VALIDATION OF PARAMETRIC OPENDRIVE ROAD SPACE MODELS

B. Schwab*, T. H. Kolbe

Chair of Geoinformatics, Technical University of Munich, Germany - (benedikt.schwab, thomas.kolbe)@tum.de

Commission IV, WG IV/9

KEY WORDS: Road Space Models, Street Space Models, ASAM OpenDRIVE, Parametric Geometries, Validation.

ABSTRACT:

The standard OpenDRIVE is widely used for exchanging road space models in order to simulate the traffic of a city or individual driving situations. For modeling continuous road courses at lane level, OpenDRIVE utilizes its own parametric geometry model. However, violations of continuity requirements due to geometric leaps or kinks can cause the vehicle dynamics simulation to fail when testing vehicle components. But also defective lane predecessor and successor relations can result in an OpenDRIVE dataset not being usable as a reference map for vehicle navigation. Since these geometric, topological, and semantic constraints go beyond the rules encoded in the schema, this article presents a framework and a first implementation for validating OpenDRIVE datasets. As the lane widths are defined parametrically relative to the reference line of the respective road, lane connectivities at road transitions are evaluated using explicit geometries derived from the parametric geometry representations. Moreover, a derived CityGML representation enables a visual inspection of the parametric models to identify unexpected but visible defects. The implemented framework is applied to examine a total of 99 OpenDRIVE datasets, where significant lane gaps were detected in the explicit representation for about 20% of the datasets.

1. INTRODUCTION

Historically, semantic 3D city models have particularly focused on building objects and terrain models. Due to the urbanization trend and the emergence of new mobility concepts, street spaces are gaining relevance. Moreover, advances in the mobile mapping domain are also contributing in obtaining more information about the street space (Dong et al., 2020; Beil et al., 2020). In the context of city modeling, this trend is addressed by version 3.0 of the CityGML standard, which comprises a revised transportation module as well as a new space concept (Open Geospatial Consortium, 2021; Kutzner et al., 2020). In order to evaluate new mobility concepts for a city, such as multimodal mobility approaches or the introduction of automated vehicles, the operation of traffic and driving simulators can be beneficial.

While traffic simulators can be used for traffic management of a city, driving simulators are utilized when individual vehicles are to be simulated in a traffic situation. Especially driving simulators require road network descriptions that are characterized by parametric geometries and by smooth road courses without leaps in the curvature. For the exchange of parametric road networks, the standard OpenDRIVE by the Association for Standardization of Automation and Measuring Systems (ASAM) is commonly used, since it is supported by driving as well as traffic simulators. Furthermore, OpenDRIVE datasets are supplied by surveying companies for the automotive sector. A model of a complex intersection is shown in Figure 1.

Despite the widespread use of OpenDRIVE datasets, there has been no open-source validation software so far that, to our knowledge, is capable of evaluating more complex model requirements. As OpenDRIVE has its own geometry model, existing methods and software projects, such as the CityDoctor (Coors et al., 2020) or val3dity (Ledoux, 2018), are not directly applicable. Hence, the question arises of how parametric road space models can be validated and whether minor inconsistencies can be removed automatically.

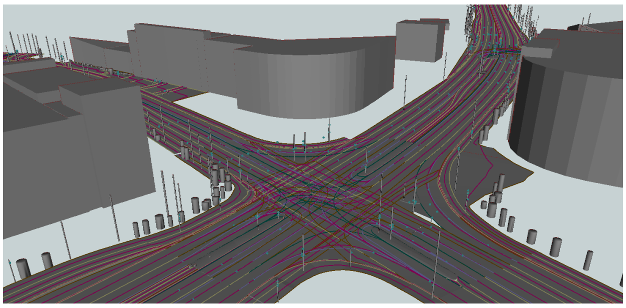

Figure 1. OpenDRIVE model of a complex intersection provided by the company 3D Mapping Solutions GmbH (2022).

2. STANDARDS AND RELATED WORKS

When modeling and simulating traffic, four categories are commonly distinguished based on the level of detail. While *macroscopic* and *mesoscopic* traffic simulations aggregate individual road users, each vehicle is distinguished and modeled separately at the *microscopic* level (Hoogendoorn and Bovy, 2001; Krauss, 1998). A microscopic traffic simulation can be used to simulate entire cities or city districts in order to evaluate traffic efficiency, emissions, or, for example, the effects of autonomous vehicles on the overall traffic flow. Software solutions for simulations at this level include, for example, PTV Vissim, AIMSUN (Casas et al., 2010) or the open-source simulator SUMO (Krajzewicz et al., 2012). At the *submicroscopic* level, the individual vehicle, its subunits, and the interaction of the subunits with their environment are modeled and simulated (Hoogendoorn and Bovy, 2001). This level is particularly relevant in the vehicle development domain, as validated environment simulations of the system under test can substantially reduce the physical testing efforts and allow tests to be repeated in a reproducible manner (Stadler et al., 2022). Examples of submicroscopic driving simulators include IPG CarMaker, Vires VTD (von Neumann-Cosel, 2014), Tesis DYNA4 and CARLA

* Corresponding author

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-X-4-W2-2022-257-2022

(Dosovitskiy et al., 2017). However, there are also coupling approaches where the ego vehicle is simulated with a submicroscopic driving simulator and the surrounding traffic is simulated using a microscopic traffic simulator (Semrau and Erdmann, 2016; Langer et al., 2021).

All simulators require a road network description as a data basis, whereby the necessary level of detail is dependent on the category of the traffic simulation. Beil et al. provide a detailed comparison between the different standards for streetspace modeling and their characteristics (Beil et al., 2020; Beil and Kolbe, 2020). Here, the main application of the standards OpenDRIVE and RoadXML is driving simulation on the submicroscopic level. The standard OpenDRIVE was originally developed by the company *Vires Simulationstechnologie GmbH* and version 0.7 was first released in the year 2005 (ASAM, 2021). All previously listed submicroscopic driving simulators as well as microscopic traffic simulators support the import of OpenDRIVE road networks. In 2018, OpenDRIVE was transferred to the standards organization ASAM and the current version 1.7 was released on August 3, 2021 (ASAM, 2021). ASAM develops several standards in the automated driving domain, such as OpenSCENARIO for the description of traffic scenarios and the Open Simulation Interface (OSI) for the interface between an automated driving function and the environment. While OpenDRIVE can be characterized by its reference line concept and its parametric geometries, which are described in more detail in section 3, RoadXML follows a similar approach (Chaplier et al., 2010). RoadXML also defines its own geometry model to construct a reference line from clothoids, splines, and straight line elements. The developers of RoadXML include French automotive companies, simulation companies as well as a research institute, and the current version 3.0.0 was released in the year 2020 (Chaplier et al., 2010).

In the city modeling domain, methods and software solutions for the validation of semantic models have already been researched and developed. Ledoux introduced a methodology for validating solid geometries against the definitions of ISO19107 and against the implementation specifications by the Open Geospatial Consortium (OGC) (Ledoux, 2013). In 2018, validation methods for all 3D primitives of ISO19107 were presented by Ledoux and implemented as part of the open-source software val3dity (Ledoux, 2018). Biljecki et al. investigated 37 CityGML datasets from 9 countries to identify the most common geometric and semantic errors. A high variance of errors was observed across the datasets, with non-planar polygons accounting for the most errors (Biljecki et al., 2016). Wagner et al. developed rules for validating geometric as well as semantical aspects of CityGML models (Wagner et al., 2013, 2015). In 2020, Coors et al. proposed a new approach for specifying application-specific requirements for 3D city models and successfully implemented them for a heating demand simulation project using the CityDoctor validation software (Coors et al., 2020). Coors et al. were able to formulate some requirements using Schematron, which is a rule-based validation language for Extensible Markup Language (XML) documents. While Schematron was sufficient for requirements describing the existence of attributes, geometric requirements could not be expressed with it (Coors et al., 2020).

To model road structures, including roads, tunnels and bridges, Lee and Kim have proposed an extension to the Industry Foundation Classes (IFC) (Lee and Kim, 2011). However, IFC Infrastructure extensions for linear infrastructure assets are still in the development phase. Amann et al. demonstrated how road cross sections can be integrated in the IFC alignment extension and validated their approach by converting from and to LandXML files (Amann et al., 2015).

3. OPENDRIVE CONCEPTS

OpenDRIVE refers to the conceptual model as well as the XML-based encoding and specifies a custom geometry model (ASAM, 2021). A dataset can be georeferenced using a projection string according to PROJ in the header element.

Geometry

OpenDRIVE uses a linear referencing concept, whereby a reference line is defined for each road by compounding parametric geometry elements, as shown in Figure 2. The reference line

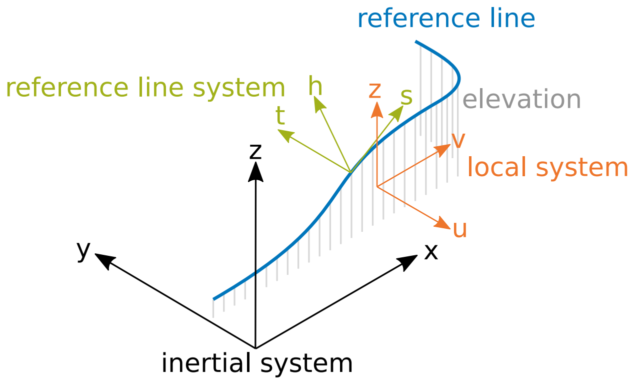

Figure 2. Three main coordinate systems used in OpenDRIVE.

opens up the tangentially adjacent reference line system with its s-, t-, and h-axis, while most objects in the road space, such as lanes, traffic signs, or even trees, are specified in this coordinate system. In the plan view, the reference line is specified by the geometry elements straight line, spiral, arc, and parametric cubic curves. Both the elevation of the reference line along the s-axis and the roll angle around the s-axis are specified with a sequence of cubic polynomials. When combining individual parametric geometry elements into a reference line, there should be no steps in the curvature, as this would lead to jumps in the steering behavior and to overlaps or gaps in the surface geometries of the lanes.

Roads and Lanes

As illustrated in Figure 3 the road is partitioned into lane sections alongside the reference line. The number of lanes remains

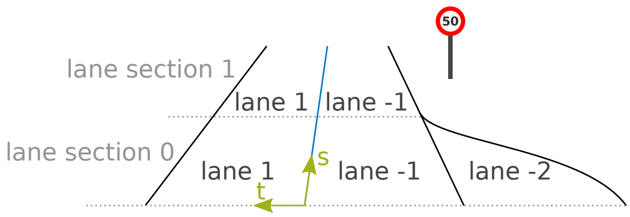

Figure 3. Road divided into lane sections and individual lanes.

stable within a lane section, and the width of a lane is given by a list of cubic polynomial functions depending on the s-axis. Each lane element holds the predecessor and successor information for the next lane section.

https://doi.org/10.5194/isprs-annals-X-4-W2-2022-257-2022 | © Author(s) 2022. CC BY 4.0 License.

Junctions

In order to connect roads in junction areas, connecting roads are placed in the junction. The connecting roads with road id 28, 61 and 64 are illustrated in Figure 4 with their road reference line. The individual lanes, which are laterally shifted to the reference line, are not visualized in the junction area due to the many overlaps. The requirements for junctions are mainly concerned with the topological aspects and can therefore be checked for the most part using the parametric representation.

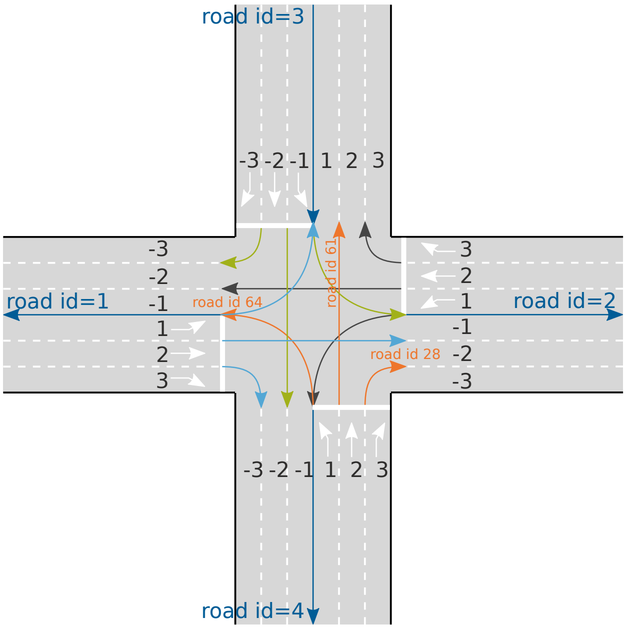

Figure 4. Road reference lines indicated as arrows with individual lanes indicated by lane ids from -3 to 3 for a right-hand traffic junction.

Objects and Signals

As shown in Figure 1, road objects can be modeled with coarse geometries, which are defined in the reference line system or in the local coordinate system. Such objects include poles, street lamps, fences, railings, and road marks. Points, parametric rectangles, circles, cuboids, and cylinders are used for simpler geometric representations. For more complex representations, outline elements with relative height values can be defined either in the local or in the reference line system.

4. VALIDATION FRAMEWORK

The challenge in processing available OpenDRIVE datasets is on the one hand the number of seven different versions (v1.1-v1.7) that have been released since 2006. On the other hand, some models contain inconsistencies according to the rules in the specification, which can lead to unexpected r esults. Inconsistencies not detected by a schema validation include, for example, unsorted element lists of a section-wise defined function and interrelated attribute assignment requirements (unit attribute being mandatory if the value attribute is assigned).

To ensure that the geometric, topological and semantic constraints are only evaluated on data structures without severe inconsistencies, the validation framework shown in Figure 5 is proposed. The input datasets are located on the left side, which are then processed in the following seven stages and exported on the right side. The first three stages are concerned with data harmonization and inconsistency removal of the input OpenDRIVE datasets. Then, starting from stage three, the evaluation of the more complex model requirements on different representations begins. If fatal errors are detected in the model during the stages, at most more errors are collected, but the overall process is terminated.

1st Stage: OpenDRIVE Schema Validation

The OpenDRIVE version of a given dataset is defined in the header element as attribute `revMajor` and `revMinor`. After the version is correctly identified, the dataset is validated using the OpenDRIVE schemas. Within the implementation, the schemas for OpenDRIVE version 1.1 to 1.7 are included and the schema validation task is delegated to the software library Jakarta XML Binding (JAXB) (Sun Microsystems, Inc., 2006).

2nd Stage: Mapping to OpenDRIVE Model Classes

After the dataset is validated against the schema, the model is unmarshalled into memory using model classes generated by JAXB. Since these model classes are generated on the basis of the schemas, they differ for each OpenDRIVE version. In order to carry out the validation tasks on the same OpenDRIVE model classes, the generated classes are mapped to the main implementation of OpenDRIVE according to the current version 1.7 (ASAM, 2021).

3rd Stage: Evaluation on OpenDRIVE Model Classes

Based on the model classes of OpenDRIVE, inconsistencies in the model are identified and removed if possible. Since attribute value ranges were not constrained in the schema until version 1.5, coordinate value ranges are checked first. For example, s-coordinates along the reference line must be greater equal zero or the length of a bounding box of a road object must be greater than zero. If inconsistencies are identified for such attributes, they are set to default values.

However, the value range can be further constrained beyond the rules of the current schema. Since cubic polynomials are used as a function of the s-coordinate to specify road heights, lane widths, road object widths, etc., it is also ensured that the s-coordinates do not exceed the total length of the reference line. In addition, the use of features may be mutually exclusive, such as the use of either lane offsets (lateral shift of the center lane relative to the reference line by cubic polynomials) or lateral road shapes (section-wise defined functions for the relative road elevation along the t-axis and linearly interpolated in-between).

If mandatory requirements of the OpenDRIVE specification are violated and no handling strategy is available, the validation process is terminated for the respective dataset with a fatal error. For example, a lane section must have exactly one center lane and at least one left or right lane. A missing left or right lane currently leads to a fatal error, since adding a new lane would involve assumptions about the lane width and the adjustment of the predecessor and successor relationships across road and junction elements. If lists of geometry elements or lists of cubic polynomials are not strictly ordered along the reference line by a tolerance, the outlier elements are removed. All modifications of the model are added to the report.

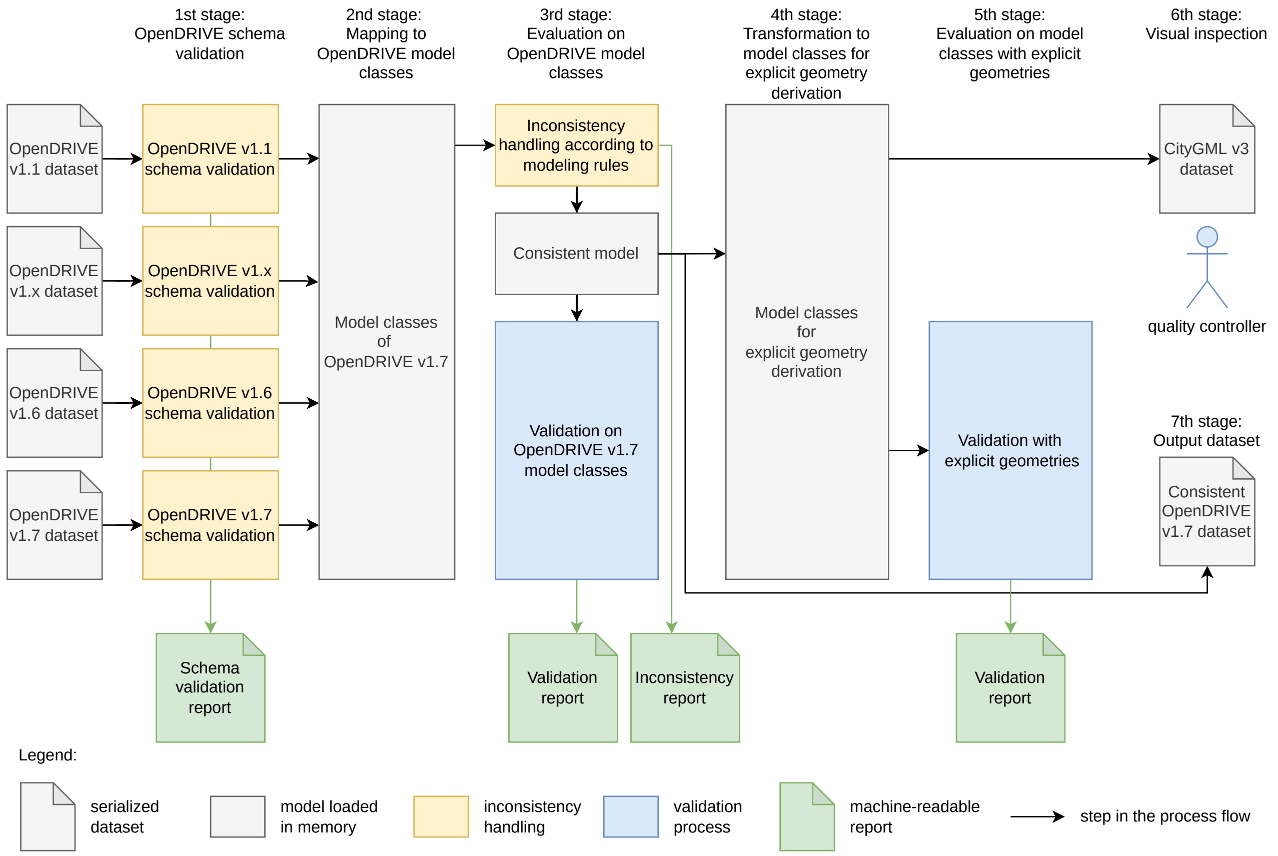

Figure 5. Framework architecture for validating OpenDRIVE datasets.

Once the inconsistencies in the model have been handled, the first geometric requirements can be examined on the implemented OpenDRIVE model classes. Since all aspects are modeled relative to the reference line in OpenDRIVE, a geometrically correct reference line is the precondition for all further validation tasks. A reference line is constructed using the geometry elements `line`, `arc`, `spiral` (clothoid), `poly3` (cubic polynomial deprecated due to ambiguities) and `paramPoly3` (parametric cubic polynomial). These geometry elements are positioned individually with a translation and rotation in 2D in the inertial system, as shown in Figure 6. According to the OpenDRIVE

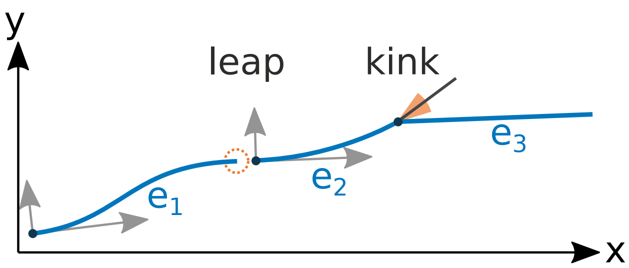

Figure 6. Plan view of concatenated geometry elements (e_1-e_3) forming a defective reference line for a single road.

specification, a reference line must strictly not contain gaps and should not contain kinks (ASAM, 2021). Since smooth reference lines are required by several applications, such as submicroscopic driving simulators, the identification o f k inks is relevant during validation. Because each geometry element is parametrically defined in a local coordinate system (colored in gray in Figure 6) the endpoint must be resolved and an affine transformation into the inertial system must be performed to identify leaps. Normalized Fresnel integrals are used for the discretization of the Euler spiral. As software solutions differ in terms of decimal accuracy, the length information per geometry element cannot be relied upon and therefore the length is calculated based on the absolute s-positions of the current and following geometry element. Then, it is tested whether the Euclidean distance between the end and start point in the inertial system exceeds a tolerance threshold.

To identify kinks at the geometry element transitions, the parametric functions are derived in their respective local coordinate systems and discretized at the start and end points. Once the curve curvature has been transformed to the inertial system, the angle difference is computed and reported if an angle threshold value is exceeded. The tolerance distance and angle ranges are highlighted in orange in Figure 6.

4th Stage: Transformation to Model Classes for Explicit Geometry Derivation

To evaluate the connectivity at road and lane transitions, distance calculations in 3D are required. Therefore, the OpenDRIVE model is mapped into an intermediate representation, which allows deriving explicit geometries from it.

If the geometries of the reference line are valid in 2D, the heights of the road space objects are calculated. The absolute height of the reference line in the inertial system is defined by a list of cubic polynomials which are placed by a sOffset value along the reference line. The lateral profile of a road is described parametrically in OpenDRIVE by the superelevations and road

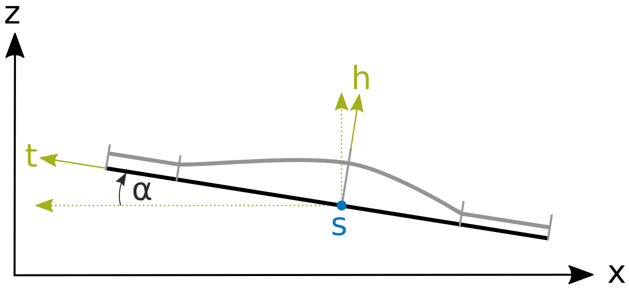

Figure 7. Superelevation α and section-wise defined height functions for the road shape.

shapes, as shown in Figure 7 (ASAM, 2021). The superelevation refers to the cross slope and is described as a list of cubic polynomials along the reference line. It is implemented as a torsion of the reference line and thus describes a continuous roll angle of the reference line system.

The road shape defines the height offset in the reference line system and is colored in gray in Figure 7. It allows the parametric modeling of road crossfalls and also general road surfaces. For a fixed s-value, cubic polynomials are defined along the t-axis to describe the height h. These lateral height functions

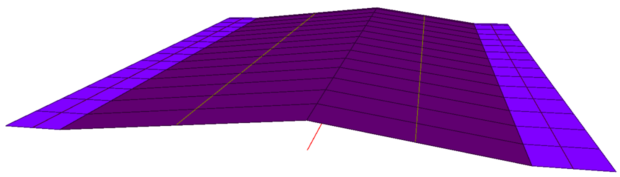

Figure 8. OpenDRIVE sample using road shapes and visualized as CityGML dataset.

can be repeated multiple times along the s-axis, while height values in-between are linearly interpolated. In order to conduct computations on such road surfaces, curve-relative parametric surfaces were implemented. Figure 8 shows a converted OpenDRIVE sample dataset from ASAM with a road surface modeled using road shapes and the red reference line below.

The calculation of the road surface elevation for distinct s- and t-coordinates is the precondition to discretize the lane boundaries. According to the specification, each road must have a center lane, which has a width of zero and an ID of zero (ASAM, 2021). As shown in Figure 9, the center lane can be shifted by a

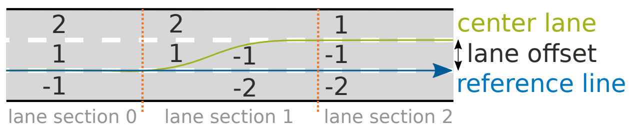

Figure 9. Lane offset functions used to laterally shift the center lane away from the road reference line.

lateral translation defined as section-wise defined cubic polynomials on the road surface. This implies that the reference lines of two successive roads do not necessarily have to be connected in the inertial system. The widths of the individual lanes are described as a list of cubic polynomials. Thus, to evaluate the explicit lane borders, several section-wise defined functions have to be aggregated, whereas the domains of those functions are usually different per lane. Moreover, individual lanes can be excluded from the superelevation and additional height offsets can be specified. Such inner and outer additional elevations are usually used for sidewalks to model the curb elevations. To evaluate the connectivity of lane transitions between different roads using explicit coordinates in the inertial system, curves were implemented which can be laterally shifted on the road surface in the reference line system.

5th Stage: Evaluation on Model Classes with Explicit Geometries

In order to assess the connectivity to the next lane, the succeeding road must first be determined. Either a road has another road as a direct successor or a list of connecting roads that belong to a junction. To check topologically linked lanes with respect to their connectivity, the left and right lane boundaries are evaluated against the boundaries of their successor lane, as shown in Figure 10. When the left and right lane boundaries

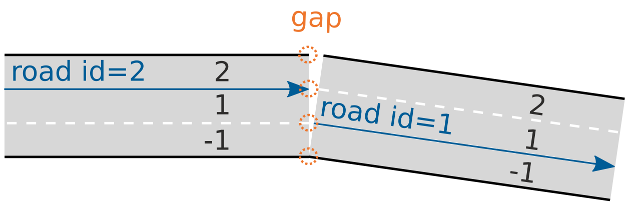

Figure 10. Gaps between the lane boundaries that exceed the orange colored tolerance.

close flush with the lanes of the following road, the entry and exit angles are the same, resulting in no curvature jumps.

Figure 11 shows lane transitions without gaps from the upper road with surface representation to six roads with linear representations located in a junction. The center lines of the lanes are also depicted but are not utilized for validation purposes.

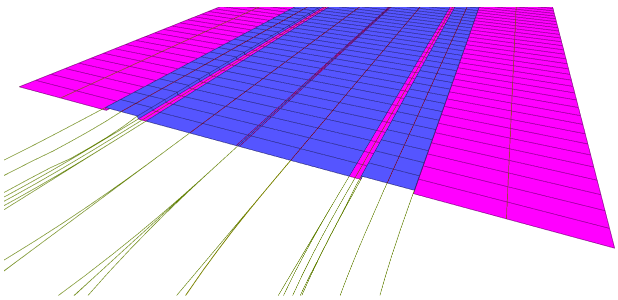

Figure 11. Connected lane boundaries at the lane transitions between multiple roads.

Furthermore, an explicit representation enables a straightforward evaluation of requirements concerning road space objects. If road space objects are geometrically represented by outlines, counter-clockwise ordering is recommended by the OpenDRIVE specification (ASAM, 2021). Figure 12 shows the triangulated surfaces of parking lots, which are defined in the reference line system and oriented downward. Faulty surface orientations due to clockwise ordered outline elements are identified in the inertial system with straightforward normal calculations.

6th Stage: Visual Inspection

In order to enable a visual inspection by a quality controller, the model is transformed to CityGML version 2.0 or 3.0 and is exported as a dataset (Schwab et al., 2020). This allows the models to be opened and viewed with established tools, such

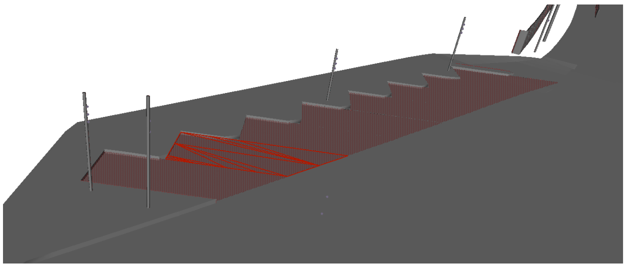

Figure 12. Downward oriented parking lots due to clockwise ordering of outline elements in the linear reference system.

as the Feature Manipulation Engine (FME) using the dedicated CityGML v2 or the generic GML reader[1]. Figure 13 shows a

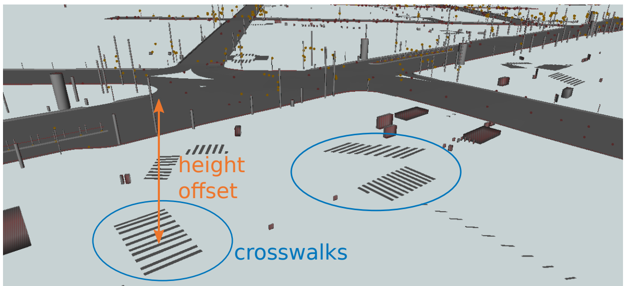

Figure 13. Systematic height offset from the road to unclassified crosswalks objects.

model where the crosswalk objects are not classified, but have a systematic height offset in orange from the road due to different interpretations of the OpenDRIVE specification. In this case, a rule-based distance check between crosswalk objects and road surfaces is not a solution due to the missing object type. Unexpected model faults can thus be identified by visual inspection.

7. Stage: Output Dataset

After the OpenDRIVE model has passed the fifth stage without fatal errors, the model is serialized as an OpenDRIVE dataset according to the current version 1.7. For this purpose, the model classes from the second stage are mapped back to the generated JAXB classes of OpenDRIVE version 1.7. Afterward, the model without inconsistencies is serialized as a dataset.

Export of the Reports

To allow other applications to read the results, the validation reports as well as the inconsistency handling reports are serialized as JavaScript Object Notation (JSON) files. Report entries include a description, severity level, exception identifier, and location.

5. IMPLEMENTATION AND RESULTS

The proposed framework was implemented by extending the open-source tool r:trån[2]. It is written in the Kotlin programming language, which is interoperable with Java and is executed using the Java Virtual Machine (JVM).

[1] XML schema: https://github.com/opengeospatial/CityGML-3.0Encodings/tree/master/CityGML/Schema

[2] https://github.com/tum-gis/rtron

Version	1.1	1.2	1.3	1.4	1.5	1.6	1.7
# compliant				28		17	21
# non-compliant	14	14		1	3	1	
Total	14	14	0	29	3	18	21

Table 1. Total number of schema compliant and non-compliant datasets grouped by versions.

The tool was originally developed for transforming OpenDRIVE datasets to CityGML2 and was presented by Schwab et al. (2020). It has been substantially extended to support CityGML3, conduct the evaluations on different model representations, support the road surface representations and generate the result reports. The developed validation method can be called via a subcommand.

In order to test the developed method, a total of 99 publicly available OpenDRIVE datasets were collected. This includes example datasets from ASAM, demo datasets from companies, open datasets for test tracks, as well as synthetic datasets used in a driving simulator.[3] Table 1 shows the aggregated results after the schema validation with respect to the OpenDRIVE version. The OpenDRIVE datasets of versions 1.1 and version 1.2 produce fatal schema validation errors. Starting with version 1.4, a large percentage of datasets are fully schema-compliant.

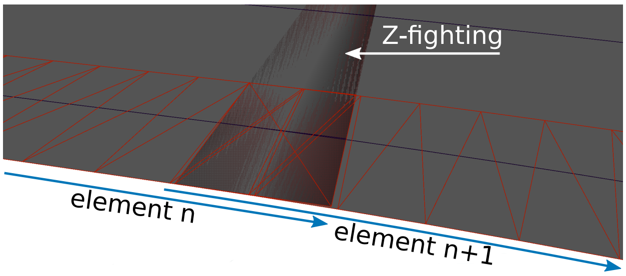

Figure 14. Lane surface overlap caused by a gap between geometry element n and $n+1$ of the reference line.

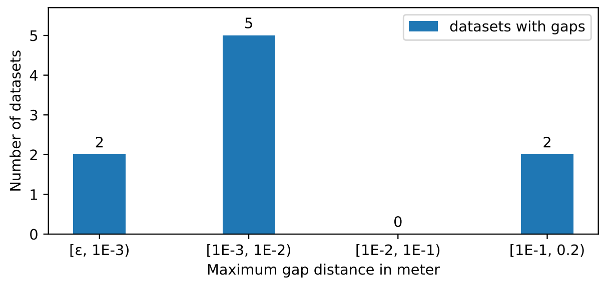

Figure 15. Distribution of maximum gap distances in reference line per dataset.

Gaps and curvature jumps in the reference line can lead to overlaps of the surface geometries of lanes. Figure 14 shows an exemplary surface overlap caused by a non-continuous reference line that potentially leads to Z-fighting in simulation applications. A histogram of the maximum gap distance of a dataset

[3] The results reports and the sources of the respective OpenDRIVE datasets are published at the following link: https://github.com/tum-gis/opendrive-dataset-validation

is shown in Figure 15, whereas a gap is identified starting at an ε of 1E-4 [m]. Overall, nine datasets have been identified with this issue. However, 33 dataset evaluations had already been stopped in a previous stage with a fatal error.

The lane boundary connectivity tests of the fifth stage in the framework identified a vertical gap of around 14.8cm at a biking lane, which is shown in Figure 16. Moreover, a gap at a junction caused by jumps in the reference line superelevation was identified and is shown in Figure 17.

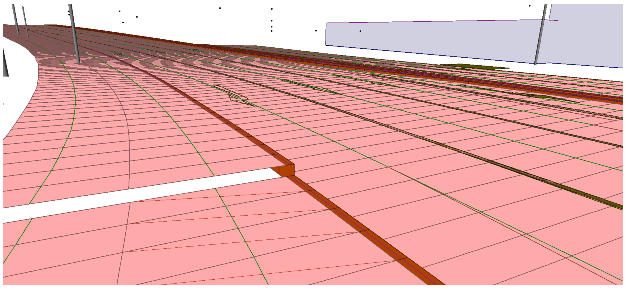

Figure 16. Dataset containing a biking lane gap at a road transition. This only becomes visible in the CityGML representation of the dataset.

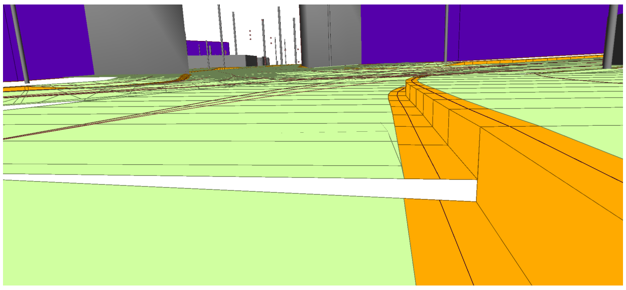

Figure 17. Dataset containing a driving lane gap of around 6cm at a road transition.

In total, 11 datasets out of 57 are identified with lane gaps at transitions, whereas a distance distribution is shown in Figure 18. Distances in the centimeter range indicate modeling inaccuracies, while distances in the larger meter range indicate erroneous predecessor and successor relationships.

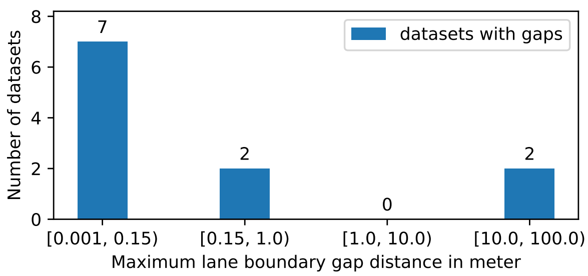

Figure 18. Distribution of maximum gap distances between linked lane boundaries.

Figure 19 shows the overview of the datasets and at which validation steps they fail. A total of 99 datasets start on the left side and 66% of them pass the schema validation. A total of around 70% out of these datasets pass the reference line continuity test on the OpenDRIVE model classes with a tolerance value of 1E-4 [m]. For approximately 20% of these, gaps are detected between lanes at road transitions using the explicit representation.

6. CONCLUSIONS AND FUTURE WORK

For the application of parametric road space models in submicroscopic driving simulators, the continuity and the absence of curvature jumps of the lane trajectories is a central aspect. To verify that OpenDRIVE datasets meet such requirements, a validation framework was developed that derives an explicit representation and conducts continuity checks on it. Here, the conducted experiments indicate that about 20% of the datasets have gaps between the lanes at road transitions.

Depending on the constraint type of the specification requirement, a parametric or explicit representation of the model is suitable. While the evaluation of lane connectivities is performed using explicit geometries, model modifications to remove inconsistencies are currently only performed on the OpenDRIVE model classes. The next steps to be investigated include, in particular, improperly oriented solids as well as excessive spacing between the road surface and road objects, such as lane markings or crosswalks. Beyond that, application-specific requirement evaluations should be further investigated considering that a traffic simulation imposes less rigorous continuity constraints compared to a driving simulation, for example. In general, the validation methods of parametric models become particularly relevant when they need to be derived from geospatial data for simulation applications, or when semantic 3D city models are to be coupled with parametric road space models.

ACKNOWLEDGEMENTS

This work is supported by the German Federal Ministry of Transport and Digital Infrastructure (BMVI) within the *Automated and Connected Driving* funding program under the Grant No. 01MM20012K (SAVeNoW).

References

3D Mapping Solutions GmbH, 2022. Ultra HD Maps. `https://www.3d-mapping.de/en/our-service/ultra-hd-maps/` [accessed 2022-04-08].

Amann, J., Singer, D., Borrmann, A., 2015. Extension of the upcoming IFC alignment standard with cross sections for road design. *Proceedings of the International Conference on Civil and Building Engineering Informatics, Tokyo, Japan, 22-24 April, 2015*.

ASAM, 2021. OpenDRIVE V1.7.0 User Guide. `https://www.asam.net/standards/detail/opendrive` [accessed 2022-04-08].

Beil, C., Kolbe, T. H., 2020. Combined Modelling of Multiple Transportation Infrastructure within 3D City Models and Its Implementation in CityGML 3.0. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, VI-4/W1-2020, 29–36.

Beil, C., Ruhdorfer, R., Coduro, T., Kolbe, T. H., 2020. Detailed Streetspace Modelling for Multiple Applications: Discussions on the Proposed CityGML 3.0 Transportation Model. *ISPRS International Journal of Geo-Information*, 9(10), 603.

Biljecki, F., Ledoux, H., Du, X., Stoter, J., Soon, K. H., Khoo, V. H. S., 2016. The Most Common Geometric and Semantic Errors in CityGML Datasets. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2/W1, 13–22.

https://doi.org/10.5194/isprs-annals-X-4-W2-2022-257-2022

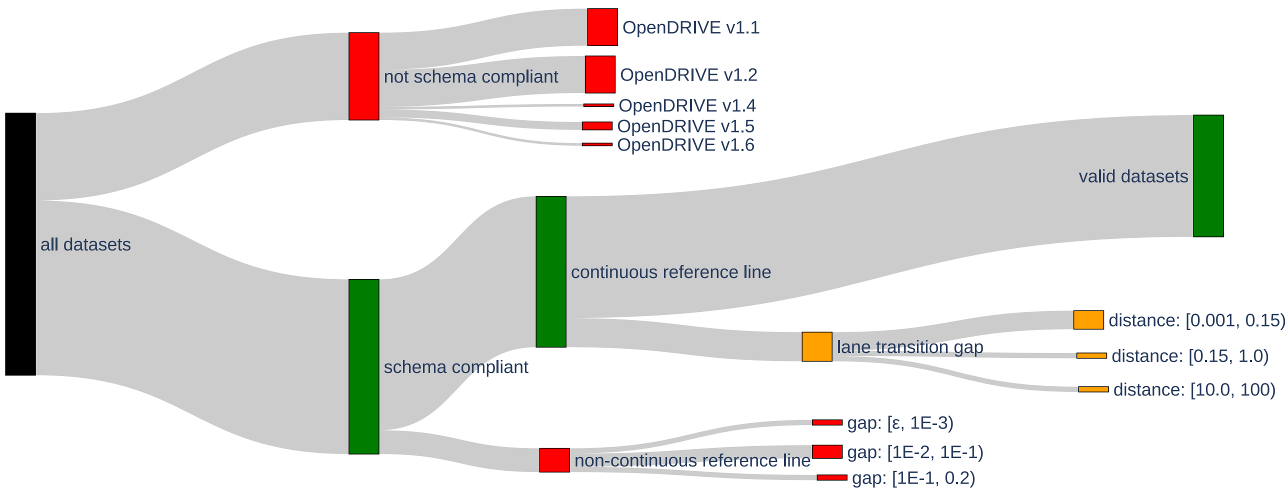

Figure 19. Overview of the tested OpenDRIVE datasets and at which validation step they fail. Green means (still) valid, red indicates not valid and orange denotes valid but not recommended according to the specification.

Casas, J., Ferrer, J. L., Garcia, D., Perarnau, J., Torday, A., 2010. Traffic Simulation with Aimsun. J. Barceló (ed.), *Fundamentals of Traffic Simulation*, 145, Springer, New York, NY, USA, 173–232.

Chaplier, J., That, T. N., Hewatt, M., Gallée, G., 2010. Toward a standard: RoadXML, the road network database format. *Proceedings of the Driving Simulation Conference Europe, Paris, France, 9-10 September 2010*, 211–221.

Coors, V., Betz, M., Duminil, E., 2020. A Concept of Quality Management of 3D City Models Supporting Application-Specific Requirements. *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 88(1), 3–14.

Dong, Z., Liang, F., Yang, B., Xu, Y., Zang, Y., Li, J., Wang, Y., Dai, W., Fan, H., Hyyppä, J., Stilla, U., 2020. Registration of Large-Scale Terrestrial Laser Scanner Point Clouds: A Review and Benchmark. *ISPRS Journal of Photogrammetry and Remote Sensing*, 163, 327–342.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V., 2017. CARLA: An open urban driving simulator. *Proceedings of the 1st Annual Conference on Robot Learning (CoRL), Mountain View, CA, USA, 13-15 November 2017*, 1–16.

Hoogendoorn, S. P., Bovy, P. H. L., 2001. State-of-the-Art of Vehicular Traffic Flow Modelling. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 215(4), 283–303.

Krajzewicz, D., Erdmann, J., Behrisch, M., Bieker, L., 2012. Recent Development and Applications of SUMO-Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4), 128-138.

Krauss, S., 1998. Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics. PhD thesis, University of Cologne, Department of Mathematics and Natural Sciences, Cologne, Germany.

Kutzner, T., Chaturvedi, K., Kolbe, T. H., 2020. CityGML 3.0: New Functions Open Up New Applications. *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 88(1), 43–61.

Langer, M., Harth, M., Preitschaft, L., Kates, R., Bogenberger, K., 2021. Calibration and assessment of urban microscopic traffic simulation as an environment for testing of automated driving. *Proceedings of the IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, USA, 19-21 September, 2021*, 3210–3216.

Ledoux, H., 2013. On the Validation of Solids Represented with the International Standards for Geographic Information. *Computer-Aided Civil and Infrastructure Engineering*, 28(9), 693–706.

Ledoux, H., 2018. Val3dity: Validation of 3D GIS Primitives According to the International Standards. *Open Geospatial Data, Software and Standards*, 3(1), 1.

Lee, S.-H., Kim, B.-G., 2011. IFC Extension for Road Structures and Digital Modeling. *Procedia Engineering*, 14, 1037–1042.

Open Geospatial Consortium, 2021. OGC City Geography Markup Language (CityGML) Part 1: Conceptual Model Standard. `http://www.opengis.net/doc/IS/CityGML-1/3.0` [accessed 2022-04-08].

Schwab, B., Beil, C., Kolbe, T. H., 2020. Spatio-Semantic Road Space Modeling for Vehicle–Pedestrian Simulation to Test Automated Driving Systems. *Sustainability*, 12(9), 3799.

Semrau, M., Erdmann, J., 2016. Simulation framework for testing adas in chinese traffic situations. *Proceedings of the SUMO 2016–Traffic, Mobility, and Logistics, Berlin, Germany, 23-25 May 2016*, 30, DLR, 103–115.

Stadler, C., Montanari, F., Baron, W., Sippl, C., Djanatliev, A., 2022. A Credibility Assessment Approach for Scenario-Based Virtual Testing of Automated Driving Functions. *IEEE Open Journal of Intelligent Transportation Systems*, 3, 45–60.

Sun Microsystems, Inc., 2006. Java architecture for XML binding (JAXB) 2.0 specification. JSR-000222, `https://download.oracle.com/otn-pub/jcp/jaxb-2.0-fr-oth-JSpec/jaxb-2_0-fr-spec.pdf` [accessed 2022-04-08].

von Neumann-Cosel, K., 2014. Virtual Test Drive. PhD thesis, Technical University of Munich, Department of Informatics, Munich, Germany.

Wagner, D., Alam, N., Wewetzer, M., Pries, M., Coors, V., 2015. Methods for Geometric Data Validation of 3D City Models. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-1/W5, 729–735.

Wagner, D., Wewetzer, M., Bogdahn, J., Alam, N., Pries, M., Coors, V., 2013. Geometric-Semantical Consistency Validation of CityGML Models. J. Pouliot, S. Daniel, F. Hubert, A. Zamyadi (eds), *Progress and New Trends in 3D Geoinformation Sciences*, Springer Berlin Heidelberg, Berlin, Heidelberg, 171–192.