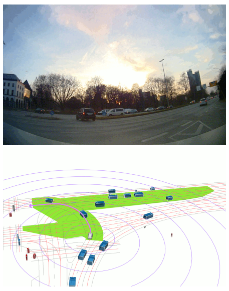
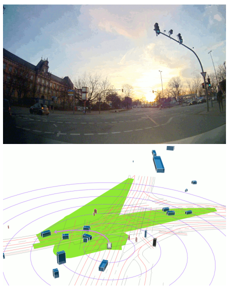
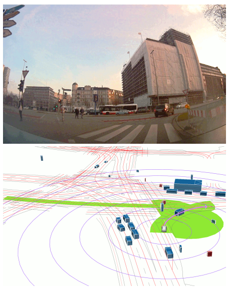
discussion[2]. The assumed maximum velocity of surrounding vehicles has a direct impact on area components like rearward distance $s_{\text{-}x}$ or merging distance s_m. While from a legal perspective the specified speed limit seems to be reasonable as a maximum velocity, most vehicles are moving slightly faster than that. Based on observations made by the European Commission [29] that most vehicles do not exceed the speed limit by 10 km h^{-1}, we also configure the maximum object velocity $\widehat{v}$ to be the active speed limit v_s plus 10 km h^{-1}. An assumption must be made regarding the maximum velocity of pedestrians for the construction of relevant areas when approaching crosswalks. Zębala et al. [127] conducted a study on behavior of pedestrians and found running pedestrians to move with a maximum speed of 4.6 m s^{-1}. This value is considered in our implementation for $\widehat{v}_v$ and reflects the worst case of a potential close pedestrian always running towards a crosswalk. Ideally, all of the parameter values should be configured based on the autonomous vehicle (reaction time) and typical behavior of traffic participants in the operational design domain (reaction times, kinematic, speed limit violations).

3.2.2. Qualitative Evaluation

To assess validity of our proposed concept and evaluate results of our exemplary implementation, we are depicting five different urban traffic situations and their generated relevant areas. We have chosen one lane change scenario, one crosswalk scenario and three intersection scenarios in the city of Hamburg, Germany. These demonstrate the complexity of urban traffic by comprising mixed traffic of motorized vehicles, pedestrians and cyclists, extended areas which need to be supervised because of broad intersections and designated infrastructure for pedestrians and cyclists like bicycle lanes or crosswalks. Table 3.3 introduces these five driving scenarios. The table shows one scene respectively to represent each of the scenarios including a camera image and a visualization of the generated relevant area. Additionally, it shows which of the six basic areas are considered in the particular scene and gives a comment explaining relevant objects in this scene.

[2] `https://intel.github.io/ad-rss-lib/ad_rss/Appendix-ParameterDiscussion/` (last accessed on 09.11.2023)

Table 3.3.: Exemplary traffic situations in the city of Hamburg and their relevant areas (©2022 IEEE)

Situation		Basic Areas	Comment
(a) Lane Change Right		☑ *Lead & Neighboring Veh.* ☑ *Appr. Veh. in LCL* ☐ *Appr. Veh. next to LCL* ☐ *Veh. Merging Lanes* ☐ *Veh. Crossing Lanes* ☐ *VRUs on Crosswalks*	Leading vehicles are relevant since they could perform a lane change or brake. Parking vehicles are relevant since they could either leave the parking spot or open a door. Following vehicles would be at fault for rear-end crashes and are thus excluded (except approaching vehicles in the lane change lane (LCL)).
(b) Approach Crosswalk		☑ *Lead & Neighboring Veh.* ☐ *Appr. Veh. in LCL* ☐ *Appr. Veh. next to LCL* ☐ *Veh. Merging Lanes* ☐ *Veh. Crossing Lanes* ☑ *VRUs on Crosswalks*	Leading vehicles are relevant since they could initiate emergency braking at any time. Pedestrians which have the intention of crossing the road or are crossing the road are considered relevant. Again, following vehicles are not deemed to be relevant since they are responsible for keeping a safe distance.
(c) Turn Left (3-Way-Intersection)		☑ *Lead & Neighboring Veh.* ☐ *Appr. Veh. in LCL* ☐ *Appr. Veh. next to LCL* ☑ *Veh. Merging Lanes* ☑ *Veh. Crossing Lanes* ☐ *VRUs on Crosswalks*	Leading vehicles on the intended path through the junction are considered relevant. Approaching vehicles from the left crossing the ego's path would be relevant as well as vehicles coming from the right merging into the left turn lanes. The area there is longer because of the later time of arrival.
(d) Turn Left (4-Way-Intersection)		☑ *Lead & Neighboring Veh.* ☐ *Appr. Veh. in LCL* ☐ *Appr. Veh. next to LCL* ☑ *Veh. Merging Lanes* ☑ *Veh. Crossing Lanes* ☑ *VRUs on Crosswalks*	Leading vehicles on the intended path through the junction are considered relevant. Oncoming vehicles crossing the path are relevant. Vehicles merging into ego's path are relevant (either oncoming or from the right). Crossing pedestrians at the entry and the exit of the junction are relevant.
(e) Turn Right		☑ *Lead & Neighboring Veh.* ☐ *Appr. Veh. in LCL* ☐ *Appr. Veh. next to LCL* ☑ *Veh. Merging Lanes* ☐ *Veh. Crossing Lanes* ☑ *VRUs on Crosswalks*	Leading vehicles also performing a right turn are relevant. Vehicles coming from the left with the intention of merging into the ego's intended path are considered relevant. Pedestrians close to the surrounding crosswalks and cyclists in the bicycle lane to be crossed (the red parallel lane behind the crosswalk) are considered relevant.

Validity and Generalizability. In summary, the areas seem to be appropriate for the identification of relevant road users in their corresponding scenes. In our opinion, relevant road users are inside the areas in all five examples. However, it is worth to be discussed whether vehicles that are following should be considered relevant or not. In each of the five examples in Table 3.3 there are vehicles closely following the ego vehicle which are currently not deemed to be as relevant. While the proposed version of our concept is based on the German road traffic regulations where §4 states that vehicles are always required to keep a safe distance in the front [38], not being required to react to following traffic is of course an idealized assumption. An ideal autonomous vehicle might even react to quickly approaching vehicles from behind by either performing an evasive maneuver or at least by already tightening safety harnesses in case that a rear-end collision is unavoidable. Another important aspect is the great extent of areas where lanes need to be looked upon like lane change lanes, merging lanes or crossing lanes (all scenarios in Table 3.3 except *Approach Crosswalk*). The deciding factor for this great extent is our behavioral requirement that the ego vehicle should not turn into the safety distance of an approaching vehicle in combination with the high reaction time for approaching vehicles of 2 s. This is also idealized and thus rather conservative. In reality, vehicles in urban traffic are moving in a more compact way and are advised to act explicitly to obviate misunderstandings between traffic participants. Lastly, not each of the objects inside the relevant areas might actually be as relevant. For instance, of the three leading vehicles in scenario *Approach Crosswalk* (cf. Table 3.3) only the closer two might be relevant. While the third one is in the same lane as the ego vehicle, it still finds itself to be occluded by one of the other leading vehicles and, therefore, does not require a direct reaction of the ego vehicle. Although the just mentioned aspects still leave room for improvements, our concept is already a promising method to automatically identify relevant road users. Moreover, the concept can be easily extended by adding further basic areas to consider either more infrastructure maneuvers (e.g., regarding roundabouts) or more groups of traffic participants (e.g., noncompliant jaywalkers or approaching emergency vehicles). Lastly, determination of relevant road users can be made even more strict by not just checking whether road users are inside the relevant area, but also by considering whether their current velocities and corresponding braking distances along their future trajectories would actually lead to overlaps with the ego vehicle path (cf. [129]). An application of our relevant areas can be found in Chapter 6, where an object detection module as part of a *Sense* component is evaluated towards its ability to detect objects inside relevant areas.

Part III.

Elicitation of Perception Requirements

Identifying Safety-critical Perception Errors

Specifying the perceptual accuracy autonomous vehicles require when interacting with surrounding traffic participants is not a trivial task. Generally speaking, the *Sense* component needs to be as accurate and robust as the *Plan* component requires to make safe decisions. This is both dependent on the error compensation capabilities of the *Plan* component and the operational driving situation the automated vehicle is in. But *Plan* components have the task of both predicting intentions of surrounding participants and subsequently deciding for a trajectory to be driven. Thus, the implementation of such components is diverse and resulting error compensation capabilities are often not explicit. Again, focusing the detection of surrounding traffic participants as one task of *Sense* components, it is known from Chapter 3 which traffic participants are relevant in which operational situations and therefore should ideally be perceived for safe planning. For instance, performing an unprotected left turn in an urban area requires an early detection of distant, quickly oncoming vehicles. But additionally, Chapter 2 shows that perception inaccuracy is manifold, and it is not only about detecting or not detecting a relevant object. So while it is clear, that oncoming vehicles during an unprotected left turn must be detected, it is not obvious how accurate corresponding object tracks need to be. The latest acceptable moment for the initial detection or required stability and accuracy of an object track is not easily specifiable. Chapter 4 deals with the simulation-based elicitation of accuracy requirements for the *Sense* component for a given *Plan* component in a defined urban scenario.

4. Simulation-based Elicitation of Accuracy Requirements for the Environmental Perception of Autonomous Vehicles[1]

In this chapter, we elicit quantifiable requirements for a *Sense* component by testing the error compensation capabilities of a prototypical *Plan* component. This comprises the definition of perceptual error models and the construction of perceptual inaccuracy spaces, as well as the exploration of these spaces and the identification of safety envelopes. Resulting safety envelopes identified by structured test processes are then used to specify requirements for a *Sense* component. To show applicability of our approach, this part is supported by a case study that utilizes the proposed methodology in a structured test process. The tested *Plan* component is part of an autonomous driving stack which has previously been deployed for real test drives in the city of Hamburg, Germany. We consider requirements regarding time of detection, tracking and the estimated position of one relevant vehicle in two concrete scenarios.

4.1. Functional Decomposition

In order to investigate the interface between environmental perception and planning, we refer to the functional decomposition of autonomous vehicles from Chapter 2 [84]. The autonomous vehicle is decomposed into *Sense*, *Plan* and *Act* and then further refined into *Environment Scanning*, *Feature Extraction*, *Scene Modeling*, *Situational Understanding*, *Behavioral Decision* and *Action*. *Plan* takes a conceptualization of the environment as input.

Since we are interested in requirements towards the *Sense* component based on the subsequent *Plan* component, we investigate what information is delivered to the *Plan* component by the *Sense* component. When composing a system and analyzing this interface towards requirements, there are in general two ways on how the system assembly can be approached: either for a given *Sense* component, deduce how robust the subsequent *Plan* component needs to be or for a given *Plan* component define how accurate the output from the *Sense* component needs to be. The focus of this chapter is the latter one. This can also be seen as a combination of assumptions and corresponding promises between these two components which is a common approach in the verification domain called assume-guarantee reasoning and also utilized in the field of contract-based system design [96, 44].

The *Sense* component is decomposed into *Environment Scanning*, *Feature Extraction* and *Scene Modeling*. *Environment Scanning* is implemented as a combination of sensors and data pro-

[1] This chapter is based on Paper II [82] and therefore contains verbatim content previously published (©2021 Springer Nature).

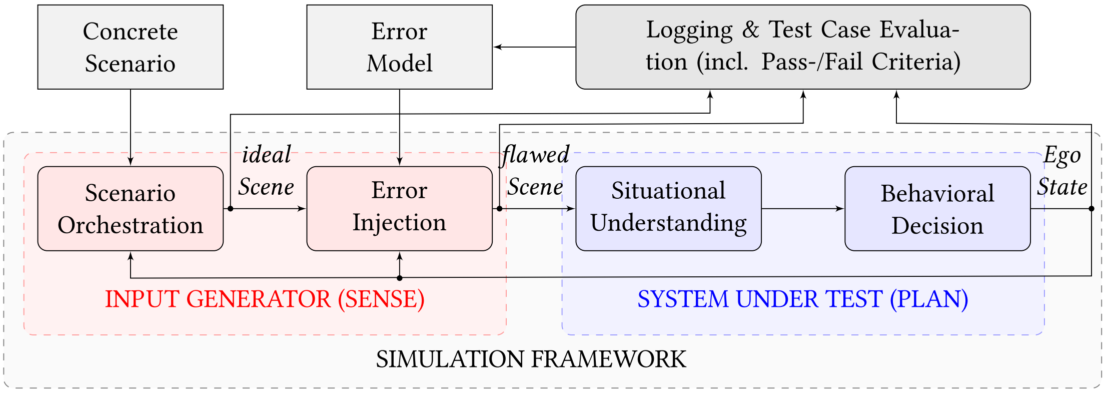


Figure 4.1.: Conceptual test design for the interface between *Sense* and *Plan*: Output of the *Sense* component is synthetically generated based on a concrete scenario and subsequently flawed to investigate the reaction of the *Plan* component (©2021 Springer Nature)

cessing and therefore consists of hardware-software systems. Environmental conditions or physical effects can cause disturbances while capturing the reality (e.g., glare of the sun) and influence the generated model of the environment (cf. environmental uncertainty [22]). An additional challenge is the association of extracted features when fusing data by multiple sensors in the *Feature Extraction* and *Scene Modeling* components.

The *Plan* component is decomposed to *Situational Understanding* and *Behavioral Decision* and can be regarded as an optimization problem with the goal of finding the trajectory with the least costs. This can be formulated as a mathematical problem and be approached with software. Especially in the aircraft domain, verification of *Plan* components as part of aircraft collision avoidance systems has been researched [110, 57, 59]. Recently, efforts have been made towards formal models for *Plan* components of autonomous vehicles [101, 12, 70].

The investigated *Plan* component of our system under test receives an object list as input. These objects represent the perceived surrounding traffic participants which are used in conjunction with an accurate ego position coming from a localization module and a highly accurate map to decide a future trajectory. Although the *Plan* component also relies on an accurate localization and detection of dynamic infrastructure elements like, e.g., traffic light states, we focus on the inaccuracy of perceived traffic participants in this chapter.

4.2. Simulation-based Testing with Error Injection

The architecture of the used test setup is shown in Figure 4.1: Components in the top row drive the exploration of variants of a concrete scenario (varying errors and recording outcomes). Components in the lower row constitute the test harness (simulation framework and augmented *Sense* component) and the system under test (the *Plan* component).

Our intention is to investigate the response of the *Plan* component to errors in the perceived scene. Since the *Plan* component comprises multiple software modules and the input coming from the *Sense* component is generated and manipulated synthetically, we analyze the *Plan* component in a closed-loop simulation framework. The scene that is given to the *Plan* component is generated by simulating the *Sense* component and augmenting its output with errors. This comprises generating the scene based on a concrete scenario specification including traffic participant behavior and infrastructure elements as well as transforming the ideal scene into a flawed scene by injecting errors. To give an example, we could perturb the velocity of a distant oncoming vehicle to be perceived as being $40\,\mathrm{km\,h^{-1}}$ while the vehicle actually drives at a speed of $50\,\mathrm{km\,h^{-1}}$.

We assess the behavior of the system under test based on the ideal scene and the states of the ego vehicle over time. By defining pass and fail criteria, the evaluation component can be used to evaluate whether a test case has passed or failed. Possible pass and fail criteria check for real collisions or unsafe distances. Due to the fact that we investigate iteratively worsening hazards, the results of the executed test cases will be taken into consideration by the error injection component for driving the variation of how the scenario is perceived. When e.g., an inaccurate measurement regarding the oncoming vehicle's velocity with a deviation of $\Delta v = 10\,\mathrm{km\,h^{-1}}$ does not result in a failed test case, a greater inaccuracy is examined in the next simulation. The evaluation component logs the entire history of all simulations. This enables the use of an exit condition, e.g., when a threshold has been found or a defined parameter range has been explored in sufficient detail.

4.3. Modeling Perceptual Inaccuracy and Errors

We identify multiple types of hazards and perceptual errors as a basis for error injection. To construct and investigate perceptual inaccuracy spaces, parameterizable error models are needed. These error models have to affect the perceived scene and therefore take the concrete interface of the system under test into consideration. Furthermore, these models should exhibit a continuity regarding resulting errors. Common hazards are introduced in Section 4.3.1, while corresponding exemplary error models are proposed in Section 4.3.2.

4.3.1. Perceptual Hazards and Errors

For the identification of the acceptable inaccuracy of a perceived object, there is the need to systematically explore its corresponding inaccuracy space. We approach that by describing common perceptual hazards which can be considered to create parameterizable error models. The hazards regarding object segmentation and their corresponding models are introduced in the following.

True Positive Inaccuracy

Surrounding traffic participants are often conceptualized as bounding box objects with attributes. These attributes can either be metric[2] variables (e.g., position, velocity) or categorical[3] variables (e.g., classification, light status). While the magnitude of an error regarding metric

[2] Metric refers to a variable defined on either an interval or ratio scale.

[3] Categorical refers to a variable defined on either a nominal or ordinal scale.

attributes can be calculated by looking at the difference between the true value and the measured value, measurements of categorical attributes can only either be true or false. Especially for the classification of other traffic participants, where some misclassifications might objectively seem worse, there is no way to assess the magnitude of the error objectively without transforming the classification attribute to either an interval or ratio scale. That can e.g., be done by establishing similarity measures between two given classifications.

Therefore, we define the true positive inaccuracy $O_\epsilon(t)$ of a conceptualized traffic participant at the time t consisting of n metric attributes m_i and k categorical attributes c_i as follows:

$$O_\epsilon(t) = [M_\epsilon(t), C_\epsilon(t)]^T \tag{4.1}$$

$$M_\epsilon(t) = \begin{bmatrix} \Delta m_1(t) \\ \Delta m_2(t) \\ \vdots \\ \Delta m_n(t) \end{bmatrix} = \begin{bmatrix} \tilde{m}_1(t) - \bar{m}_1(t) \\ \tilde{m}_2(t) - \bar{m}_2(t) \\ \vdots \\ \tilde{m}_n(t) - \bar{m}_n(t) \end{bmatrix} \tag{4.2}$$

$$C_\epsilon(t) = \begin{bmatrix} S_{c_1}(t) \\ S_{c_2}(t) \\ \vdots \\ S_{c_k}(t) \end{bmatrix}, \text{where } S_{c_i}(t) = \begin{cases} 0, & \text{if } \tilde{c}_i(t) = \bar{c}_i(t) \\ 1, & \text{if } \tilde{c}_i(t) \neq \bar{c}_i(t) \end{cases} \tag{4.3}$$

$\Delta m_i(t)$ is the difference of the measured value $\tilde{m}_i(t)$ and true value $\bar{m}_i(t)$ regarding the metric object attribute m_i. $S_{c_i}(t)$ specifies whether the measured value of a categorical variable c_i is erroneous or not. Based on this definition, error models can be established that affect the different components of the attribute inaccuracy.

Field of View Delimiting

The field of view of an autonomous vehicle is made up out of the scanning areas of its different sensors. Traffic participants that are not within the range of the field of view can therefore not be perceived. Moreover, environmental conditions like sun glare, rainfall or occlusions by other traffic participants can temporarily limit the field of view. The field of view that is required by an autonomous vehicle moving in an operational design domain (ODD) is conditioned by the occurring infrastructure (e.g., size of junctions) and surrounding traffic participant behavior (e.g., travelling speed). While highly automated driving on the highway especially necessitates perceiving objects in the far longitudinal distance, urban scenarios require a more uniform surround view due to cross traffic. Hence, eliciting the required field of view for a concrete ODD is a non-trivial task. By systematically delimiting an ideal field of view, safety-critical areas can potentially be deduced. Delimiting the field of view can e.g., be implemented by defining maximum ranges and specific opening angles or by individually defining sensor scanning areas that are subsequently aggregated.

Object Track Instability

When perceiving traffic participants it is not only important to capture them accurately in a scene, but to also track them over the course of scenes. By tracking an object and therefore knowing what its past trajectory looked like, the *Situational Understanding* component can be more reliable in predicting possible and likely future behavior. For that, object tracks of past scenes need to be associated with the current object tracks, which is usually implemented by giving these object tracks a similar identifier. Ideally, as long as a traffic participant is in immediate range of the autonomous vehicle, the corresponding object track should not cease to exist. However, this can happen due to faults like extensive computation time for the association or occlusion by other traffic participants. Dealing with unstable object tracks is usually addressed by the *Situational Understanding* within the *Plan* component.

Object Track Decay/Multiple Track

Ideally, one traffic participant is conceptualized with one consistent track (one bounding box). However, when dealing with larger traffic participants such as busses, it can happen that their bounding box decays into multiple smaller ones. While these bounding boxes might still occupy around the same space as the previous larger one, this hazard results in more separate objects for the *Situational Understanding* to deal with. One traffic participant that is conceptualized by more than one object track is defined as *Multiple Track* by Brahmi et al. [18].

Multiple Object

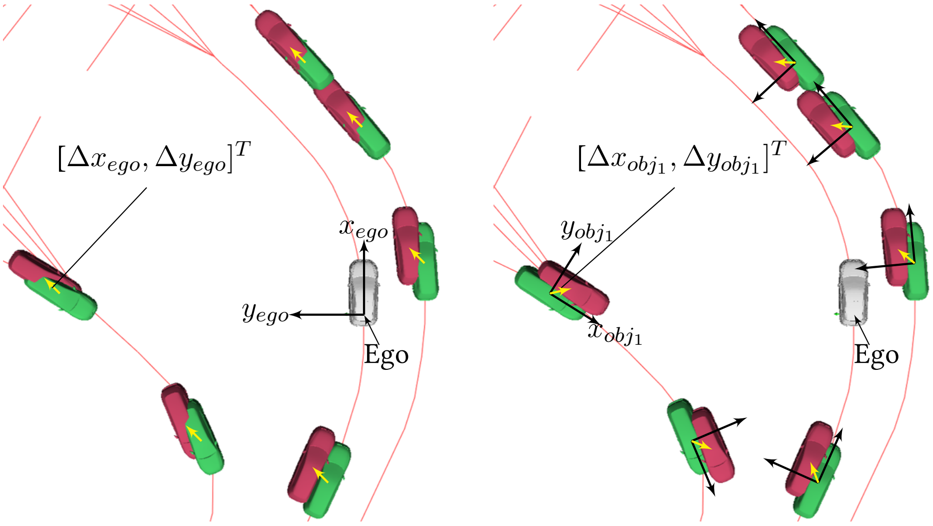
In contrast to the error *Multiple Track*, it can also happen that multiple traffic participants are captured as one object track, e.g., when being close to each other. Such multiple object tracks are likely to decay into object tracks when they start moving resulting in the fact that these multiple object tracks are of special relevance to the *Situational Understanding* component. This is defined as *Multiple Object* by Brahmi et al. [18].

4.3.2. Error Models

We propose three exemplary parameterizable error models based on hazards and errors discussed previously. The proposed error models are kept simple. Moreover, we do not claim for these to model all facets of their corresponding hazard. They are rather a proof of concept of general error models which both result in meaningful errors based on perceptual hazards and are limited in their complexity to still be analyzable. The investigated perceptual hazards in our initial application are *True Positive Inaccuracy*, *Field of View Delimiting* and *Object Track Instability*.

True Positive Inaccuracy: Object Position Shift

The position accuracy of surrounding objects is essential for the *Plan* component. The perceived longitudinal position is crucial for estimating the time until an oncoming or incoming vehicle arrives at a potential conflict zone or until a leading vehicle is close. The perceived lateral position is of importance when the corresponding object is passing or is being passed. Additionally, it affects lane matching and thus has an impact on the predicted future behavior of the corresponding object.


$[\Delta x_{ego}, \Delta y_{ego}]^T$
x_{ego}
y_{ego}
Ego
$[\Delta x_{obj_1}, \Delta y_{obj_1}]^T$
y_{obj_1}
x_{obj_1}
Ego

(a) Position errors (red) specified in the ego vehicle coordinate system ($\Delta x_{ego,\mu} = 1\,\text{m}, \Delta y_{ego,\mu} = 1\,\text{m}$)

(b) Position errors (red) specified in the respective object coordinate system ($\Delta x_{obj,\mu} = 1\,\text{m}, \Delta y_{obj,\mu} = 1\,\text{m}$)

True Positive Inaccuracy: Object Position

Parameter	Unit	Description
$\Delta x_{ego,\mu}$	$[m]$	x-Position mean error (in ego coordinates)
$\Delta x_{ego,\sigma}$	$[m]$	x-Position standard deviation (in ego coordinates)
$\Delta x_{obj,\mu}$	$[m]$	x-Position mean error (in object's coordinates)
$\Delta x_{obj,\sigma}$	$[m]$	x-Position standard deviation (in object's coordinates)
$\Delta y_{ego,\mu}$	$[m]$	y-Position mean error (in ego coordinates)
$\Delta y_{ego,\sigma}$	$[m]$	y-Position standard deviation (in ego coordinates)
$\Delta y_{obj,\mu}$	$[m]$	y-Position mean error (in object's coordinates)
$\Delta y_{obj,\sigma}$	$[m]$	y-Position standard deviation (in object's coordinates)

(c) Parameterization

Figure 4.2.: Error model *True Positive Inaccuracy: Object Position* (©2021 Springer Nature)

Regarding a metric attribute $m(t)$ like, e.g., an object's position component, we propose to split up the actual measurement error into three parts:

$$\Delta m(t) = \Delta m_s(t) + \Delta m_r(t) + \Delta m_o(t) \tag{4.4}$$

where $\Delta m_s(t)$ is the systematic error component, $\Delta m_r(t)$ is the random error component and $\Delta m_o(t)$ describes coarse outliers. While random errors and outliers can be detected and at times taken care of by smoothing and stabilization considering past measurements, systematic offsets are more challenging to detect and can therefore be highly relevant to safe planning.

The object's position is a metric attribute defined in a Cartesian coordinate system consisting of two components. While an object's position is usually captured in an egocentric coordinate system, it may also be of interest to not directly manipulate the ego-relative position components, but to define position shifts based on an object-centric coordinate system. The object-centric coordinate system resembles the egocentric coordinate system rotated by the object's relative heading Ψ to the ego. In that way it is possible to separate an object's position shift distinctly into a longitudinal and lateral shift. Both coordinate systems and exemplary shifts in these are visualized in Figure 4.2.

Our proposed error model for the object position covers both systematic errors μ and random errors σ and offers the possibility to define the deviation either in the ego-relative or object-relative coordinate system. Thus, the perceived position shift of the object $[\Delta x, \Delta y]^T$ can comprise a shift specified in the ego-coordinate system $[\Delta x_{ego}, \Delta y_{ego}]^T$ and a shift specified in the object-coordinate system $[\Delta x_{obj}, \Delta y_{obj}]^T$. It is defined as follows:

$$\begin{bmatrix}\Delta x\\ \Delta y\end{bmatrix} = \begin{bmatrix}\Delta x_{ego}\\ \Delta y_{ego}\end{bmatrix} + \begin{bmatrix}cos(\Psi) & -sin(\Psi)\\ sin(\Psi) & cos(\Psi)\end{bmatrix}\begin{bmatrix}\Delta x_{obj}\\ \Delta y_{obj}\end{bmatrix} \tag{4.5}$$

$$\begin{bmatrix}\Delta x_{ego}\\ \Delta y_{ego}\end{bmatrix} = \begin{bmatrix}\Delta x_{ego,\mu} + \Delta x_{ego,\sigma}\\ \Delta y_{ego,\mu} + \Delta y_{ego,\sigma}\end{bmatrix}, \qquad \begin{bmatrix}\Delta x_{obj}\\ \Delta y_{obj}\end{bmatrix} = \begin{bmatrix}\Delta x_{obj,\mu} + \Delta x_{obj,\sigma}\\ \Delta y_{obj,\mu} + \Delta y_{obj,\sigma}\end{bmatrix}, \tag{4.6}$$

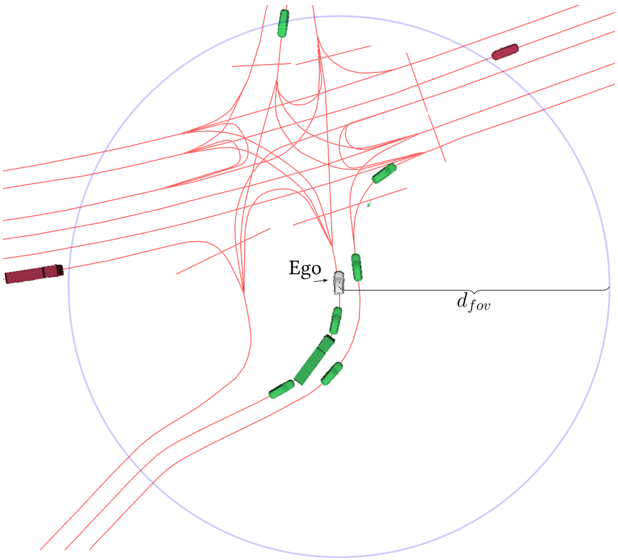
where the systematic error components $\Delta\{x, y\}_{\{ego,obj\},\mu}$ are defined by a constant value and the random error components $\Delta\{x, y\}_{\{ego,obj\},\sigma}$ are defined by a variable value. In our initial application later on, we make use of the systematic error components.

Field of View Delimiting: Distance Thresholding

Since driving in urban scenarios generally premises a surround view around the ego vehicle, investigating the actual needed field of view ranges is required when designing the sensor setup. For our initial application, we propose a simple error model which uniformly delimits the ideal field of view by a defined maximum range. All object tracks beyond that range are cut off and therefore result in a *False Negative*. This leads to surrounding traffic participants being perceived at a later point in time with decreasing field of view range. The error model consequently only consists of one parameter d_{fov}, which serves as a distance threshold and thus decides whether traffic participants are perceived or not based on their position (cf. Figure 4.3).

Object Track Instability: Lifetime and Downtime

When dealing with object tracking in general, it is of interest how stable the object tracks generated by the *Sense* component have to be. Regarding a single object in one scene, there either exists a corresponding object track (*True Positive*) and thus the object is perceived or there is no object track and the object is therefore not perceived (*False Negative*). Cumulating these over time and associating object tracks of two consecutive timestamps with one another allows the subsequent planning to consider the past behavior of a perceived object. The two main aspects of object tracks we are therefore interested in is loss of an object track as well as the time gap until there is a new track for the same object.

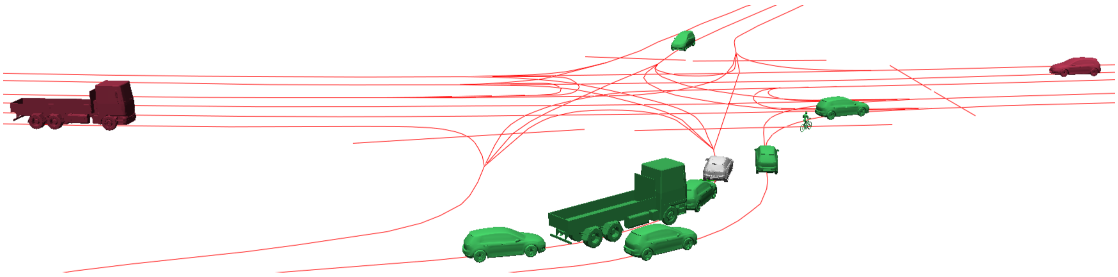


(a) Field of view limitation by range

Field of View Delimiting

Parameter	Unit	Description
d_{fov}	$[m]$	Perception Range

(b) Parameterization

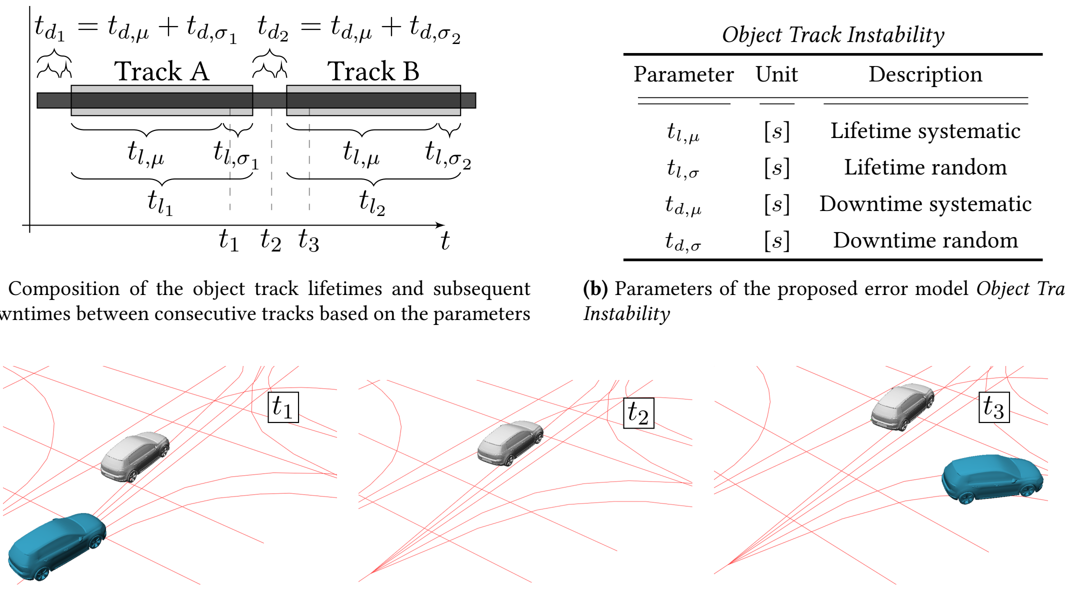


(c) Example: Perceived (green) and missed (red) objects for $d_{fov} = 50\,\mathrm{m}$

Figure 4.3.: Error model *Field of View Delimiting: Distance Thresholding*

The proposed error model (cf. Figure 4.4) varies both the lifetime of object tracks and the downtime between two consecutive object tracks for the same object. Additionally, a new object track also comes with a new ID. Therefore, it is not trivial for the system under test to associate the new object track with the old one and by that deduct the past behavior of the traffic participant. Both the lifetime and the downtime consist of a systematic component $t_{\{l,d\},\mu}$ and a random component $t_{\{l,d\},\sigma}$. While the systematic component is defined by a constant time, the random component is generated by a folded normal distribution for our initial application. Consequently, since the folded random distribution always produces a positive value, the systematic component also equals the minimum lifetime (or minimum downtime respectively). The subsequent equations for object track lifetimes t_l and downtimes t_d are therefore defined as follows:

$$t_l = t_{l,\mu} + t_{l,\sigma}, \qquad t_d = t_{d,\mu} + t_{d,\sigma}. \tag{4.7}$$



(a) Composition of the object track lifetimes and subsequent downtimes between consecutive tracks based on the parameters

Object Track Instability

Parameter	Unit	Description
$t_{l,\mu}$	$[s]$	Lifetime systematic
$t_{l,\sigma}$	$[s]$	Lifetime random
$t_{d,\mu}$	$[s]$	Downtime systematic
$t_{d,\sigma}$	$[s]$	Downtime random

(b) Parameters of the proposed error model *Object Track Instability*

(c) Ego (white) tracks a vehicle (blue)

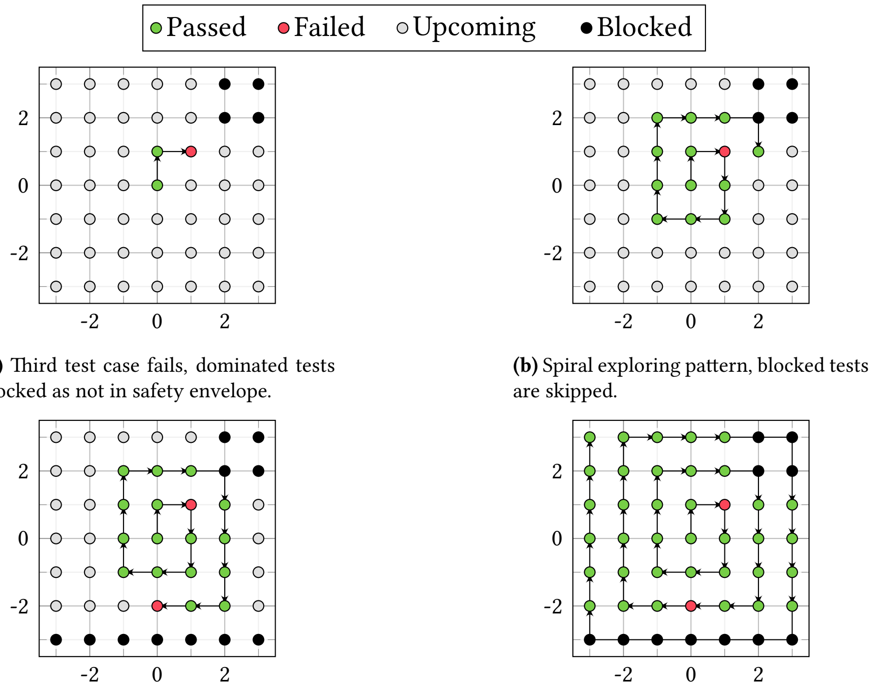
(d) Track of vehicle is lost

(e) The vehicle is perceived again

Figure 4.4.: Error model *Object Track Instability: Lifetime and Downtime* (©2021 Springer Nature)

4.3.3. Inaccuracy Space Exploring

To utilize the proposed test setup and make use of the error models, a strategy for exploring the resulting perceptual inaccuracy spaces is needed. We consider a multidimensional space of inaccuracies regarding a perceived object where each dimension describes a specific inaccuracy (e.g., position, velocity, etc.). In this work, we use different approaches for sampling this space (grid-based and exploration around the origin) in different experiments (cf. Section 4.4). Moreover, we consider the inaccuracy space to be metric, meaning that the aggregated error becomes objectively worse when being farther away from the origin. The origin marks perfect accuracy, meaning that no error is existent (existing object is perceived with $O_\epsilon(t) = \vec{\mathbf{0}}$). To avoid exhaustive sampling of this space, we propose the idea to systematically identify safety envelopes around the origin. An exemplary method for that is explained in the following considering an exemplarily two-dimensional inaccuracy space. The space is explored in a spiral manner around the origin, meaning that the perceptual inaccuracy is continuously increased (cf. Figure 4.5). Once an error is declared as not acceptable (e.g., based on testing), exploring farther in the same direction is blocked. In this way, errors of greater magnitude of the same type are not assessed, resulting in less checks. Even if these errors may be regarded as acceptable, they can not be part of a safety envelope because of its inevitable enclosing of an already unaccepted error. The exploring is set up to stop when either all prospective checks are blocked or the boundaries of the space are reached. Optionally, after a first safety envelope is identified, the specified discretization step size(s) can be adjusted for a second exploration process in the area of the previously identified safety envelope boundaries. This can lead to a refinement of the previous safety envelope. We follow this principle later for the application of our one-parametric error model *Field of View: Distance Thresholding*.



(a) Third test case fails, dominated tests blocked as not in safety envelope.

(b) Spiral exploring pattern, blocked tests are skipped.

(c) Another failed test leads to more test cases being blocked.

(d) The test process finishes. A safety envelope can now be defined.

Figure 4.5.: Exemplary application of exploring a two-dimensional inaccuracy space with boundaries $[-3, 3] \times [-3, 3]$ and step sizes $s_x = s_y = 1$

4.4. Case Study

We are using the three proposed error models and inaccuracy space exploring methods for eliciting accuracy requirements in multiple series of experiments based on two urban scenarios. Each error model is utilized in both scenarios, thus resulting in six different experiments. This section presents results from these experiments before discussing generalizability and validity.

4.4.1. Limitations of Test Framework and System under Test

The *Plan* component under test is part of a prototypical automated driving stack which has been deployed in urban test drives in the city of Hamburg, Germany throughout 2019. The simulation framework is self-developed and is based on ADTF[4], which is an established framework for development, visualization and testing of automated driving systems in the automotive industry. The simulation framework also comprises a plugin which can translate OpenSCENARIO files to ideally perceived traffic participants (ideal *Sense* component), a parameterizable error injection plugin which covers our proposed error models and an idealized *Act* component which features simple vehicle dynamic models. A self-written Python program wraps the sim-

[4] Automotive Data and Time-Triggered Framework

ulation framework and is thus able to start new simulation runs, adjust the configuration of injected perception errors and evaluate simulated test cases. Since both the simulation framework and the system under test are experimental, the whole setup has limitations. E.g., we observed rarely occurring crashes or communication failures of simulation runs. To compensate these minor reliability issues leading to non-deterministic test results, we come up with the countermeasure of repeating the same test case several times instead of only executing it once. When one repetition of a test case is failed, we conservatively consider the whole test case as failed. As explained in the upcoming sections of our case study, we find the results of our experiments to be reasonable. We thus conclude, that the technical limitations do not affect the general validity of both our concept and trend of our elicited requirements.

4.4.2. Scenario 1: Unprotected Left Turn

We firstly investigate a left turn scenario which is illustrated in Figure 4.6. The simulated scenario comprises an unprotected left turn maneuver of the ego vehicle with one oncoming passenger car at a real intersection between *Jungiusstraße* and *Gorch-Fock-Wall* in the city of Hamburg, Germany. With the oncoming vehicle maintaining a velocity of 50 km h^{-1}, the ego vehicle arrives right at the time when it needs to decide whether to turn in front of the oncoming vehicle or to wait for it to pass. Making that decision requires a proper understanding of the scene and thereby an accurate perception. That encompasses an early and accurate detection of the oncoming vehicle. Position and velocity are both essential attributes for predicting the future trajectory and estimating the time of arrival at the potential point of intersection. The object's width and its position are crucial for estimating how far the ego vehicle can already pull into the intersection. The object conceptualization as part of the interface between *Sense* and *Plan* components of our system under test is specified in Figure 4.7.

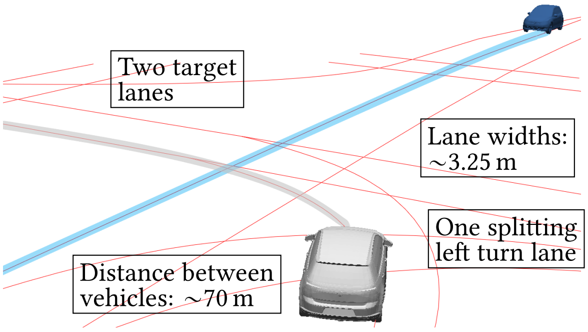

Two target lanes
Lane widths: ~3.25 m
One splitting left turn lane
Distance between vehicles: ~70 m

Figure 4.6.: Scenario 1: Unprotected left turn (©2021 Springer Nature)

Object Conceptualization

Attribute	Unit	Description
x, y	[m]	Position
Ψ	[rad]	Yaw
v_x, v_y	$[m/s]$	Velocity
a_x, a_y	$[m/s^2]$	Acceleration
l, w	$[m]$	Length, Width
c		Classification

Figure 4.7.: Object attributes (©2021 Springer Nature)

Scenario 1 - Error Model 1 (True Positive Inaccuracy: Object Position Shift)

For an initial application of the proposed error model, we investigate the effect of a systematic object shift based on the object coordinate system. This means that the two parameters $\Delta x_{obj,\mu}$ and $\Delta y_{obj,\mu}$ are being utilized during the test process. We specify an initial step size of 1 m for

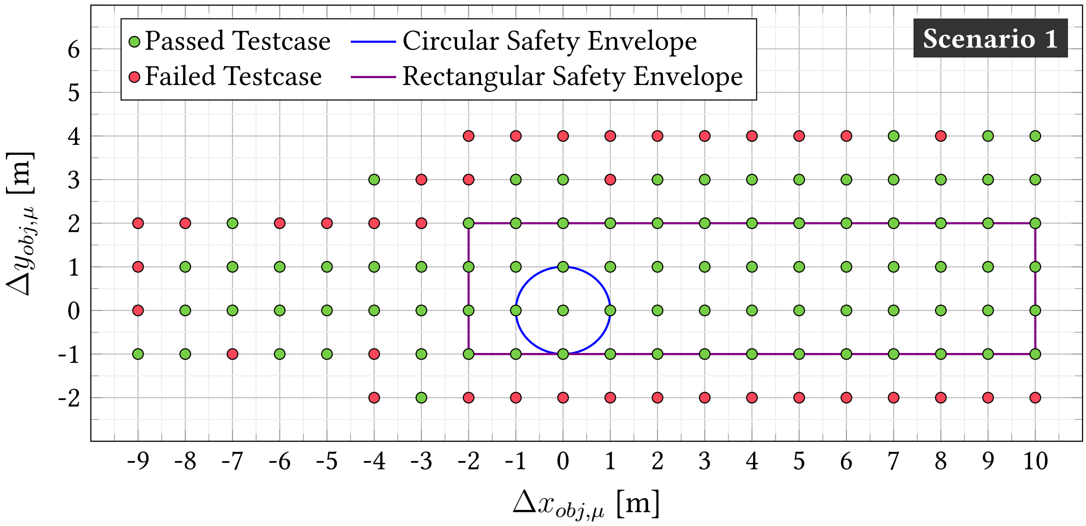


Figure 4.8.: Scenario 1: Test cases and resulting safety envelope regarding position inaccuracy (©2021 Springer Nature)

both parameters and the range to be $[-10\,\text{m}, 10\,\text{m}] \times [-10\,\text{m}, 10\,\text{m}]$. Collisions of the ego vehicle with the ground truth object bounding box are considered for evaluating whether a test case is declared as either passed or failed. Also, one test case is repeated several times to cope with potentially occurring non-deterministic effects caused by the prototypical system under test or the experimental simulation framework. When only one of the test case repetitions is declared as failed, the whole test case is declared as failed. The results of the test process comprising a perceived systematic object position shift are visualized in Figure 4.8.

Figure 4.8 shows that there is a higher tolerance for an inaccurate longitudinal position while smaller errors regarding the lateral position component already propagate up to a failure earlier. This effect seems plausible since the lateral position of the oncoming vehicle is relevant to how far the ego vehicle can enter the intersection while it waits for the oncoming vehicle to pass (see Figure 4.6). Even longitudinal position errors of higher severity do not necessarily result in a failure. The higher tolerance for longitudinal position errors can be explained by the cautious behavior of the ego vehicle. This behavior is observable by the time needed to accelerate again and finish the left turn maneuver after initially braking for the inaccurately perceived oncoming vehicle. Moreover, there is also a higher tolerance for a positive longitudinal shift than for a negative one. This does also seem plausible due to negative longitudinal shifts tricking the ego vehicle into overestimating the time gap between itself and the oncoming vehicle. These negative longitudinal shifts can therefore be the reason for the ego vehicle to decide for a quick left turn before the apparent arrival of the oncoming vehicle. Based on the resulting circular safety envelope, we state the following observation:

Requirement 1

The tested *Plan* component requires a *Sense* component that reports an oncoming vehicle's position with less than 1 m inaccuracy in the simulated scenario of an unprotected left turn in an urban setting.

Alternatively, a rectangular safety envelope can be considered for the elicitation of individual requirements for both the allowed perceived longitudinal and lateral object shift. However, that demands a translation of the shifts defined in the object-centric coordinate system back into the egocentric coordinate system which is used for the object position in our object conceptualization.

Scenario 1 - Error Model 2 (Field of View: Distance Thresholding)

The second perceptual hazard we investigate is a limited field of view. We do this by utilizing our simplistic one-parametric error model from the previous section where the field of view is defined by a circular perception range. For the identification of the minimum required field of view range in our scenario, we iteratively lower the range (and thus increase the false negative rate) as long as the ego vehicle is passing the simulated test case repetitions for this parameter value. Once a test case is declared as failed, the range is increased again (meaning that the false negative rate decreases) and sampling of the perceptual inaccuracy space with a higher resolution (smaller step size) is triggered. Again, a test case is passed if there was no collision of the ego vehicle and the oncoming vehicle in any of the test case repetitions. For an initial application of the proposed error model we specify two different phases. At first, d_{fov} is iteratively scaled down from 150 m to 10 m with a defined step size in subsequently executed test cases. After the entire range has been checked, the downscaling of d_{fov} starts again but with a value greater than the one used in the test case that first failed. From there on d_{fov} is again iteratively reduced, but with a more precise step size. When a pre-defined number of test cases is declared as passed, the test orchestration is stopped. The executed and evaluated test cases are visualized in Figure 4.9.

The results show that the strictest limitation of the field of view and therefore greatest inaccuracy not leading to a collision corresponds to $d_{fov} = 56\,\text{m}$. Therefore, we elicit a requirement regarding the field of view considering our scenario as follows:

Requirement 2

The tested *Plan* component requires a *Sense* component that detects an oncoming vehicle at latest at a distance of 56 m in the simulated scenario of an unprotected left turn in an urban setting.

Scenario 1 - Error Model 3 (Object Track Instability: Lifetime and Downtime)

The proposed error model can be utilized to investigate two aspects for a defined scenario. Firstly, whether the tested *Plan* component is sensitive to changing object IDs caused by in-

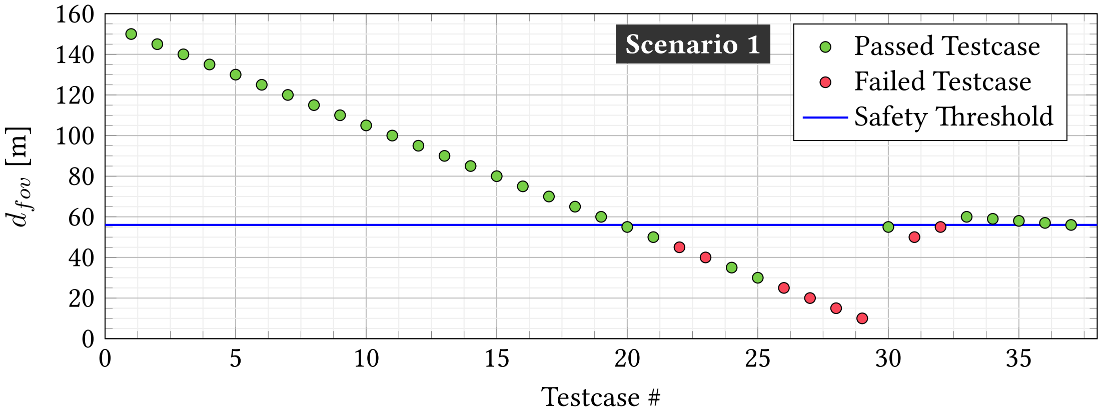


Figure 4.9.: Scenario 1: Test cases and resulting safety threshold regarding field of view

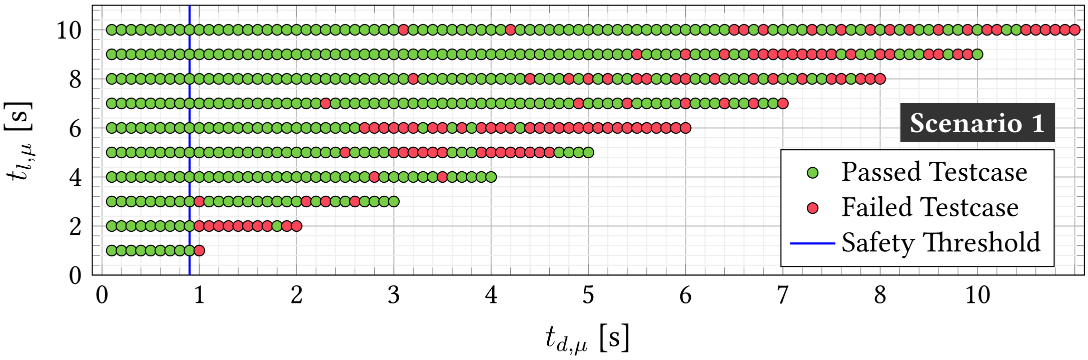


Figure 4.10.: Scenario 1: Test cases and resulting safety threshold regarding fragmentary object tracks (©2021 Springer Nature)

sufficient tracking consistency. Secondly, when exactly a loss of track and the subsequent misdetection leads to a failure (collision).

A first test process focuses on the former, considering a systematic lifetime of object tracks with no downtime in between ($t_{l,\mu} \in [1\,\text{s}, 10\,\text{s}]$ with a step size of 1 s, $t_d = 0\,\text{s}$). Each concrete test case is repeated several times. However, none of the executed test case repetitions fail. This shows that even a frequently changing object ID ($t_{l,\mu} = 1\,\text{s}$) of the oncoming vehicle is not leading to a collision. Therefore, we consider a varying lifetime as well as a downtime between consecutive tracks in a subsequent test process. Whereas this combination constitutes a two-dimensional parameter space, the parameter t_l is not positively related to error severity. Hence, we choose a grid-based exploration instead of a spiral exploration around the origin.

For a first analysis of the *Plan* components response regarding a fragmentary object track, a test process with parameters $t_{d,\mu} \in [0\,\text{s}, 10\,\text{s}]$ (step size 0.1 s) and $t_{l,\mu} \in [1\,\text{s}, 10\,\text{s}]$ (step size 1 s) under the condition $t_{d,\mu} < t_{l,\mu}$ is conducted. This results in 550 concrete test cases (cf. Figure 4.10). While there is still no obvious influence coming from frequently changing object IDs, collisions occur first for downtimes of 1 s when track lifetimes are set to 1 s, 2 s and 3 s.

Not detecting the oncoming vehicle for 1 s means that the *Plan* component is not perceiving it for nearly 14 m of its covered distance. Longer downtimes become acceptable with increased lifetimes. This seems plausible, since the object track lifetime $t_{l,\mu}$ serves as an indirect trigger for the track loss. Losing track of the oncoming vehicle is less critical, when it is either at a farther distance or has already crossed the intersection. This emphasizes, that timing of such errors in conjunction with the concrete scenario plays an important role regarding acceptable track instability. Based on the results, we state the following observation regarding object track instability in the investigated scenario:

Requirement 3

The tested *Plan* component requires a *Sense* component to never miss an oncoming vehicle for more than 0.9 s in the simulated scenario of an unprotected left turn maneuver in an urban setting.

4.4.3. Scenario 2: Lane Change and Passing

The second investigated scenario constitutes changing the lane to pass a preceding, slowly moving vehicle. This scenario can usually be seen in urban environments, when a vehicle drops off or picks up a passenger at the side of the road and then drives away. Initially, the ego vehicle drives on a road segment consisting of two lanes with the preceding vehicle moving in the same lane in front of it at a velocity of 10 km h^{-1}. Since the specified target pose lies far ahead of the preceding vehicle, the ego vehicle is forced to change lane, pass the preceding vehicle and reach the target as soon as possible. The scenario is illustrated in Figure 4.11.

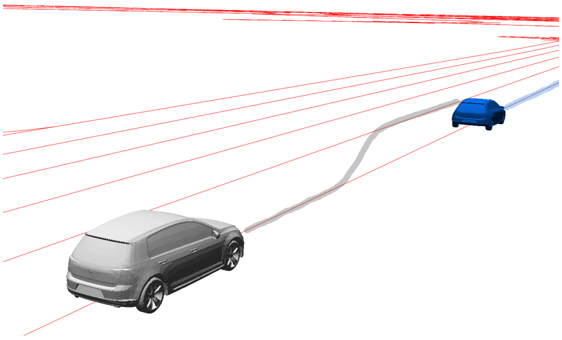


Figure 4.11.: Scenario 2: Overtaking a slowly moving vehicle

Scenario 2 - Error Model 1 (True Positive Inaccuracy: Object Position Shift)

Again, the effect of an inaccurately perceived object position is analyzed. The perceived position of the preceding vehicle is flawed by injecting a continuously increasing systematic error. Thus, the two parameters $\Delta x_{obj,\mu}$ and $\Delta y_{obj,\mu}$ of the corresponding error model are utilized. The resulting inaccuracy space is sampled in a spiral manner with an initial step size of 1 m for both parameters inside the range $[-10\,\text{m}, 10\,\text{m}] \times [-10\,\text{m}, 10\,\text{m}]$. As in the investigated left turn scenario before, the fail criterion for executed test cases is a collision of the ego vehicle with the ground truth object bounding box. One failed test case repetition leads to the corresponding test case being classified as failed. Results of the test process investigating the response of the ego vehicle to an inaccurate object position in the overtaking scenario are depicted in Figure 4.12.

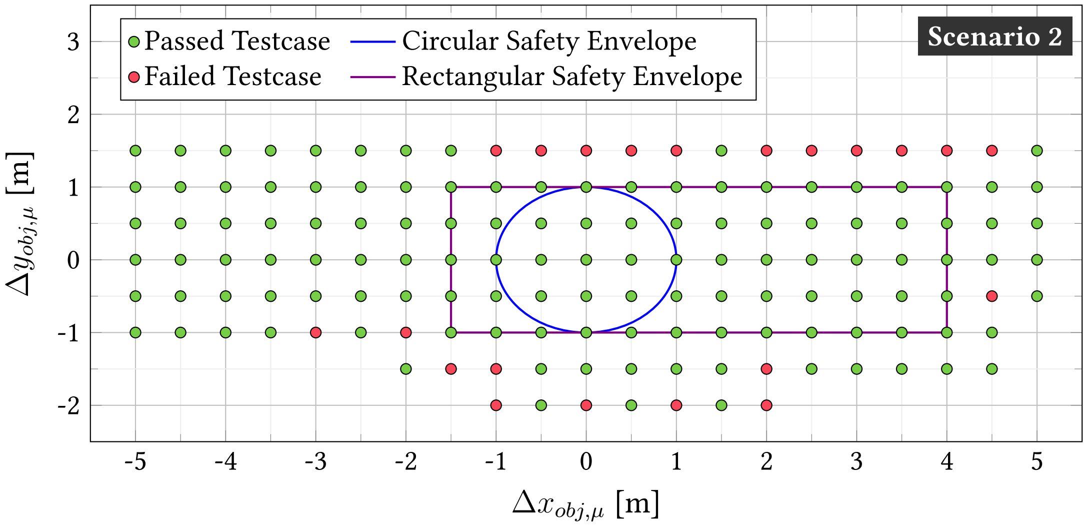


Figure 4.12.: Scenario 2: Test cases and resulting safety envelope regarding position inaccuracy

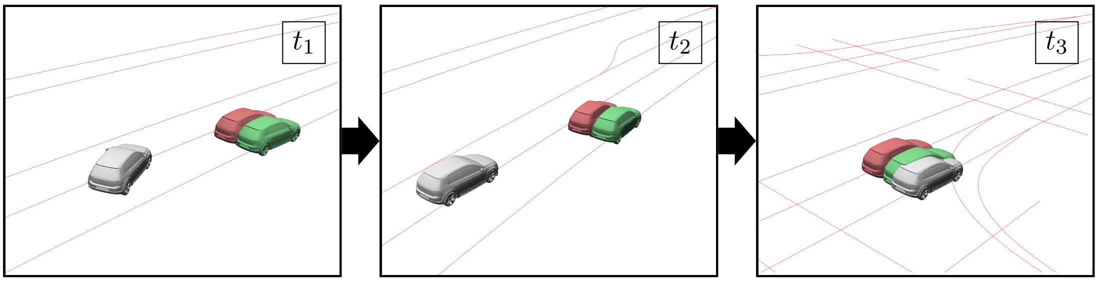


Figure 4.13.: Exemplary failed Test Case $(\Delta x_{obj,\mu}, \Delta y_{obj,\mu}) = (0\,\text{m}, 1.5\,\text{m})$: Ego perceives the preceding vehicle (green) with an inaccurate lateral position (red). Firstly, Ego decelerates when approaching the standing vehicle, but then decides for a change to the most left lane to pass it (t_1). At the same time, the preceding vehicle starts moving with a higher acceleration than Ego, thus leading to Ego aborting its *Passing* maneuver (t_2). A few seconds later at the exit of the crossed intersection, the preceding vehicle - now being slower - is presumptively moving in the center lane. Consequently, Ego decides to pass it on the right, which leads to a collision due to an overlap with the real vehicle (t_3).

The sensitivity analysis of inaccurate object positions in the overtaking scenario shows a higher tolerance of the *Plan* component regarding longitudinal inaccuracies. While the elicited thresholds for lateral inaccuracies are similar to the elicited inaccuracies in the left turn scenario, this scenario comprises a stricter requirement regarding the preceding vehicles' longitudinal position. This seems plausible, since the ego vehicle is firstly following and then approaching the preceding vehicle before cutting out. The elicited lateral inaccuracies also seem reasonable, since a lateral shift to the right or to the left can already lead to the preceding vehicle being matched to either the left lane (cf. Figure 4.13) or not being on the road at all. This directly tricks the planner into not considering a lane change at all, since the ego vehicles driving lane seems to be empty in front. Based on the results, another safety envelope is defined (cf. Figure 4.12). The requirement corresponding to the circular safety envelope is as follows:

Requirement 4

The tested *Plan* component requires a *Sense* component that reports an oncoming vehicle's position with less than 1 m inaccuracy in the simulated overtaking scenario in an urban setting.

Scenario 2 - Error Model 2 (Field of View: Distance Thresholding)

For the identification of a minimal perception range in the overtaking scenario, we follow the same approach of multiple phase sampling as for the previous left turn scenario. The structured testing process starts with a rough sampling and is set up to firstly perform tests in the range of [50 m,5 m]. We select 50 m as an upper limit due to the relatively low speed of the preceding vehicle. Afterwards, based on the results of the first phase, a more refined sampling with a smaller range and smaller step size is performed. Since the first failed test case occurred with $d_{fov} = 5\,\text{m}$, the testing process starts at the last passed test case ($d_{fov} = 10\,\text{m}$). It comes to a stop when the test case for $d_{fov} = 7\,\text{m}$ fails. Subsequently, d_{fov} is increased again and another test process is started, which finds at least one test case repetition for $d_{fov} = 8\,\text{m}$ to fail. Ultimately, the range [9 m,8 m] is sampled and $d_{fov} = 8.1\,\text{m}$ is identified as the smallest parameter value not leading to a failure in any of the corresponding test case repetitions. The results are visualized in Figure 4.14.

Of special interest is the non-deterministic behavior of the system under test for the parameter value $d_{fov} = 8\,\text{m}$. While none of the corresponding test case repetitions fails in the second phase (# 13), at least one of the repetitions fails in the third phase (# 19). We elicit the following safety requirement for the field of view:

Requirement 5

The tested *Plan* component requires a *Sense* component that detects an oncoming vehicle at latest at a distance of 8.1 m in the simulated overtaking scenario in an urban setting.

Scenario 2 - Error Model 3 (Object Track Instability: Lifetime and Downtime)

Again, we investigate the previous same two aspects regarding the robustness of the *Plan* component under test in the overtaking scenario with help of the object track error model. Firstly,

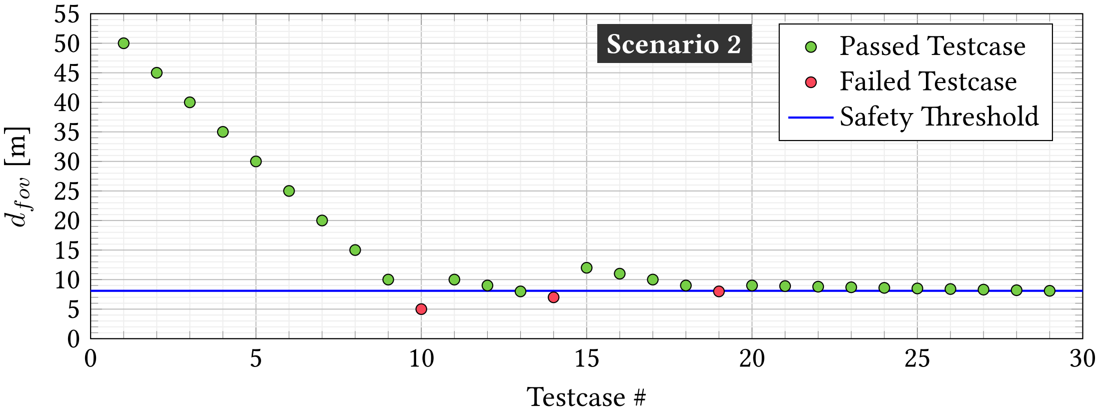


Figure 4.14.: Scenario 2: Test cases and resulting safety threshold regarding field of view

the sensitivity towards changing object IDs caused by an object tracking insufficiency is analyzed. Secondly, the effect of missing the preceding vehicle for an increasing amount of time is examined. Like in the left turn scenario before, no visible reaction regarding a changing object ID of the preceding vehicle can be observed based on tests where only $t_{l,\mu}$ is varied. These results further confirm the robustness of the *Plan* component under test towards tracking errors. Subsequently, detection errors are injected within the next structured testing process to investigate robustness towards detection insufficiencies. Also like in the analyzed scenario before, this testing process is conducted with parameter settings $t_{d,\mu} \in [0\,\mathrm{s}, 10\,\mathrm{s}]$ (step size 0.1 s) and $t_{l,\mu} \in [1\,\mathrm{s}, 10\,\mathrm{s}]$ (step size 1 s) under the condition $t_{d,\mu} < t_{l,\mu}$. The results are depicted in Figure 4.15.

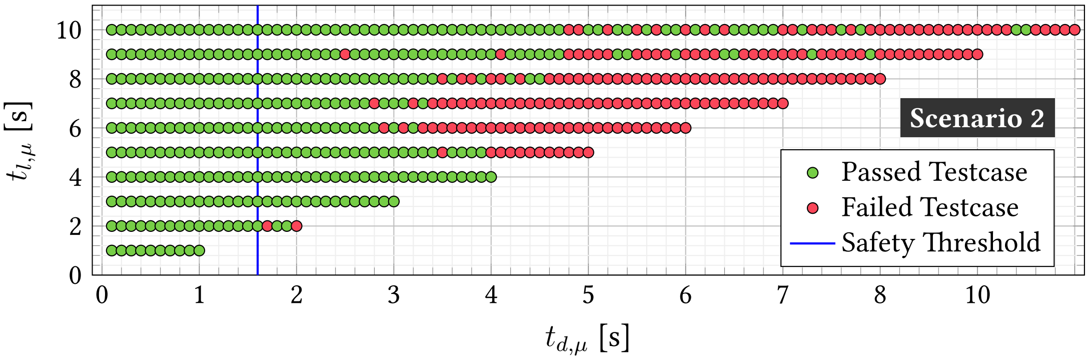


Figure 4.15.: Scenario 2: Test cases and resulting safety threshold regarding fragmentary object tracks

Since the parameter $t_{l,\mu}$ alone is not regarded as a hazardous influence based on the tests before, its purpose in the test process corresponding to Figure 4.15 is rather to shift the activation of the *False Negative* error through time. The failed test case with the smallest value for $t_{d,\mu}$ appears at $t_{l,\mu} = 2$ s and $t_{d,\mu} = 1.7$ s. Therefore, we elicit the following requirement based on the test results:

Requirement 6

The tested *Plan* component requires a *Sense* component to never miss a preceding vehicle for more than 1.6 s in the simulated overtaking scenario in an urban setting.

4.4.4. Discussion

Let us briefly discuss the obtained results, the generalizability of the approach and threats to validity of the presented test setup and results.

Obtained assumptions / guarantees

We conjecture that the left turn scenario is valuable towards eliciting a meaningful requirement for the minimum range of the *Sense* component because of the relatively large distance that the oncoming vehicle covers in a relatively short amount of time in the scenario. The lower speed of both the preceding vehicle and the ego vehicle in the overtaking scenario result in a more lenient requirement regarding the minimum field of view range. Thus, this requirement is dominated by the *at least required 56 m* of the left turn scenario.

Both scenarios are adequate for identifying meaningful requirements regarding inaccuracy of the perceived lateral object position. An unprotected left turn maneuver can comprise keeping a relatively small lateral distance to an oncoming vehicle for a short amount of time (even when executed perfectly safely). Estimating the lateral position of a preceding vehicle accurately is mandatory to keep a safe lateral distance while performing a safe passing maneuver. For the elicitation of a meaningful requirement regarding the acceptable longitudinal position shift, we deem the overtaking scenario to be more suitable than the left turn scenario. While the *Plan* component under test shows a higher tolerance regarding longitudinal position shifts in the left turn scenario, the overtaking scenario leads to a stricter requirement (cf. Table 4.1). Object position inaccuracy requirements elicited for the overtaking scenario dominate the corresponding requirements from the left turn scenario.

Table 4.1.: Tolerated perceptual insufficiencies of the *Plan* component under test in both simulated scenarios. The lateral and longitudinal object position inaccuracy values are specified within the corresponding objects coordinate system. Dominating inaccuracies which correspond to stricter requirements are highlighted.

Perceptual insufficiency	Scenario 1 *"Left Turn"*	Scenario 2 *"Overtaking"*
Field of view upper range limitation	> 56 m	> 8.1 m
Object position inaccuracy (lat.)	> -1 m, < 2 m	> -1 m, < 1 m
Object position inaccuracy (long.)	> -2 m, < 10 m	> -1.5 m, < 4 m
Object position inaccuracy (absolute)	< 1 m	< 1 m
Object misdetection duration	< 0.9 s	< 1.6 s

As emphasized earlier, timing is an important factor regarding loss of an object track. Objectively seen, the faster a vehicle is and the more relevant it becomes while not being perceived, the more severe the corresponding error of losing track is. This statement is supported by the stricter requirement regarding object track loss of the fast oncoming vehicle in the left turn scenario. The deciding concrete test cases for accepted time gaps (downtimes) between two object tracks comprises the situation that the system under test was not able to detect the oncoming vehicle right before it entered the crossing area. While this can already be seen as critical, it cannot be ruled out that there exists a shorter downtime in the scenario linked to another situation that would subsequently lead to a stricter requirement. For the automatic identification of critical situations including such errors, reward functions [77] or optimization [20] could be considered as strategies for further exploring the inaccuracy space. Table 4.1 summarizes the elicited accepted perceptual inaccuracies for the prototypical *Plan* component under test based on the two investigated scenarios.

Generalizability

Our initial application shows that our proposed perception error injection testing process for a given *Plan* component can generate results which can be utilized for identifying accuracy requirements for a *Sense* component in concrete driving scenarios. By applying the technique to multiple scenarios and by investigating more perceptual hazards, it would certainly be possible to specify initial quantifiable and measurable quality criteria for a *Sense* component which are needed for safety validation — either per scenario or aggregated, e.g., for road types or for an urban setting vs. the motorway. A key question to address will then be the right level of aggregation, trading many, bespoke requirements in individual scenarios for fewer, more stringent requirements covering a range of different scenarios.

In a next step, the identified acceptable errors need to be analyzed together and not isolated from one another to identify potential dependencies. Together with more classes of perceptual errors and hazards, the space to explore grows quickly and it will not be possible to explore it exhaustively. Scalability will hinge on adding other techniques than simulation-based testing: e.g., a theory of combining the effects of errors or a feedback loop with software verification that can actually discharge guarantees and use assumptions in formal proofs. For a decomposed safety validation, it is indispensably necessary, that the elicited requirements for *Sense* components of automated driving functions in urban settings can actually be verified. An exemplary verification of requirements for a flight-critical system is conducted by Brat et al. [19].

Validity

The obtained results pertain to a hypothetical *Sense* component and one concrete *Plan* component and as such can only be understood as a first step.

Concept Validity. As discussed above, many more steps will be required in order to arrive at an actual assume/guarantee-style safety argument for autonomous vehicles. The presented approach seems suitable for providing sensible initial accuracy requirements.

Internal Validity. Safety validation relying on black-box methods has its limitations due to the fact that there can never be a proof of correctness and only statements about the performance based on expert-knowledge and statistics will be available. What this work has shown, however, is that sensible assumptions/guarantees seem to exist for which safe behavior can be achieved consistently in the presence of perceptual inaccuracies.

External Validity. While we did only analyze two concrete scenarios in conjunction with three different error models, we can already see that different scenarios can lead to differently strict requirements regarding the same kind of perception error (cf. Table 4.1). A major challenge will be the compilation of a minimum scenario set which leads to the most strict accuracy requirements for a *Sense* component to be implemented and verified.

Part IV.

Evaluation of Perception Performance

Reference Generation and Task-oriented Evaluation

Conducting a task-oriented evaluation of *Sense* components entails two prerequisites. The first prerequisite is to have reference data, which corresponds to ideal perception results without any errors or inaccuracies. Often, this data is also called ground truth. An example would be the true position and dimensions of an existing surrounding vehicle or the precise location of a stop sign. Results generated by the *Sense* component must then be compared to ground truth data for determining the actual perception performance. The delta between results and ground truth is made of the perception errors (cf. Chapter 2). The acquisition of ground truth data for environmental elements can be differently challenging. Static elements and their attributes (e.g., traffic signs and their position or lane markings) are mostly constant and change only rarely due to temporal modifications like traffic constructions. Thus, corresponding ground truth data can be obtained by mapping efforts. The acquisition of ground truth data for dynamic elements, however, is cumbersome. Surrounding traffic participants change their velocity and thus their position all the time and their dimensions are diverse. Capturing these with a high accuracy requires to either have reference sensor systems with a higher resolution equipped to the automated vehicle or to consider traffic monitoring methods like drones following the automated vehicle. One example would be ,e.g., the validation of a neural network-based depth estimation on a camera image by comparing the result to corresponding lidar measurements. Another example would be the consideration of drone data to evaluate how well a *Sense* component detects objects which are partially or fully occluded. Finally, the result to be generated by the *Sense* component (e.g., object bounding boxes) must be labeled within this reference data - either manually by experts or semi-automatically/fully-automatically by revision and reconstruction processes - so that performance of the *Sense* component can be assessed. In Chapter 5, we propose an automatic revision process for the generation of reference values for the width, height and classification of perceived objects. The second prerequisite for the task-oriented evaluation of *Sense* components is to have a set of task-oriented quality criteria, which are considered for an assessment of perception quality. Common performance metrics in the field of object detection are, e.g., precision and recall scores, average precision or the CLEAR MOT metrics [11]. However, these metrics are not task-oriented per se, since they either consider all surrounding objects or only differentiate relevance based on attributes like heading or distance (cf. Section 1.3.1). A first step towards a task-oriented evaluation of *Sense* components is already made in Chapter 3, where the term *relevant road users* is broken down, and a further step is made in Chapter 4, where exemplary verifiable requirements for a *Sense* component are elicited. In a concluding step, we evaluate a *Sense* component based on the just mentioned preliminary contributions. Thus, Chapter 6 contains both an exemplary task-oriented evaluation regarding relevant objects and regarding requirements from Chapter 4.

5. Automated 3D Object Reference Generation for the Evaluation of Autonomous Vehicle Perception[1]

In this chapter, we address the challenge of generating reference values for the evaluation of online perception systems. For that, we propose an offline, rule-based revision process to generate reference dimension and classification values, which are based on the on-board perceived objects subjected to inaccuracy. This comprises both a reclassification of objects and a reestimation of object width and length to acquire accurate reference values. The process utilizes a map for inferring infrastructure information of given positions. The perceived objects are processed by iterating over consecutive object states. Indications and confident measurements are then exposed for generating classification and dimension references. The reclassification considers defined indications like explicit kinematic progressions or specific behavior patterns including reaction to the present infrastructure. Reestimation of object width and length considers favorable observing situations which are identified with respect to features like distance, relative heading and occlusion.

The research question we address here is whether an automated process that considers objectively defined rules and indications can be utilized to post-process perceived objects to obtain more accurate attributes. For that purpose, we implement the proposed method and apply it on datasets recorded during test drives in the city of Hamburg, Germany. Furthermore, we evaluate results of our approach by comparison to manually labeled ground truth object classifications and dimensions. This addresses accuracy of the references compared to original object attribute values (onboard) as well as correctness of the generated references compared to the reality.

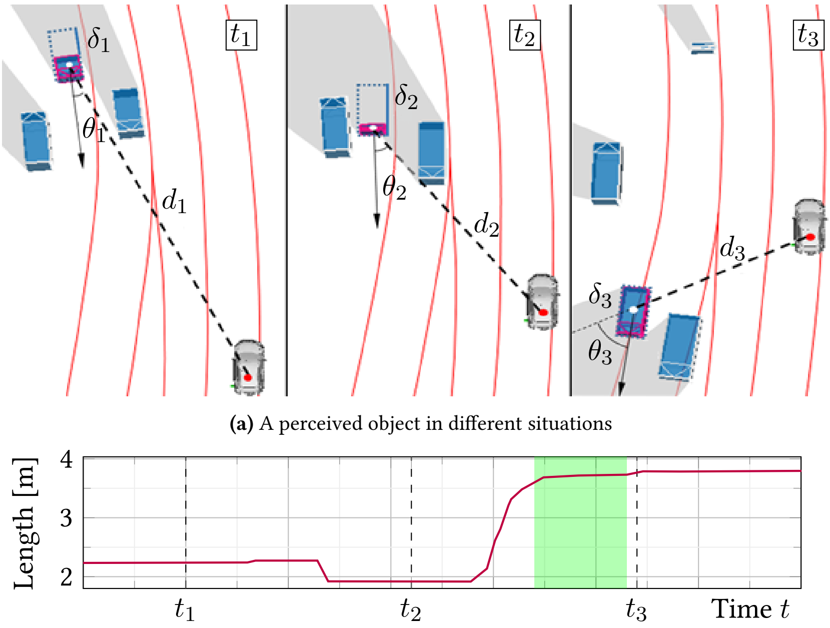
Our contribution is threefold:

1. We propose an automated offline dimension and classification estimation including concrete situations indicating accurate dimensional measurements and a decision tree for the classification of traffic participants.

2. We apply the process on a dataset collected while conducting automated test drives in the city of Hamburg, Germany, evaluate results and show process feasibility.

3. We touch on the applicability of such an automated process for facilitating perception evaluation and discuss existing limitations.

[1] This chapter is based on Paper III [85] and therefore contains verbatim content previously published (©2021 IEEE).



(a) A perceived object in different situations

(b) Length measurements of the perceived object

Figure 5.1.: Reference generation concept exemplarily demonstrated by online length measurements of a perceived object. (a) Real dimensions are indicated by the dotted box, estimated online dimensions are marked in magenta. (b) Most reliable measurements (green area) that are considered for a reference value appear around t_3 due to the relatively great projection angle θ, smaller distance d and minor occlusion rate δ. (©2021 IEEE)

5.1. Preprocessing & Reference Generation

Our proposed method relies on a set of traffic participants perceived over time (often referred to as object list). These objects exhibit attributes like position, velocity, dimensions and a classification. In conjunction with an HD map these objects can be accurately located and linked to specific infrastructure elements they are interacting with (turning on a junction, using a crosswalk, etc.). The main idea is as follows: Since the view on observed objects changes over time and thus affects sensor recording conditions and subsequent object segmentation, measurements vary in reliability. While width of a close leading vehicle might be well ascertainable, the length measurement of it might be distorted at the same moment as the corresponding side is less captured. Another static attribute of an object is its classification. Finding specific patterns in object lifetimes like usage of a crosswalk (likely a pedestrian) or a bicycle-sized object heavily accelerating at a traffic light (likely a motorbike) reveals object classifications. By systematically analyzing the occurred situations for each object and attaching confidences to measurements, the method aims at generating more accurate references for length, width and the classification of objects. Our proposed method consists of a preprocessing stage and the reference generation stage (cf. Figure 5.2). The process is fully rule-based.

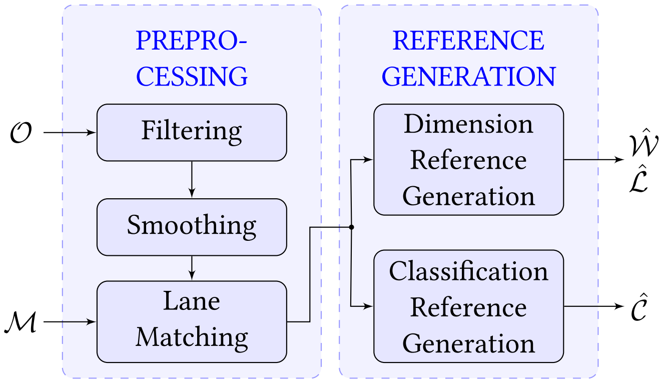


Figure 5.2.: Processing steps of proposed revision process: Objects $\mathcal{O}$ including all of their perceived states over time with corresponding attributes and an HD map $\mathcal{M}$ are taken into account to automatically generate more accurate width, length and classification references ($\hat{\mathcal{W}}, \hat{\mathcal{L}}, \hat{\mathcal{C}}$). These references are a set of tuples consisting of an object ID and a value. (©2021 IEEE)

5.1.1. Preprocessing

Firstly, the preprocessing takes care of object filtering. For our initial application later, we focus on objects that are potentially of higher relevance and perceived in diverse situations. Therefore, only objects that are closer than 40 m [78] to the ego vehicle at any point during their lifetime are considered. Secondly, the estimated heading measurements of objects are mended. This comprises unreasonable glitches or confusing the side and front of vehicles, which results in a swap of width and length estimation. The restoration of a smooth heading signal over time is not only important for inferring dimension references later on, but also for proper lane matching as the last step of preprocessing. Fusing the lane matching result with object position and velocity attributes over time helps to distinguish different types of objects according to their interaction with diverse infrastructure elements. While motorized vehicles are usually driving on the road, pedestrians are mainly walking on side- and crosswalks. After the preprocessing is done, the remaining objects are analyzed for the intended reference generation.

5.1.2. Dimension Reference

Dimension references are calculated considering reliable measurements. Objectively, reliability of measurements is indicated by the sensor recording condition. In our work this is quantitatively described by the three features: distance d, projection angle θ and occlusion rate δ. While the relative distance d to a surrounding object is an intuitive factor, the later two are briefly introduced in the following.

Projection Angle

The projection angle θ_o of an object O is the aggregation of both object position and orientation relatively to the ego-vehicle. The projection angle is the acute angle spanned by the object heading vector (roll axis) $\overrightarrow{x_o}$ and the path between ego position P_e and object position P_o:

$$\theta_o = \min\left(\angle\left(\overrightarrow{P_e P_o}, \overrightarrow{x_o}\right), \angle\left(\overrightarrow{P_o P_e}, \overrightarrow{x_o}\right)\right). \tag{5.1}$$

Based on the definition, $\theta \in [0, \frac{\pi}{2}]$ applies. While a large projection angle indicates an object being oriented orthogonally to the ego vehicle, a small projection angle corresponds to an object being headed either in the same or directly the opposite direction. Subsequently, a small projection angle of an object results in a more favorable situation for measuring its width, while a large projection angle is advantageous for measuring the object's length.

Occlusion Rate

The occlusion rate denotes to which extent an object might be occluded. For the identification whether an object might be occluded by another, we utilize a simplified geometrical analysis. Given a target object, the area between the ego vehicle and the target object is checked for other objects. If there is an obstructing object in between, the horizontal area that is obstructed by it is considered. As a next step, it is analyzed which portion of the edge of interest (width or length) of the target object lies inside the obstructed area. To cope with the effect of the target object possibly being small because of the existing occlusion, the maximum value for the edge of interest over the lifetime of the target object is considered. Note that the sensor set with which the corresponding objects were captured is located higher than the usual passenger car height. This means, that an accurate length or width measurement can also exist for an object that is horizontally occluded. Therefore, our utilized geometrical analysis is rather strict and over-approximates a possible occlusion. An object with an occlusion rate of $\delta = 0$ is directly perceived without any other objects being in the view, while an occlusion rate of $\delta = 1$ denotes an object being fully behind an obstructing object.

Confidence Scores

The proposed quantitative description of recording conditions is represented as confidence scores. For indicating influences on reliability of dimensional measurements by distance d, occlusion rate δ and projection angle θ, confidence score functions are designed:

$$S_d\left(d\right) = \begin{cases} \exp\left(-\frac{d^{1.7}}{{\mu_1}^2 - \mu_1(1-\varepsilon)d}\right) & , d \in [0, \mu_1] \\ 0 & , \text{else} \end{cases} \tag{5.2}$$

$$S_\delta\left(\delta\right) = \begin{cases} \exp\left(-\frac{\delta^{2}}{{\mu_2}^2 - \mu_2(1-\varepsilon)\delta}\right) & , \delta \in [0, \mu_2] \\ 0 & , \text{else} \end{cases} \tag{5.3}$$

$$S_\theta^l\left(\theta\right) = \begin{cases} \frac{4}{\pi^2\left(1-\frac{2\alpha}{\pi}\right)^2}\left(\theta^2 - 2\alpha\theta + \alpha^2\right) & , \theta \in \left[\alpha, \frac{\pi}{2}\right] \\ 0 & , \text{else} \end{cases} \tag{5.4}$$

$$S_\theta^w\left(\theta\right) = \begin{cases} \frac{1}{\beta^2}\theta^2 - \frac{2}{\beta}\theta + 1 & , \theta \in [0, \beta] \\ 0 & , \text{else} \end{cases}, \tag{5.5}$$

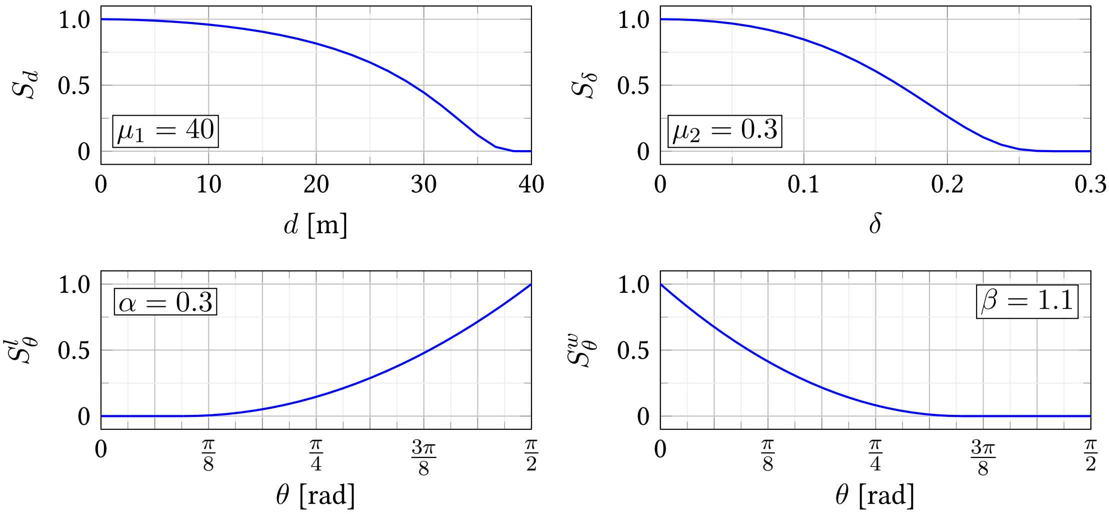


Figure 5.3.: Confidence score functions: Confidence based on relative distance d and occlusion rate δ is calculated similarly for length and width measurements (S_d, S_δ). Confidence based on the object projection angle θ differs for length and width measurements (S_θ^l, S_θ^w respectively). (©2021 IEEE)

where $\varepsilon > 0$. The nature of the functions and their parameterization (acceptance thresholds $\mu_1, \mu_2, \alpha, \beta$) were chosen empirically and could be tailored to measurement reliability of other systems (Figure 5.3). However, the trend of the confidence functions should be valid in general (measurement reliability is higher for less distance and less occlusion, and length/width measurements are more reliable for a greater/smaller projection angle). The aggregated confidence of a width or length measurement of an object at a given time is the sum of the three confidence scores. After conducting experiments, we found the projection angle θ having the biggest influence on the measurement accuracy and therefore weight the three confidence scores with a ratio of 1:1:3. Subsequently, measurements for width and length are sorted decreasing in confidence. Up to five most confident measurements are averaged and result in a generated dimensional reference.

5.1.3. Classification Reference

Classifications considered by our proposed approach are *Passenger Car*, *Van*, *Truck*, *Bus*, *Vehicle Group*, *Motorbike*, *Bicycle*, *Pedestrian*, *Parking Group* and *Unknown*. While most of these are common, there are also the two vague object classifications *Vehicle Group* and *Parking Group*. Considering the dataset later used in this work and that the goal of this approach is not to take care of multi-object errors [18], sensor objects comprising more than one traffic participant should also obtain a more accurate classification reference. We define a *Vehicle Group* to be a larger object comprising several vehicles, e.g., occurring when vehicles stand closely next to each other at a traffic light. These objects usually decay into single vehicle objects when starting to move with a different kinematic profile. A *Parking Group* is considered to be a larger object next to the road comprising several static traffic participants, infrastructure elements like

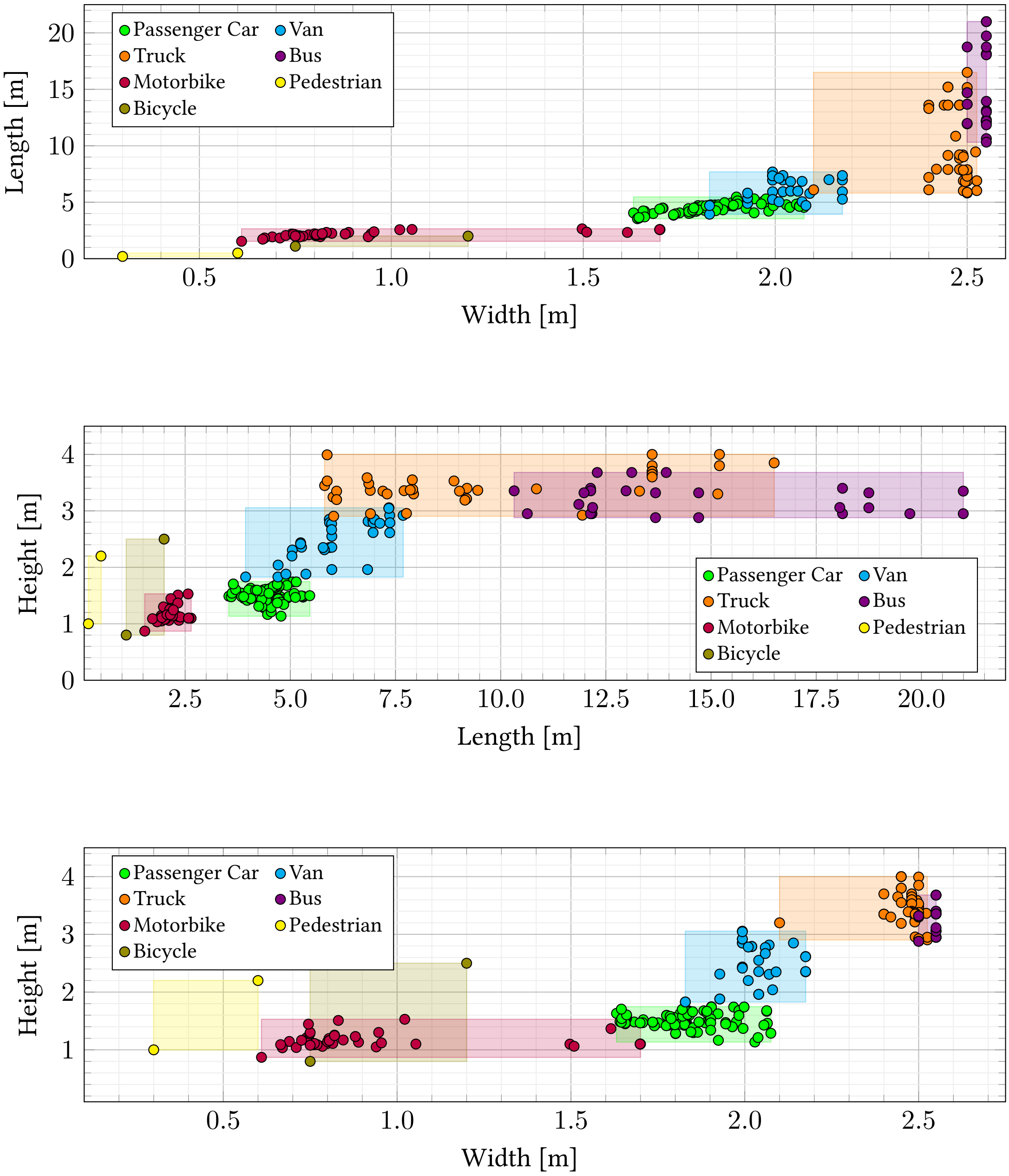


Figure 5.4.: Visualization of gathered object dimensions and derived approximate ranges: The vehicle dataset consists of 90 passenger cars, 33 vans, 42 trucks, 25 busses and 40 motorbikes. The information is based on manufacturer specifications. The length of trucks is defined by the cab and the truck trailer. Since trailers can vary in length and are not easily identifiable, plotted aggregated truck lengths are exemplary. The width specification usually does not consider side mirrors, thus creating a margin. The specified motorbike dimensions do not consider the drivers size. (©2021 IEEE)

poles or scenery elements like bushes. In the utilized data, such objects can, e.g., be observed comprising sidewalk construction elements and a parking vehicle. Due to the variety and changing of elements being included in these objects, *Parking Group* objects tend to have an unusual velocity profile and quickly changing dimensions.

The classification reference generation is implemented by a decision tree. It relies on indications deducted while iterating over the perceived states of an investigated object. Features that are considered by this tree are rough dimensional information, kinematic behavior and interaction with infrastructure elements. Since the classification reference generation is independent of the dimension reference generation, there are no dimension references available for the processed objects at this point in our process. Therefore, for the initial width, length and height of an object, the averages of the measurements from the longest interval where the value changes the least is taken into consideration. In the following, we will summarize our identified features and finally explain how we designed the decision tree.

Feature - Object Dimensions

Classifying objects based on their estimated width, length or height requires an understanding about the typical dimension ranges of different object classifications. To that end, we collected a dataset of vehicle dimensions based on manufacturer specification (cf. Figure 5.4). Additionally, we refer to anthropometric studies to elicit dimension ranges for *Pedestrian* and technical guidelines to derive dimension ranges for *Bicycle*. Length of a pedestrian is mainly determined by body depth, which can, e.g., be measured by chest depth or abdominal depth. Width of a pedestrian is decided by body breadth, which considers, e.g., elbow to elbow, shoulders or the hip. Hanson et al. [47] provide dimension ranges of 15 cm to 42 cm for length and 30 cm to 57 cm for width (for ages 18 to 65). We derive a height range from the study by Fryar et al. [42], where measured heights for humans vary between 92.2 cm and 189.6 cm (for ages 2 to 60+). Regarding dimension ranges for *Bicycle*, we refer to a guideline by the American Association of State Highway and Transportation Officials (AASHTO) for the development of bicycle facilities [4] and the technical regulation from the Union Cycliste Internationale (UCI) [116]. According to the AASHTO, the physical width of a cyclist in specified as 75 cm, while the height is estimated to be between 1.5 m (eye height) and 2.5 m (operating height of an adult standing upright on the pedals) [4, p. 3-2]. Typical lengths of typical bicycles range from 1.1 m to 2 m [4, p. 3-3] [116, p. 10]. All elicited dimension ranges are visualized in Figure 5.4. We observe following indications from our dataset:

- *Width* × *Length* combination is suitable to distinguish *Motorbike* from other vehicles.
- All combinations are suitable to distinguish *Passenger Car* from *Truck* and *Bus*.
- *Height* is suitable to distinguish *Passenger Car* from *Van*.
- *Height* is promising to distinguish *Van* from *Truck* and *Bus*.
- *Bus* tends to be longer than *Truck*.
- *Length* is suitable to distinguish *Pedestrian* from *Bicycle* and motorized vehicles.

Regarding the remaining three object classifications *Vehicle Group*, *Parking Group* and *Unknown*, no quantitative dimension ranges can be determined since these classifications do not represent one specific type of traffic participant. *Vehicle Group* and *Parking Group* objects can comprise multiple traffic participants or infrastructure elements over time, which leads to implausible, but also abruptly changing dimensions. They are the product of object segmentation insufficiencies of the environmental perception component. Similarly, objects being detected by the perception component under test can already be classified as *Unknown*. When the perception component is not able to deduce a reasonable classification for a detected object in real time, the class *Unknown* serves as a placeholder for the subsequent prediction and decision-making. For our purpose, *Unknown* refers to tiny objects like cones, bicycle stands or bushes or tall objects like traffic light poles, for which we have not considered an individual classification.

Feature - Kinematic Behavior

Another important aspect when differentiating classes of traffic participants is their motion profile. Traffic participants are not allowed to move freely along urban road networks due to the traffic management. Traffic elements like red lights, stop signs or traffic jams demand traffic participants to slow down and wait, until they are required to move again. The corresponding travel speed and acceleration/deceleration profiles of each traffic participant class differs significantly. While motorized vehicles in cities usually travel by going the speed limit (e.g., $30\,\mathrm{km\,h^{-1}}$, $50\,\mathrm{km\,h^{-1}}$) when possible, pedestrians are walking or running with significantly lower speed. While passenger cars and motorbikes are able to accelerate quickly, pedestrians and cyclists are slower and their respective velocity does not change as abruptly. To derive typical velocity and acceleration ranges for the different traffic participants classifications, we refer to multiple studies in the following where various vehicle types and vulnerable road users have been observed regarding their motion profile.

Pedestrian. Zębala et al. [127] state that pedestrians usually move with a velocity between $1.1\,\mathrm{m\,s^{-1}}$ (ordinary pace) and $2.5\,\mathrm{m\,s^{-1}}$ (fast pace). When sprinting, pedestrians can reach velocities of $6.9\,\mathrm{m\,s^{-1}}$ (male aged 31-40). Additionally, they discuss that pedestrian acceleration usually ends after a distance of about 1 m is covered. Their subsequent movement is typically executed with a constant velocity (cf. [127]).

Bicycle. Depending on whether designated bike lanes exist, cyclists in urban areas travel either on the driving lane or in parallel to it. In a study by Romanillos and Gutiérrez [94], travel speeds of cyclists are separated in clusters from $0\,\mathrm{km\,h^{-1}}$ up to $35\,\mathrm{km\,h^{-1}} = 9.7\overline{2}\,\mathrm{m\,s^{-1}}$. Karakaya et al. [63] provide an analysis regarding acceleration and deceleration behavior of cyclists based on the SimRa dataset [62]. They show that cyclists usually do not go slower than $2\,\mathrm{m\,s^{-1}}$. Regarding the acceleration/deceleration capabilities behavior of cyclists, they extract from the dataset that most acceleration values lie in a range from $-2\,\mathrm{m\,s^{-2}}$ to $2\,\mathrm{m\,s^{-2}}$.

Motorized vehicles. When not specified otherwise, the common speed limit in German urban areas is defined as $50\,\mathrm{km\,h^{-1}} = 13.\overline{8}\,\mathrm{m\,s^{-1}}$. This is also the case for the domain where our investigated datasets were recorded in (cf. Section 5.2). Traffic participants like *Passenger Car*, *Van*, *Truck*, *Bus* and *Motorbike* are commonly observed there travelling with speeds between $0\,\mathrm{m\,s^{-1}}$ and $60\,\mathrm{km\,h^{-1}} = 16.\overline{6}\,\mathrm{m\,s^{-1}}$. Bokare and Maurya [13] provide typical acceleration values for various motorized vehicles based on their study using Global Positioning System (GPS).

According to their observations, vehicles of type *Passenger Car* and *Van* typically show acceleration values between $-5\,\mathrm{m\,s^{-2}}$ and $2.87\,\mathrm{m\,s^{-2}}$. Acceleration values for *Truck* are ranging from $-0.88\,\mathrm{m\,s^{-2}}$ to $1.00\,\mathrm{m\,s^{-2}}$, which we also consider for the classification *Bus*.

Vehicle Group, **Parking Group**, **Unknown.** Again, only qualitative statements regarding kinematics of *Vehicle Group*, *Parking Group* and *Unknown* objects can be made. As long as a *Vehicle Group* object is tracked and refers to the same traffic participants, velocity and acceleration values should be similar to the other motorized vehicles. Objects of class *Parking Group* and *Unknown* can show implausible velocity or acceleration values, but do not have to all the time. Taking all the above into consideration, we observe following indications:

- High velocities (greater than $10\,\mathrm{m\,s^{-1}}$) are promising to distinguish motorized vehicles from *Pedestrian* and *Bicycle*.
- When not sprinting, *Pedestrian* typically moves not faster than $2.5\,\mathrm{m\,s^{-1}}$, which is also close to the typical minimum velocity of a moving *Bicycle*.
- The acceleration ranges overlap, which makes it difficult to differentiate traffic participants solely based on acceleration.

Feature - Interaction with Infrastructure

The last aspect we assess for deducing object classification hypotheses is where different traffic participant's typically move and wait in German urban traffic. According to the German road traffic regulation pedestrians are required to use the sidewalk (cf. StVO §25 [36]). Crossing the road shall be executed as quickly as possible or by using respective pedestrian crossing aids (crosswalks, pedestrian traffic lights). In general, cyclists share the road with other motorized traffic participants, except if Sign 237, 240 or 241 demand cyclists to use the existing bike lanes (cf. StVO Annex 2 [34]). These bike lanes are then usually located next to the sidewalk, part of the sidewalk or designated areas on the road. Traffic lights for cyclists are either separated or integrated into the traffic lights of pedestrian crossing aids (showing both a pedestrian and cyclist symbol). The corresponding crossing aid shall then be used by cyclists (cf. StVO §37 [37]). If these do not exist, cyclist traffic is regulated by the same traffic lights that are also valid for motorized vehicles, since they all share the same road then. The aforementioned designated infrastructure elements and crossing aids for cyclists and pedestrians like bike lanes or the sidewalk shall generally not be used by motorized vehicles (cf. StVO Annex 2 [34]). We define the following four variables which can be true or false for an object:

- *OnRoad*: The object is located on the driving lane (typically true for *Passenger Car*, *Van*, *Truck*, *Bus*, *Motorbike*, *Bicycle* and *Vehicle Group*).
- *InVRUArea*: The object is located on crossing aids, bike lanes or sidewalks (typically true for *Pedestrian* and *Bicycle*).
- *OnJunction*: The object is located in a junction area (typically true for any classification except *Parking Group*).
- *OnSidewalk*: The object is located on a sidewalk (typically true for *Pedestrian*).

Design of Decision Tree for Object Classification

The decision tree for creating classification references of objects is designed by considering the aforementioned features and our perception component's performance. Before we explain how the tree is constructed, we want to make three short remarks, that need to be kept in mind:

1. Not every feature of every traffic participant is always captured when travelling along or passing these traffic participants. Subsequently, not every feature needs to be fulfilled in order to deduce a likely object classification.

2. There can be exceptions to the classification dependent features: A cyclist might travel on the driving lane while being quicker than $35\,\mathrm{km\,h^{-1}}$, a car might park illegally on the sidewalk or a motor scooter might use a cyclist crossing aid.

3. Dimension and kinematic estimations of the processed objects are subject to inaccuracies. For example, some bounding boxes generated by the perception component consider the corresponding vehicle's mirrors, while others do not. Acceleration values are not reliable enough since these are not directly measured but derived from the velocity.

That being said, the collected features from the preceding sections should be generally valid, while our decision tree is tailored to suit our perception component's performance and thus the accuracy of objects. Taking all of this into account, our decision tree considers mostly dimensional features, but also maximum and median velocity as well as quartiles of all measured velocities of an object. Additionally, the location variables *OnRoad*, *InVRUArea*, *OnJunction* and *OnSidewalk* are used with different degree of strictness. While *OnRoad* and *InVRUArea* are only true for an object when it is more than 70 % of its lifetime in the corresponding area, *OnJunction* and *OnSidewalk* are already true for an object when it is inside a junction area or on the sidewalk for just one state.

The first step of the decision tree is to separate very small and very tall objects (*Unknown*) from the rest. Subsequently, the remaining objects are divided into moving and standing objects by evaluating each object's third quartile of all its observes velocity values. Based on the standstill objects initially determined dimensions, the decision tree leads either to the *Vulnerable Road Users (VRU)* subtree, to the classification *Vehicle Group* or to the *Car* subtree if the object is mainly on the driving lane and either wide or long. The moving objects are separated into fast moving and slowly moving objects. Fast moving objects then end up being a *Parking Group* or a *Vehicle Group*, *Motorbike* or car (*Passenger Car*, *Van*, *Bus* or *Truck*) if they are located on the driving lane. The type of motorized vehicle is again decided by the object's dimensions. The decision tree leads slowly moving objects to the VRU subtree, to become a *Parking Group* if the object is neither on the driving lane nor in a junction area, to become a *Vehicle Group* or to the car subtree if the object is mainly on the driving lane and either wide or long. The VRU subtree distinguishes objects into *Pedestrian*, *Motorbike* or *Bicycle*. For this purpose, the median and maximum velocity, width, length and height values as well as whether the object is located on the sidewalk or on the driving lane. The car subtree decides for *Passenger Car*, *Van*, *Bus* or *Truck* based on the object's dimensions. There exist several forks within the tree where the tree ends up with the decision to not decide for a classification due to lack of indications or evidence. The decision tree is shown in detail in Figure 5.5.

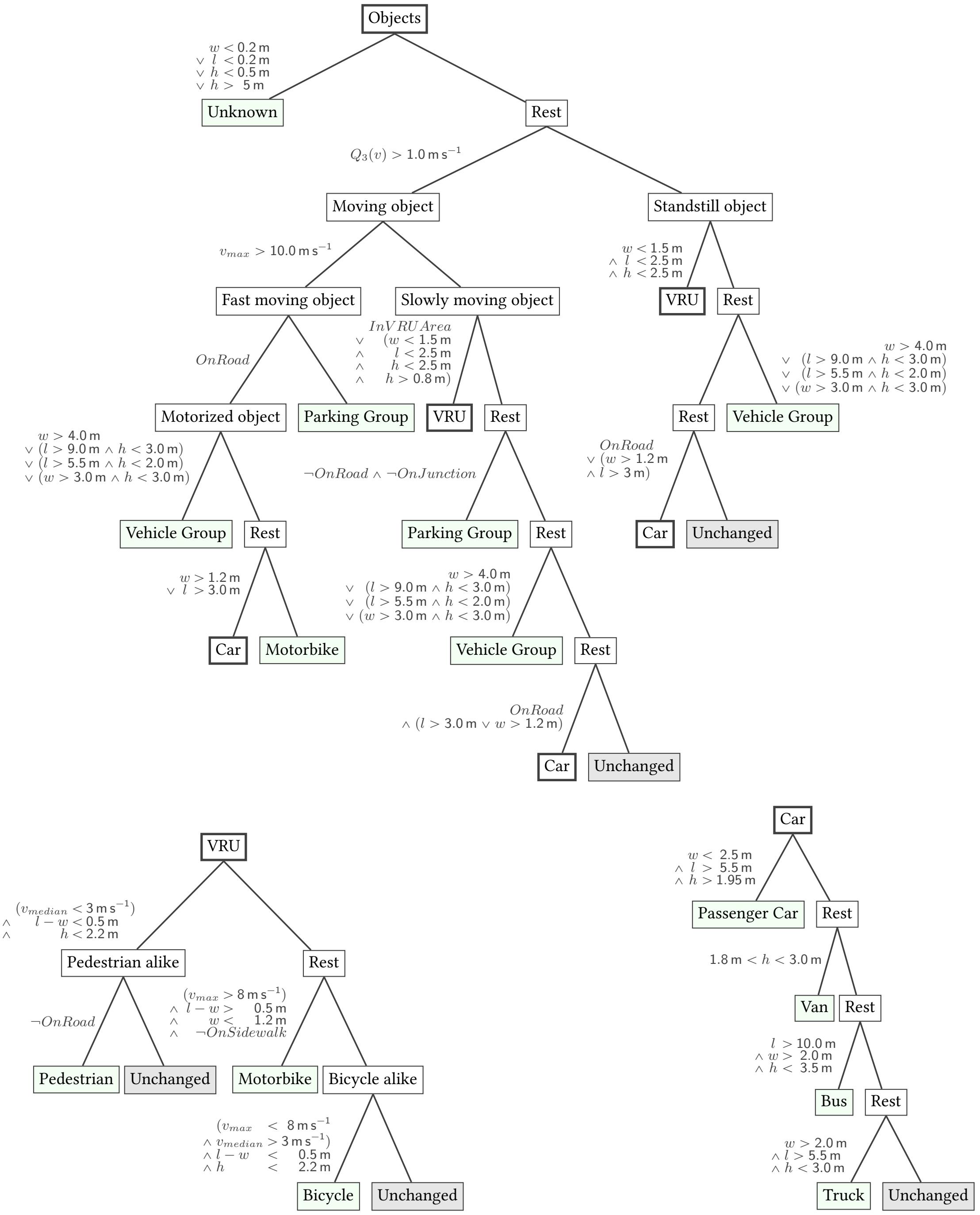

Objects
$w < 0.2\,\mathrm{m} \lor l < 0.2\,\mathrm{m} \lor h < 0.5\,\mathrm{m} \lor h > 5\,\mathrm{m}$
Unknown
Rest
$Q_3(v) > 1.0\,\mathrm{m\,s^{-1}}$
Moving object
Standstill object
$v_{max} > 10.0\,\mathrm{m\,s^{-1}}$
Fast moving object
Slowly moving object
$w < 1.5\,\mathrm{m} \land l < 2.5\,\mathrm{m} \land h < 2.5\,\mathrm{m}$
VRU
Rest
$OnRoad$
$InVRUArea \lor (w < 1.5\,\mathrm{m} \land l < 2.5\,\mathrm{m} \land h < 2.5\,\mathrm{m} \land h > 0.8\,\mathrm{m})$
$w > 4.0\,\mathrm{m} \lor (l > 9.0\,\mathrm{m} \land h < 3.0\,\mathrm{m}) \lor (l > 5.5\,\mathrm{m} \land h < 2.0\,\mathrm{m}) \lor (w > 3.0\,\mathrm{m} \land h < 3.0\,\mathrm{m})$
Motorized object
Parking Group
VRU
Rest
Rest
Vehicle Group
$w > 4.0\,\mathrm{m} \lor (l > 9.0\,\mathrm{m} \land h < 3.0\,\mathrm{m}) \lor (l > 5.5\,\mathrm{m} \land h < 2.0\,\mathrm{m}) \lor (w > 3.0\,\mathrm{m} \land h < 3.0\,\mathrm{m})$
$\neg OnRoad \land \neg OnJunction$
$OnRoad \lor (w > 1.2\,\mathrm{m} \land l > 3\,\mathrm{m})$
Vehicle Group
Rest
Parking Group
Rest
Car
Unchanged
$w > 1.2\,\mathrm{m} \lor l > 3.0\,\mathrm{m}$
$w > 4.0\,\mathrm{m} \lor (l > 9.0\,\mathrm{m} \land h < 3.0\,\mathrm{m}) \lor (l > 5.5\,\mathrm{m} \land h < 2.0\,\mathrm{m}) \lor (w > 3.0\,\mathrm{m} \land h < 3.0\,\mathrm{m})$
Car
Motorbike
Vehicle Group
Rest
$OnRoad \land (l > 3.0\,\mathrm{m} \lor w > 1.2\,\mathrm{m})$
Car
Unchanged
VRU
$(v_{median} < 3\,\mathrm{m\,s^{-1}}) \land l - w < 0.5\,\mathrm{m} \land h < 2.2\,\mathrm{m}$
Pedestrian alike
Rest
$\neg OnRoad$
$(v_{max} > 8\,\mathrm{m\,s^{-1}}) \land l - w > 0.5\,\mathrm{m} \land w < 1.2\,\mathrm{m} \land \neg OnSidewalk$
Pedestrian
Unchanged
Motorbike
Bicycle alike
$(v_{max} < 8\,\mathrm{m\,s^{-1}} \land v_{median} > 3\,\mathrm{m\,s^{-1}}) \land l - w < 0.5\,\mathrm{m} \land h < 2.2\,\mathrm{m}$
Bicycle
Unchanged
Car
$w < 2.5\,\mathrm{m} \land l > 5.5\,\mathrm{m} \land h > 1.95\,\mathrm{m}$
Passenger Car
Rest
$1.8\,\mathrm{m} < h < 3.0\,\mathrm{m}$
Van
Rest
$l > 10.0\,\mathrm{m} \land w > 2.0\,\mathrm{m} \land h < 3.5\,\mathrm{m}$
Bus
Rest
$w > 2.0\,\mathrm{m} \land l > 5.5\,\mathrm{m} \land h < 3.0\,\mathrm{m}$
Truck
Unchanged

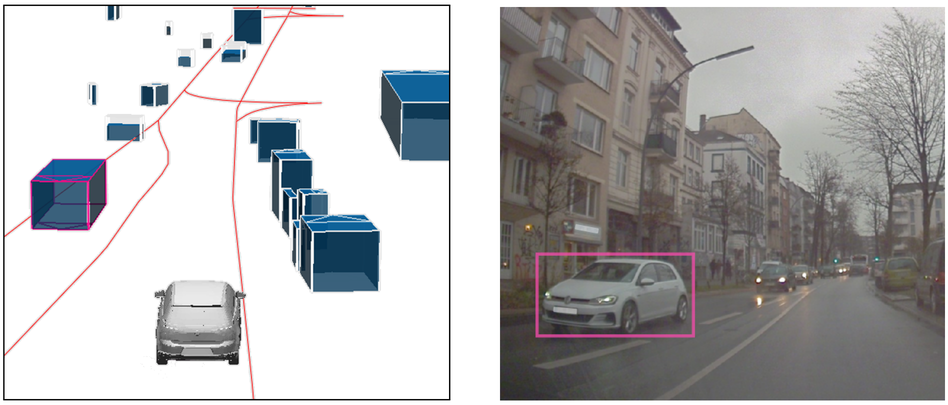
Figure 5.5.: Classification tree: The tree consists of a main tree and two sub-trees. Conditions are located left of the edge (except for *Vehicle Group*), corresponding neighbor edges hold the negated condition. $Q_3(v)$ is the third quartile of all measured velocities. *OnRoad* (driving lane) and *InVRUArea* (crossing aid, bike lane, sidewalk) are true when the object is in the respective area for more than 70 %. *OnJunction* and *OnSidewalk* are true when the object is inside a junction area or on the sidewalk for just one state. v_{max} and v_{median} are maximum/median speed and w, l, h initial dimensions of the investigated object. (©2021 IEEE)

5.2. Case Study

The proposed methodology is evaluated within the scope of three datasets. The datasets were collected during test drives conducted in the city of Hamburg, Germany in late 2019. Therefore, the data comprises mixed traffic consisting of both motorized and vulnerable road users operating in an urban environment. A brief overview of the datasets is given in the following section.

5.2.1. Dataset

The utilized datasets consist of ego states coming from the localization module, object states coming from the perception module and raw camera images recorded by cameras that are part of the perception module. The object states are the result of an object segmentation based on lidar sensors. Both a front and rear camera as well as multiple cameras equipped around the prototype vehicle enable a surround view of the environment. In the scope of this work, these datasets have been enriched by manually labeling identifiable object classes and dimensions. This is carried out by manually associating perceived objects with traffic participants in the corresponding camera images. Subsequently, motorized vehicles are further analyzed by identifying the exact model. Finally, exact dimension specification are elicited by considering manufacturer information. These manually acquired labels for both classification and width and length are needed to evaluate the accuracy of automatically generated references later on. An example of this process is given in Figure 5.6. Table 5.1 summarizes the utilized datasets.



(a) 3D-Visualization **(b)** Front camera

Figure 5.6.: Exemplary object with labeled dimensions: Golf Mk7 GTI (2017) with Length = 4.268 m, Width = 1.799 m, Height = 1.442 m (regarding manufacturer specification). Acquired labeled dimensions can exhibit minor inaccuracies due to slightly different model variants. (©2021 IEEE)

5.2.2. Evaluation

We propose a set of metrics to evaluate the generated dimensional and classification references. Both generated reference accuracy regarding the reality (ground truth) and accuracy improve-

Table 5.1.: Dimension and Classification Revision: Datasets Information (©2021 IEEE)

Set	Duration	Objects	Class Labels	Dimension Labels
$\mathcal{S}_1$	2 min 40 s	785	182	18
$\mathcal{S}_2$	2 min 30 s	691	150	14
$\mathcal{S}_3$	4 min 04 s	1460	295	55

ment regarding the online measurements and estimations are considered. The evaluation of generated dimensional and classification references only considers objects for which manual labels were created as described earlier. Dimensional references of vehicles are compared to ground truth labels for a quantitative evaluation. Generated dimension references of objects for which only a classification label exists are evaluated qualitatively. All generated classification references are compared to the created classification labels and thus evaluated quantitatively.

Quantitative Dimension Reference Evaluation

The automatically generated dimension references are compared to the manually labelled real dimensions. Based on the comparison, the distribution of absolute errors as well as the root mean squared error (RMSE) are calculated.

Moreover, it is of interest to which extent the dimension estimation error has decreased compared to the online measurements. To calculate that, we are considering the ratio of the integrals of reference error and online measurement error over time. We define this to be the residual normalized error $G_{\varepsilon,x,o}$ regarding dimensional attribute x of object O:

$$G_{\varepsilon,x,o} = \frac{\int_{t_{o,b}}^{t_{o,e}} |\bar{x}_o - \tilde{x}_{o,t}| \, dt}{\int_{t_{o,b}}^{t_{o,e}} |\bar{x}_o - \hat{x}_o| \, dt} = \frac{\int_{t_{o,b}}^{t_{o,e}} |\bar{x}_o - \tilde{x}_{o,t}| \, dt}{(t_{o,e} - t_{o,b}) \, |\bar{x}_o - \hat{x}_o|}, \tag{5.6}$$

where $\bar{x}_o$ is the true dimensional attribute for object O existing in the time interval $[t_{o,b}, t_{o,e}]$, $\tilde{x}_{o,t}$ denotes the online measurement of dimensional attribute x at time t of object O and $\hat{x}_o$ is the corresponding automatically generated dimensional reference. An exemplary application of this metric regarding the length of two objects is visualized in Figure 5.7. Subsequently, averaging all dimensional residual errors results in the two performance indicators mean length residual normalized error (MLRE) and mean width residual normalized error (MWRE).

Qualitative Dimension Reference Evaluation

Generated dimension references for which there exists no true dimension information (e.g., pedestrians, cyclists, trucks) are checked for plausibility regarding the respective classification-dependent dimension ranges that were introduced in Figure 5.4. We define the dimension reference plausibility rate as follows:

$$p_x = \frac{|\{O \in \mathcal{O}_{rev} \mid \exists \, \bar{c}_o \land \hat{x}_o \in [x_{\bar{c}_o,l}, x_{\bar{c}_o,u}]\}|}{|\{O \in \mathcal{O}_{rev} \mid \exists \, \bar{c}_o\}|}, \tag{5.7}$$

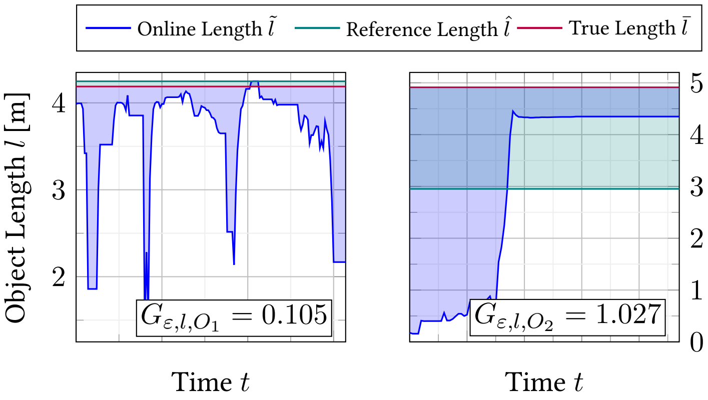


Figure 5.7.: Exemplary length residual error estimation: The generated length reference of O_1 (left) approximates the true object length accurately ($\bar{l}_1$ =4.188 m, $\hat{l}_1$ =4.248 m). The residual error just covers 10.5 % of the original error ($G_{\varepsilon,l,1} = 0.105$). The generated length reference of O_2 (right) results in a slight amplification of the error by 2.7 % ($G_{\varepsilon,l,2} = 1.027$) since the online measurement is more accurate over time than the generated reference ($\bar{l}_2$ =4.913 m, $\hat{l}_2$ =2.952 m). (©2021 IEEE)

where $\mathcal{O}_{rev}$ are all revised objects and $x_{c,l}$ and $x_{c,u}$ are the lower and upper limit of the investigated dimensional attribute x regarding object classification c.

Quantitative Classification Reference Evaluation

Correctness of generated classification references is checked for objects with corresponding classification ground truth labels. Classification correctness rate R_c is calculated as follows:

$$R_c = \frac{|\{O \in \mathcal{O}_{rev} \,|\, \exists\, \bar{c}_o \wedge \hat{c}_o = \bar{c}_o\}|}{|\{O \in \mathcal{O}_{rev} \,|\, \exists\, \bar{c}_o\}|}, \tag{5.8}$$

where the numerator defines a subset of all revised objects $\mathcal{O}_{rev}$ for which corresponding generated classification references $\hat{c}$ equal the correct classification $\bar{c}$ respectively.

Analogously to estimating the dimension accuracy improvement, we are considering to which extent the generated classification references result in a more accurate object representation. While the estimated classification of an object during a drive might change, our method decides for one classification after an object is revised. Thus, all object states of the respective object either exhibit a correct or an incorrect classification. Due to the different lifetimes of objects and changing online classifications, we are considering the total amount of improved and impaired object classifications regarding each object state to evaluate to which extent the reclassification is more accurate than the online estimation:

$$N_+ = |\{\tilde{c} \in \tilde{\mathcal{C}}_O \,|\, O \in \mathcal{O}_{rev}, \exists\, \bar{c}_o, \hat{c}_o = \bar{c} \,\wedge\, \tilde{c} \neq \bar{c}\}|, \tag{5.9}$$

$$N_- = |\{\tilde{c} \in \tilde{\mathcal{C}}_O \,|\, O \in \mathcal{O}_{rev}, \exists\, \bar{c}_o, \hat{c}_o \neq \bar{c} \,\wedge\, \tilde{c} = \bar{c}\}|, \tag{5.10}$$

where $\tilde{\mathcal{C}}_O$ is the set of all online classifications of object O.

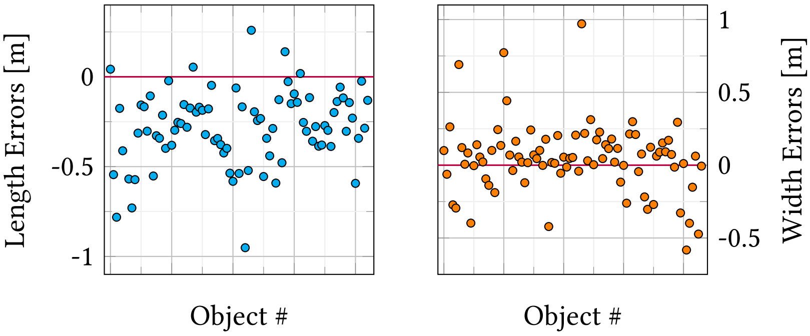


Figure 5.8.: Absolute errors of generated width and length references: Length references tend to under-approximate true length (**RMSE** = 37.51 cm). In contrast, generated width references rather over-approximate true width (**RMSE** = 24.14 cm). (©2021 IEEE)

Additionally, the average correct rate improvement considering all revised objects O that have a ground truth classification label $\bar{c}_o$ is calculated by considering all of their online classifications $\tilde{c}$:

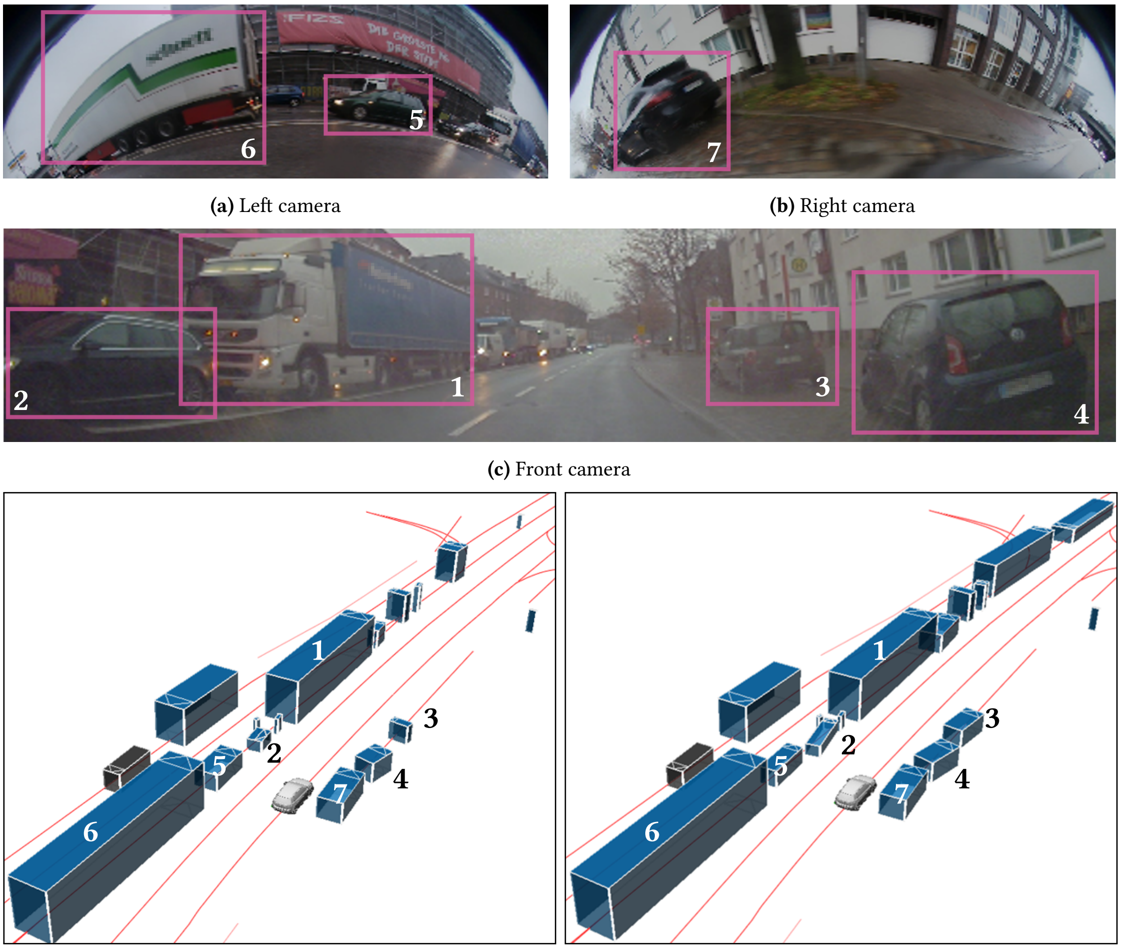
$$I_c = \frac{N_+ - N_-}{|\{\tilde{c} \in \tilde{\mathcal{C}}_O \,|\, O \in \mathcal{O}_{rev},\ \exists\, \bar{c}_o\}|}. \tag{5.11}$$

5.2.3. Results

Dimension Reference Results

Generated dimension references that are verifiable (ground truth dimensions are known for a set of 87 vehicles) achieve an accuracy with a mean error of 37.51 cm regarding length and 24.14 cm regarding width. While length references rather under-approximate true length, width references tend to be larger than true width. Since the generated dimension references can only be as accurate as the most accurate online measurements, the trend for shorter lengths might result from inaccuracies of the online perception. Scanning effects caused by considerably lower lidar scanning speed compared to motor vehicle moving speed can affect dimensional measurements. Specified widths for models documented by the manufacturer often do not include the width extension by the side mirrors. A possible reason for the over-approximation could therefore be that the online object segmentation might consider perceived side mirrors for the bounding box modeling. Absolute errors of the verifiable generated dimension references are visualized in Figure 5.8. Comparing generated references to online measurements, the original error is reduced for both width and length for all three datasets on average. Exemplarily, the error of length measurements is reduced down to 33.4 % ($\mathcal{S}_1$) and the error of width measurements is reduced down to 51.5 % ($\mathcal{S}_3$) of their respective original size. Table 5.2 shows quantitative results for each vehicle class where true lengths and widths are known.

Generated dimension references for which true dimensions are unknown, but true classifications, are evaluated regarding plausibility (cf. Table 5.3). While most of the generated length references for each classification lie within the specified ranges, the class pedestrian with a



(a) Left camera

(b) Right camera

(c) Front camera

(d) Original dimensions

(e) Reference dimensions

Figure 5.9.: Comparison of original (d) and generated reference dimensions (e). Figures (a-c) show camera images. (©2021 IEEE)

factor of 0.281 is an exception. By investigating corresponding objects, it is noticed that a lot of pedestrian objects are actually a group of pedestrians. A possible approach to cope with this issue is to define another classification *Pedestrian Group*. However, due to the dense traffic it is frequently observed that one pedestrian object comprises a varying amount of pedestrians over time. Taking care of these *Multi Object* [18] errors and subsequent segmentation is challenging based on objects alone and not considered in our approach. Another noticeable result is that the width reference plausibility rate p_w is rather low for larger objects like *Truck* and *Bus*. Possible reasons for that could either be inaccurate online dimensional measurements (short lifetimes, larger objects can potentially exhibit more occlusion) or that the specified dimensional ranges are not reasonable enough. The qualitative evaluation of dimension references is summarized in Table 5.3. A comparison of an exemplary scene with both original object dimensions and generated dimension references is visualized in Figure 5.9.

Table 5.2.: Dimension Revision: Evaluation (©2021 IEEE)

Class	**LRMSE**		**WRMSE**		**LMRE**	**WMRE**
	[m]	[%]	[m]	[%]		
Passenger Car	0.339	8.08	0.231	13.3	0.386	0.579
Van	0.363	7.18	0.178	8.9	0.359	0.461
Bus	0.842	5.03	0.611	24.0	0.869	1.100

LRMSE: Length Root Mean Squared Error; **WRMSE**: Width Root Mean Squared Error; **LMRE**: Length Mean Residual Normalized Error; **WMRE**: Width Mean Residual Normalized Error

Table 5.3.: Dimension Revision: Plausibility Evaluation (©2021 IEEE)

Class	p_l	Ref. Lengths [#]	p_w	Ref. Widths [#]
Pedestrian	0.281	57	0.404	57
Bicycles	0.722	54	1.000	53
Motorbike	1.000	3	1.000	3
Passenger Car	0.932	324	0.808	334
Van	0.915	71	0.907	75
Truck	0.964	28	0.580	31
Bus	1.000	6	0.500	6

p_l: Length Ref. Plausibility Rate; p_w: Width Ref. Plausibility Rate;

Classification Reference Results

The correct rate of the generated classification references achieves around 93 % and 88 % for the processed datasets (cf. Table 5.4). Compared to the online classifications of perceived objects, the corresponding generated classification references improve the accuracy by 20 %. More online classifications are corrected instead of falsified for all the datasets. These results are shown in Table 5.4. Additionally, since the original dataset distinguishes between *Passenger Car*, *Bicycle* and *Pedestrian*, the generated classifications offer a more refined classification (e.g., differentiating *Passenger Cars* also into other motorized vehicle types). A confusion matrix of the object classification correctness is shown in Figure 5.10.

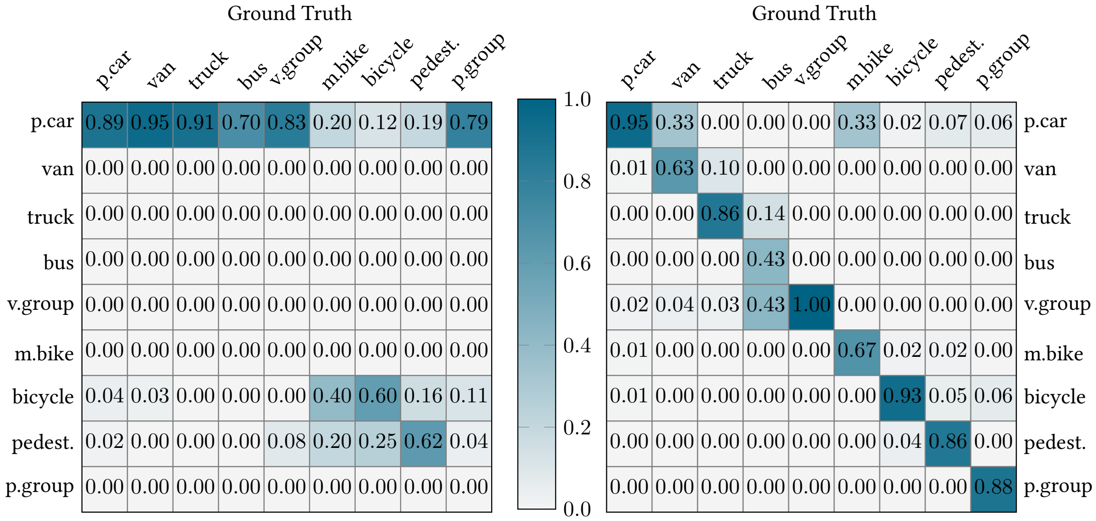
5.3. Discussion

Results. The case study presents that automatically generated dimension and classification references are close to the reality, thus showing feasibility of the approach. The references can therefore serve as an indicator for analyzing online classification and dimension estimation quality of the online perception. For instance, it might be of interest how quickly the online measurement of an object's width becomes reliable. Moreover, other attributes of certain traffic participant types can be investigated more reliably by utilizing the references. When there

Table 5.4.: Classification Revision: Evaluation (©2021 IEEE)

Set	$\boldsymbol{N_+}$	$\boldsymbol{N_-}$	$\boldsymbol{I_c}$	$\boldsymbol{R_c}$
$\mathcal{S}_1$	4862 (28.4 %)	207 (1.2 %)	0.237	0.933
$\mathcal{S}_2$	4884 (41.6 %)	46 (0.4 %)	0.231	0.937
$\mathcal{S}_3$	6881 (30.2 %)	525 (2.3 %)	0.159	0.886

$\boldsymbol{N_+}$: Corrected Object Classifications; $\boldsymbol{N_-}$: Impaired Object Classifications; $\boldsymbol{I_c}$: Mean Correct Rate Improvement; $\boldsymbol{R_c}$: Correct Rate



	p.car	van	truck	bus	v.group	m.bike	bicycle	pedest.	p.group
p.car	0.89	0.95	0.91	0.70	0.83	0.20	0.12	0.19	0.79
van	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
truck	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
bus	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
v.group	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
m.bike	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
bicycle	0.04	0.03	0.00	0.00	0.00	0.40	0.60	0.16	0.11
pedest.	0.02	0.00	0.00	0.00	0.08	0.20	0.25	0.62	0.04
p.group	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

(a) Online Classification Correctness

p.car	van	truck	bus	v.group	m.bike	bicycle	pedest.	p.group	
0.95	0.33	0.00	0.00	0.00	0.33	0.02	0.07	0.06	p.car
0.01	0.63	0.10	0.00	0.00	0.00	0.00	0.00	0.00	van
0.00	0.00	0.86	0.14	0.00	0.00	0.00	0.00	0.00	truck
0.00	0.00	0.00	0.43	0.00	0.00	0.00	0.00	0.00	bus
0.02	0.04	0.03	0.43	1.00	0.00	0.00	0.00	0.00	v.group
0.01	0.00	0.00	0.00	0.00	0.67	0.02	0.02	0.00	m.bike
0.01	0.00	0.00	0.00	0.00	0.00	0.93	0.05	0.06	bicycle
0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.86	0.00	pedest.
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.88	p.group

(b) Classification Reference Correctness

Figure 5.10.: Classification correctness comparison: Columns denote ground truth, rows describe original classifications and generated references respectively. The visible diagonal on the right depicts an accurate refined classification. Moreover, correctness rate of already considered classifications *Passenger Car* (+6 %), *Bicycle* (+33 %) and *Pedestrian* (+24 %) improves. (©2021 IEEE)

is, e.g., the interest to analyze measured velocity profiles of perceived cyclists in our dataset, querying all objects with the classification reference *Bicycle* would result in 92.6 % of all real cyclists, while considering the online classification would only end up with 59.5 % of all real cyclists (cf. Figure 5.10).

Validity. However, the process is limited to true positive accuracy of perceived objects and therefore only focuses on one part of perception. Sensor objects that comprise multiple traffic participants or change identity over time can not be easily corrected. Additionally, missed objects cannot be covered. We therefore think, that a proper automatic ground truth generation process needs to rely on raw data by, e.g., reprocessing it without real-time constraints and being able to consider both preceding and subsequent scannings. Another potential improvement is dealing with sensor-specific effects like the scanning speed of lidars which can cause distortions when measuring vehicle dimensions and thus need to be compensated [75]. Nevertheless, our proposed approach already accelerates perception evaluation by being fully automated and not being dependent on object association. We consider the approach also useful and scalable as a preceding step before automatically analyzing measurement data for the generation of test scenarios in the context of safety validation [50, 49, 48]. Identification of already experienced driving situations is mandatory for estimating residual risk regarding SOTIF [55]. Finally, we also think that parts of our proposed approach can be beneficial for improving online perception or situational understanding components by either neglecting or adapting the prospective aspects of the introduced concepts.

6. Exemplary Perception Evaluation[1]

To conclude this work, we exemplarily evaluate a prototypical *Sense* component. This *Sense* component is a lidar-based object detection module for the use-case of perceiving surrounding traffic participants in urban situations. In a first step, we exemplarily evaluate this component towards its ability to perceive relevant objects. Secondly, we investigate the component's performance in exemplary situations regarding requirements elicited in Chapter 4.

6.1. Evaluation of Relevant Object Detection

In this Section, we assess how well relevant objects are detected. To that end, generated object hypotheses of the *Sense* component are compared to manually labeled objects on the same point cloud. We firstly explain which kind of perception errors we identify in our sample dataset. Secondly, we compare perception errors regarding all surrounding objects and only regarding objects inside the relevant areas.

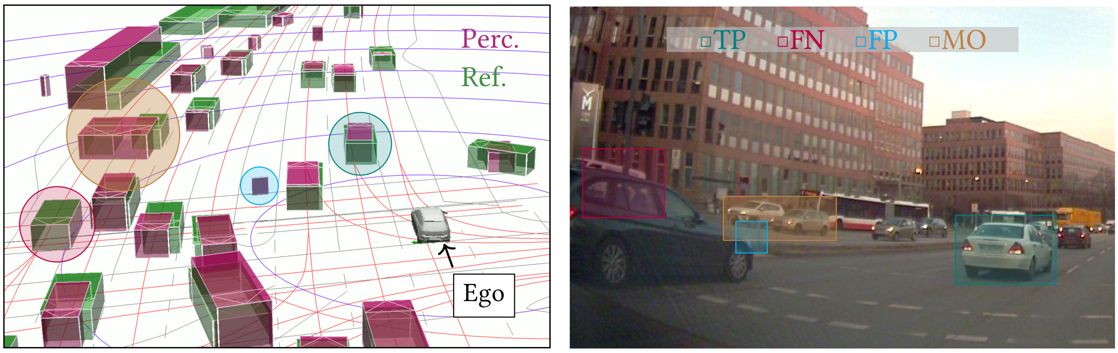
6.1.1. Identification of Perception Errors

For the identification of perception errors, manually labeled objects must firstly be matched to the object hypotheses generated by the perception component. Since point clouds of the lidar sensors served both as basis for the object detection algorithm as well as for manual labeling, the minimum euclidean distance between labeled and generated objects is used. For the matched objects, we consider five different types of perception errors, namely: *True Positive Inaccuracy* (existing object is perceived but subject to inaccuracies), *False Negative* (existing object is not perceived), *False Positive* (a non-existing object is perceived) and *Multiple Object* (multiple existing objects are perceived as one object) [18, 84] The error type *Multiple Track* does not occur in the sample dataset. Figure 6.1 shows the four perception errors.

6.1.2. Evaluation of Object Detection Performance

We consider errors introduced in Figure 6.1 for a performance assessment of the investigated perception component. We investigate both an extract of a test drive over 214.3 s conducted by an autonomous prototype in the city of Hamburg, Germany and situation (c) and (e) from Table 3.3 from Chapter 3 (11.6 s and 11.3 s long respectively). The intention of this exemplary evaluation is not to show a sufficient correctness of the investigated system under test, but rather to expose significant differences regarding number of errors over different groups of

[1] This chapter uses results from Paper IV [83] and therefore contains verbatim content previously published (©2022 IEEE).



(a) 3D-Visualization

(b) Snippet of front camera

Figure 6.1.: Exemplary perception errors in our dataset (green objects in the visualization are labeled reference objects, red objects are generated hypotheses of the perception) (©2022 IEEE)

objects (all objects vs. relevant objects). Our line of reasoning is as follows: if there are significant differences of the *Sense* component capability regarding detection of objects inside and outside the relevant areas, then this argues for a differentiated perception evaluation. In the context of safety validation, we are interested in an accurate perception performance regarding environment elements that need to be seen to operate safely in traffic. To spin this thought out even further, perception errors regarding irrelevant objects should ideally not affect system safety at all. Consequently, the object detection module of a *Sense* component should only be investigated towards its capability of detecting relevant road users. Results of our exemplary perception evaluation are depicted in Table 6.1.

Table 6.1.: Perception evaluation regarding all objects and relevant objects (©2022 IEEE)

	Whole drive		**Scenario (c)**		**Scenario (e)**	
	# Objects		# Objects		# Objects	
Error	All	$\in \mathcal{R}$	All	$\in \mathcal{R}$	All	$\in \mathcal{R}$
TP	478	132	39	12	22	4
FN	302	48	31	10	5	0
FP	1341	105	97	13	88	5
MO	18	5	3	3	0	0

Columns show all objects and objects in rel. areas ($\mathcal{R}$).
Objects in the far distance are not labeled but perceived, thus leading to a lot of supposed phantom objects (FP).

Findings of the analysis are that less than one third of correctly perceived objects are located in our specified relevant areas. Also, significantly less *False Negatives*, *False Positives* and *Multiple Object* errors occur inside relevant areas. For example, we see that five objects have not been perceived in situation *Right Turn* (Table 3.3 (e)), but none of these objects are inside the relevant areas (cf. Table 6.1). Our conclusion here is twofold:

1. A task-oriented perception evaluation incorporating relevant areas leads to a more precise characterization of the relevant detection performance.

2. A task-oriented perception evaluation focusing on individual maneuvers helps to understand in which situations the perception performs better or worse.

However, while the results of this task-oriented evaluation already expose errors regarding relevant objects, this evaluation does not yet consider whether these errors (*False Negative*, *False Positive*, *Multiple Object*) also propagate up to a failure of the overall system over time. Just counting the errors (as in Table 6.1) is obviously not sufficient. E.g., missing ten different relevant objects in just one scene (and thus for up to 0.2 s) account for the same overall result as missing one relevant object in ten consecutive scenes (and thus for up to 1.1 s). While an object just missing for a short moment of time can be compensated by a *Plan* component, coping with a missing relevant object for 1.1 s can already lead to hazardous behavior[2]. Therefore, the evaluation in the next section moves away from a scene-based assessment and focuses requirements elicited in Chapter 4, which are based on perception errors leading to a failure in the corresponding situation.

6.2. Evaluation of Perception Requirements

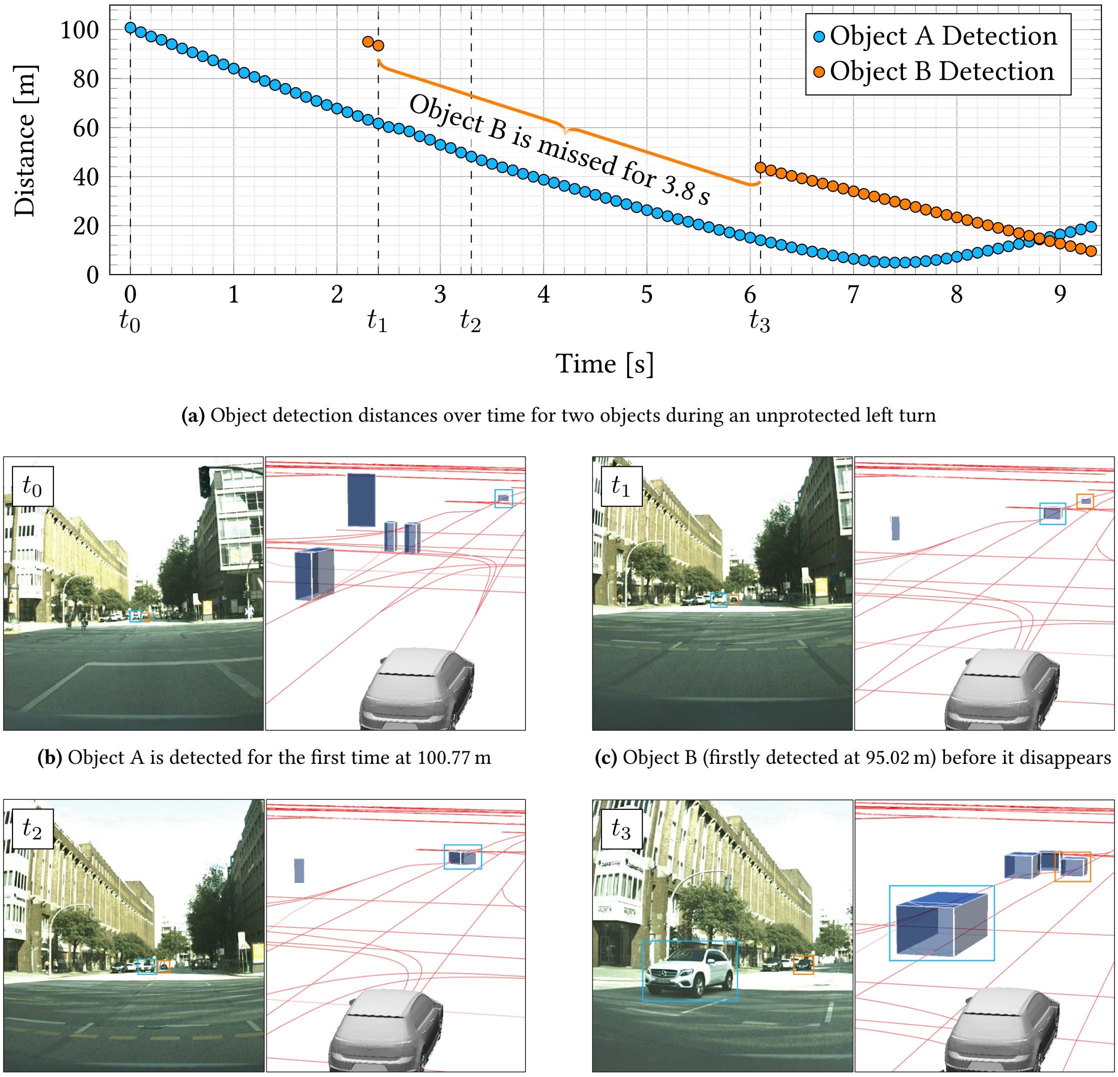
In this Section, exemplary situations are shown where specific perception requirements are met or not met. The aim of this evaluation is not to prove correctness of a given *Sense* component, but rather show how compliance or non-compliance with some requirements elicited before is determined.

6.2.1. Detection Distance & Track Stability

The first investigation deals with an unprotected left turn scenario, which is comparable to the scenario analyzed previously in Subsection 4.4.2. The ego vehicle approaches a crossing during a green traffic light, thus drives onto the junction and then needs to wait for two oncoming vehicles in order to turn left. For a safe left turn, these oncoming vehicles must be perceived with sufficient accuracy. A part of that is how early the object is detected and how stable its corresponding track is. We know based on the simulation results from Subsection 4.4.2 that an oncoming vehicle must be detected at latest at a distance of 56 m and cannot be missed for longer than 0.9 s in order for our prototypical *Plan* component to safely perform. For an automatic verification of these requirements in a big dataset, accurate ground truth data is needed in order to firstly verify the existence of a detected object before evaluating track stability. This requires a matching of ground truth objects to object hypotheses generated by the *Sense* component. Additionally, oncoming vehicles which are not detected at all (in that case the object's track downtime is equal to the object's lifetime) can be identified and the assessment of

[2] As a reminder, the strictest object misdetection duration tolerated by our prototypical *Plan* component based on the two investigated scenarios in Chapter 4 is 0.9 s (cf. Table 4.1).

non-existing (but detected) objects can be avoided. However, in our case the existence of both oncoming vehicles is proven by manually checking the corresponding camera images. Here, both objects are detected by the *Sense* component. The detections and tracks for both objects are plotted and accentuated by images in Figure 6.2.



(a) Object detection distances over time for two objects during an unprotected left turn

(b) Object A is detected for the first time at 100.77 m

(c) Object B (firstly detected at 95.02 m) before it disappears

(d) Object A is detected at 56 m

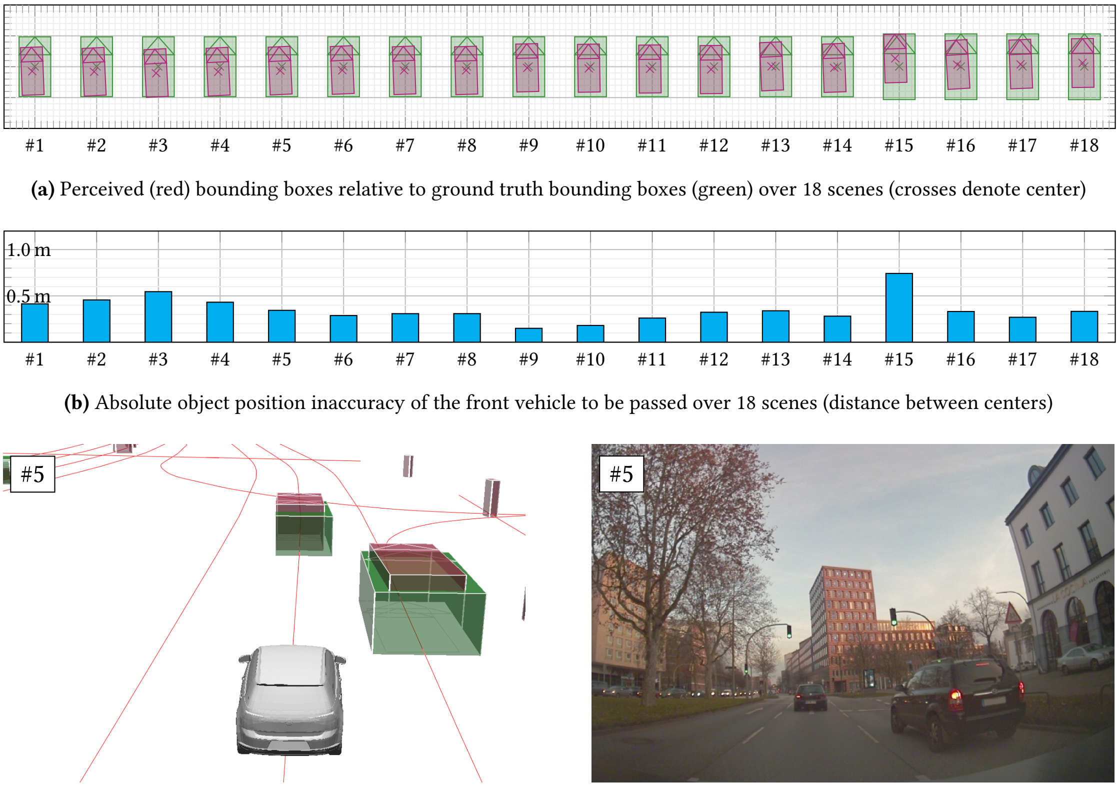
(e) Object B appears again at 43.68 m

Figure 6.2.: Tracks of two oncoming objects during an unprotected left turn: While both Object A and B are detected early enough, Object B is missed intermittently for 3.8 s.

It can be seen, that Object A is detected early enough (100.77 m $\geqslant$ 56 m) with no downtime of the corresponding track (0 s $\leqslant$ 0.9 s). Object B is also detected early enough (95.02 m $\geqslant$ 56 m). However, after two subsequent detections, the *Sense* component loses track of Object B and it is only detected again after a downtime of around 3.8 s. Thus, the track stability requirement regarding Object B is not met in this situation (3.8 s $\nleqslant$ 0.9 s).

6.2.2. Object Position Accuracy

The second investigation focuses the passing of a neighboring vehicle in front, which is part of the second scenario analyzed in Section 4.4.3. The ego vehicle is required to pass the neighboring vehicle while maintaining a safe lateral distance. To that end, the neighboring vehicle's position and dimensions need to be detected accurately enough. One corresponding, quantified requirement addresses the inaccuracy of the perceived lateral and longitudinal position of the neighboring vehicle. Based on the simulation results from Section 4.4.3, we know that the absolute inaccuracy in this situation regarding the vehicle's position shall not be greater than 1 m in any direction. Here, ground truth data is required for the evaluation. Since even inaccuracies of a few centimeters can make a difference regarding passing or failing such a perception requirement, the degree of quality which the ground truth data needs to exhibit shall not be underestimated. For the exemplary evaluation of our requirement, we look into 18 subsequent scenes, where a neighboring vehicle is passed. Figure 6.3 shows the corresponding track (perceived and ground truth bounding box), the absolute position error (distance between ground truth bounding box center and perceived bounding box center) and both visualization and camera image for one of the scenes. Our accuracy requirement is met in all scenes.



(a) Perceived (red) bounding boxes relative to ground truth bounding boxes (green) over 18 scenes (crosses denote center)

(b) Absolute object position inaccuracy of the front vehicle to be passed over 18 scenes (distance between centers)

(c) Exemplary visualization of perceived (red) and ground truth (green) bounding box during the *Passing* scenario

(d) Exemplary front camera image during the *Passing* scenario (neighboring vehicle is on the right)

Figure 6.3.: Perceived and ground truth bounding boxes corresponding to a front vehicle which is to be passed by the ego vehicle. The absolute object position inaccuracy is sufficiently small all the time (⩽1.0 m)

It can be seen that the perceived bounding boxes match their corresponding ground truth bounding boxes closely over the 18 scenes. The distance between perceived bounding box centers and corresponding ground truth bounding box centers is always smaller than 1 m. However, due to varying dimensions (length and width) and slightly different headings of both perceived and ground truth bounding boxes, we conclude that the distance between the bounding box centers should not be investigated exclusively when addressing bounding box accuracy. While the first few scenes represent the ego vehicle coming closer (longitudinally) to the neighboring vehicle, the subsequent scenes cover the *Passing* maneuver and finally the ego vehicle pulling away. These different situations and the corresponding perspective the ego vehicle has on the neighboring vehicle also have an impact on the accuracy of the perceived bounding box. Rear edges of the bounding boxes match mainly in the first scenes. This changes over the course of the scenes, and it can be observed in Figure 6.3b that the front edges gradually get closer. At all times, the perceived bounding boxes match with the left side of the ground truth bounding boxes. This is crucial for maintaining a lateral distance while passing the neighboring vehicle. An offset between the right edge of the perceived and the right edge of the ground truth bounding box should ideally not have an effect on the performance of the *Plan* component. Therefore, it is expedient to rather focus the relevant sides of the bounding boxes than the bounding box centers (in our case firstly rear/left side, then left side and finally front/left side). Note how scene #15 features the greatest position inaccuracy, but also the most accurate matching of the front left corner. Assessing the maximum distances between perceived and ground truth bounding boxes at their situation-dependent, relevant sides each scene seems also like a more suitable evaluation approach in our case. In Figure 6.3b it can be seen, that the left side of the perceived bounding box always matches nicely and firstly the rear side and lastly the front side are much closer than 1 m to the respective sides of the ground truth bounding boxes. Thus, our position accuracy requirement is also met following the more goal-oriented evaluation approach.

Part V.

Conclusion and Outlook

7. Conclusion

In this chapter, we present a brief summary of our results. To that end, we point out our main findings and link them to the research questions raised in Chapter 1. Afterwards, we follow with a reflection on our contributions. This chapter is closed by giving a brief overview about how the concepts and results of our other contributions (cf. *List of Scientific Contributions*), that were not directly created in the scope of this thesis, can be linked to this thesis.

7.1. Summary

Research Questions

1. How can dependability threats to automated driving systems pertaining to perception components be characterized?
2. What types of perception errors do exist and how can they be classified?
3. Which perception errors are of relevance for a prediction & planning module?
4. How can requirements for the perception component be elicited?
5. How can the costly generation of reference data, which is needed for the evaluation of perception modules, be approached?

The first part of this thesis investigates the specification of perception performance. In the scope of this part, we distinguish the terms fault, error and failure (cf. Section 2.1) and show how the qualitative requirement of detecting relevant road users with a certain quality in a fixed time interval can be further refined. This includes the systematic derivation of possible perceptual errors based on well-defined interfaces to the *Plan* component to show different dimensions of *certain quality* (cf. Section 2.2). To that end, we refer to commonly used conceptualizations of the environment and show corresponding false positive and false negative errors and - if existing - possible inaccuracies of true positive inferences. Furthermore, in this part, the term *relevant road users* is decomposed into six different road user groups (cf. Chapter 3). We define corresponding relevant areas and show how these can be constructed given the geometry of the traffic infrastructure and assuming compliant worst-case behavior of surrounding traffic participants. These results contribute directly to Research Question 1 and 2, which address the impact of the *Sense* component on the subsequent processing and the classification of perceptual errors. Additionally, Research Question 3 is partially answered since the results enable differentiation of error relevance by considering the proposed relevant areas.

However, we do not yet define quantitative requirements regarding perceptual errors. This is investigated further in the second part of this thesis, where we elicit perception requirements for a prototypical *Plan* component (cf. Chapter 4). Utilizing our error injection approach, we define six different requirements based on simulation results of two different scenarios. These requirements address the distance at which an object needs to be detected latest, its track stability and the accuracy of its estimated position during an unprotected left turn scenario and an overtaking scenario. It can be observed, that requirements of the same type are differently strict in these two scenarios. We see our approach as suitable to assess error compensation capabilities and then derive a corresponding set of perception requirements for a given black-box *Plan* component (Research Question 4). The third part of this thesis deals with the evaluation of object detection modules as part of a *Sense* component. Research Question 5 is touched on in Chapter 5, where we show an approach to generate reference data for object dimensions and classifications based on already perceived objects. However, this approach focuses only on certain aspects of object detection and can therefore not be seen as a complete solution. The third part concludes with exemplary evaluations of an object detection module. While this chapter does not directly correspond to one of the five research questions, it complements the concepts and ideas from Part II and III. The chapter shows an evaluation featuring the relevant areas from Chapter 3 and demonstrates how exemplary requirements from Chapter 4 can be verified.

7.2. Reflection

We reflect on each research question individually before providing a more general reflection on the overall thesis and our contributions.

7.2.1. Reflection on Research Questions

Research Question 1

How can dependability threats to automated driving systems pertaining to perception components be characterized?

In Chapter 2, a taxonomy for the characterization of dependability threats to perception components is established. To that end, we applied the concept of faults, errors and failures by Avižienis et al. [9] to the environmental perception component. This introduced causalities between the threats along the processing chain of environmental perception. Faults are causes to errors and errors can lead to the occurrence of failures. Firstly, we map these threats to the processing chain of environmental perception, which consists of environment scanning, feature extraction and scene modeling. We further illustrate the causalities between these threats in the case study (cf. Section 2.3), where a low-hanging sun (external fault) leads to an overexposed camera image (error) due to the camera configuration settings (internal fault). The lane marking detection algorithm is not able to detect all lane markings on the overexposed camera image (internal fault), which results in an incomplete set of detected lane markings (error) and the lane network being insufficiently modeled (perception failure). Subsequently, the lane keeping assistance system is not able to keep the vehicle inside its lane anymore (planning failure and actuator failure). While our hypothesized lane keeping assistance system in the case study is able to perform a reconfiguration, adjust the system to the low-hanging sun and can thus cope with the perception failure, one of the key questions that is discussed later in this thesis becomes apparent: when does a perception error become a failure, that leads to an overall system failure? The need for more concise terms regarding dependability threats of automated driving systems and its components is also backed by the ISO 21448 [55]. The ISO 21448 invents the terms *triggering condition* and *functional insufficiency*, which can be interpreted as external faults (related to the environment) and internal faults (related to the system). Distinguishing into faults, errors and failures also structures the testing process. External faults are part of the test input (like scenarios or triggering conditions), which have the intention to reveal internal faults (functional insufficiencies) of the component under test. Based on defined pass/fail-criteria regarding either the component's output or the system's output, it can be determined whether the resulting error of the component leads then to a failure of the component and thus a failure of the overall system. This results in a concrete challenge regarding safety validation of components (here *Sense* component): it needs to be determined which perception errors lead to a system failure and which faults trigger these errors in the first place.

We answer this research question by concluding that our proposed taxonomy seems to be meaningful from both an analysis and testing perspective, is also partly supported by the ISO 21448 concepts and supports answering of the remaining research questions.

Research Question 2

What types of perception errors do exist and how can they be classified?

Following the differentiation into fault, error and failure and the already mentioned challenge of determining both the effect and occurrence of perception errors, an approach to derive possible perception errors for an automated driving system is needed. As errors are part of the total state of the system (cf. Avižienis [9]) and we assume that errors can occur in every step and at any time when processing environmental data, we again considered the whole processing chain of environmental perception. To that end, we functionally decomposed the task of environmental perception into environment scanning, feature extraction and scene modeling with their corresponding well-defined interfaces raw scan, features and scene. Subsequently, for the classification of perceptual error types, we briefly discussed errors on raw scan level for cameras, lidar and radar sensors in Subsection 2.2.1 before investigating errors on feature level in Subsection 2.2.2. For the definition of perceptual errors on feature level, possible errors are derived by splitting up the environment into its subsequent parts and considering in which aspects features that need to be extracted can be flawed. We find that features of the environment that are necessary for scene modeling include, but are not limited to, movable objects, traffic signs and lights, curbs and lane markings as well as any obstacles or vertical elevation on the road that leads to parts of the road being impassable. Following, we further derive perceptual errors by constructing error trees for these features (cf. Figures 2.4 to 2.8). These error trees all follow the same pattern: an existing feature in the environment can either be detected (true positive) or not detected (false negative / type II error) and a non-existing feature can either be detected (false positive / type I error) or not detected (true negative). Additionally, we define that a true positive of any feature with attributes can be subject to inaccuracies, which we also understand as an error. For instance, an existing oncoming vehicle can be correctly detected by the environment perception, but its estimated velocity value might differ from its real velocity. How features of the environment are actually conceptualized by the environment perception also defines the possible dimensions of true positive inaccuracy errors. In general, the variety of possibly occurring perception errors that might also affect the subsequent *Plan* component is dependent on which information this very *Plan* component actually relies on. Based on that observation, it becomes apparent that the complexity of the interface between the *Sense* and *Plan* component also shapes the effort of validating both corresponding components. Thus, we conclude, that the interface between the *Sense* and *Plan* component shall be as simple as possible and as detailed as necessary. By following that principle, the variety of possibly occurring perception errors could be reduced, which directly translates to verification and validation efforts.

We answer this research question with our introduced approach to derive perception errors based on decomposing the processing chain of the *Sense* component under analysis, considering the different elements of urban environments and by constructing corresponding error trees.

Research Question 3

Which perception errors are of relevance for a prediction & planning module?

As already pointed out by Avižienis et al. [9], many errors do not affect the system's external state. In our context, this means that not every perception error leads to a failure of the overall automated driving system. This statement seems plausible, since driving in urban traffic also does not require the human driver to always perceive the whole environment with perfect accuracy at all times. In this thesis, we approach this research question with the following hypotheses:

1. The elements in the environment, which need to be detected for safely participating in traffic, mainly depend on the actual scenario.
2. The acceptable severity of corresponding perception errors of these elements mainly depends on the robustness of the subsequent *Plan* component, which is responsible for situational understanding and deciding for a trajectory.

We investigated the first hypothesis in the scope of this research question, while findings and experiments regarding the second hypothesis correspond to Research Question 4. To cope with the variety and complexity of urban driving scenarios, we provided a modular concept for the automated generation of relevant areas in urban driving scenarios in Chapter 3. The purpose of this is to systematically differentiate between relevant and less relevant elements in the environment. Movable objects, traffic lights, lane markings and other elements within these relevant areas are understood as relevant. Thus, they need to be detected and reacted to by an automated driving system. This also enables a task-oriented perception evaluation by specifically assessing the detection accuracy of relevant elements instead of all elements within the environment. Our concept comprises the decomposition of the urban traffic environment into six basic areas which become relevant while performing certain driving maneuvers. We show how these areas can be constructed by taking into consideration infrastructure information from a map and worst-case assumptions about surrounding traffic participant behavior. In each situation, these areas are merged and result in a polygon surrounding the ego vehicle where elements in these areas can be considered relevant. To show the applicability of our concept, we demonstrated generated relevant areas for five different urban traffic situations in the city of Hamburg, Germany.

As already discussed in Section 3.2.2, our concept relies on a modular design and can thus be easily extended by adding more basic areas or modifying the construction of existing ones. While preliminary results already look promising and are evaluated qualitatively (cf. Table 3.3), a proper validation is required to show correctness and completeness of defined relevant areas in the future. This could, e.g., be approached by closed-loop testing of a *Plan* component in a simulation framework, where elements outside the relevant areas are filtered out and not considered as input for the system under test. A resulting safe behavior of the tested *Plan* component (e.g., no collisions, safe distances) could then be argued as an evidence for validity of the relevant areas.

While we are not able to prove our first hypothesis in response to this research question, we provide evidences that a scenario-based differentiation into relevant and less relevant elements seems feasible and can be implemented. Moreover, different evaluation results when either considering relevant traffic participants or all traffic participants emphasize that there is a need for a task-oriented evaluation of *Sense* components. We conclude, that our approach regarding perception evaluation constitutes an improvement and very needed addition to state-of-the-art metrics like receiver operating characteristic curves or false negative rates (cf. Section 1.3.1). As stated above, our second hypothesis is examined in the scope of Research Question 4.

Research Question 4

How can requirements for the perception component be elicited?

Following our second hypothesis in reply to Research Question 3, the investigation of a *Plan* component's robustness is required for further answers. From another perspective, acceptable and non-acceptable perception errors decide which exact accuracy requirements a *Sense* component needs to meet. To that end, we have presented a structured testing process for the elicitation of requirements based on closed-loop testing of a given *Plan* component in Chapter 4. This comprises the concept of subsequently executing test cases while degrading the perception accuracy based on consecutive evaluation. Here, we focused on the accuracy regarding the detection of surrounding traffic participants. For that, we also introduced a non-exhaustive set of perceptual hazards and error models for the simulation. Our results show that it is possible to elicit measurable and quantifiable initial requirements for a *Sense* component that shall be used in conjunction with the *Plan* component under test. We elicited six requirements based on two different scenarios which address the position inaccuracy of a detected object, the flickering of a detected object as well as at which distance an object needs to detected (cf. Table 4.1). To the best of our knowledge, we are among the first ones to show that such areas of acceptable perceptual inaccuracies exist and can be quantified, even for prototypical systems where its components were not implemented with having well-defined quantifiable performance goals in mind.

Future work should address modeling of further perceptual errors, also in a combined manner and including random errors. Moreover, various different scenarios should be utilized and investigated to further specify the assumption of *Plan* on *Sense*. Our approach for the elicitation of accuracy requirements is solely based on structured testing and thus corresponds to a black-box perspective. It is therefore essential, that future research also deals with methods for actually proving safety of the *Plan* component under accuracy assumptions — or at least lifting (black-box) simulation-based testing to a grey-box approach or a white-box approach, increasing the validity of obtained safety results.

We answer this research question with our introduced structured testing concept and the corresponding case study, where we elicited accuracy requirements for a perception component. Validity, limitations and required follow-up research has been discussed by us in detail in both Section 4.4.4 and reflection of this research question. In our opinion, accuracy requirements such as ours are a first step towards assume/guarantee-style decomposition of system validation at the interface between *Sense* and *Plan*.

Research Question 5

How can the costly generation of reference data, which is needed for the evaluation of perception modules, be approached?

Specifying accuracy requirements for a perception component is only one side of the coin, making sure these are actually met is the other one. When evaluating the performance of a *Sense* component regarding object detection, a set of reference objects (ground truth) is needed. As already pointed out in the introduction, generation of ground truth is cumbersome since it is often the result of manual human labelling. The quest for automated generation of reference data can be understood as developing high fidelity perception algorithms, which — in comparison to the ones integrated in automated driving systems — do not need to adhere to real-time constraints and can go back and forth in time. Since the development of a so-called offline perception module is an enormous engineering challenge, we focused on one aspect in the scope of this thesis. We asked ourselves whether object states generated by a *Sense* component of an automated driving system can be revised in order to approximate a ground truth in some aspects. Thus, in Chapter 5, a methodology for the automated generation of object dimension and classification references based on object states coming from an online perception module in conjunction with an HD map is established. This comprises the concept of ranking measurements regarding reliability as well as looking for features indicating the true classification of perceived objects. For that purpose, we proposed a concept on how to calculate confidence scores for width and length measurements. Additionally, we introduced a classification tree that distinguishes into nine different object classes. To show the applicability of the proposed methodology we generated references for objects perceived in the traffic of Hamburg, Germany and evaluated these with a set of metrics that also includes a comparison to the actual ground truth. To that end, a set of vehicle dimensions was manually labeled by identification of models and consideration of manufacturer specifications. We are able to show, that generated references for dimensions and classification come close to reality (cf. Subsection 5.2.2) and that our approach supports aspects of perception evaluation (cf. Section 5.3).

However, our approach has its limitations and only covers two aspects of true positive inaccuracies: object dimensions and classifications. As already pointed out in Section 5.3, we believe that a proper process for automated generation of reference data needs to consider the raw sensor data as input (e.g., camera images, point clouds). While research towards offline 3D object detection is already ongoing, development of a proper process for automated generation of reference data requires more time and effort and is beyond the scope of this thesis.

We can thus answer this research question only with a hypothesis: offline processing of sensor data and online perception data shows the potential to make human labeling work obsolete in the long term. Along with the continuous advancements of machine learning, we believe that it is only a matter of time until the problem of generating high fidelity reference data is solved. In the scope of this thesis, we are however not able to prove that.

7.2.2. General Reflection

In the introductory part of this thesis (cf. Chapter 1), we are referring to the design criteria problem faced by avionics engineers at the beginning of the 20th century. Throughout the 1920s and 1930s, they pointed the way with their initial specifications of flying-quality and by that started to come from qualitative, ill-defined specifications to quantitative, well-defined design criteria [118]. This thesis is also meant to point the way towards well-defined and verifiable requirements - but in our case for *Sense* components, so that a *Plan* component is able to make safe decisions. Our contributions provide a novel perspective on the interface between *Sense* and *Plan* components and thus on a decomposed safety validation of automated driving systems. By investigating both a *Sense* and a *Plan* component, that were part of the same prototypical automated driving stack, we are able to show that scenario-dependent thresholds regarding perceptual inaccuracy exist and that corresponding requirements for a *Sense* component can be both specified and evaluated. The systematization of relevant road users helps to further assess relevance of perception errors. Both the quantification of acceptable inaccuracy and the differentiation of relevant objects exhibit the potential to have a lasting effect on how object detection modules will be benchmarked and evaluated in the future. We show that static attributes like dimensions and classifications of objects can be generated automatically with a certain quality by solely post-processing the result of an online object detection module. While this approach does not yet cover all aspects of objects, the potential of offline object detection for the purpose of generating reference data becomes evident.

While our analyses and experiments show the potential of our proposed concepts and our overall approach, there are several advancements to be made to both complete and complement our approach. These further advancements and investigations are crucial before a statement can be made regarding the feasibility of a decomposed safety validation approach in an industrial context. One necessary line of research is whether white-box *Plan* components can be either analyzed towards their perceptual error compensation capabilities or whether they need to be directly constructed by considering well-defined perceptual inaccuracies. Additionally, robustness of black-box *Plan* components should also be investigated further by testing more scenarios and injecting more perception errors like, e.g., misclassifications or phantom objects (*False Positives*). To determine which perceived surrounding road users need to be assessed regarding their accuracy, further systematization of relevant road users and a corresponding validation of this systematization is required. Only then, traffic participants considered as irrelevant can be actually discarded during the evaluation of an object detection module. Furthermore, our exemplary evaluation of an object detection module towards one elicited position accuracy requirement (cf. Subsection 6.2.2) shows the need for reference data of high accuracy. Thus, offline object detection methods need to be further developed and investigated towards their applicability of generating reference data. Evaluation of *Sense* components on a large scale will require as much automation as possible in regard to data processing. Last but not least, while Part II of this thesis deals with perceptual inaccuracies regarding various use cases of a *Sense* component (object detection, lane marking detection, etc.), Part III and IV only investigate elicitation and evaluation of requirements corresponding to object detection. Translation of the underlying concepts and methods of Part III and IV to use cases like lane marking detection or landmark detection for localization should also be a focus of future work.

7.3. Relation to Other Contributions

We conclude this chapter by briefly relating our other contributions (cf. *List of Scientific Contributions*), that have not been created directly in the scope of this thesis, to the concepts and methods of this thesis.

The maneuver model established by Hartjen et al. [49, 48] serves as a basis for our systematization of relevant road users (cf. Chapter 3). Only by decomposing the driving task into well-defined maneuvers, the construction of situation-dependent relevant areas becomes possible. Adjustments and extensions to the maneuver model shall also be reflected in the systematization of relevant road users (e.g., when adding maneuvers related to roundabouts).

The search for critical parameter combinations within logical scenarios as shown by Bussler et al. [20] makes use of evolutionary algorithms. Criticality is determined by the reaction of a *Plan* component under test towards the surrounding traffic participants. Evolutionary algorithms could also be used to not only vary parameters of a scenario but also to adjust the parameters of injected perception errors (cf. Chapter 4). This could be of particular interest for injection of perception errors where the effect of specific parameters on error severity is not directly known (like timing or random components of the error).

Zhu et al. [128] analyze the concept of triggering conditions, which is introduced in the ISO 21448 [55]. They define triggering conditions as external conditions which trigger functional insufficiencies and further result in hazardous behavior. Triggering conditions can affect *Sense*, *Plan* and *Act* components of automated driving systems. Triggering conditions related to *Sense* components can be seen as external faults (referring to our taxonomy in Chapter 2). Thus, these should be considered as test input when evaluating *Sense* components towards specified requirements.

Zhu et al. [129] develop and implement an automatic process for the reconstruction of scenarios based on perceived object states. Their work integrates both our automated revision for object dimensions and classification (cf. Chapter 5) and our concept for determining relevant areas (cf. Chapter 3). Both processes are further built upon and applied for their use case of reconstructing scenarios from test drive data. While the automated revision process is improved towards partial filtering of *false positive* objects and partial mending of instable object tracks, the relevant areas are extended by a rearward area behind the ego vehicle and criteria are defined for an even stricter filtering of objects inside the relevant areas (e.g., object braking distances shall intersect with a braking distance along the ego path).

8. Outlook

We quickly talk through a few observations we have made along our line of research and highlight some implications of our results. All of these points should be the focus of future research.

Complexity of Sense-Plan-Interface. The different types of perceptual errors (cf. Subsection 2.2.2) the *Plan* component can be exposed to, is dependent on how the environment is conceptualized by the *Sense* component. For instance, the more different classifications an object can be associated with by the *Sense* component, the more possible misclassification errors and their impact need to be taken into account when assessing robustness of the *Plan* component. Any continuous state variable (e.g., position, velocity) can potentially have an enormous number of different values - depending on how it is stored[1]. Subsequently, the number of possible different values that the corresponding measurement inaccuracy can have is huge. The impact of all of these different inaccuracies on the *Plan* component would theoretically need to be investigated to ensure complete test coverage. The question arises whether the environment conceptualization (and thus the interface between *Sense* and *Plan*) can both be kept simple and still be detailed enough for a *Plan* component to make safe and comfortable decisions. Only considering the different types of object classifications that are essential because of legal regulations (for instance, there exist individual safety distances for cyclists in Germany) and storing state variables with less precision (e.g., positions on a grid and kinematics discretely), while still being able to predict and interpret surrounding traffic accurately might be a desirable design criterion to research into.

Different Scenario Sets for Sense & Plan Components. As already mentioned in the introduction, the popularity surrounding scenario-based safety verification and validation has increased significantly in the last few years. To that end, identification of scenario catalogs has become an active field of research. Following the idea of this thesis to verify *Sense* and *Plan* components individually before validating the complete integrated system, we see a need for different test cases. Test cases for the *Sense* component refer to the real world and test for a certain required accuracy. E.g., different colors of surrounding vehicles could result in different performance regarding object detection accuracy. In contrast, test cases for a *Plan* component build on an environmental conceptualization which needs to be generated by a *Sense* component. When detected objects do not have an attribute regarding color, then the *Plan* component can also not treat these differently when deciding for a trajectory. In other words, two different scenarios in the real world could be identically conceptualized by a *Sense* component. Thus, a differentiation between scenario sets for the purpose of testing a *Sense* component and for the purpose of testing a *Plan* component should be made. Research regarding the identification and systematization of both types of scenario sets needs to be the focus of further research.

[1] As an example, a single-precision floating point (32 bit) can store $2^{32} = 4294967296$ different values.

Refined Pass/Fail Criteria for Plan Components. The experiments regarding testing of *Plan* components in this thesis consider the absence or occurrence of a collision as exclusive pass/fail criterion. But moving through traffic is challenging and only avoiding collisions is not sufficient for a safe and compliant driving behavior. Maintaining longitudinal and lateral safety distances, reducing velocity when passing a stopping bus or simply halting at a stop line are further examples for behavioral rules which shall also be considered when evaluating test cases. Currently, there does not yet exist an explicit set of behavioral rules for each situation. At the moment, most of the corresponding laws are formulated on a qualitative level, like, e.g., to show constant care and mutual respect (cf. StVO §1 [35]). However, engineering a *Plan* component for an autonomous vehicle requires concrete design goals and thus explicit behavioral boundaries, which are quantitative and thus verifiable. Structuring and refining existing behavioral boundaries is going to be a major topic for both the specification and evaluation of autonomous vehicles. This does not only require further research, but also mutual efforts by politics (e.g., governments and city authorities) and distributors of autonomous systems (e.g., automotive OEMs or mobility providers).

Statistical Verification of Sense Components. The *Plan* component usually consists of several software modules and the specification of its interface can be diverse. As motivated at the beginning of this chapter, the complexity of this interface has a direct impact on the amount of possible different input signals. In theory, this amount is exhaustive. In practice, the less complex this interface is and thus the less choice it provides, the more feasible it will be to consider every possible different sequence of input signals (cf. model checking). Contrary to that, the input of the *Sense* component is the complex, real world. Here, possible combinations of different road actors, infrastructure and environmental influences are endless. Showing that a *Sense* component consisting of different sensor modalities and processing algorithms can provide the required performance in the required situations with a given level of statistical confidence is a major challenge. To that end, the underlying physical principles of different sensors and their interaction with the environment as well as how raw sensor data is processed by perception algorithms needs to be extensively investigated and completely understood. Finally, the verification of *Sense* components also requires accurate reference data. Improvement of existing approaches and development of novel ideas surrounding the automated generation of reference data should also be a major focus of future research activities.

Bibliography

[1] 2018. *SAE J3016: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles.*

[2] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. 2018. Efficient Interactive Annotation of Segmentation Datasets with Polygon-RNN++. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Salt Lake City, UT, USA, 859–868. https://doi.org/10.1109/CVPR.2018.00096

[3] ADAC (Wolfgang Rudschies). 2023. Autonomes Fahren Level 3: Freihändig durch den Stau (Mercedes-Benz Drive Pilot Test Report). https://www.adac.de/rund-ums-fahrzeug/ausstattung-technik-zubehoer/autonomes-fahren/technik-vernetzung/autonomes-fahren-staupilot-s-klasse/ (last accessed on 19.10.2023).

[4] American Association of State Highway and Transportation Officials. 2012. *Guide for the Development of Bicycle Facilities.*

[5] Christian Amersbach. 2020. *Functional decomposition approach-reducing the safety validation effort for highly automated driving.* Ph. D. Dissertation. Technische Universität Darmstadt.

[6] Christian Amersbach and Hermann Winner. 2017. Functional Decomposition: An Approach to Reduce the Approval Effort for Highly Automated Driving. In *8. Tagung Fahrerassistenz*. Lehrstuhl für Fahrzeugtechnik mit TÜV SÜD Akademie, Munich, Germany.

[7] Eduardo Arnold, Omar Y. Al-Jarrah, Mehrdad Dianati, Saber Fallah, David Oxtoby, and Alex Mouzakitis. 2019. A Survey on 3D Object Detection Methods for Autonomous Driving Applications. *IEEE Transactions on Intelligent Transportation Systems* 20, 10 (2019), 3782–3795. https://doi.org/10.1109/TITS.2019.2892405

[8] Automotive News (Hans Greimel). 2023. Honda's Level 3 system for automated driving has limits (Honda SENSING Elite Test Report). https://www.autonews.com/technology/honda-legend-hybrid-ex-sedans-level-3-automated-driving-system-has-limits (last accessed on 19.10.2023).

[9] Algirdas Avižienis, Jean-Claude Laprie, Brian Randell, and Carl Landwehr. 2004. Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Transactions on Dependable and Secure Computing* 1, 1 (2004), 11–33. https://doi.org/10.1109/TDSC.2004.2

[10] Sagar Behere and Martin Törngren. 2015. A Functional Architecture for Autonomous Driving. In *First International Workshop on Automotive Software Architecture (WASA '15)*. ACM, Montréal, QC, Canada, 3–10. `https://doi.org/10.1145/2752489.2752491`

[11] Keni Bernardin and Rainer Stiefelhagen. 2008. Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. *EURASIP Journal on Image and Video Processing* 2008 (2008), 1–10. `https://doi.org/10.1155/2008/246309`

[12] Brandon Bohrer, Yong Kiam Tan, Stefan Mitsch, Andrew Sogokon, and Andre Platzer. 2019. A Formal Safety Net for Waypoint-Following in Ground Robots. *IEEE Robotics and Automation Letters* 4, 3 (2019), 2910–2917. `https://doi.org/10.1109/lra.2019.2923099`

[13] P.S. Bokare and A.K. Maurya. 2017. Acceleration-Deceleration Behaviour of Various Vehicle Types. *Transportation Research Procedia* 25 (2017), 4733–4749. `https://doi.org/10.1016/j.trpro.2017.05.486`

[14] Jan-Aike Bolte, Andreas Bar, Daniel Lipinski, and Tim Fingscheidt. 2019. Towards Corner Case Detection for Autonomous Driving. In *2019 IEEE Intelligent Vehicles Symposium (IV)*. Paris, France, 438–445. `https://doi.org/10.1109/IVS.2019.8813817`

[15] Amol Borkar, Monson Hayes, and Mark T. Smith. 2012. A Novel Lane Detection System With Efficient Ground Truth Generation. *IEEE Transactions on Intelligent Transportation Systems* 13, 1 (2012), 365–374. `https://doi.org/10.1109/tits.2011.2173196`

[16] John R. Boyd. 1996. The essence of winning and losing. *Unpublished lecture notes* (1996).

[17] Marius Bozga and Joseph Sifakis. 2022. Correct by Design Coordination of Autonomous Driving Systems. In *Leveraging Applications of Formal Methods, Verification and Validation. Adaptation and Learning*. Springer Nature Switzerland, 13–29. `https://doi.org/10.1007/978-3-031-19759-8_2`

[18] Mohamed Brahmi, K-H Siedersberger, Andreas Siegel, and Markus Maurer. 2013. Reference Systems for Environmental Perception: Requirements, Validation and Metric-based Evaluation. In *6. Tagung Fahrerassistenz*. Lehrstuhl für Fahrzeugtechnik mit TÜV SÜD Akademie, Munich, Germany. `https://mediatum.ub.tum.de/doc/1187193/file.pdf`

[19] Guillaume Brat, David Bushnell, Misty Davies, Dimitra Giannakopoulou, Falk Howar, and Temesghen Kahsai. 2015. Verifying the Safety of a Flight-Critical System. In *FM 2015: Formal Methods*. Springer International Publishing, 308–324. `https://doi.org/10.1007/978-3-319-19249-9_20`

[20] Andreas Bussler, Lukas Hartjen, Robin Philipp, and Fabian Schuldt. 2020. Application of Evolutionary Algorithms and Criticality Metrics for the Verification and Validation of Automated Driving Systems at Urban Intersections. In *2020 IEEE Intelligent Vehicles Symposium (IV)*. Las Vegas, NV, USA, 128–135. `https://doi.org/10.1109/IV47402.2020.9304662`

[21] Lluis Castrejon, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. 2017. Annotating Object Instances With a Polygon-RNN. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI, USA, 4485–4493. `https://doi.org/10.1109/CVPR.2017.477`

[22] Marsha Chechik, Rick Salay, Torin Viger, Sahar Kokaly, and Mona Rahimi. 2019. Software Assurance in an Uncertain World. In *Fundamental Approaches to Software Engineering*. Springer International Publishing, 3–21. `https://doi.org/10.1007/978-3-030-16722-6_1`

[23] Jiayun Chu, Tingdi Zhao, Jian Jiao, Yuan Yuan, and Yongfeng Jing. 2023. SOTIF-Oriented Perception Evaluation Method for Forward Obstacle Detection of Autonomous Vehicles. *IEEE Systems Journal* (2023), 1–12. `https://doi.org/10.1109/JSYST.2023.3234200`

[24] Mike Daily, Swarup Medasani, Reinhold Behringer, and Mohan Trivedi. 2017. Self-Driving Cars. *Computer* 50, 12 (2017), 18–23. `https://doi.org/10.1109/MC.2017.4451204`

[25] Apurba Das, Upendra Suddamalla, and Siva Srinivas. 2016. M^{2}BMT: An Automated Ground Truth Generation Algorithm for Lane Detection. In *International Conference on Pattern Recognition Systems (ICPRS-16)*. Institution of Engineering and Technology, Talca, Chile, 1–6. `https://doi.org/10.1049/ic.2016.0028`

[26] Klaus Dietmayer. 2016. *Predicting of Machine Perception for Automated Driving*. Springer Berlin Heidelberg, Berlin, Heidelberg, 407–424.

[27] Riccardo Donà and Biagio Ciuffo. 2022. Virtual Testing of Automated Driving Systems. A Survey on Validation Methods. *IEEE Access* 10 (2022), 24349–24367. `https://doi.org/10.1109/ACCESS.2022.3153722`

[28] Tommaso Dreossi, Alexandre Donzé, and Sanjit A. Seshia. 2019. Compositional Falsification of Cyber-Physical Systems with Machine Learning Components. *Journal of Automated Reasoning* 63, 4 (2019), 1031–1053. `https://doi.org/10.1007/s10817-018-09509-5`

[29] European Commission. 2021. The frequency of speed limit violations. `https://road-safety.transport.ec.europa.eu/eu-road-safety-policy/priorities/safe-road-use/safe-speed/archive/frequency-speed-limit-violations_en` (last accessed on 22.09.2022).

[30] European Commission. 2022. Commission Implementing Regulation (EU) 2022/1426 of 5 August 2022 laying down rules for the application of Regulation (EU) 2019/2144 of the European Parliament and of the Council as regards uniform procedures and technical specifications for the type-approval of the automated driving system (ADS) of fully automated vehicles (Text with EEA relevance). `http://data.europa.eu/eli/reg_impl/2022/1426/oj` (last accessed on 26.09.2023).

[31] European New Car Assessment Programme (Euro NCAP). 2023. TEST PROTOCOL – AEB/LSS VRU systems - Implementation 2023 - Version 4.4. `https://www.euroncap.com/en/for-engineers/protocols/vulnerable-road-user-vru-protection/`

[32] Markus Fach, Frank Baumann, Jörg Breuer, Alexander May, and Stephan Mücke. 2010. Bewertung der Beherrschbarkeit von Aktiven Sicherheits- und Fahrerassistenzsystemen an den Funktionsgrenzen. In *26. VDI-VW-Gemeinschaftstagung Fahrerassistenz und Integrierte Sicherheit: Tagung Wolfsburg, 6. und 7. Oktober 2010*. VDI-Verlag, 425–436.

[33] Federal Ministry for Digital and Transport (Germany). 2022. Verordnung zur Regelung des Betriebs von Kraftfahrzeugen mit automatisierter und autonomer Fahrfunktion und zur Änderung straßenverkehrsrechtlicher Vorschriften. `https://www.bundesrat.de/bv.html?id=0086-22` (last accessed on 22.09.2022).

[34] Federal Ministry of Justice and Consumer Protection (Germany). 2013. Road traffic regulations (StVO) - Annex 2 (to §41(1)) - Regulatory signs. Online. (2013).

[35] Federal Ministry of Justice and Consumer Protection (Germany). 2013. Road traffic regulations (StVO) - §1 Basic rules. Online. (2013).

[36] Federal Ministry of Justice and Consumer Protection (Germany). 2013. Road traffic regulations (StVO) - §25 Pedestrians. Online. (2013).

[37] Federal Ministry of Justice and Consumer Protection (Germany). 2013. Road traffic regulations (StVO) - §37 Traffic light signals, lane control signals and green arrow. Online. (2013).

[38] Federal Ministry of Justice and Consumer Protection (Germany). 2013. Road traffic regulations (StVO) - §4 Distance. Online. (2013).

[39] Di Feng, Xiao Wei, Lars Rosenbaum, Atsuto Maki, and Klaus Dietmayer. 2019. Deep Active Learning for Efficient Training of a LiDAR 3D Object Detector. In *2019 IEEE Intelligent Vehicles Symposium (IV)*. Paris, France, 667–674. `https://doi.org/10.1109/IVS.2019.8814236`

[40] Duarte Fernandes, António Silva, Rafael Névoa, Cláudia Simões, Dibet Gonzalez, Miguel Guevara, Paulo Novais, João Monteiro, and Pedro Melo-Pinto. 2021. Point-cloud based 3D object detection and classification methods for self-driving applications: A survey and taxonomy. *Information Fusion* 68 (2021), 161–191. `https://doi.org/10.1016/j.inffus.2020.11.002`

[41] Daniel J. Fremont, Johnathan Chiu, Dragos D. Margineantu, Denis Osipychev, and Sanjit A. Seshia. 2020. Formal Analysis and Redesign of a Neural Network-Based Aircraft Taxiing System with VerifAI. In *Computer Aided Verification*, Shuvendu K. Lahiri and Chao Wang (Eds.). Springer International Publishing, 122–134. `https://doi.org/10.1007/978-3-030-53288-8_6`

[42] Cheryl D. Fryar, Qiuping Gu, Cynthia L. Ogden, and Katherine M. Flegal. 2016. Anthropometric Reference Data for Children and Adults: United States, 2011-2014. *Vital and health statistics. Series 3, Analytical studies* 39 (2016). `http://europepmc.org/abstract/MED/28437242`

[43] Bernd Gassmann, Fabian Oboril, Cornelius Buerkle, Shuang Liu, Shoumeng Yan, Maria Soledad Elli, Ignacio Alvarez, Naveen Aerrabotu, Suhel Jaber, Peter van Beek, Darshan Iyer, and Jack Weast. 2019. Towards Standardization of AV Safety: C++ Library for Responsibility Sensitive Safety. In *2019 IEEE Intelligent Vehicles Symposium (IV)*. Paris, France, 2265–2271. `https://doi.org/10.1109/IVS.2019.8813885`

[44] Houssem Guissouma, Simon Leiner, and Eric Sax. 2019. Towards Design and Verification of Evolving Cyber Physical Systems Using Contract-Based Methodology. In *2019 International Symposium on Systems Engineering (ISSE)*. Edinburgh, UK, 1–8. `https://doi.org/10.1109/ISSE46696.2019.8984478`

[45] Hatem Hajri and Mohamed-Cherif Rahal. 2019. Real Time Lidar and Radar High-Level Fusion for Obstacle Detection and Tracking with evaluation on a ground truth. arXiv:1807.11264v2 [cs.RO, cs.PF] `http://arxiv.org/abs/1807.11264v2`

[46] Timo Hanke, Nils Hirsenkorn, Bernhard Dehlink, Andreas Rauch, Ralph Rasshofer, and Erwin Biebl. 2016. Classification of Sensor Errors for the Statistical Simulation of Environmental Perception in Automated Driving Systems. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. Rio de Janeiro, Brazil, 643–648. `https://doi.org/10.1109/ITSC.2016.7795621`

[47] Lars Hanson, Lena Sperling, Gunvor Gard, Staffan Ipsen, and Cindy Olivares Vergara. 2009. Swedish anthropometrics for product and workplace design. *Applied Ergonomics* 40, 4 (2009), 797–806. `https://doi.org/10.1016/j.apergo.2008.08.007`

[48] Lukas Hartjen, Robin Philipp, Fabian Schuldt, and Bernhard Friedrich. 2020. Saturation Effects in Recorded Maneuver Data for the Test of Automated Driving. In *13. Uni-DAS e.V. Workshop Fahrerassistenz und automatisiertes Fahren*. Walting, Germany, 74–83. `https://www.uni-das.de/images/pdf/fas-workshop/2020/FAS_2020_HARTJEN.pdf`

[49] Lukas Hartjen, Robin Philipp, Fabian Schuldt, Falk Howar, and Bernhard Friedrich. 2019. Classification of Driving Maneuvers in Urban Traffic for Parametrization of Test Scenarios. In *9. Tagung Automatisiertes Fahren*. Lehrstuhl für Fahrzeugtechnik mit TÜV SÜD Akademie, Munich, Germany. `https://mediatum.ub.tum.de/doc/1535131/file.pdf`

[50] Lukas Hartjen, Fabian Schuldt, and Bernhard Friedrich. 2019. Semantic Classification of Pedestrian Traffic Scenarios for the Validation of Automated Driving. In *2019 IEEE 22nd International Conference on Intelligent Transportation Systems (ITSC)*. Auckland, New Zealand, 3696–3701. `https://doi.org/10.1109/ITSC.2019.8917485`

[51] Franziska Henze, Dennis Faßbender, and Christoph Stiller. 2021. Identifying Admissible Uncertainty Bounds for the Input of Planning Algorithms. *IEEE Transactions on Intelligent Vehicles* (2021), 1–1. `https://doi.org/10.1109/TIV.2021.3119352`

[52] Martin Friedrich Holder, Clemens Linnhoff, Philipp Rosenberger, Christoph Popp, and Hermann Winner. 2019. Modeling and Simulation of Radar Sensor Artifacts for Virtual Testing of Autonomous Driving. In *9. Tagung Automatisiertes Fahren.* Lehrstuhl für Fahrzeugtechnik mit TÜV SÜD Akademie, Munich, Germany. `https://mediatum.ub.tum.de/doc/1535151/file.pdf`

[53] Bardh Hoxha, Adel Dokhanchi, and Georgios Fainekos. 2017. Mining parametric temporal logic properties in model-based design for cyber-physical systems. *International Journal on Software Tools for Technology Transfer* 20, 1, 79–93. `https://doi.org/10.1007/s10009-017-0447-4`

[54] Rui Huang, Wanyue Zhang, Abhijit Kundu, Caroline Pantofaru, David A. Ross, Thomas Funkhouser, and Alireza Fathi. 2020. An LSTM Approach to Temporal 3D Object Detection in LiDAR Point Clouds. In *Computer Vision – ECCV 2020.* Springer International Publishing, 266–282. `https://doi.org/10.1007/978-3-030-58523-5_16`

[55] ISO 21448. 2022. *Road vehicles — Safety of the intended functionality.* Standard. International Organization for Standardization (ISO), Geneva, Switzerland.

[56] Boris Ivanovic and Marco Pavone. 2022. Injecting Planning-Awareness into Prediction and Detection Evaluation. In *2022 IEEE Intelligent Vehicles Symposium (IV).* Aachen, Germany, 821–828. `https://doi.org/10.1109/IV51971.2022.9827101`

[57] Jean-Baptiste Jeannin, Khalil Ghorbal, Yanni Kouskoulas, Ryan Gardner, Aurora Schmidt, Erik Zawadzki, and André Platzer. 2015. A Formally Verified Hybrid System for the Next-Generation Airborne Collision Avoidance System. In *Tools and Algorithms for the Construction and Analysis of Systems.* Springer Berlin Heidelberg, 21–36. `https://doi.org/10.1007/978-3-662-46681-0_2`

[58] Xiaoqing Jin, Alexandre Donze, Jyotirmoy V. Deshmukh, and Sanjit A. Seshia. 2015. Mining Requirements From Closed-Loop Control Models. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 34, 11, 1704–1717. `https://doi.org/10.1109/TCAD.2015.2421907`

[59] Kyle D. Julian and Mykel J. Kochenderfer. 2019. Guaranteeing Safety for Neural Network-Based Aircraft Collision Avoidance Systems. In *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC).* San Diego, CA, USA, 1–10. `https://doi.org/10.1109/DASC43569.2019.9081748`

[60] Just Auto (Kurt Robson). 2023. Fully self-driving cars unlikely before 2035, experts predict. `https://www.just-auto.com/news/fully-self-driving-cars-unlikely-before-2035-experts-predict/` (last accessed on 26.10.2023).

[61] Nidhi Kalra and Susan M. Paddock. 2016. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice* 94 (dec 2016), 182–193. https://doi.org/10.1016/j.tra.2016.09.010

[62] Ahmet-Serdar Karakaya, Jonathan Hasenburg, and David Bermbach. 2020. SimRa: Using crowdsourcing to identify near miss hotspots in bicycle traffic. *Pervasive and Mobile Computing* 67 (2020), 101197. https://doi.org/10.1016/j.pmcj.2020.101197

[63] Ahmet-Serdar Karakaya, Ioan-Alexandru Stef, Konstantin Köhler, Julian Heinovski, Falko Dressler, and David Bermbach. 2023. Achieving realistic cyclist behavior in SUMO using the SimRa dataset. *Computer Communications* 205 (2023), 97–107. https://doi.org/10.1016/j.comcom.2023.04.015

[64] Björn Klamann, Moritz Lippert, Christian Amersbach, and Hermann Winner. 2019. Defining Pass-/Fail-Criteria for Particular Tests of Automated Driving Functions. In *2019 IEEE 22nd International Conference on Intelligent Transportation Systems (ITSC)*. Auckland, New Zealand, 169–174. https://doi.org/10.1109/ITSC.2019.8917483

[65] Robert Krajewski, Michael Hoss, Adrian Meister, Fabian Thomsen, Julian Bock, and Lutz Eckstein. 2020. Using drones as reference sensors for neural-networks-based modeling of automotive perception errors. In *2020 IEEE Intelligent Vehicles Symposium (IV)*. Las Vegas, NV, USA, 708–715. https://doi.org/10.1109/IV47402.2020.9304615

[66] Jungwook Lee, Sean Walsh, Ali Harakeh, and Steven L. Waslander. 2018. Leveraging Pre-Trained 3D Object Detection Models for Fast Ground Truth Generation. In *2018 IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*. Maui, HI, USA, 2504–2510. https://doi.org/10.1109/itsc.2018.8569793

[67] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. 2018. Deep Continuous Fusion for Multi-sensor 3D Object Detection. In *Computer Vision – ECCV 2018*. Springer International Publishing, Munich. Germany, 663–678. https://doi.org/10.1007/978-3-030-01270-0_39

[68] Bob Lightsey. 2001. *Systems Engineering Fundamentals*. Technical Report. Department of Defense - Systems Management College, Fort Belvoir, VA, USA.

[69] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler. 2019. Fast Interactive Object Annotation With Curve-GCN. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA, 5252–5261. https://doi.org/10.1109/CVPR.2019.00540

[70] Matt Luckcuck, Marie Farrell, Louise A. Dennis, Clare Dixon, and Michael Fisher. 2019. Formal Specification and Verification of Autonomous Robotic Systems. *Comput. Surveys* 52, 5 (2019), 1–41. https://doi.org/10.1145/3342355

[71] Wenjie Luo, Bin Yang, and Raquel Urtasun. 2018. Fast and Furious: Real Time End-to-End 3D Detection, Tracking and Motion Forecasting With a Single Convolutional Net. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Salt Lake City, UT, USA, 3569–3577. `https://doi.org/10.1109/CVPR.2018.00376`

[72] Maria Lyssenko, Christoph Gladisch, Christian Heinzemann, Matthias Woehrle, and Rudolph Triebel. 2021. From Evaluation to Verification: Towards Task-Oriented Relevance Metrics for Pedestrian Detection in Safety-Critical Domains. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Nashville, TN, USA, 38–45. `https://doi.org/10.1109/CVPRW53098.2021.00013`

[73] Zoltan Ferenc Magosi, Hexuan Li, Philipp Rosenberger, Li Wan, and Arno Eichberger. 2022. A Survey on Modelling of Automotive Radar Sensors for Virtual Test and Validation of Automated Driving. *Sensors* 22, 15 (jul 2022), 5693. `https://doi.org/10.3390/s22155693`

[74] Leandro Masello, German Castignani, Barry Sheehan, Finbarr Murphy, and Kevin McDonnell. 2022. On the road safety benefits of advanced driver assistance systems in different driving contexts. *Transportation Research Interdisciplinary Perspectives* 15 (2022), Article 100670. `https://doi.org/10.1016/j.trip.2022.100670`

[75] Pierre Merriaux, Yohan Dupuis, Rémi Boutteau, Pascal Vasseur, and Xavier Savatier. 2017. LiDAR point clouds correction acquired from a moving car based on CAN-bus data. arXiv:1706.05886v1 [cs.RO] `http://arxiv.org/abs/1706.05886v1`

[76] Gregory P. Meyer, Ankit Laddha, Eric Kee, Carlos Vallespi-Gonzalez, and Carl K. Wellington. 2019. LaserNet: An Efficient Probabilistic 3D Object Detector for Autonomous Driving. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA, 12669–12678. `https://doi.org/10.1109/CVPR.2019.01296`

[77] Mehrdad Moradi, Bentley James Oakes, Mustafa Saraoglu, Andrey Morozov, Klaus Janschek, and Joachim Denil. 2020. Exploring Fault Parameter Space Using Reinforcement Learning-based Fault Injection. In *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. Valencia, Spain, 102–109. `https://doi.org/10.1109/DSN-W50199.2020.00028`

[78] Eshed Ohn-Bar and Mohan Manubhai Trivedi. 2017. Are all objects equal? Deep spatio-temporal importance prediction in driving videos. *Pattern Recognition* 64 (2017), 425–436. `https://doi.org/10.1016/j.patcog.2016.08.029`

[79] Corina S. Păsăreanu, Divya Gopinath, and Huafeng Yu. 2018. Compositional Verification for Autonomous Systems with Deep Learning Components. In *Safe, Autonomous and Intelligent Vehicles*. Springer International Publishing, 187–197. `https://doi.org/10.1007/978-3-319-97301-2_10`

[80] Raphael Pfeffer and Tobias Leichsenring. 2016. Continuous Development of Highly Automated Driving Functions with Vehicle-in-the-Loop Using the Example of Euro NCAP Scenarios. In *Simulation and Testing for Vehicle Technology*. Springer International Publishing, 33–42. https://doi.org/10.1007/978-3-319-32345-9_4

[81] Jonah Philion, Amlan Kar, and Sanja Fidler. 2020. Learning to Evaluate Perception Models Using Planner-Centric Metrics. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA, 14052–14061. https://doi.org/10.1109/CVPR42600.2020.01407

[82] Robin Philipp, Hedan Qian, Lukas Hartjen, Fabian Schuldt, and Falk Howar. 2021. Simulation-Based Elicitation of Accuracy Requirements for the Environmental Perception of Autonomous Vehicles. In *Leveraging Applications of Formal Methods, Verification and Validation*. Rhodes, Greece, 129–145. https://doi.org/10.1007/978-3-030-89159-6_9

[83] Robin Philipp, Jana Rehbein, Felix Grün, Lukas Hartjen, Zhijing Zhu, Fabian Schuldt, and Falk Howar. 2022. Systematization of Relevant Road Users for the Evaluation of Autonomous Vehicle Perception. In *2022 IEEE International Systems Conference (SysCon)*. Montreal, Canada, 1–8. https://doi.org/10.1109/SysCon53536.2022.9773877

[84] Robin Philipp, Fabian Schuldt, and Falk Howar. 2020. Functional Decomposition of Automated Driving Systems for the Classification and Evaluation of Perceptual Threats. In *13. Uni-DAS e.V. Workshop Fahrerassistenz und automatisiertes Fahren*. Walting, Germany, 93–105. https://www.uni-das.de/images/pdf/fas-workshop/2020/FAS_2020_PHILIPP.pdf

[85] Robin Philipp, Zhijing Zhu, Julian Fuchs, Lukas Hartjen, Fabian Schuldt, and Falk Howar. 2021. Automated 3D Object Reference Generation for the Evaluation of Autonomous Vehicle Perception. In *2021 5th International Conference on System Reliability and Safety (ICSRS)*. IEEE Computer Society, Palermo, Italy. https://doi.org/10.1109/ICSRS53853.2021.9660660

[86] Andrea Piazzoni, Jim Cherian, Martin Slavik, and Justin Dauwels. 2020. Modeling Perception Errors towards Robust Decision Making in Autonomous Vehicles. In *29th International Joint Conference on Artificial Intelligence*. Yokohama, Japan, 3494–3500. https://doi.org/10.24963/ijcai.2020/483

[87] Andrea Piazzoni, Jim Cherian, Martin Slavik, and Justin Dauwels. 2020. Modeling Perception Errors towards Robust Decision Making in Autonomous Vehicles. arXiv:2001.11695v2 [cs.AI, cs.RO, C.4; I.2; I.6] http://arxiv.org/abs/2001.11695v2

[88] Charles R. Qi, Yin Zhou, Mahyar Najibi, Pei Sun, Khoa Vo, Boyang Deng, and Dragomir Anguelov. 2021. Offboard 3D Object Detection From Point Cloud Sequences. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Nashville, TN, USA, 6134–6144. https://doi.org/10.1109/CVPR46437.2021.00607

[89] Deepak Rao, Plato Pathrose, Felix Huening, and Jithin Sid. 2019. An Approach for Validating Safety of Perception Software in Autonomous Driving Systems. In *Model-Based Safety and Assessment*. Springer International Publishing, 303–316. https://doi.org/10.1007/978-3-030-32872-6_20

[90] Govind Rathore, Wan-Yin Lin, and Ji Eun Kim. 2019. DeepBbox: Accelerating Precise Ground Truth Generation for Autonomous Driving Datasets. In *2019 IEEE 22nd International Conference on Intelligent Transportation Systems (ITSC)*. Auckland, New Zealand, 1871–1876. https://doi.org/10.1109/itsc.2019.8917133

[91] Stefan Riedmaier. 2022. *Model Validation and Uncertainty Aggregation for Safety Assessment of Automated Vehicles.* Ph. D. Dissertation. Technische Universität München.

[92] Stefan Riedmaier, Jonas Nesensohn, Christian Gutenkunst, Bernhard Schick, Tobias Düser, and Houssem Abdellatif. 2018. Validation of X-in-the-Loop Approaches for Virtual Homologation of Automated Driving Functions. In *11. Grazer Symposium VIRTUAL VEHICLE (GSVF) Graz, 15./16.05.2018.* https://www.researchgate.net/publication/338392015_Validation_of_X-in-the-Loop_Approaches_for_Virtual_Homologation_of_Automated_Driving_Functions

[93] Jens Rieken, Richard Matthaei, and Markus Maurer. 2015. Benefits of using explicit ground-plane information for grid-based urban environment modeling. In *18th International Conference on Information Fusion (Fusion 2015)*. IEEE, Washington, DC, USA, 2049–2056.

[94] Gustavo Romanillos and Javier Gutiérrez. 2019. Cyclists do better. Analyzing urban cycling operating speeds and accessibility. *International Journal of Sustainable Transportation* 14, 6 (mar 2019), 448–464. https://doi.org/10.1080/15568318.2019.1575493

[95] Philipp Rosenberger, Martin Holder, Sebastian Huch, Hermann Winner, Tobias Fleck, Marc Rene Zofka, J. Marius Zöllner, Thomas D'hondt, and Benjamin Wassermann. 2019. Benchmarking and Functional Decomposition of Automotive Lidar Sensor Models. In *2019 IEEE Intelligent Vehicles Symposium (IV)*. Paris, France, 632–639. https://doi.org/10.1109/IVS.2019.8814081

[96] Alberto Sangiovanni-Vincentelli, Werner Damm, and Roberto Passerone. 2012. Taming Dr. Frankenstein: Contract-Based Design for Cyber-Physical Systems. *European Journal of Control* 18 (2012), 217–238. https://doi.org/10.3166/ejc.18.217-238

[97] Nicolas Scheiner, Nils Appenrodt, Jürgen Dickmann, and Bernhard Sick. 2019. Automated Ground Truth Estimation of Vulnerable Road Users in Automotive Radar Data Using GNSS. In *IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*. Detroit, MI, USA, 1–5. https://doi.org/10.1109/icmim.2019.8726801

[98] Fabian Schuldt. 2017. *Ein Beitrag für den methodischen Test von automatisierten Fahrfunktionen mit Hilfe von virtuellen Umgebungen.* Ph. D. Dissertation. Technische Universität Braunschweig.

[99] Valerij Schönemann, Mara Duschek, and Hermann Winner. 2019. Maneuver-based Adaptive Safety Zone for Infrastructure-Supported Automated Valet Parking. In *5th International Conference on Vehicle Technology and Intelligent Transport Systems*. SCITEPRESS - Science and Technology Publications, Heraklion, Crete, Greece, 343–351. https://doi.org/10.5220/0007689503430351

[100] Alexandru Constantin Serban, Erik Poll, and Joost Visser. 2018. A Standard Driven Software Architecture for Fully Autonomous Vehicles. In *2018 IEEE International Conference on Software Architecture Companion (ICSA-C)*. Seattle, WA, USA, 120–127. https://doi.org/10.1109/ICSA-C.2018.00040

[101] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. 2018. On a Formal Model of Safe and Scalable Self-driving Cars. arXiv:1708.06374v6 [cs.RO, cs.AI, stat.ML] http://arxiv.org/abs/1708.06374v6

[102] Rebecca Spicer, Amin Vahabaghaie, George Bahouth, Ludwig Drees, Robert Martinez von Bülow, and Peter Baur. 2018. Field effectiveness evaluation of advanced driver assistance systems. *Traffic Injury Prevention* 19, sup2 (2018), S91–S95. https://doi.org/10.1080/15389588.2018.1527030

[103] Christoph Stadler, Francesco Montanari, Wojciech Baron, Christoph Sippl, and Anatoli Djanatliev. 2022. A Credibility Assessment Approach for Scenario-Based Virtual Testing of Automated Driving Functions. *IEEE Open Journal of Intelligent Transportation Systems* 3 (2022), 45–60. https://doi.org/10.1109/OJITS.2022.3140493

[104] Tim Stahl and Frank Diermeyer. 2021. Online Verification Enabling Approval of Driving Functions—Implementation for a Planner of an Autonomous Race Vehicle. *IEEE Open Journal of Intelligent Transportation Systems* 2 (2021), 97–110. https://doi.org/10.1109/OJITS.2021.3078121

[105] Tim Stahl, Matthis Eicher, Johannes Betz, and Frank Diermeyer. 2020. Online Verification Concept for Autonomous Vehicles – Illustrative Study for a Trajectory Planning Module. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. Rhodes, Greece, 1–7. https://doi.org/10.1109/ITSC45102.2020.9294703

[106] Jan Erik Stellet, Matthias Woehrle, Tino Brade, Alexander Poddey, and Wolfgang Branz. 2020. Validation of automated driving–a structured analysis and survey of approaches. In *13. Uni-DAS e.V. Workshop Fahrerassistenz und automatisiertes Fahren*. Walting, Germany, 64–73. https://www.uni-das.de/images/pdf/fas-workshop/2020/FAS_2020_STELLET.pdf

[107] Jan Erik Stellet, Marc Rene Zofka, Jan Schumacher, Thomas Schamm, Frank Niewels, and J. Marius Zollner. 2015. Testing of Advanced Driver Assistance Towards Automated Driving: A Survey and Taxonomy on Existing Approaches and Open Questions. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE. https://doi.org/10.1109/ITSC.2015.236

[108] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. 2020. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA, 2443–2451. `https://doi.org/10.1109/CVPR42600.2020.00252`

[109] TechCrunch (Darrell Etherington). 2017. BMW's self-driving car will aim for full Level 5 autonomy by 2021. `https://techcrunch.com/2017/03/16/bmws-self-driving-car-will-aim-for-full-level-5-autonomy-by-2021` (last accessed on 26.10.2023).

[110] Claire Jennifer Tomlin, Ian Mitchell, Alexandre M. Bayen, and Meeko Oishi. 2003. Computational techniques for the verification of hybrid systems. *Proc. IEEE* 91, 7 (2003), 986–1001. `https://doi.org/10.1109/JPROC.2003.814621`

[111] Sever Topan, Karen Leung, Yuxiao Chen, Pritish Tupekar, Edward Schmerling, Jonas Nilsson, Michael Cox, and Marco Pavone. 2022. Interaction-Dynamics-Aware Perception Zones for Obstacle Detection Safety Evaluation. In *2022 IEEE Intelligent Vehicles Symposium (IV)*. Aachen, Germany, 1201–1210. `https://doi.org/10.1109/IV51971.2022.9827409`

[112] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. 2000. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E* 62, 2 (2000), 1805–1824. `https://doi.org/10.1103/PhysRevE.62.1805`

[113] Cumhur Erkan Tuncali, Georgios Fainekos, Hisahiro Ito, and James Kapinski. 2018. Simulation-based Adversarial Test Generation for Autonomous Vehicles with Machine Learning Components. In *2018 IEEE Intelligent Vehicles Symposium (IV)*. Changshu, China, 1555–1562. `https://doi.org/10.1109/IVS.2018.8500421`

[114] Cumhur Erkan Tuncali, Georgios Fainekos, Danil Prokhorov, Hisahiro Ito, and James Kapinski. 2020. Requirements-Driven Test Generation for Autonomous Vehicles With Machine Learning Components. *IEEE Transactions on Intelligent Vehicles* 5, 2 (2020), 265–280. `https://doi.org/10.1109/TIV.2019.2955903`

[115] Simon Ulbrich, Till Menzel, Andreas Reschka, Fabian Schuldt, and Markus Maurer. 2015. Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems (ITSC)*. Las Palmas, Spain, 982–988. `https://doi.org/10.1109/itsc.2015.164`

[116] Union Cycliste Internationale. 2021. Clarification Guide of the UCI Technical Regulation. (2021).

[117] United Nations Economic Commission for Europe (UNECE). 2023. UN Regulation No. 157 - Uniform provisions concerning the approval of vehicles with regard to

Automated Lane Keeping Systems (E/ECE/TRANS/505/Rev.3/Add.156/Amend.4). https://unece.org/transport/vehicle-regulations-wp29/standards/addenda-1958-agreement-regulations-141-160 (last accessed on 24.10.2023).

[118] Walter G. Vincenti. 1990. *What Engineers Know and How They Know It*. Vol. 141. The Johns Hopkins University Press, Baltimore, USA.

[119] Georg Volk, Jörg Gamerdinger, Alexander von Bernuth, and Oliver Bringmann. 2020. A Comprehensive Safety Metric to Evaluate Perception in Autonomous Systems. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. Rhodes, Greece, 1–8. https://doi.org/10.1109/ITSC45102.2020.9294708

[120] Walther Wachenfeld and Hermann Winner. 2016. The Release of Autonomous Vehicles. In *Autonomous Driving*. Springer Berlin Heidelberg, 425–449. https://doi.org/10.1007/978-3-662-48847-8_21

[121] Mirja Wolf, Luiz R. Douat, and Michael Erz. 2021. Safety-Aware Metric for People Detection. In *2021 IEEE 24th International Conference on Intelligent Transportation Systems (ITSC)*. Indianapolis, IN, USA, 2759–2765. https://doi.org/10.1109/ITSC48978.2021.9564734

[122] Bin Yang, Min Bai, Ming Liang, Wenyuan Zeng, and Raquel Urtasun. 2021. Auto4D: Learning to Label 4D Objects from Sequential Point Clouds. arXiv:2101.06586v2 [cs.CV] http://arxiv.org/abs/2101.06586v2

[123] Bin Yang, Wenjie Luo, and Raquel Urtasun. 2018. PIXOR: Real-time 3D Object Detection from Point Clouds. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Salt Lake City, UT, USA, 7652–7660. https://doi.org/10.1109/CVPR.2018.00798

[124] Bin Yang, Wenjie Luo, and Raquel Urtasun. 2019. PIXOR: Real-time 3D Object Detection from Point Clouds. arXiv:1902.06326v3 [cs.CV] http://arxiv.org/abs/1902.06326v3

[125] Junbo Yin, Jianbing Shen, Chenye Guan, Dingfu Zhou, and Ruigang Yang. 2020. LiDAR-Based Online 3D Video Object Detection With Graph-Based Message Passing and Spatiotemporal Transformer Attention. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA, 11492–11501. https://doi.org/10.1109/CVPR42600.2020.01151

[126] Sergey Zakharov, Wadim Kehl, Arjun Bhargava, and Adrien Gaidon. 2020. Autolabeling 3D Objects With Differentiable Rendering of SDF Shape Priors. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA, 12221–12230. https://doi.org/10.1109/CVPR42600.2020.01224

[127] Jakub Zębala, Piotr Ciepka, and Adam Reza. 2012. Pedestrian acceleration and speeds. *Problems of Forensic Sciences* 91 (2012), 227–234.

[128] Zhijing Zhu, Robin Philipp, Constanze Hungar, and Falk Howar. 2022. Systematization and Identification of Triggering Conditions: A Preliminary Step for Efficient Testing of Autonomous Vehicles. In *2022 IEEE Intelligent Vehicles Symposium (IV)*. Aachen, Germany, 798–805. `https://doi.org/10.1109/IV51971.2022.9827238`

[129] Zhijing Zhu, Robin Philipp, Yongqi Zhao, Constanze Hungar, Jürgen Pannek, and Falk Howar. 2023. Automatic Disengagement Scenario Reconstruction Based on Urban Test Drives of Automated Vehicles. In *2023 IEEE Intelligent Vehicles Symposium (IV)*. Anchorage, AK, USA, 1–8. `https://doi.org/10.1109/IV55152.2023.10186640`