ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume IV-4/W8, 2019
14th 3D GeoInfo Conference 2019, 24–27 September 2019, Singapore

REQUIREMENT ANALYSIS OF 3D ROAD SPACE MODELS FOR AUTOMATED DRIVING

B. Schwab[1], T. H. Kolbe[2]

[1] AUDI AG, 85045 Ingolstadt, Germany – benedikt.schwab@audi.de
[2] Chair of Geoinformatics, Technical University of Munich (TUM), 80333 Munich, Germany – thomas.kolbe@tum.de

KEY WORDS: Automated Driving, Road Space, Street, Environment, Requirements, Sensors, CityGML, OpenDRIVE

ABSTRACT:

Automated driving has received a high degree of public attention in recent years as it will lead to profound changes in mobility, society and urban development. Despite several product announcements from automobile manufacturers and mobility providers, many questions have not yet been answered completely. The need of lane-level HD maps was widely discussed and has been the reason for company acquisitions. HD maps are tailored towards supporting the operation of an automated vehicle. However, the development of this technology also requires road space models, but with a completely different focus and level of detail. Therefore, this article investigates the system development and testing challenges of automated driving. Based on this, requirements of road space models for developing automated driving are derived and gaps to current standards are indicated.

1. INTRODUCTION

Automating the tasks of a driver has several potentials. Full automation will pave the way towards mobility-on-demand services, where a circulating fleet offers rides and overall costs are reduced due to high utilization rates. It shifts the vehicle interior to a living space, in which passengers can focus on other aspects than driving. First and foremost, the technology will enable the reduction of traffic accidents and subsequent fatalities. However, the development challenges not only automobile manufacturers and mobility provider companies, but also legislators and certification services. A main reason for this involves the exceptionally high number of different traffic scenarios that can occur on public roads. Since the simulation is conceived as a central element for testing the correct functioning of the system, the modeling of the vehicle's environment becomes inevitable. The objective of this paper is to analyze the requirements for road space models in the context of automated driving. Thus, the functioning and testing of automated driving systems is discussed and the main requirements for road space models are derived. Current standards and formats capable of modeling road spaces are examined and evaluated against those requirements.

2. AUTOMATED DRIVING SYSTEMS

Realizing technologies for automated driving requires representations of the vehicle's environment. However, the requirements for road space models very much depend on the objective to be achieved. A distinction has to be made between the operation and the testing of the developed technologies. Section 2.1 gives an overview of the functional components and Section 2.2 discusses the testing of them.

2.1 System Architecture

Automating the different tasks of a driver requires the development and implementation of several functional components. These components can be structured into the three main tasks *localization and map provision*, *environment and self perception* and *planning and control*, which are represented as columns in Figure 1. Each main task is further subdivided into three ver-

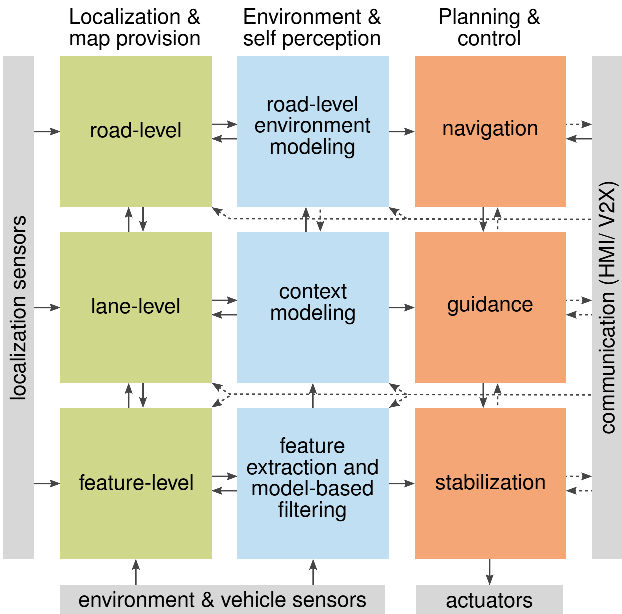


Figure 1. Functional system architecture of an automated vehicle proposed by Ulbrich et al., 2017

tical abstraction layers representing action planning (strategic layer, top row), maneuver decisions (tactical layer, middle row) and reactive action monitoring (operational layer, bottom row) (Ulbrich et al., 2017).

HD maps constitute not only the information base for routing algorithms to reach the desired destination, but also support the vehicle's interpretation of the situation. In order to provide map information about the vehicle's environment to subsequent components in the processing chain, the vehicle has to localize itself within its environment. Whereas road-level localization can be achieved by means of conventional GPS receivers with positioning errors of 10–20 m, lane-level localization requires an accuracy of around 1.5 m. Stabilizing the vehicle on the operational layer requires a quasi-continuous localization with an accuracy of approx. 10 cm. However, this challenges localization sensors

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-4-W8-99-2019 | © Authors 2019. CC BY 4.0 License.

in general and causes a problem in urban areas, in which occlusion and multi-path effects lead to significant positional errors. Therefore, the localization is supported by utilizing features like walls and poles, which are detected by the vehicle's sensors and are also represented in the HD map (Matthaei, 2015).

The vehicle monitors the environment with various sensors including cameras, LiDAR, RADAR and ultrasonic sensors. Subsequently, features are extracted from the raw sensor data and objects are recognized. This includes static objects like traffic signs and dynamic objects, such as other cars, pedestrians and cyclists. The information of the sensors is fused and extended with information from the HD map to model the context in which the vehicle is situated. On the strategic layer, road-level environment information like traffic flow is identified and provided to the other columns.

The third column involves the mission and route planning of the actual journey. On the tactical level, relevant objects comprised in the context model are selected and augmented with goals and values like labeling the relevancy of a pedestrian or plastic bag. Based on the situational information, the vehicle's behavior is planned by selecting maneuvers, such as overtaking or following another vehicle, and by generating target poses. The stabilization layer receives the set of target poses and has to plan an adequate trajectory accounting for restrictions like drivable areas or reference corridors. Low level controllers command the actuators to follow the trajectory (Ulbrich et al., 2017).

2.2 Testing

Validating and verifying the correct functioning of automated driving systems proposes a fundamental challenge. This is due to the high number of possible traffic scenarios, which arise from varying environmental conditions, different weather conditions, unusual and complex situations. Wachenfeld and Winner have estimated that testing an automated driving function for highways requires 6.6 billion kilometers of driving to statistically undercut the currently expected distance between two fatal accidents (Wachenfeld et al., 2016). Numbers of such magnitudes are impossible to achieve for different hardware and software versions by the means of real test drives, as depicted in Figure 2. In order to approve vehicles on a market, it is

(a) Testing in public

(b) Testing on proving ground

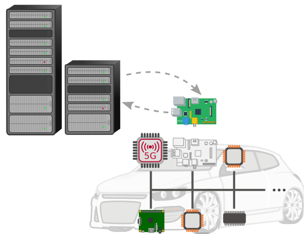
Figure 2. Vehicle-in-the-Loop testing in real environments[1]

mandatory to provide proof that the systems and components meet the respective requirements of national and international regulatory bodies. This process is referred to as homologation and already allows simulation for verifying assistance systems. It is not yet conclusively clear how the safety of autonomous driving systems is to be proven, but the environment simulation of the system under test will enable efficient system tests.

[1] source: AUDI AG

[2] source: `https://commons.wikimedia.org/wiki/File:Autohaus_Spaett_2008_5.JPG` by Slotosch is licensed under CC BY-SA 3.0

(a) Driver-in-the-Loop[2]



(b) Hardware-in-the-Loop

(c) Software-in-the-Loop

Figure 3. Testing within simulated environments

Everything-in-the-Loop (XiL) approaches, depcited in Figure 2 and 3, are utilized for virtual homologation and system testing during development (Riedmaier et al., 2018). Here, different components up to the complete system are exposed to different environments. Validation is a prerequisite for applying and testing with simulated environments, which are shown in Figure 3.

Virtual testbeds contain the relevant information to simulate the vehicle's environment and to provide ground truth for analyzing the performances of newly developed functions. Thus, a virtual testbed has to contain information for simulating road users, such as other cars, trucks, bicycles and pedestrians. In addition to the dynamic elements of traffic scenarios, scenery information is needed, which serves as static data basis for the road users. This could be the geometries of bicycle lanes or usage information about buildings for pedestrian simulation. Thereby, the objectives to be achieved with virtual testbeds can be of different nature. A virtual testbed can model either environments of reality or fictional design alternatives to reality—up to complete fiction. Once models are validated against reality, the static and dynamic elements of the testbed can be varied to cover potentially possible traffic scenarios. The automated driving system and its components are then exposed to critical traffic scenarios for system verification and validation (Menzel et al., 2018).

Sensors are the interface between the vehicle's environment and its internal system. Thus, valid modeling of the vehicle's sensors is a core prerequisite for adequate testing by means of simulation. As a consequence, the virtual testbed has to be enriched with information required for sensor simulation. Table 1 gives an overview of relevant sensors, which all measure electromagnetic waves in different spectra, with the exception of the ultrasonic sensor, which measures sound waves (Winner et al., 2016).

Generally sensor models can be differentiated into two main categories—object-based and physically based sensor models. Figure 4 illustrates the information processing stages of a real sensor and both categories of sensor models. Sensing can be further divided into the subsequent steps: measurement and processing. The measurement unit of an active sensor measures the signals transmitted by the sensor and reflected by the objects ϕ_n in the physical environment Φ. Distances are then estimated by the time of flight between transmitting and receiving the sig-

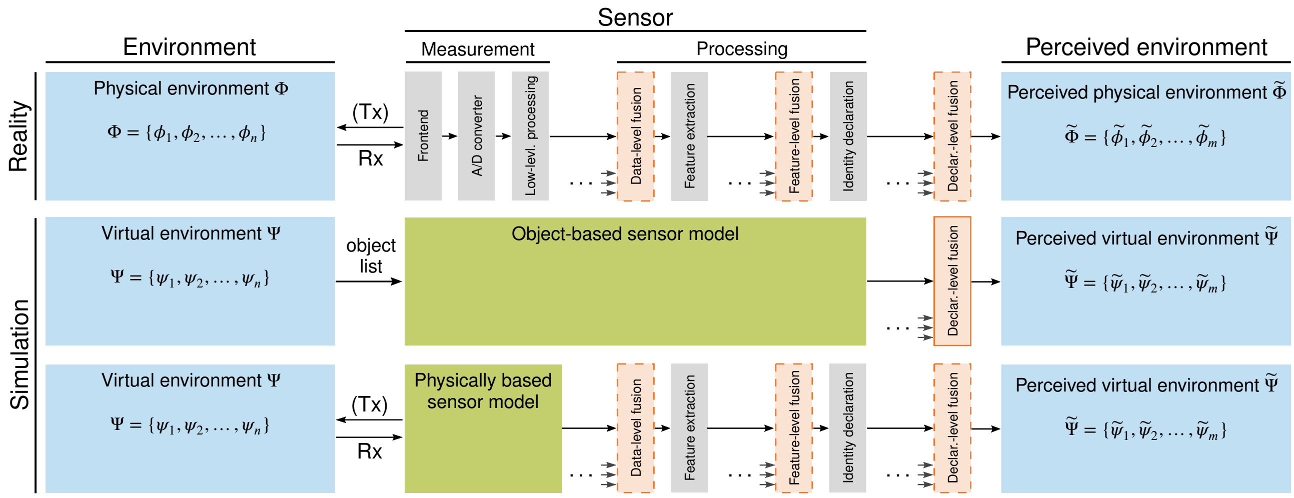


Figure 4. Architectural overview over a real sensor and the different types of sensor models, whereas the objects in the physical world Φ are denoted as ϕ_n and the objects in the virtual world Ψ are denoted as ψ_n

Sensor type	frequency	range
ENVIRONMENT		
Camera	430–770 THz (visible spec.)	300 m
	300–4300 GHz (infrared)	
LiDAR	330–350 THz	150 m
RADAR	24–24.25 GHz,	30 m (short),
	76–77 GHz	100 m (medium),
		250 m (long)
Ultrasonic	40–50 kHz	5 m
COMMUNICATION		
Car2X (802.11p)	5.855–5.925 GHz	300–1000 m
Cellular (5G)	0.6–70 GHz	
LOCALIZATION		
GNSS	1.1–1.7 GHz	

Table 1. Relevant vehicle sensors for automated driving (properties depend on model, manufacturer and country)

nal. Active environment sensors include RADAR, LiDAR and ultrasonic sensors. In contrast, passive sensors rely on externally emitted energy. For example, image sensors in cameras detect the amount of light received and process this information into pixel color values. The output of the measurement unit is referred to as sensor raw data. The subsequent processing unit uses the raw data to extract features. Then, objects are identified with bounding boxes, velocity vectors, classification probabilities and so on. Despite the fact that these steps are functionally separated, the measurement and processing units are usually located in the sensor housing. Thus, the information fusion over all installed environmental sensors takes place at object list level. However, there are two other basic architectural alternatives for fusing multisensor data, which are indicated in orange boxes in Figure 4: first, direct fusion of the sensors' raw data and second, the fusion of feature vectors (Liggins et al., 2009). Each fusion architecture poses advantages and disadvantages. Object list fusion requires significantly less communication bandwidth, since only objects and not sensor raw data need to be transferred from the sensors to the fusion unit. Furthermore, the fusion unit itself is computationally less intensive. On the other hand, fusion at sensor raw data level allows object identification algorithms to run on maximum available information. Centralizing processing also simplifies software updates as opposed to distributed sensor processing units, which are controlled by intellectual property rights of different sensor manufacturers. It is not yet clear which architectural approach will prevail. However, deep learning algorithms for object recognition utilize sensor raw data. Furthermore, dedicated chips for centralized processing have been introduced by well-known chip manufacturers.

Object-based sensor models receive the object list $\psi_1, \ldots, \psi_n$ of the virtual environment Ψ and return the perceived object list towards the fusion unit, as shown in the second row of Figure 4. An ideal object-based sensor model passes through the objects of the virtual environment, so that the perceived virtual environment $\widetilde{\Psi} = \{\psi_1, \psi_2, \ldots, \psi_m\}$ is a subset $\widetilde{\Psi} \subseteq \Psi$, since not all objects in the simulation are relevant to the perception system of an automated vehicle. Generally, test concepts for automated driving systems should enable the testing of subcomponents under perfect conditions and the testing of connected subcomponents to investigate interactional effects. Phenomenological sensor models use stochastics to model sensor behavior. These models include, for example, Gaussian noise models for errors in the position measurement of other road users (Hanke et al., 2015). The standard ISO 23150 is currently being developed to describe the logical interface between sensors and the fusion unit for automated driving. In order to facilitate the connection of simulation environments and automated function development frameworks, the project Open Simulation Interface (OSI) addresses the open source implementation of the emerging ISO standard.[3] Since this communication and its implementation describe what an automated driving system can perceive at object-level, it also lists the objects and attributes required to be modeled in a virtual testbed.

Object-based sensor models allow the covering of the main sensor characteristics, are parameterizable and real-time capable. However, object-based sensor models are limited by the ability to model sensor effects, such as ghost targets, the detection of hidden targets due to multipath propagation or combinations thereof (Schick et al., 2017). Hirsenkorn et al. state that object-based statistical models can be realistic for simpler environments, but have difficulties to achieve realistic results in more complex environments, such as urban situations (Hirsenkorn et al., 2017). Physically based sensor models attempt to capture these effects by simulating the propagation of waves within the virtual environment. This means modeling physical effects of

[3] github.com/OpenSimulationInterface

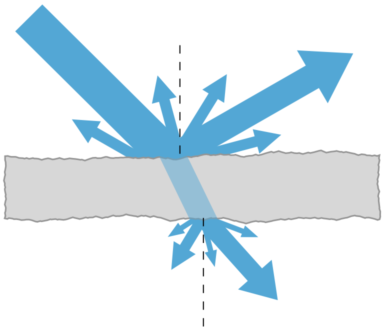


Figure 5. Light interacting with rough material

wave propagation, such as reflection, refraction and scattering, as illustrated in Figure 5. Therefore, physically based sensor models require material and surface information about the objects represented in the virtual testbed. Furthermore, the description of and information about the materials should be designed towards supporting the different sensor types, which are listed in Table 1. Thus, objects located in the range of the sensor become relevant to be modeled in virtual testbeds.

3. ROAD SPACE MODELING STANDARDS

Modeling road spaces is part or purpose of various standards and formats. Application areas involve civil engineering, surveying, driving simulation, as well as maps for navigation, city models and also game engines. The objective of the standards presented below is to provide an overview of relevant model categories and to indicate their characteristics for discussing the requirements. Therefore, the list given here is not exhaustive and comprises only a selection of relevant standards.

3.1 Maps for Navigation

Maps for navigation systems have entered the automotive mass market in the 2000s. Due to the need of lane-level maps, the data models of traditional maps are being refined.

3.1.1 Navigation Data Standard: This standard was first introduced in 2012 and was developed to completely separate the navigation data from the application. It is structured in building blocks, which include not only routing and basic map display, but also orthoimages and speech. A SQLite database is used to manage and access the data. In order to meet the increased needs of automated driving, a lane building block with lane geometries, boundary and road marking representations has been added. The specification of the Open Lane Model (OLM) has been published with API implementations and a database inspector[4] (Navigation Data Standard e.V., 2016).

3.1.2 HERE HD Live Map: The HERE HD Live Map is developed by the company HERE and aims to provide self-healing mechanisms using the sensors of automated vehicles for updating. In 2018, the OneMap Alliance was founded by HERE, SK Telecom, NavInfo and Pioneer Corporation. This consortium plans to offer a global, consistent and dynamic HD map according to the HERE HD Live Map specification (HERE Technologies, 2018). The HD Live Map comprises a *road centerline model* and a *lane model*, whereas each of them is enriched with attributes grouped in different layers. The layers include attributes for speed, routing and dedicated automated driving attributes, which detail the road geometries with parameters. Google's Protocol Buffer data format is utilized to describe the schema of the data model (HERE Global B.V., 2018).

[4] olm.nds-association.org

3.2 Driving Simulation

Traffic simulation is used for various fields of application, such as planning transportation infrastructure and analyzing traffic flow dynamics. Thereby, the application fields require differently detailed traffic simulations, which can be classified into macro-, mesos-, micro- and submicroscopic levels and require differently detailed road space models. Submicroscopic vehicle simulators are utilized for testing automotive related components and functions in several development stages.

3.2.1 OpenDRIVE & OpenCRG: The OpenDRIVE standard logically describes road networks and was originally developed by the company *Vires Simulationstechnologie GmbH*. It serves as the input format for their submicroscopic vehicle simulator *Virtual Test Drive* and was transferred to the Association for Standardization of Automation and Measuring Systems (ASAM) in 2018. ASAM offers a platform to further develop the standard with members from academia, OEMs, first tier suppliers and tool manufacturers. A right-handed inertial coordinate system according to ISO 8855 is used and can be georeferenced with a PROJ.4 string. Roads are modeled by means of reference lines, which are compound by geometric primitives and allow modeling with continuos curvatures. Road side objects, road signals, road markings, superelevation, etc. are referenced relative to the reference line in a track coordinate system. Lanes are also added relative to the reference line and can be assigned with attributes, such as material, speed, access information. Junctions are modeled via linking the predecessors and successors of roads and their corresponding lanes (Dupius et al., 2019). However, there exists no open source libraries for the OpenDRIVE standard.

OpenCRG is a complementary format to OpenDRIVE and models detailed road surfaces, which might be needed for tire, vibration or driving simulations for example. A curved regular grid is utilized to model the road surface along reference line. Open source tools in Matlab and C are made available for creating and evaluating OpenCRG-datasets.[5]

3.2.2 RoadXML: RoadXML also describes road networks for driving simulator applications and is the result of combining proprietary formats, which were originally developed by OEMs, research and other industry partners. Its documentation and data schema were published in version 2.4.1. in 2016. The driving simulator *SCANeR™ studio* by *AVSimulation* reads the format RoadXML and is used for in-the-loop testing setups by OEMs and suppliers (AVSimulation, 2019). RoadXML offers four main information layers—a topological, logical, physical and visual layer. A RoadXML file is structured as a patchwork of sub-networks consisting of tracks and intersections. The tracks are modeled by stringing together geometric primitives in the XY-plane with continuos curvatures. Road signs with externally referenced 3D objects can be clipped along the track. Lanes, lane borders and lane markings are described as profiles and linked to the tracks (Ducloux, 2016).

3.3 City Modeling – CityGML

CityGML is an open standard for modeling, storing and exchanging semantic 3D city models. The standard is issued by the

[5] opencrg.org

Open Geospatial Consortium (OGC) and implemented as an application schema of the Geography Markup Language (GML). It defines classes and their relations of the most relevant objects present in city and landscape environments. Thereby, CityGML models objects with respect to their geometrical, topological, semantical and appearance properties (Kolbe, 2009). Furthermore, open source libraries and database solution exist to read, process and store city models. CityGML comprises a concept to model different Level of Details (LoDs). It is structured in thematic modules and contains a dedicated transport model. The main class *TransportationComplex* is thematically specialized into the subclasses *Track*, *Road*, *Square* and *Railway*. Transportation complexes are geometrically modeled by line objects within a linear network in the coarsest LoD0. In LoD2-4 the transportation complex class is thematically subdivided into traffic areas and auxiliary traffic areas with geometrical multi surface representations. The class *TrafficArea* allows the modeling of lanes, which can be attributed by their function, and the class *AuxiliaryTrafficArea* allows the modeling of kerbstones and road markings for example. Beil and Kolbe have proposed a refined and more detailed transportation module, which *inter alia* enables centerline representation at all LoDs (Beil and Kolbe, 2017). If specific applications require the modeling of additional information, CityGML supports Application Domain Extensions (ADEs), which are extension schema based on the CityGML schema definitions.

3.4 3D Graphics – Filmbox

Filmbox (FBX) is a proprietary file format by the company Autodesk and is used to exchange 3D scenes. The format can be serialized as binary or ASCII file and processed in C++ and Python by means of the FBX SDK that is also provided by Autodesk. Furthermore, current game engines like Unity or the Unreal Engine support the importing and exporting to this format. 3D scenes are structured as graphs containing a collection of nodes, which are organized in a tree structure and can be efficiently traversed. Nodes include light sources, cameras and meshes for example. Since scene graphs are optimized towards graphical application, they mainly contain systematic semantics that are related to the graphics domain (Autodesk, Inc., 2019).

3.5 Further Standards

Further standards include Geographic Data Files (GDF), INSPIRE, LandInfra, OpenStreetMap and standardized national catalogues for consistent road provisioning, such as Anweisung Straßeninformationsbank (ASB) and Objektkatalog für Straßen- und Verkehrswesen (OKSTRA) for Germany. These standards are discussed by Beil & Kolbe and Labetski et al. in more detail (Beil and Kolbe, 2017, Labetski et al., 2018).

4. REQUIREMENTS

Modeling road spaces can be achieved in different ways, while applications define the requirements for them. In the case of automated driving, several requirements have been identified and grouped into four main categories, which will be further discussed in the subsequent subsections. The category *ground truth* involves information the automated driving system is attempting to understand about its environment. In order to test the performance of landmark localization functions, a ground truth of the respective landmarks is required for benchmarking. The second category *environment simulation* groups requirements that are necessary to simulate other road users like cars, pedestrians or bicycles. The third and fourth groups deal with the different ways in which a vehicle can perceive its environment in simulation. Whereas *object-based sensor simulation* lists the required objects for modeling sensors at object level, the group *physically based sensor simulation* comprises the required objects predominately occurring within sensor sight.

Table 2 gives an overview of the identified requirements and their respectively required geometrical, topological, semantical and appearance attributes. Furthermore, the suitability of the standards, discussed in section 3, is compared to the listed requirements. However, an in-depth evaluation would require access to the data models of each standard.

4.1 Ground Truth

As every functional system component, discussed in section 2, aims to either understand the traffic situation or base decisions on it, requirements 1–6 regard ground truths for testing and benchmarking these components and functions.

1. *Road-level linear graph* with attributes for routing (e.g. access restrictions, road types and live information for traffic congestion) for navigation (strategic layer).
2. *Representation of landmarks* (e.g. pylons, traffic signs and fences) for feature-level localization (operational layer).
3. *Surface-based lane representations* of possibly passable areas for behavior generation (e.g. lane changing, following) and target pose generation with constraints (tactical layer). For this purpose Bender et al. have introduced the concept of "lanelets", which are atomic and efficiently computable lane segments (Bender et al., 2014). Lanelets are bound by a left and right polyline, in which traffic rules do not change. As shown in Figure 6, lanelets can be utilized

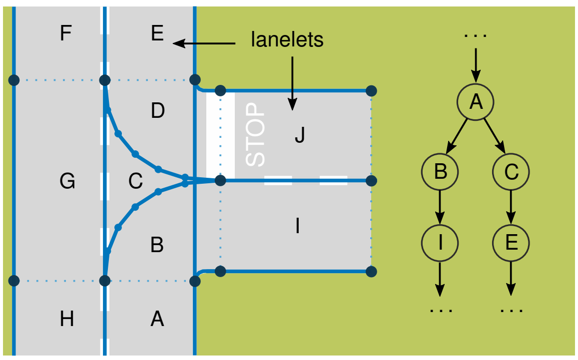


Figure 6. Surface-based representation as lanelets of a road crossing

for lane-level navigation. Beil and Kolbe have proposed a similar approach for a refined transportation model in CityGML (Beil and Kolbe, 2017).
4. *Traffic rule information* by either attributing or preferably referencing traffic infrastructure objects (e.g. traffic signs, traffic lights) to their applicable lanes or stopping lines for situation assessment (tactical layer). A road space model requires the capability to enrich roads, lanes and lines with rule-related information. This includes speed limits, priority rules and maximized permitted vehicle measures, but also conditional rules like time- or weather dependent speed limits. Rule-based traffic objects should be classified in a manner that allows an international interpretation.
5. *Lane boundaries and stopping lines* with classifications for behavior generation (e.g. lane changing, stopping due to red traffic light) and target pose generation with constraints

https://doi.org/10.5194/isprs-annals-IV-4-W8-99-2019

	Required information				Navigation		Driving Simulation		City Modeling	3D Graphics
	Geom.	Top.	Sem.	App.	NDS OLM ver. 1.0	HD Live Map as of 2019	OpenDRIVE ver. 1.5	RoadXML ver. 2.4.1	CityGML ver. 2.0	FBX 2019
GENERAL										
Developer					NDS Association	HERE Technologies	ASAM	RoadXML board	OGC	Autodesk
Schema language					SQL	Protocol Buffers	XML	XML	GML	ASCII
Type					closed standard	privately controlled API	open standard	open standard	open standard	proprietary format
GROUND TRUTH										
1. Road-level linear graph	2.5D	◪	◪	□	+	+	~	~	~	✗
2. Representation of landmarks	3D	■	□	□	~	~	~	~	+	+
3. Surface-based lane representations	3D	■	■	□	~	~	✗	✗	+	✗
4. Traffic rule information	□	■	■	□	+	~	+	+	✗	✗
5. Lane boundaries and stopping lines	3D	■	■	□	+	+	+	+	~	~
6. Road markings	3D	■	■	□	~	~	+	+	~	✗
ENVIRONMENT SIMULATION										
7. Parametric lane-level representations	3D	■	■	□	~	~	+	+	✗	✗
8. Traffic areas used by other road users	3D	■	■	□	~	~	~	~	+	✗
9. Attributes for OD matrix estimations	□	■	■	□	✗	✗	✗	✗	+	✗
10. Attributes for ageing processes	□	□	◪	■	✗	✗	✗	✗	~	~
OBJECT-BASED SENSOR SIMULATION										
11. Lanes	3D BB & orien.	◪	■	◪	~	~	+	+	+	✗
12. Lane boundaries		◪	■	◪	~	~	+	+	+	✗
13. Traffic signs		◪	■	◪	~	~	+	+	✗	✗
14. Traffic lights		◪	■	◪	~	~	+	+	✗	✗
15. Road side objects (e.g. landmarks)		◪	■	◪	~	~	+	+	+	✗
PHYSICALLY BASED SENSOR SIMULATION										
16. Lanes	3D	◪	□	■	✗	✗	✗	✗	~	~
17. Lane boundaries	3D	◪	□	■	✗	✗	✗	✗	~	~
18. Traffic signs	3D	◪	□	■	✗	✗	✗	✗	~	~
19. Traffic lights	3D	◪	□	■	✗	✗	✗	✗	~	~
20. City furniture	3D	◪	□	■	✗	✗	✗	✗	~	~
21. Vegetation (e.g. trees, bushes)	3D	◪	□	■	✗	✗	✗	✗	~	~
22. Buildings (esp. facade)	3D	◪	□	■	✗	✗	✗	✗	~	~
23. Tunnels, bridges	3D	◪	□	■	✗	✗	✗	✗	~	~

Table 2. Evaluation of the automated driving development requirements to the road space models, whereas + denotes suitability, ~ somewhat suitable, ✗ not directly suitable

(tactical layer). Lane boundaries are usually marked lines, whereas the line type allows or restricts maneuvers like lane changes. However, lanes can also be bounded by curbs, structures, gravel, etc. or be completely invisible. Labetski et al. have also suggested to model stopping lines in CityGML (Labetski et al., 2018).

6. *Road markings* with classifications for feature-level localization (operational layer) and context modeling (tactical layer). As stated in the latter requirements, road markings are used to communicate lane boundaries and can contain textual or symbolic traffic rules painted on the surface. However, road markings can temporarily be not valid in terms of their rule applicability as in the case of construction works. Furthermore, road markings can contain navigational suggestions or attention warnings.

4.2 Environment Simulation

The requirements 7–10 consider information necessary to simulate the more dynamic elements of the vehicle's environment.

7. *Parametric lane-level representations* with continuous curvatures for submicroscopic driving simulations. Road space models for submicroscopic driving simulations usually follow a parametric modeling approach. The standard OpenDRIVE uses lines, clothoids, arcs and (parametric) cubic curves for constructing a road's reference line, as shown in Figure 7. This enables the modeling of roads with continuous curvatures and also adheres to the process of planning. Continuous curvatures avoid abrupt changes of the steering angle during driving and thus increase driving safety. The elevation profile, superelevation and crossfalls are described with polynomial functions of third order in OpenDRIVE. Moreover, lane widths and borders are also modelled with polynomial function relative to the reference line (Dupius et al., 2019). Since road network models for submicroscopic vehicle simulators follow this parametric approach, either a conversion towards the supported formats or extended read-in capabilities of simulators are required.

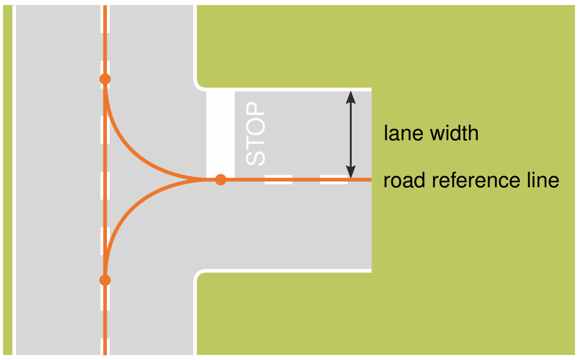


Figure 7. Parametric representation of a road crossing

8. *Traffic areas used by other road users* (e.g. pedestrians, cyclists and trams) for context modeling (tactical layer) and simulating route choices of road users.
9. *Information for OD matrix estimations* (e.g. traffic countings, number of inhabitants and usage of buildings) for simulating destination choices and activity sequences of road users.
10. *Attributes for ageing processes* (e.g. washed out lane markings, curved pole of a traffic sign) for simulating the variety of different environment conditions.

4.3 Object-based Sensor Simulation

To support object-based sensor simulation all objects and attributes have to be represented, which are possibly communicated between the sensors and the sensor fusion unit. As discussed in subsection 2.2, the approach of the OSI project is to implement standardized interfaces for object-based sensor models and thus serves as indication of relevant objects. Requirements 11–15 in Table 2 list the most relevant objects. Hereby, the objects are geometrically modeled as a 3D Bounding Box (BB) or base polygon with an orientation. Attributes involve coarse material, density and color descriptions.

4.4 Physically based Sensor Simulation

In the case of physically based sensor models, every object within sensor sight becomes relevant, as sound and electromagnetic waves can be reflected by those objects. Thus, city furniture, vegetation and also buildings have to be modeled in addition to the objects listed for object-based sensor models. However, the geometrical requirements exceed bounding box simplifications, since the object's 3D shape influences the wave propagation. Furthermore, objects have to be enriched with material information in order to model the appearances to the different sensors listed in Table 1.

5. DISCUSSION

Clearly, a HD map for supporting the operation implies a different set of requirements compared to development and testing. This is not only the case from a technical perspective, but also from an economic perspective. Since highly accurate and thematically rich mapping constitutes a costly endeavor, automated driving systems designed for area-wide operation should only require *a priori* information that can be mapped in an economically feasible way on a broader scale. A mapping process involving manual labour is not viable due to the scale and update rate in the long term. The reasoning stands in stark contrast to the situation encountered when developing and testing automated driving systems. Here, every aspect, which needs to be tested, will benefit from environment simulation, as it offers reproducible, early and variable testing. Especially the efficiency obtained by simulation justifies high costs in the building up of virtual environments, since real test drives are not feasible due to the extensively high number of traffic scenarios.

However, a virtual testbed only provides effective testing, if sensor models and road user models are valid. In order to validate those models against reality, virtual testbeds of real road spaces are required. The building up of such virtual testbeds enables the gradual shifting of tests, which are currently performed in reality, towards simulation. Moreover, virtual testbeds of real environments can facilitate method development, such as translating measurement data from vehicles into machine-readable scenario descriptions for resimulation or supporting the labelling of training data sets. Since standards originating from driving simulation applications are focusing on roads and traffic infrastructure, their data models are very limited regarding the more extended road space. But the extended 3D road space becomes particularly relevant for physically based sensor simulation. Therefore, semantic city model standards like CityGML are suited to complement driving simulation standards, as they can provide the information basis for the simulation of other road users and ageing processes. Furthermore, city models could be enriched with material information necessary for physically based sensor simulation. This in turn, could enable the testing of different sensor configurations and consequences on automated driving functions in the long term.

6. CONCLUSION AND FUTURE WORK

This article discussed the approaches of how an automated driving system functions and its testing challenges. Thereby, the main focus was to derive requirements for road space models from a development perspective, as they are inherently different and more comprehensive compared to HD maps. In order to model the vehicle's sensors, different approaches can be followed, whereas physically based models promise to cover sensor behavior that object-based models cannot capture. Despite the complexity of simulating sensors in virtual testbeds, simulation offers a way for efficient testing compared to test drives in reality. Therefore, future work will focus on transforming currently used HD maps to sematic 3D city models, i.e. CityGML. Furthermore, the enriching of 3D city models with information required for physically based sensor simulation constitutes a further field for future work. Another open question involves the surveying and acquiring of the additionally required information.

ACKNOWLEDGEMENTS

This work is supported by the German Federal Ministry of Transport and Digital Infrastructure (BMVI) through the *Automated and Connected Driving* funding program under funding code 16AVF2145A (SAVe).

REFERENCES

Autodesk, Inc., 2019. FBX SDK. `https://help.autodesk.com/view/FBX/2019/ENU` (May 2019).

AVSimulation, 2019. SCANeR studio. `https://www.avsimulation.fr` (May 2019).

Beil, C. and Kolbe, T. H., 2017. CityGML and the Streets of New York – A Proposal for detailed Street Space Modelling. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* IV-4/W5, pp. 9–16.

Bender, P., Ziegler, J. and Stiller, C., 2014. Lanelets: Efficient map representation for autonomous driving. In: *Intelligent Vehicles Symposium (IV 2014)*, IEEE, Dearborn, MI, pp. 420–425.

Ducloux, P., 2016. [RoadXML 2.4.1] Road Network Description XML Format Specification. `https://www.road-xml.org` (May 2019).

Dupius, M., Hekele, E. and Biehn, A., 2019. OpenDRIVE – Format Specification, Rev. 1.5. VIRES Simulationstechnologies GmbH. `http://opendrive.org` (May 2019).

Hanke, T., Hirsenkorn, N., Dehlink, B., Rauch, A., Rasshofer, R. and Biebl, E., 2015. Generic architecture for simulation of ADAS sensors. In: *16th International Radar Symposium (IRS 2015)*, IEEE, Dresden, Germany, pp. 125–130.

HERE Global B.V., 2018. HD Live Map – Data Specification – Protobuf Version 2018.9. `https://developer.here.com/automotive/documentation` (May 2019).

HERE Technologies, 2018. OneMap Alliance – A global mapping system for autonomous vehicles. `https://www.here.com/vision/onemap-alliance` (April 2019).

Hirsenkorn, N., Subkowski, P., Hanke, T., Schaermann, A., Rauch, A., Rasshofer, R. and Biebl, E., 2017. A ray launching approach for modeling an FMCW radar system. In: *18th International Radar Symposium (IRS 2017)*, IEEE, Prague, Czech Republic, pp. 1–10.

Kolbe, T. H., 2009. Representing and Exchanging 3D City Models with CityGML. In: J. Lee and S. Zlatanova (eds), *3D Geo-Information Sciences*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 15–31.

Labetski, A., van Gerwen, S., Tamminga, G., Ledoux, H. and Stoter, J., 2018. A Proposal for an Improved Transportation Model in CityGML. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLII-4/W10, pp. 89–96.

Liggins, M. E., Hall, D. L. and Llinas, J. (eds), 2009. *Handbook of Multisensor Data Fusion: Theory and Practice*. The Electrical Engineering and Applied Signal Processing Series, 2nd edn, CRC Press, Boca Raton, FL.

Matthaei, R., 2015. Wahrnehmungsgestützte Lokalisierung in fahrstreifengenauen Karten für Assistenzsysteme und automatisches Fahren in urbaner Umgebung. PhD thesis, Technical University Braunschweig, Department of Electrical Engineering, Information Technology, Physics, Braunschweig.

Menzel, T., Bagschik, G. and Maurer, a. M., 2018. Scenarios for Development, Test and Validation of Automated Vehicles. In: *Intelligent Vehicles Symposium (IV 2018)*, IEEE, Changshu, pp. 1821–1827.

Navigation Data Standard e.V., 2016. Navigation Data Standard – Open Lane Model Documentation. `http://www.openlanemodel.org` (May 2019).

Riedmaier, S., Nesensohn, J., Gutenkunst, C., Duser, T., Schick, B. and Abdellatif, H., 2018. Validation of X-in-the-Loop Approaches for Virtual Homologation of Automated Driving Functions. In: *11th Graz Symposium Virtual Vehicle (GSVF 2018)*, Graz, Austria.

Schick, B., Herz, G., Hettel, R. and Meinel, H., 2017. Sophisticated Sensor Model Framework Providing Realistic Radar Sensor Behavior in Virtual Environments. In: *8. Tagung Fahrerassistenz (2017)*, Munich, Germany.

Ulbrich, S., Reschka, A., Rieken, J., Ernst, S., Bagschik, G., Dierkes, F., Nolte, M. and Maurer, M., 2017. Towards a functional system architecture for automated vehicles. *arXiv preprint arXiv:1703.08557*.

Wachenfeld, W., Junietz, P., Wenzel, R. and Winner, H., 2016. The worst-time-to-collision metric for situation identification. In: *Intelligent Vehicles Symposium (IV 2016)*, Gothenburg, Sweden, pp. 729–734.

Winner, H., Hakuli, S., Lotz, F. and Singer, C. (eds), 2016. *Handbook of Driver Assistance Systems*. Springer International Publishing.