

CAPABILITY-BASED ROUTES FOR DEVELOPMENT, TESTING AND OPERATION OF SAFE AUTOMATED VEHICLES

Moritz Lippert

Institute of Automotive Engineering, Technical University of Darmstadt
Germany

Hermann Winner

Former professor for Automotive Engineering, Technical University of Darmstadt
Germany

Paper Number 23-0121

ABSTRACT

Ensuring safety of automated vehicles (AVs) within their operational design domain (ODD) is essential for a release. In some parts of the ODD, ensuring safe operation is more challenging, requiring more sophisticated driving capabilities. For example, the same intersection requires different capabilities depending on the selected turn, i.e., if driving right, left, or straight ahead. To guarantee safe operation, only route sections for which capabilities for safe driving are available and validated should be selected. So far, the direct relationship between routes within ODDs and the driving capabilities of AVs has not been explicitly addressed. This paper presents for the first time an approach to identify routes with driving requirements that do not exceed driving capabilities of AVs. To this end, this approach builds on the Behavior-Semantic Scenery Description (BSSD), which links behavioral demands for AVs directly to the scenery as a central element of the ODD. Based on the BSSD, route-based behavioral requirements are derived. Geometric characteristics of the scenery are used to specify driving requirements and driving capabilities that can be matched as a function of route and developed matching criteria. This matching is integrated into a conventional route planner, which as a result determines routes that are drivable based on the driving capabilities of an AV. The application to a real road network shows that the identification of capability-based routes is generally possible. Different intersections demand different requirements and lead to different routes. Nevertheless, several challenges are discussed that need to be overcome for a real-world application for development, testing, and operation of AVs.

INTRODUCTION

According to the current state of research and technology, the development and release process of automated vehicles (AVs) is a wide field of unsolved problems. One core challenge is the safety validation of these vehicles, which can no longer be performed using a conventional statistical approach [1] due to the high complexity of the overall system. Therefore, alternatives are needed to partially replace the statistical safety validation and thus reduce the overall effort in the development of AVs. One approach, which is additionally reinforced by various norms and standards such as ISO/TR 4804 [2] or ISO 21448 [3], is the safety-by-design approach. With the help of this approach, safety-relevant aspects are to be explicitly addressed in the development from the beginning, so that the final proof is facilitated. The goal of a safety or functional validation is to prove that the AV behaves according to the functional specification and does not exhibit any safety-critical deviations from this behavior. In this context, the functional scope is defined in the operational design domain (ODD), which describes the operational conditions for which an AV is specified and designed to function [4]. As the only final published standard for the specification of ODDs so far, PAS 1883 [5] defines the three main components *scenery*, *environmental conditions* and *dynamic elements* to describe an ODD. As a core element, the scenery describes all non-movable elements of the ODD. According to Ulbrich et al. [6], the scenery includes, for example, the road and lane network, but also stationary objects such as traffic lights or curbs. Thus, the scenery describes the space for the vehicle motion and behavior.

The proof of safe function and operation is thus closely linked to the defined scenery. In order to fulfill its function, an AV requires driving capabilities that can cope with the driving requirements arising from the scenery within the ODD. Many problems regarding development and safety validation of AVs are considered in related work, but the essential minimal task of AVs is often neglected - accomplishing an actual route within the ODD. The route forms the central element of AVs' mission accomplishment, accurately mapping the interdependencies between driving capabilities and location-based, different driving requirements of the ODD. Different routes potentially demand different driving capabilities that must be proven. Therefore, it seems reasonable to design the development, testing, and operation of AVs based on routes within the ODD.

For this purpose, it is necessary to establish a relationship between the scenery to be driven on and the resulting driving requirements as well as driving capabilities. The dependency between driving requirements and specific scenery areas within the ODD is explicitly addressed for the first time in a previous work [7] of the author team

of this paper. We show how behavioral requirements for AVs can be derived based on routes within the scenery. For deriving the behavioral requirements, the Behavior-Semantic Scenery Description (BSSD) is used, which was co-developed by the same author team [8, 9]. The BSSD represents the behavioral limits imposed on AVs based on the scenery and applicable traffic rules enabling a route-based analysis of the ODD. Based on this previous work, the present paper develops a method for identifying capability-based routes. The driving requirements of these routes do not exceed the driving capabilities of selected AVs - no matter if already released or still under development. These routes and their associated driving requirements as well as required driving capabilities can be used for both the development process and operation of safe AVs.

This paper is structured as follows. First, the fundamental basics for a route-based scenery analysis based on the previous work are explained. Subsequently, an overall concept for the identification of capability-based routes is developed and implemented. Using a real road network as a basis, the developed concept is applied and evaluated. Finally, the remaining need for research is derived. A detailed description of the related work is intentionally not provided in this paper, as it can be found in the recently (co-)published papers of this author team [7-9].

FUNDAMENTALS

The dependency between scenery and resulting driving requirements for AVs is already described by Lippert and Winner [7]. The basis is the externally observable behavior of an AV while driving through a scenery. Based on this behavior, it can basically be determined how safe or unsafe an AV behaves during operation. This behavior is significantly influenced by the scenery, which imposes behavioral limits on an AV when combined with the applicable traffic rules. For example, a stop sign means that an AV must stop at the associated stop line before driving any further. It does not matter whether or not priority is given to another traffic participant during the stop. With the help of the BSSD, these behavioral limits are explicitly linked to the scenery and represented in a map. In this way, route-based behavioral requirements can be derived that result from the behavioral limits of the BSSD. The behavioral demands serve the present work as a basis for identifying the capability-based routes. Therefore, the BSSD and subsequently the derivation of route-based behavioral requirements is explained in the following.

Behavior-Semantic Scenery Description [8, 9]

The BSSD represents the legal behavioral limits based on scenery and traffic rules. These so-called *behavioral demands* are represented using directional *behavior spaces*, which usually describe a lane segment and within which the behavioral demands do not change. A behavior space is always described with four behavioral attributes that reflect the behavioral demands: *Speed* (S), *Boundary* (B), *Reservation* (R), and *Overtake* (O). Each of these attributes has additional properties that concretize the behavioral demands. Speed describes any limitations on travel speed. Boundary describes limitations when crossing the boundaries of the behavior spaces. Behavioral demands regarding priority and residence within the behavior spaces is described by the reservation attribute. Permission or prohibition of overtaking is represented by the attribute overtake.

For illustration, a behavior space is concretely considered using a real scenery. Figure 1 shows an aerial view of a real scenery in Darmstadt, Germany, in the upper left. The remaining part of the figure is ignored for the explanation of the behavior space. The behavior spaces are labeled with capital letters. Each behavior space has exactly one longitudinal (entry) boundary (black dashed line) and two lateral (exit) boundaries (black solid line). For each behavior space, there is an inverted behavior space that represents the behavioral demands for the opposite direction of travel (only one direction shown here). The principle of the behavior space is explained using behavior space K, which represents a right turn at a T-intersection with a stop sign. A speed limit of 30 km/h applies in this area. This limit is stored in the speed attribute. Before entering the behavior space, the vehicle must stop due to the stop sign. Therefore, the attribute boundary contains the behavioral demand *stop* for the longitudinal boundary. Laterally, the behavior space should not be left when turning, so the behavioral demands for the lateral boundaries prohibit crossing. For example, between behavior spaces H and E is a dashed lane marking, so the behavioral demands for these behavioral boundaries allow passing at this point. Behavior space K may only be traversed if priority is given to traffic participants approaching from F and C. Therefore, the attribute reservation shows *externally-reserved* for the road user types *motor vehicle* and *bicyclist* potentially coming from C and F (these *reservation links* are not shown here). Overtaking is allowed in the behavior space, so the attribute overtake is set to the value *yes*. In the same way, all other behavior spaces are defined. The concrete, remaining properties of the four behavioral attributes are not further relevant in the context of this work, which is why they are not discussed in more detail.

Route-Based Behavioral Requirements [7]

Lippert and Winner show how behavioral requirements are derived based on BSSD. For this purpose, they first define the terms *global* and *local behavioral requirements*. Global behavioral requirements apply everywhere regardless of the scenery and are, for example, requirements regarding collision avoidance (e.g. [10]). Local

behavioral requirements, on the other hand, have a local scope and are scenery-specific. Thus, they do not apply everywhere, but only at specific locations within a scenery. The derivation of the behavioral requirements refers only to the local behavioral requirements, which is why they are simply called behavioral requirements in the following. According to Lippert and Winner, the complete behavioral requirements result only from a concatenation of the behavior spaces. Thus, lane-accurate routes are considered that concatenate the behavior spaces of the BSSD and thus generate a sequence of the different behavioral demands. The concatenation of the different behavioral demands can lead to new behavioral demands depending on the *transition* (*longitudinal* or *lateral*). Based on the behavioral spaces concatenated within a route, behavioral requirements are derived for each behavioral space and the resulting behavioral demands.

Figure 1 shows the resulting behavioral requirements for an example concatenation of behavior spaces. The lane-accurate route leads from behavior space I to behavior space H. A total of four behavior spaces are traversed, of which behavior space K represents the right turn considered earlier. The concatenated behavior spaces E_i are numbered sequentially. For each behavior space E_i , the associated transition $T_{i-1,i}$, the behavioral demands D_i of the BSSD, and the behavioral demands $D_{i-1,i}$ resulting from concatenation are shown. For each behavior space, a speed limit of 30 km/h applies. For the transitions between the behavior spaces, only when entering E_3 there is a condition that the AV must stop before proceeding. As described in the previous example, the AV must additionally give priority to motor vehicles and cyclists at E_3 and must not obstruct them. From the concatenation it additionally follows, among other things (not shown here), that the traffic participants entitled to priority must be indicated that priority is actually granted to them. In the other behavior spaces, the AV does not have to grant priority, which is why the reservation attribute shows *own-reserved* (own). Overtaking is allowed in any behavior space (yes). In the lower part of the figure, the resulting behavioral requirements are listed and marked with a cross according to their validity in each behavior space. This brief example is used to understand the route-based behavioral requirements as a basis for capability-based routes, and therefore it is not elaborated further.

		E_i	$i = 1$	$i = 2$	$i = 3$	$i = 4$
		$T_{i-1,i}$	longitud.	longitud.	longitud.	longitud.
		D_i	S: 30 km/h R: own O: yes	S: 30 km/h B: allowed R: own O: yes	S: 30 km/h B: stop R: ext. O: yes	S: 30 km/h B: allowed R: own O: yes
		$D_{i-1,i}$	-	-	R: indication of giving priority	-
Requirements of E_i	SR1: The AV shall not exceed the maximum permissible speed limit.		x	x	x	x
	BR1: The AV shall stop at the longitudinal boundary before proceeding.				x	
	RR1: The AV shall not obstruct traffic participants with reservation entitlement for the space.				x	
	RR1.1: The AV shall indicate in advance by adjusting the driving speed reasonably that it will give priority to traffic participants who have priority.				x	

Figure 1. Example for behavior spaces and corresponding route-based behavioral requirements based on [7]. SR: Speed Requirement; BR: Boundary Requirements; RR: Reservation Requirement.

CONCEPT OF MATCHING REQUIREMENTS AND CAPABILITIES

The main goal of this work is to identify routes that can be accomplished by AVs based on their driving capabilities. Consequently, the driving requirements of the route must not exceed the driving capabilities of the vehicles. In order to identify an exceeding of driving capabilities it is necessary that they can be matched with the driving requirements of the route. This matching determines whether the route can be mastered by an AV or not. In order to match, the requirements and capabilities must be compatible with each other. This means that for each driving requirement there must also be a corresponding driving capability. It is important that the driving capabilities of the AVs can be proven. According to the state of the art, the proof of driving functions is typically achieved with the help of various tests [11]. Therefore, it is assumed that test certificates exist for driving capabilities that have been tested and thus proven. If a driving capability has been successfully tested and proven,

a test certificate exists for this driving capability. Whether the driving capabilities really meet the driving requirements is determined by the matching process. This process requires matching criteria that determine a match based on appropriate metrics. The following argumentation results from these considerations:

Let DR_i be the set of necessary driving requirements in order to drive in a concatenated behavior space E_i . Furthermore, let $\bigcup_{m=1}^{n_{DC}} DC_m$ be the superset of $n_{DC} \in \mathbb{N}$ sets of driving capabilities of an AV that is proven with a corresponding superset of test certificate sets $\bigcup_{m=1}^{n_{DC}} TC_m$ and compatible with DR_i . Then an AV shall only be allowed to drive in E_i if at least one set of proven driving capabilities DC_m matches the set of driving requirements DR_i . A match of DR_i and DC_m is determined with a matching process based on matching criteria.

Therefore, for matching to be possible, the following conditions must be met:

- For each set of driving requirements DR_i of a concatenated behavior space E_i , there is at least one compatible set of driving capabilities DC_m .
- The set of driving capabilities DC_m must be provable using tests in order to provide an associated set of test certificates TC_m for AVs.
- There are matching criteria for identifying matches between driving requirements of DR_i and driving capabilities of DC_m .

Consequently, driving requirements and driving capabilities must be defined in a way that allows for matching. This matching is performed based on matching criteria. Thus, the identification of drivable behavior spaces is enabled. Specifications for driving requirements and driving capabilities as well as matching criteria are developed below.

Specification of Driving Requirements

So far, only behavioral requirements have been derived based on the BSSD. These behavioral requirements can be used for matching with driving capabilities, but there is a problem with the level of abstraction. If only the behavioral information from the BSSD is used, potentially many driving requirements are identical, even though significantly different requirements are imposed on the dynamic driving task (DDT) and thus on the driving capabilities. A simple example of this is the behavioral requirements of the behavioral attribute reservation (cf. Figure 1). The requirements only demand that priority can be granted to certain types of road users from certain areas and that the granting of priority is indicated accordingly. These behavioral requirements potentially apply to many different intersections, since no reference to the geometry of the scenery has been made so far. Geometric information such as position, dimensions, or orientation of relevant areas would solve this problem. For example, right-before-left priority intersections with different relative orientations of their intersection arms would be distinguishable. Based only on the behavioral requirements, they would be the same.

Thus, the goal is to identify necessary geometry-based specifications for the driving requirements that form a basis for matching with driving capabilities. For this purpose, the relevant scenery properties are divided into specification categories. The specification category serves as a container for the concrete specifications of behavior spaces. For the demonstration of the methodical procedure of this work, the specification and subsequent work steps are performed and described on the basis of the reservation behavioral requirements. The reservation behavioral requirements are chosen because they have a high complexity compared to other behavioral requirements. Therefore, they are particularly suitable for demonstrating the procedure. The reservation behavioral requirements RR1 and RR1.1 (cf. Figure 1) are considered in the following from the DDT perspective referring to a considered behavior space with these requirements.

Probably the most obvious specification category is the type of traffic participant. This information already exists explicitly within the behavior space in the BSSD, but must still be included in the specification. Otherwise, it would not be explicitly specified that a corresponding driving capability must meet this specification. Different road user types require different capabilities of an AV, as they must not simply be recognized as a dynamic object, but must necessarily be classified according to the reservation as well. Reservation requirements necessitate this classification, as it determines which traffic participants are entitled to a reservation.

The speed limit of the traffic participants entitled to reservation is also relevant for the specification of the requirements. The capability to perceive these traffic participants must be tested and demonstrated based on different speeds of movement. It may well make a difference whether traffic participants are potentially approaching at 30 km/h or 50 km/h. In this case, a different behavior is demanded of the complete automation chain, which must be explicitly demonstrated. For this purpose, it is additionally necessary to include the speed limit of the AV before entering the behavior space under consideration. Different relative speeds between AV and the other traffic participants require explicit proof and therefore explicit specification for the same reason.

The perception of traffic participants entitled to reservation is only successful if they are also sensed in the relevant areas of the scenery. For this purpose, the direction of origin of the traffic participants entitled to reservation is explicitly stored in the BSSD. Based on the BSSD map, the absolute positions of the linked areas are thus available. However, only the direction of origin is indicated by the BSSD and not the entire area of origin potentially to be monitored. It is necessary for the specification to define the position of these relevant origin areas. Thereby, the information about the road course from the respective direction of origin should not be lost, so that potential paths along which reservation-authorized traffic participants move can be represented.

According to the above mentioned requirements an AV shall not travel through the considered behavior space if any other traffic participant with reservation-entitlement is present. This applies both to traffic participants who are already in the behavior space under consideration and to traffic participants who want to enter the space. In addition, if an AV is in the space itself, it must leave the space as soon as possible. Differences in the fulfillment of these requirements arise from the geometry of the considered behavior space. Different lengths of the behavior space mean different distances that the AV must travel through. But also different curved shapes of the behavior space possibly influence the driving behavior of the AV. Therefore, it is necessary that these geometric properties are part of the requirements specification.

In addition to the geometry of the behavior space under consideration, the geometry of the preceding behavior space(s) is also relevant. Depending on which curvature is present in the previously concatenated behavior spaces, for example, the AV will approach the considered behavior space with a different orientation. Depending on the orientation, the relevant perceptual areas for the relevant traffic participants differ. However, since based on the requirement specification it is not yet specified how exactly the vehicle aligns in the behavior spaces, the orientation of the AV cannot be part of the requirement specification. Rather, the orientation of the behavior spaces must be considered. The design and proof of the specific driving capabilities can thus be unrestricted, so that the actual orientation of the AV in the behavior space is defined in the development process.

Finally, if the behavior space under consideration is highly curved, as is the case with behavior spaces for turning, occlusion may occur. Depending on the position of potentially present planted areas, walls or buildings, the area to be monitored might not be completely visible or only visible at a late stage. If such occlusion is present, the driving behavior must be adjusted so that the reservation requirements are not violated. For example, depending on the type of occlusion, it may be necessary for an AV to slowly move into the considered behavior space so that the relevant areas can be observed. These cases have to be specified explicitly, since an extra proof has to be provided accordingly.

Overall, the following specification categories result for the reservation requirements RR1 and RR1.1:

- Type of relevant traffic participant
- Speed limit of relevant traffic participant
- Geometry and position of relevant area of origin
- Geometry and position of considered behavior space
- Geometry and position of relevant area of preceding behavior space(s)
- Speed limit of relevant preceding behavior space(s)
- Geometry and position of relevant area of occlusion

Specification of Driving Capabilities

What do the capabilities look like to enable matching? Simple driving capabilities that are tailored exactly to the driving requirements may be used. In this way, a capability always meets exactly one requirement. The following generic example illustrates the relationship between driving requirement and driving capability:

- Driving requirement: *The AV shall/ shall not perform a certain action under certain conditions.*
- Driving capability: *The AV is capable of performing/ not performing a certain action under certain conditions.*

The advantage of this very direct matching is that the capabilities fit the requirements in every case. There is no need for reasoning that assigns different capabilities to single requirements. This leads to the fact that the driving capabilities can be addressed by different vehicle-specific solutions. However, the proof of the capabilities must then be vehicle specific. In this way, driving capabilities are defined universally and uniformly without excluding specific technical solutions or developments. For this reason, this direct assignment appears not only intuitive but also practicable with regard to a uniform specification of driving capabilities.

The alternative to this approach is to further decompose the driving capabilities. It is possible to break down the driving requirements to subsets of the DDT. The result is a set of capabilities that contribute to the main capability

being met at the behavioral level. Various approaches are suitable for decomposing the capabilities. Basically, at the beginning of the decomposition, the decision has to be made how fine granular to decompose. In order to remain as abstract and generic as possible, the Sense-Plan-Act paradigm [12] is suitable, since it only provides for a decomposition of the DDT on three layers. For a more detailed decomposition, which allows a deeper analysis of the capabilities, the six layers of functional decomposition according to Amersbach and Winner [13, 14] or skill graphs introduced by Reschka et al. [15] are suitable.

However, these approaches have two fundamental drawbacks in terms of matching requirements to capabilities. First, the complexity of the functional relationships between the individual sub-capabilities increases with each step of the decomposition. Second, there is a certain arbitrariness in the choice of decomposed sub-capabilities. Both phenomena lead to the fact that selected capabilities can no longer be assigned to requirements in a simple way. Another problem is that the choice of partial capabilities often already assumes technical solutions, such as concrete vehicle setups. This would lead to the need to specify different capabilities for different AVs. However, the present approach aims to be as solution-neutral as possible and thus independent of a concrete vehicle specification. This increases the universal applicability of this approach. A solution to these problems is not known to the authors of this work. Nevertheless, if these problems are mitigated or even eliminated by appropriate approaches, capability decomposition would be another option for the matching process.

Due to the difficulties pointed out for the decomposition of driving capabilities, the method of driving capabilities analogous to driving requirements presented before is chosen in the present work. This is done by reformulating the requirements into capabilities as shown. In order to ensure that the comparison between driving requirements and driving capabilities contributes to a statement about the drivability of the behavior spaces, matching criteria must be defined that are as clear as possible. For this purpose, the aforementioned considered reservation requirements are chosen, which have a high potential for route-specific differences. Thus, the reservation requirements RR1 and RR1.1 are selected. For these requirements, the analogous reservation capabilities (RCs) are formulated and the derived specification categories are assigned:

- RC1: The AV is capable of avoiding obstructions of traffic participants with reservation entitlement for the behavior space.
- RC1.1: The AV is capable of indicating in advance by adjusting the driving speed reasonably that it will give priority to traffic participants who have priority.

Matching Criteria

For the matching between the driving requirements RR1 and RR1.1 and the driving capabilities RC1 and RC1.1, each specification category is considered individually. To ensure that the capabilities meet the requirements, the matching criteria of all specification categories must be met. To illustrate the reasoning, a X-intersection in a 30 km/h speed zone is considered in Figure 2. In Germany, the right-of-way rule "right-before-left" applies at such intersections. This means that traffic participants coming from the right from the perspective of a vehicle entering the intersection have priority. Additionally, left-turners must generally give priority to oncoming traffic. The *geometry and position of considered behavior space* as well as the *geometry and position of relevant area of occlusion* will be neglected in the following. The remaining specification categories are sufficient for an evaluation of the overall approach in a first implementation, since sufficient variations are to be expected. Additionally, areas reserved by pedestrians are not considered. The focus of the matching criteria is placed on areas of origin for motor vehicles, bicyclists and rail vehicles.

The specification categories of the reservation requirements have dependencies that must be considered in the nomenclature of the matching criteria. The following variables are introduced and partially shown in Figure 2a:

- For each externally-reserved behavior space E_i , there exist $n_{\text{orig},i} \in \mathbb{N}$ areas of origin $(A_{\text{orig},i,k})_{k=1,2,\dots,n_{\text{orig}}}$ (orange areas).
- Each area of origin $A_{\text{orig},i,k}$ is associated with a set of traffic participant types $P_{i,k}$.
- For the set of traffic participant types $P_{i,k}$ assigned to an area of origin $A_{\text{orig},i,k}$, there is a maximum speed limit $v_{\text{lim,orig},i,k}$ (speed limits of road user types within an area rarely differ).
- The relevant speed limit $v_{\text{lim,pre},i}$ of the AV for approaching the considered behavior space E_i is assigned to the relevant area of preceding behavior space(s) $A_{\text{pre},i}$ (blue area).

Type of traffic participant The matching criterion for the type of traffic participant is based on a nominal scale. This criterion is satisfied only if a successfully proven set of traffic participant types P_{proof} related to the associated proven area of origin matches the required set of traffic participant types $P_{i,k}$. The following matching criterion results:

$$P_{\text{proof}} = P_{i,k} \quad (\text{Equation 1})$$

Possible traffic participant types are *motor vehicle*, *pedestrian*, *bicyclist*, and *rail vehicle*. Therefore, the following closed set is defined for the road user types: $P \in \{\text{motor vehicle, pedestrian, bicyclist, rail vehicle}\}$.

Speed limit of relevant traffic participant For the speed limit of relevant traffic participant types, the maximum speed limit within the set of traffic participant types $P_{i,k}$ is chosen. The following assumption is made: If an AV has a successful proof of granting priority to a traffic participant coming from $A_{\text{orig},i,k}$ with a speed limit $v_{\text{lim,orig,proof}}$, then it is able to grant priority even with equal or lower speed limits $v_{\text{lim,orig},i,k}$ of traffic participants. The following matching criterion results:

$$v_{\text{lim,orig,proof}} \geq v_{\text{lim,orig},i,k} \quad (\text{Equation 2})$$

Speed limit of relevant preceding behavior space(s) For the speed limit within the relevant area of preceding behavior space(s) $A_{\text{pre},i}$, the maximum speed limit $v_{\text{lim,pre},i}$ among these behavior space(s) is chosen conservatively. The following assumption is made: If an AV has a successful proof of the required capabilities with a speed limit $v_{\text{lim,pre,proof}}$ for approaching, then the proof is valid even for equal or lower speed limits $v_{\text{lim,pre},i}$. The following matching criterion results:

$$v_{\text{lim,pre,proof}} \geq v_{\text{lim,pre},i} \quad (\text{Equation 3})$$

Geometry and position of relevant area of preceding behavior space(s) and relevant area of origin The matching criteria for the geometries and positions of the different areas are considered together. Basically, the relevant areas of origin must always be considered relative to the relevant area of preceding behavior space(s). This is because an AV approaches the considered behavior space within the area of preceding behavior space(s) and meanwhile already has to execute the DDT to grant priority. Accordingly, the relative positions of the relevant areas of origins to the AV's approach are crucial. One way of matching successfully proven and required combinations of the relevant areas is to superimpose the areas based on an equal reference system. Thus, an overlap of the matched areas can be identified. The same principle of this matching can be applied using a geometric parameterization of the relevant areas. The advantage here is a simpler and more efficient identification of the geometries of the areas as well as the matching itself. Therefore, with regard to the application of the matching criteria, a geometric parameterization of the relevant areas is performed. For this purpose, the following assumption is made: *The relevant road areas can be approximated by rectangles.*

Intersections are scenery components that predominantly contribute to externally-reserved behavior spaces. In urban areas, intersections and junctions are designed so that the associated road arms are straight with sufficient distance to the intersection. This must be taken into account during the design and construction of roads by ensuring that all intersection accesses (in the sense of sufficient distance) are identifiable in good time [16, p. 109]. The extensive implementation of this layout principle can be easily confirmed by looking at suitable aerial images, such as those from Google Earth [17]. Additionally, it is noticeable that the lane or road widths do not change significantly in the areas around the intersections. Based on these findings, the assumption made is retained. However, it must be assumed that there are exceptions that are not correctly represented due to this assumption (cf. Discussion of Results).

With this assumption, the following simplifications result:

- The alignment of the relevant areas is determined by longitudinal and lateral offsets and a constant angle relative to each other.
- The geometry of the relevant areas need only be specified by a constant width. The length is no longer needed, since if the alignment - and thus the course - of the area is known, the length along this course can be chosen according to the associated speed limits for a proof. The proof thus confirms the driving capability regardless of the length of the area.

Note 1: In the case of a non-rectilinear course of the areas, the length is relevant because the alignment changes along the course.

Note 2: This simplification does not apply to the identification of occlusion, since the complete geometry of the relevant area is required for this process.

Therefore, the relevant areas can be parameterized as shown in Figure 2b. Each area is characterized by a width:

- Width $w_{\text{orig},i,k}$ of relevant area of origin $A_{\text{orig},i,k}$
- Width $w_{\text{pre},i}$ of relevant area of preceding behavior space(s) $A_{\text{pre},i}$

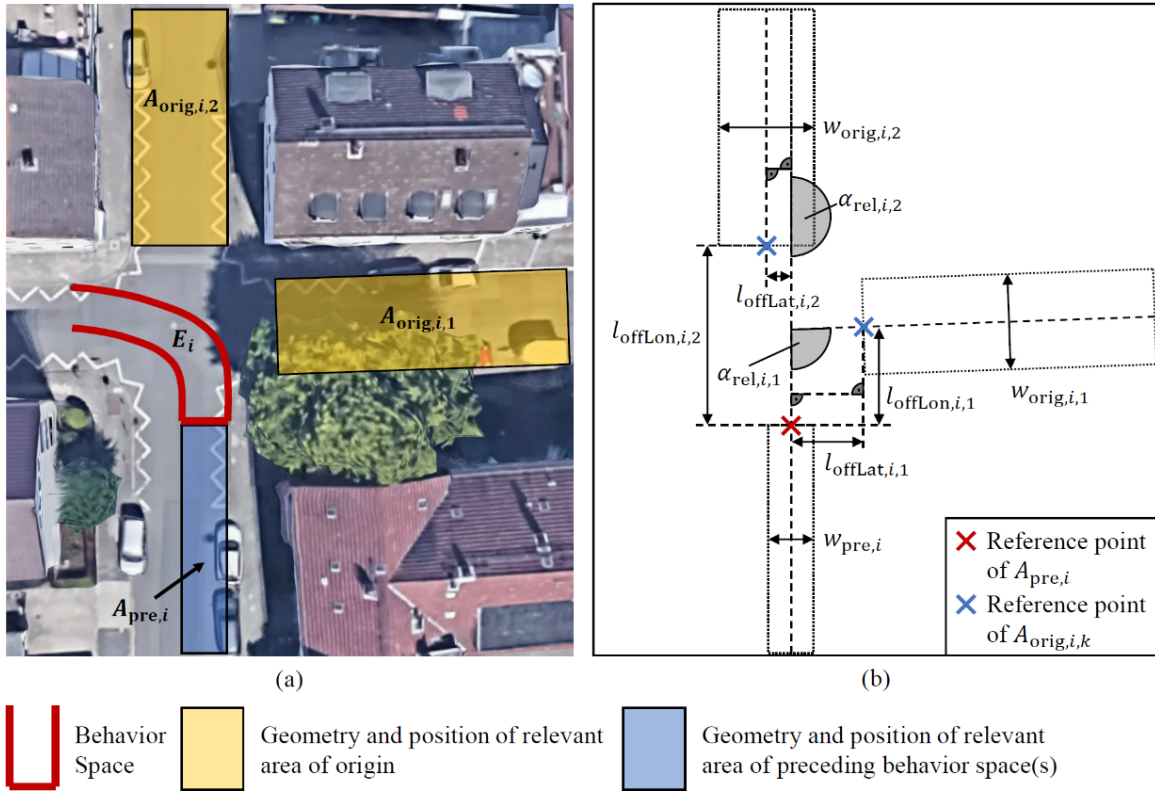


Figure 2. Specification for externally-reserved behavior spaces (Aerial image ©2022 Google/Aero West).

The alignment of the areas is defined using reference points and relative angles. The reference points result from the geometric analysis of the scenery based on the BSSD map. They mark the end of the areas in the direction of the intersection. They form the crossing point between the virtual center line of an area and the transition line to the intersection area. All areas are positioned relative to the reference point of the relevant area of preceding behavior space(s) (red cross). In this way, all relevant areas are represented relative to the approach area of the AV. For this purpose, the reference points of these areas (blue cross) are defined using the longitudinal and lateral offsets $l_{\text{offLon},i,k}$ and $l_{\text{offLat},i,k}$ relative to reference point (red cross) and virtual centerline of $A_{\text{pre},i}$. The orientation of the areas is defined by a relative angle $\alpha_{\text{rel},i,k}$ describing the angle between the virtual centerlines of $A_{\text{pre},i}$ and $A_{\text{orig},i,k}$. With the help of this parameterization, a matching is made possible, for which the matching criteria are defined below.

Basically, all parameters are matched. Only if all required parameters are within the tested parameter spaces (index: proof), the capabilities are considered sufficient. The following assumptions are made:

- *Widths of the areas:* If the AV has a successful proof for a certain width, then this proof is also valid for smaller widths. This assumption results from the consideration that a smaller area to be observed or driven on within the proven range is equally covered as a subset.
- *Offsets and relative angles between the areas:* If the AV has a successful proof for a range of values of these parameters, then this proof is valid for all values within this range. Also in this case, the different parameters within the range of values are a subset of the proven values.

The following matching criteria result:

- Width of $A_{\text{pre},i}$:

$$w_{\text{pre,proof}} \geq w_{\text{pre},i} \quad (\text{Equation 4})$$

- Width of $A_{\text{orig},i,k}$:

$$w_{\text{orig,proof}} \geq w_{\text{orig},i,k} \quad (\text{Equation 5})$$

- Longitudinal and lateral offset between $A_{\text{pre},i}$ and $A_{\text{orig},i,k}$:

$$l_{\text{offLon,proof,min}} \leq l_{\text{offLon},i,k} \leq l_{\text{offLon,proof,max}} \quad (\text{Equation 6})$$

$$l_{\text{offLat,proof,min}} \leq l_{\text{offLat},i,k} \leq l_{\text{offLat,proof,max}} \quad (\text{Equation 7})$$

- Relative angle between $A_{\text{pre},i}$ and $A_{\text{orig},i,k}$:

$$\alpha_{\text{rel,proof,min}} \leq \alpha_{\text{rel},i,k} \leq \alpha_{\text{rel,proof,max}} \quad (\text{Equation 8})$$

CAPABILITY-BASED ROUTING

In this section, the elaborated matching concept is used to identify capability-based routes. Capability-based routes are routes with driving requirements that do not exceed the driving capabilities of the AVs. Thus, the driving requirements of the scenery potentially lead to routes that deviate from conventionally planned routes. Therefore, a novel route search must be developed that takes into account this new criterion of *capability-exceeding* of AVs. Conventional route planners do not accomplish this so far, but are still suitable as a basis for this approach.

Concept of Capability-Based Route Search

A route planner searches for routes based on selected optimization criteria [18]. Conventional optimization criteria are the shortest or even the fastest route. For this purpose, the road network is typically divided into edges and nodes. Nodes are equivalent to intersections or junctions and represent the connection points of edges. Edges represent the individual road segments that are connected via the nodes. For the planning of routes within traffic networks, however, this division between nodes and edges is not always appropriate. In order to consider concrete turns or lane changes and thus lane-accurate paths within the network, this division is reversed. In so-called *edge based routing*, finer granular road sections (e.g. lane sections) become nodes and pairs of adjacent road sections become edges [19, p. 127]. In this way, lane-accurate route planning is enabled. Depending on the selected criterion, the edges of the road network are weighted with different costs based on a cost function. From a starting node to a destination node, there are different paths depending on the network size, alternating nodes and edges. As a result, the combination of edges with the lowest total cost represents the optimal route with respect to the selected criterion.

The BSSD road network does not necessarily consist of a graph that is suitable for routing. For the capability-based route search, a preprocessing is necessary to transform the BSSD road network into a graph. In this process, the lateral and longitudinal connections of the individual behavior spaces are explicitly represented as edges and the behavior spaces themselves as nodes. In this conversion process, the explicit BSSD information is lost. However, this loss of information is intentional, since the road network should be reduced to the minimum necessary information for efficient route search. With the help of a cost function the explicit BSSD information is transferred into the edge weighting. The result is a routing graph that enables explicit routing based on the weighted edges without exceeding the driving capabilities of AVs. For the identification of capability-based routes, a conventional route search algorithm is needed in addition to the weighted routing graph. This searches for the route with the lowest total costs within the weighted routing graph based on a starting point and a destination point. Since the route search algorithm is state of the art, the focus in this section is on the generation of the weighted routing graph. This is crucial for route identification since it defines the routing cost. The route search algorithm simply sums up the costs based on the routing graph and selects the edge combination with the minimum costs. The following modules are developed to create the routing graph:

- *Capability-Based Cost* module: Calculates the edge weights of the routing graph based on the matching results (further described in the following).
- *Matching* module: Matches driving requirements with existing driving capabilities based on the defined matching criteria (not further described in the following).
- *Requirement Generation* module: Generates the driving requirements of the concatenated behavior spaces based on the defined specification (not further described in the following).

The principle of finding an optimal route from origin to destination also applies to capability-based route search. A new optimization criterion is needed for the intended function of identifying routes that are feasible for AVs. This does not mean that the conventional criteria must be discarded. Even if a new criterion is used based on the new search function, the determined route should still be optimal with respect to conventional vehicle navigation. Accordingly, the route found should be the shortest or fastest possible despite further optimization criteria, for example. The basis of the new route planning should therefore be based on conventional route planning, so that these criteria and cost functions can be adopted. For the new criterion, however, a different or adapted cost function is required to weight the edges. In principle, there are two extreme forms of edge weighting. A weight can become minimal in the optimal case, i.e., theoretically assume a value of zero. The other extreme is an infinitely high weight assigned to edges that are maximally far from an optimum based on the evaluation of the cost function. Depending on the cost function, all other values are conceivable within these extremes. Negative costs are not allowed. From the previous findings, it is clear that the new cost function must be based on the previously presented matching of driving requirements and driving capabilities. Since the matching is performed for concatenated behavior spaces, it is suitable for *edge based routing*. The matching can either fail - driving capabilities are exceeded - or succeed - driving capabilities are not exceeded. Therefore, a cost function must be defined for the transitions between concatenated behavior spaces, following the matching concept.

The cost function to be defined must produce only two values. In the case of exceeding, the cost must be maximum so that the behavior space under consideration is excluded from the planning. In the case of a successful match, the cost must be minimal. In order to identify a successful match, the matching criteria of the specification categories of driving requirements and driving capabilities are applied. Thus, the matching cost function $c_{\text{match},i}$ for a concatenated behavior space E_i is defined as a function based on the set of driving requirements DR_i and all available, certified sets of driving capabilities $\bigcup_{m=1}^{n_{\text{DC}}} DC_m$ as follows:

$$c_{\text{match},i} = f(DR_i, \bigcup_{m=1}^{n_{\text{DC}}} DC_m) \quad (\text{Equation 9})$$

The matching cost function has the following range of values according to the considerations:

$$c_{\text{match},i} = \begin{cases} 0 & \text{for not exceeding the driving capabilities} \\ \infty & \text{for exceeding the driving capabilities} \end{cases} \quad (\text{Equation 10})$$

To avoid that the matching cost function $c_{\text{match},i}$ interferes with the conventional cost function $c_{\text{conv},i}$ in an undesired way, the cost functions have to be separated unambiguously. This means for the total cost function $c_{\text{tot},i}$:

- As long as driving capabilities are not exceeded, the total cost function $c_{\text{tot},i}$ is determined by the conventional cost function $c_{\text{conv},i}$.
- Once exceeding is identified, the total cost function $c_{\text{tot},i}$ is determined by the matching cost function $c_{\text{match},i}$.

Using the binary range of values of the matching cost function $c_{\text{match},i}$, these requirements can be realized via a simple addition of the cost functions. This results in the following total cost function $c_{\text{tot},i}$ for E_i :

$$c_{\text{tot},i} = c_{\text{conv},i} + c_{\text{match},i} \quad (\text{Equation 11})$$

Since the routing graph is generated and weighted in a preprocessing as described above, it is practical to remove edges with an infinite weight directly in this process. Thus, the route search algorithm does not have to visit these edges at all resulting in a more efficient calculation. This is taken into account in the implementation. For very large road networks with frequently updated data, an on-the-fly calculation of the edge weights within the iterations of the routing algorithm would also be suitable. This way, the entire road network would not always have to be preprocessed. Since the road network considered for the implementation in this work is rather small, this approach is not pursued further.

Implementation

The described concept of capability-based route planning is implemented based on the high-definition map framework Lanelet2 [20] for which a BSSD map instantiation was developed by Lippert et al. [9]. Besides providing a map format, another advantage of this framework is the availability of a comprehensive C++ software library for handling Lanelet2 map data [21]. The following software modules are used from this library as a basis for capability-based route planning:

- *lanelet2_core*: Basic module for handling Lanelet2 maps as well as all related primitives like points, linestrings, or lanelets (= lane sections). Extensive functions for geometric calculations are provided.
- *lanelet2_io*: Module for reading and writing Lanelet2 maps.
- *lanelet2_traffic_rules*: Module for interpreting selected traffic rules in Lanelet2 maps such as passability or speed limits of lanelets based on country and road user type. This module is used in this work as input for the generation of a routing graph, since traffic passable lanelets must be known for this purpose.
- *lanelet2_routing*: Module for route search within Lanelet2 maps. This module generates a routing graph based on conventional optimization criteria such as shortest or fastest path. Within the routing graph the optimal route is subsequently searched. It is possible to modify and extend the cost function as desired. To ensure that only passable lanelets are used from the point of view of traffic rules, the *lanelet2_traffic_rules* module is included in addition to the cost function to generate the routing graph. Thus, a basic function is already given, which generates routes that are correct from the traffic point of view based on the conventional optimization criteria.

For the handling of the BSSD data within the Lanelet2 framework the *BSSD data handler* is used, which was developed especially for this purpose. When reading the maps, this handler parses the BSSD data and makes it available to the software environment according to the generic BSSD structure [9]. In this way, arbitrary queries regarding all BSSD information are made possible. The *capability-based cost, matching and requirement generation* modules are implemented as part of the *lanelet2_routing* module.

Application in a Real-World Road Network

For the demonstration of capability-based route search based on a BSSD map, a road network specifically selected for this purpose is used. The road network covers large parts of Darmstadt's city center and includes both large, multi-lane roads and small residential streets. In doing so, the network connects specific intersections that were selected based on their different characteristics in terms of behavioral demands and geometry. These intersections therefore form the key component of the network for the following application.

Specific sets of reservation capabilities are first defined as the basis for the routing application. These capabilities are exemplary and serve to demonstrate the approach developed in this work. Since many intersections do not require priority to be granted when turning right, the capabilities are defined based on parameter values for straight and left turns. Therefore, when traveling on the road network under consideration, some intersections do not result in reservation requirements. However, as soon as an intersection is to be crossed straight ahead, it may be necessary to give priority from the right in 30 km/h speed zones. Therefore, the first capability set $DC_{\text{reserv},1}$ is designed to grant priority from the right ($\alpha_{\text{rel,proof,min|max}} = 80^\circ|100^\circ$). This set is valid for reservation-entitled road user types motor vehicle and bicyclist and speed limits of 30 km/h. As shown in the considered data, right is not necessarily equal to right. Thus, $DC_{\text{reserv},1}$ is extended by a second capability set $DC_{\text{reserv},2}$, which includes an additional angle range for traffic from the right ($\alpha_{\text{rel,proof,min|max}} = 50^\circ|70^\circ$). In order to use the road network even more efficiently, another capability set $DC_{\text{reserv},3}$ is defined, which specifically includes granting priority from the front ($\alpha_{\text{rel,proof,min|max}} = 170^\circ|190^\circ$). Compared to the previously defined capability sets, the longitudinal and lateral offsets differ. This is necessary because the offsets depend on the angular orientation of the areas. Last, the capability set $DC_{\text{reserv},3}$ is extended to a new set $DC_{\text{reserv},4}$ so that rail vehicles are covered in addition to motor vehicles and bicyclists. Additionally, the covered speed limits are increased to 50 km/h. The resulting driving capability sets including the defined specification parameters are summarized in Table 1.

Table 1. Specified reservation capabilities (MV: motor vehicle / B: bicyclist / RV: rail vehicle).

Parameter	$DC_{\text{reserv},1}$	$DC_{\text{reserv},2}$	$DC_{\text{reserv},3}$	$DC_{\text{reserv},4}$	Unit
$v_{\text{lim,pre,proof}}$	30	30	30	50	km/h
$w_{\text{pre,proof}}$	4	4	4	4	m
P_{proof}	{MV, B}	{MV, B}	{MV, B}	{MV, B, RV}	-
$v_{\text{lim,orig,proof}}$	30	30	30	50	km/h
$l_{\text{offLon,proof,min max}}$	3 10	3 10	10 30	10 30	m
$l_{\text{offLat,proof,min max}}$	3 10	3 10	-10 0	-10 0	m
$\alpha_{\text{rel,proof,min max}}$	80 100	50 70	170 190	170 190	°

Based on the defined driving capabilities, the capability-based route search is applied to the present road network. In order to demonstrate the functionality of the implemented route search, the start and destination points are defined with respect to diverse route options. For this purpose, the route planning should have different route options that require different driving capabilities. The following route requests are performed (driving capability set m corresponds to $DC_{\text{reserv},m}$):

- Route 1: Shortest route without driving capability sets
- Route 2: Shortest route with driving capability set 1
- Route 3: Shortest route with driving capability set 1 and 2
- Route 4: Shortest route with driving capability set 1, 2 and 3
- Route 5: Shortest route with driving capability sets 1, 2, 3, and 4

Figure 3 shows the road network with selected start and destination points and the calculated routes. For a clear representation of the different routes, the road network is shown twice. The presented road network is colored yellow and the detailed modeled intersections within it are highlighted in magenta. Start and destination points are shown with corresponding markers. As a reference for the calculated routes, the shortest route based on the conventional optimization criterion *shortest path* of the Lanelet2 algorithm is shown in green. The focus will be on the intersections modeled in detail, so *all other (non-magenta) intersections* in the road network will be handled *without* a capability-based matching. Route sections without the modeled intersections therefore always represent the shortest path based on the conventional optimization criterion.

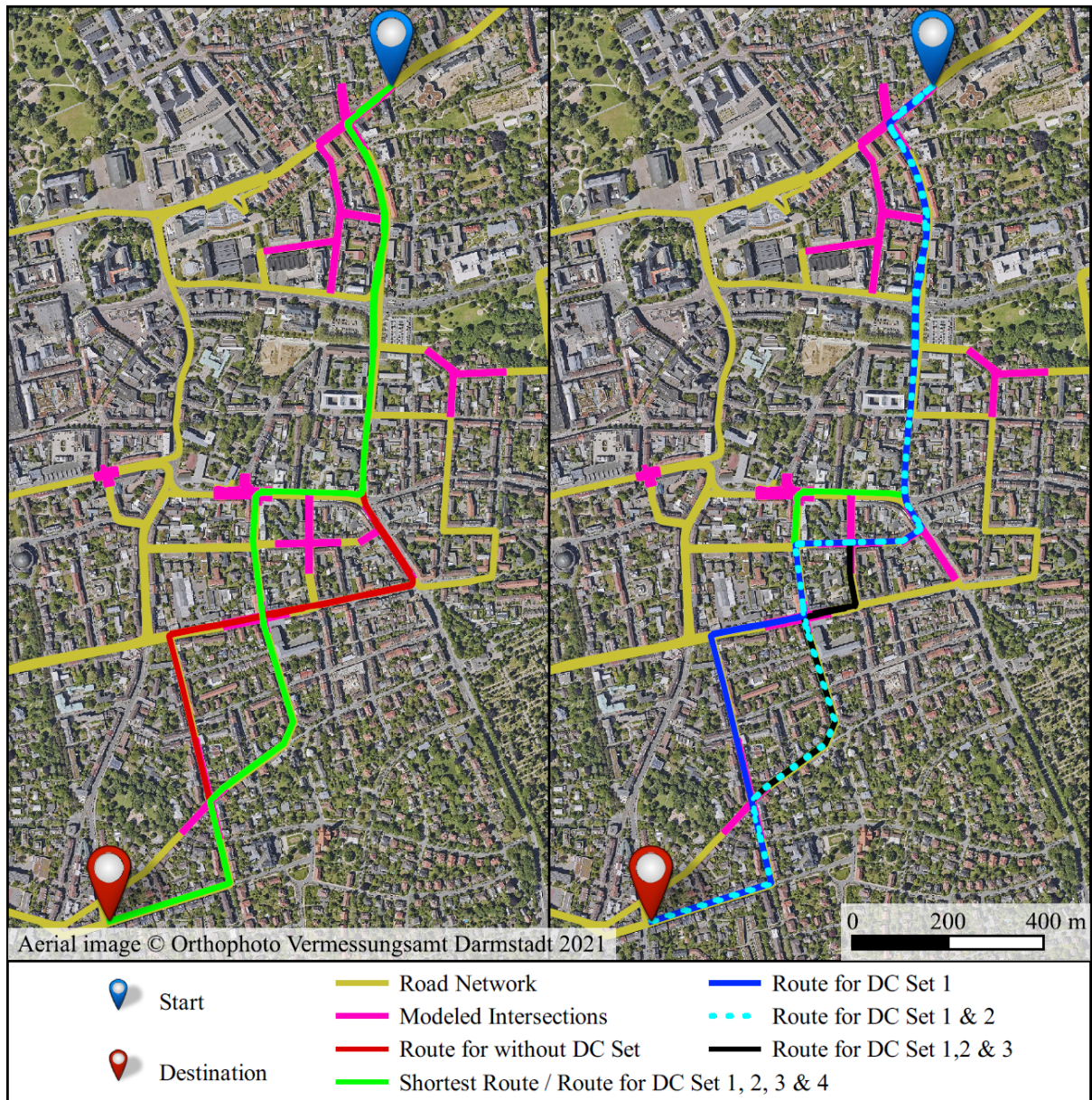


Figure 3. Routing results for the defined driving capability sets.

Route 1: This route is colored red and is partially below the green route as it follows the same course in these sections. The red route only contains behavior spaces that have no driving requirements in terms of reservation. Thus, all externally-reserved behavior spaces are excluded in the generated routing graph. The route therefore only passes through intersections where the AV has priority. At almost all intersections, routing is therefore straight ahead or to the right. Of particular note is the first intersection in the route where a left turn is made. This is possible without the appropriate capability sets because protected left turns are provided at this intersection (green turn arrow at the traffic signal). The last intersection on the route is in a residential area where the right-before-left rule applies. Nevertheless, it is allowed to cross this intersection because the intersection arm on the right in the direction of the route is a one-way street. Since the one-way street is for both motor vehicles and bicyclists and leads away from the intersection, no priority needs to be given.

Route 2: The blue route is based on driving capability set 1, which gives priority to motor vehicle and bicyclists in 30 km/h speed zones from the right. Therefore, the priority road within this route is exited into a residential area, as this path is shorter than that described in the red route. When turning right from the priority road, priority does not need to be granted and priority in the residential area is covered by the capability set. Finally, the residential area is exited by a right turn at an intersection with traffic signals. From this point on, the blue route again coincides with the red route.

Route 3: If the driving capabilities of the blue route are extended by capability set 2, an even shorter route succeeds. The dotted route colored in cyan avoids a detour at the end of the route by crossing the penultimate intersection with traffic lights straight ahead. This crossing requires no special capabilities, but a left turn is made at the intersection after it. However, the associated behavior space requires that priority can be granted from the right from a previously uncovered angular area. The new capability set 2 gained adds this missing angular range to the overall driving capabilities.

Route 4: If the previous driving capabilities are extended by set 3, the even shorter black route is found. The new capability set now allows driving in externally-reserved behavior spaces that require granting priority from the front. This enables left turns at many intersections. Accordingly, compared to Route 2 or 3, it is now possible for the AV to make a left turn in a residential area as well as at a major intersection. Despite the intersection being controlled by a traffic signal, the AV must yield priority from the front because left turns are not protected. Thus, although this intersection is geometrically similar to the first intersection in the route, different driving capabilities are required despite the traffic signal.

Route 5: With the driving capabilities available, Route 4 is not far from the shortest route shown in green. In order for the shortest route to be mastered by the AV, further driving capabilities are lacking. Part of the green route is on a road with rail traffic. The left turn required with this Start-Destination combination must also cross rail traffic. This means that in addition to motor vehicles and bicyclists, the AV must also give priority to rail vehicles from the front. Furthermore, both the associated behavior space and the linked areas of traffic participants with priority are in a 50 km/h speed zone. Driving capability set 4 covers these new requirements, so that this part of the network can also be driven on. If the AV possesses all defined capability sets, it is thus able to travel the shortest route.

Discussion of Results

With the application of the implemented route planning, it is shown that the identified routes are dependent on driving requirements and driving capabilities. Based on the reservation requirement specifications developed in this work, the route planning performs a comparison with the driving capabilities defined for the road network at hand. It is shown that the route planning only calculates routes with driving requirements that do not exceed the available driving capabilities. The performed variation of the available driving capabilities supports this result. Nevertheless, the demonstration initially only shows that capability-based routes can be identified based on the available specifications (for RR1 and RR1.1), but not whether these routes do not exceed the actual available driving capabilities of a real AV. Besides the omitted specifications for the remaining behavioral requirements, the following sources of errors and general uncertainties potentially prevent or complicate a real-world application of the approach.

Map-Based Errors A digital map is necessary for the derived approach. Regardless of map format, it is possible that the map data is incorrect. The data may contain inaccuracies with regard to *geometry*. These include simple offsets as well as shape errors or incorrect angles of individual map elements. Another source of errors is incorrectly *labeled data*. This primarily affects the validity of the behavioral demands of the BSSD, resulting in incorrect driving requirements, for example. In addition, the *actuality* of the map data is a potential source of error. A road network is often subject to many changes, such as road works. Changed scenery elements or temporary changes that are not stored in the map lead to errors of the approach presented here. In the road network modeled in this work, there are no noticeable geometry or label errors. This was verified using highly accurate and georeferenced aerial imagery. In addition to the aerial imagery, the accuracy of the labeled information was partially verified with the help of on-site inspections and publicly available map data. However, especially with regard to the actuality of the data, errors cannot be completely excluded.

Model-Based Errors The aforementioned sources of error related to the maps used are independent of the specification approach chosen in this work. However, errors also arise based on the models used. For the purpose of developing an initial implementation, the simple assumption was made that the areas to be specified (A_{orig} , A_{pre}) are rectangles. However, based on this assumption, errors may occur if the mentioned areas deviate significantly from a rectangular shape in reality. The routing results presented are based only on intersections where there is little to no deviation of the rectangle approximation from reality. For a higher accuracy of the approximation in general, it is possible to assume other shapes as a basis for the specification. However, it should be noted that the specification of driving requirements is always a trade-off. On the one hand, the real world should be approximated as closely as possible so that the specification is as close to reality as possible. On the other hand, the level of abstraction of the driving requirements should be high enough so that common test certificates can be created for similar behavior spaces. Thus, the closer the specification is to the real world, the more difficult it becomes to harmonize similar behavior spaces. Furthermore, a detailed modeled world requires many more

specification parameters than shown in this work. This is another drawback to modeling the real world in too much detail.

General Uncertainties In addition to the aforementioned sources of error that directly lead to a faulty specification, there are uncertainties in the overall approach that need to be critically questioned. Uncertainties refer to circumstances that may cause capability-based route finding to be infeasible in the manner presented in this work, regardless of implementation. A well-known problem in the field of automated driving is proving *completeness*. The approach described here is also affected by this, since it is not possible to identify with certainty whether potentially important information is lost during a development step. Thus, although the presented approach is systematically structured, it cannot be spoken of as being complete without further ado. The choice of *specification* represents another uncertainty. It has not been conclusively clarified whether the identified specification parameters are suitable, on the one hand, for mapping the real requirements of the routes and, on the other hand, for providing proof with test certificates. Finally, the *matching criteria* are subject to a number of assumptions. For example, for all specified widths of the different areas, it is assumed that the proof of a certain width automatically implies the proof of lower widths. It has not been demonstrated that the assumptions apply to a real-world proof. For this reason, not only the suitability of the specification, but also the suitability of the matching criteria must be proven for a realization.

OUTLOOK

The present work shows that the identification of capability-based routes is generally possible. This also implies that they can potentially be used for development, testing, and operation of safe automated vehicles. However, based on the discussion of the results, the following challenges need to be overcome before this approach may be used in real-world application.

Map creation and update: A correct map is essential for the realization of capability-based route search. Particularly important is the fidelity of the geometry, the labeled context knowledge (e.g., types of geometry elements), and the actuality of the data.

Proof of specification: The suitability of the identified reservation requirement and capability specifications must be proven (the same applies to yet missing specifications). This will require performing and evaluating tests in simulations and the real world. The specification is falsified if an AV successfully tested for a particular driving capability set is unable to drive a route segment with equally specified driving requirements. Based on the results, it may be necessary to adjust the specifications.

Proof of matching criteria: The suitability of the identified marching criteria must be proven (the same applies to yet missing criteria). This requires demonstrating that an AV with certified capability sets is actually capable of driving the route segments identified as non-exceeding. If route sections identified as passable exceed the driving capabilities of the AV, the matching criteria are falsified and must be revised.

Creation of test certificates: A general challenge becomes the creation and proof of test certificates for capability sets. The tests must prove that an AV actually possesses the capabilities that are certified using the test certificates. This challenge is closely coupled with the proof of specification and matching criteria.

ACKNOWLEDGEMENT

This research is accomplished within the project “UNICARagi” (FKZ 16EMO0286). We acknowledge the financial support for the project by the Federal Ministry of Education and Research of Germany (BMBF).

REFERENCES

- [1] Wachenfeld, Walther; Winner, Hermann: The Release of Autonomous Vehicles, in: Maurer, Markus; Lenz, Barbara; Winner, Hermann; Gerdes, J. Christian (PUB): Autonomous Driving: Technical, Legal and Social Aspects, Springer, 2016
- [2] International Organization for Standardization: ISO/TR 4804:2020: Road vehicles - Safety and cybersecurity for automated driving systems - Design, verification and validation, 2020
- [3] International Organization for Standardization: ISO 21448:2022: Road vehicles - Safety of the intended functionality, 2022
- [4] Society of Automotive Engineers: SAE J3016: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles, 2021
- [5] British Standards Institution: PAS 1883:2020: Operational design domain (ODD) taxonomy for an automated driving system (ADS) - Specification, 2020

- [6] Ulbrich, Simon; Menzel, Till; Reschka, Andreas; Schuldt, Fabian; Maurer, Markus: Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving, in: 2015 IEEE 18th International Conference on Intelligent Transportation Systems (ITSC 2015), pp. 982–988, 2015
- [7] Lippert, Moritz; Winner, Hermann: How to Derive Behavioral Requirements for Automated Driving from a Behavior-Semantic Scenery Description, in: 14. Uni-DAS e.V. Workshop Fahrerassistenz und automatisiertes Fahren: 09.–11.05.2022, pp. 217–229, 2022
- [8] Glatzki, Felix; Lippert, Moritz; Winner, Hermann: Behavioral Attributes for a Behavior-Semantic Scenery Description (BSSD) for the Development of Automated Driving Functions, in: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), pp. 667–672, 2021
- [9] Lippert, Moritz; Glatzki, Felix; Winner, Hermann: Behavior-Semantic Scenery Description (BSSD) of Road Networks for Automated Driving, URL: <https://arxiv.org/abs/2202.05211>, 2022
- [10] Shalev-Shwartz, Shai; Shammah, Shaked; Shashua, Amnon: On a Formal Model of Safe and Scalable Self-driving Cars, URL: <http://arxiv.org/pdf/1708.06374v6>, 2017
- [11] Junietz, Philipp; Wachenfeld, Walther; Klonecki, Kamil; Winner, Hermann: Evaluation of Different Approaches to Address Safety Validation of Automated Driving, in: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 491–496, 2018
- [12] Nilsson, Nils J: Principles of artificial intelligence, Springer Science & Business Media, 1982
- [13] Amersbach, C.; Winner, H.: Functional Decomposition, An Approach to Reduce the Approval Effort for Highly Automated Driving. In: 8. Tagung Fahrerassistenz, 22.-23. November, München, 2017
- [14] Amersbach, C.; Winner, H.: Functional Decomposition - A Contribution to Overcome the Parameter Space Explosion during Validation of Highly Automated Driving In: Traffic Injury Prevention sup1, Issues 20, pp. 52–57, 2019
- [15] Reschka, Andreas; Bagschik, Gerrit; Ulbrich, Simon; Nolte, Marcus; Maurer, Markus: Ability and skill graphs for system modeling, online monitoring, and decision support for vehicle guidance systems, in: 2015 IEEE Intelligent Vehicles Symposium (IV), pp. 933–939, 2015
- [16] Forschungsgesellschaft für Straßen- und Verkehrswesen: Richtlinien für die Anlage von Stadtstraßen (RASt), 2006
- [17] Google LLC: Google Earth, URL: <https://earth.google.com/>, 2022, visited on 07/14/2022
- [18] Bast, Hannah; Dellling, Daniel; Goldberg, Andrew; Müller-Hannemann, Matthias; Pajor, Thomas; Sanders, Peter; Wagner, Dorothea; Werneck, Renato F.: Route Planning in Transportation Networks, in: Kliemann, Lasse; Sanders, Peter (PUB): Algorithm Engineering: Selected Results and Surveys, Springer International Publishing, 2016
- [19] Dellling, Daniel; Sanders, Peter; Schultes, Dominik; Wagner, Dorothea: Engineering Route Planning Algorithms, in: Lerner, Jürgen; Wagner, Dorothea; Zweig, Katharina A. (PUB): Algorithmics of Large and Complex Networks: Design, Analysis, and Simulation, Springer Berlin Heidelberg, 2009
- [20] Poggenhans, Fabian; Pauls, Jan-Hendrik; Janosovits, Johannes; Orf, Stefan; Naumann, Maximilian; Kuhnt, Florian; Mayr, Matthias: Lanelet2: A high-definition map framework for the future of automated driving, in: 2018 IEEE Intelligent Transportation Systems Conference, pp. 1672–1679, 2018
- [21] FZI Forschungszentrum Informatik: Lanelet2 GitHub repository, URL: <https://github.com/fziforschungszentrum-informatik/Lanelet2>, 2018, visited on 11/01/2022