CommonRoad Scenario Designer: An Open-Source Toolbox for Map Conversion and Scenario Creation for Autonomous Vehicles

Sebastian Maierhofer*, Moritz Klischat*, and Matthias Althoff

***Abstract*— Maps are essential for testing autonomous driving functions. Several map and scenario formats are available. However, they are usually not compatible with each other, limiting their usability. In this paper, we address this problem using our open-source toolbox that provides map converters from different formats to the well-known CommonRoad format. Our toolbox provides converters for OpenStreetMap, Lanelet/Lanelet2, OpenDRIVE, and SUMO. Additionally, a graphical user interface is included, which allows one to efficiently create and manipulate CommonRoad maps and scenarios. We demonstrate the functionality of the toolbox by creating CommonRoad maps and scenarios based on other map formats and manually-created map data.**

I. Introduction

The validation of motion planning and prediction algorithms for autonomous vehicles requires many test scenarios. Maps are indispensable for their creation. However, there is a lack of open-source maps. Additionally, available map sources and scenario datasets are often given in different map formats, forcing users to develop custom conversion algorithms to obtain the desired map format. Therefore, a simple-to-use framework for converting different map formats into a common representation is desirable. Researchers and developers often want to adapt existing maps or even create them from scratch, requiring an intuitive user interface. Our CommonRoad Scenario Designer solves the above issues. It provides converters to a common representation and an easy-to-use interface for creating and manipulating maps. The toolbox addresses different research and development areas for intelligent transportation systems, e.g., traffic simulation, motion planning, and prediction.

A. Related Work

The creation of traffic scenarios for virtual tests of autonomous vehicles is often facilitated by real-world data collection. Data are usually collected using drones [1]–[5], stationary cameras [6], or test drives using onboard sensors [7]. Although these datasets provide naturalistic data, they are also more costly to generate. An alternative is to use traffic simulators, e.g., SUMO [8], SVL [9], or Carla [10]. These simulators can also simulate traffic using intelligent agents. Simple, single-lane scenarios can be created with the editor presented in [11].

For representing maps, several formats have emerged for motion planning [12]. Lanelet-based formats, such as Lanelet2 [13] and CommonRoad [14] have been widely used in research. OpenDrive[1] is prevalent in the industry and is used by the simulator Carla and the platform Apollo[2]. The simulator SVL supports both, Lanelet2 and OpenDRIVE. A map format used by commercial map providers is NDS[3].

Maps for virtual test drives or real driving can be created through aerial imagery [15], reconstructed from sensor data [16], or created manually using commercial tools, such as MATLAB RoadRunner[4] or open-source tools[5]. Another source for creating maps is OpenStreetMap (OSM)[6] [17], which provides geographic data for most locations in the world. In the context of autonomous driving, OSM data are used for self-localization [18], [19] or fused with sensor data [20], [21] for road detection in images. However, raw OSM data describe roads on a coarse level and do not provide the lane-level information required for navigating autonomous vehicles. To create highly detailed maps, a converter from OSM to road networks exists for the traffic simulator SUMO [22], which can also generate OpenDRIVE maps.

B. Contributions

Based on the integration and extension of our previously presented map converters from OpenDRIVE [23], SUMO [24], and OSM [25] to lanelets, we present the first unified open-source toolbox, which

1) converts OSM data into lanelets;
2) converts between different map formats;
3) provides a graphical user interface (GUI) to manually create and edit scenarios using lanelet-based maps;
4) offers easy-to-use interfaces and tutorials for getting started quickly.

The remainder of the paper is organized as follows. Sec. II presents an overview of the CommonRoad Scenario Designer. The CommonRoad format, the different supported map conversions, and the GUI are presented in Sec. III and Sec. IV. In Sec. V, we evaluate the toolbox. Finally, Sec. VI presents the conclusion.

II. Overview

Fig. 1 shows an overview of the CommonRoad Scenario Designer toolbox. The toolbox integrates several map converters and is based on the CommonRoad format. We support

*The first two authors have contributed equally to this work.

All authors are with the Department of Informatics, Technical University of Munich, 85748 Garching, Germany.

{sebastian.maierhofer, moritz.klischat, althoff}@tum.de

[1] https://www.asam.net/standards/detail/opendrive/
[2] https://apollo.auto
[3] https://nds-association.org
[4] https://mathworks.com/products/roadrunner.html
[5] https://github.com/autocore-ai/MapToolbox
[6] https://www.openstreetmap.org

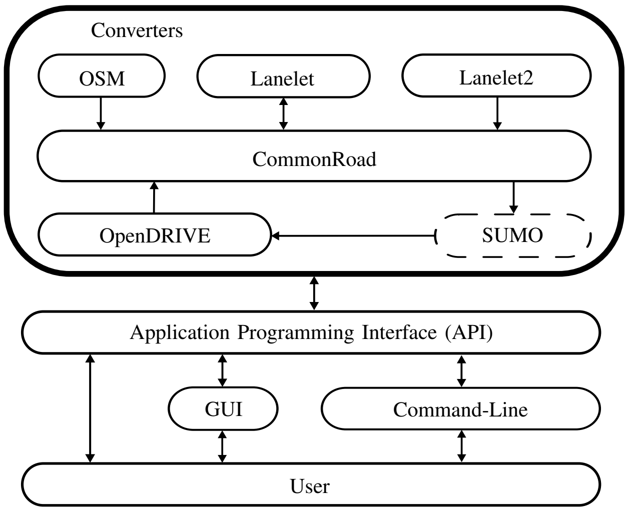


Fig. 1: Overview about the CommonRoad Scenario Designer. The arrows indicate the data flow, where the integration of the third-party tool SUMO is optional.

conversions from the OSM, Lanelet, Lanelet2, OpenDRIVE and SUMO map format to the CommonRoad format. For the Lanelet and SUMO format, we also support the conversion in the other direction. Users can work with the CommonRoad Scenario Designer in different ways: 1) an application programming interface (API), which can be directly integrated into the user's software; 2) a GUI, which provides many functionalities for editing and creating CommonRoad scenarios and for converting maps; 3) a command-line interface for executing the converters or GUI. The toolbox is provided as an easy-to-install Python package. For a detailed documentation and information on downloading and installing the CommonRoad Scenario Designer, please visit `https://commonroad.in.tum.de/`.

III. Map Conversion

We present the CommonRoad 2020a map format version and different map conversions to the CommonRoad format and vice versa. Note that we do not present detailed conversion steps since it is not in the scope of this paper. Detailed conversion steps can easily be traced in the documented code.

A. *CommonRoad Map Format*

For a description of CommonRoad elements that are not related to maps, we refer the reader to [14]. The CommonRoad map format is based on lanelets [26]. A lanelet is a combination of two polylines, which model the left and right lane boundary; the boundaries can also have line markings. The driving direction is implicitly defined by the two polylines. In addition to predecessor and successor lanelets, a lanelet can have an adjacent right and left lanelet. Fig. 2 shows the relationships between lanelets. A lanelet can reference traffic signs, traffic lights, a stop line, allowed road user types, and lanelet types, e.g., access ramp, main carriageway, or sidewalk.

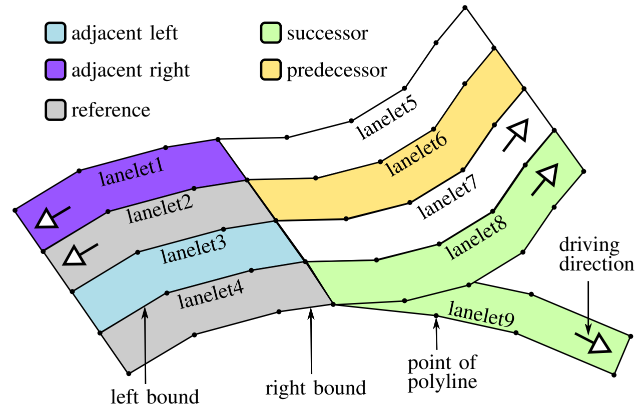


Fig. 2: Schematic visualization of a lanelet network with lanelet relationships. Only the adjacency in the same driving direction is shown.

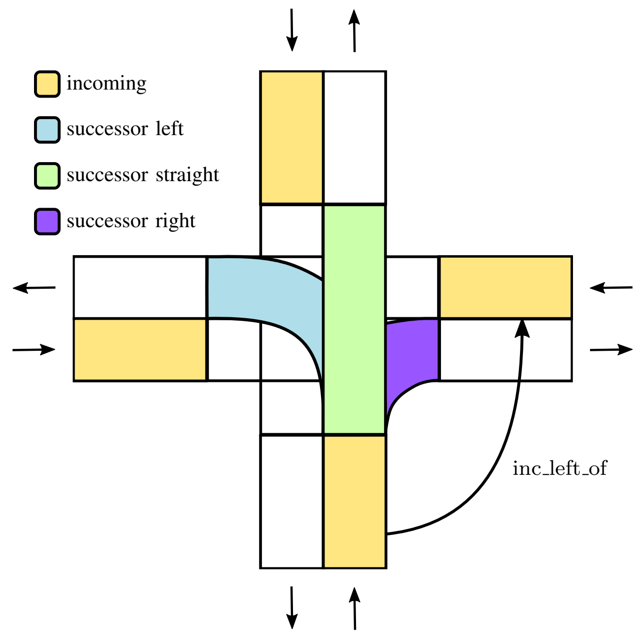


Fig. 3: Schematic intersection (CommonRoad version 2020a), where the successors are only visualized for one incoming lanelet.

Intersections are essential items of road networks. We model an intersection by grouping incoming lanelets, defining the successor lanelets of incoming lanelets, and listing the crossing lanelets of an intersection, e.g., pedestrian crossings (cf. Fig. 3). Additionally, an incoming group has a reference to the next counterclockwise-oriented incoming group (cf. inc_left_of in Fig. 3).

B. *OpenDRIVE to CommonRoad*

OpenDRIVE models lanes using various mathematical representations, such as spirals (clothoids), arcs, straight lines, or parametric cubic curves. In our previous work [23], we developed a converter of these representations into a lanelet-based representation. We extended this conversion by considering traffic signs, line markings, stop lines, traffic lights, and intersections.

C. Lanelet/Lanelet2 to CommonRoad

The CommonRoad format is based on the lanelet format [26]. Another extension is the Lanelet2 format [13], which also considers regulatory elements and adds several layers of abstraction to the original lanelet format. Besides supporting the conversion of lanelets in both directions, the CommonRoad Scenario Designer can convert regulatory elements from Lanelet2 to CommonRoad.

D. OpenStreetMap to CommonRoad

The road data of OSM are valuable assets for creating maps. For instance, geometric road information, traffic signs, and traffic lights are of great interest. The maps in OSM consist of mainly three elements: *nodes*, *ways* (ordered lists of nodes), and *relations* between arbitrary elements. Such a relation can define a turn restriction between two ways on an intersection. Features of elements, such as the speed limits of ways, are provided by *tags*.

Geometric conversion: Since the level of detail of this representation is not yet sufficient for highly-detailed maps in autonomous driving, we have to enhance this representation on the lane level and deduce missing information. The conversion undergoes three major steps described below. Fig. 9 shows an example, which also visualizes intermediate results for a complex intersection.

1) The road network topology is extracted from the nodes and ways in OSM (cf. Fig. 9b). The number of lanes is provided either through tags of the ways or deduced from the road type if no tags are provided. The level of detail is further enhanced by fitting splines to the center lines of lanes and resampling the lanelet vertices adapted to the spline curvature.
2) At intersections in OSM, all ways usually cross at the center of the intersection and links between lanes of incoming and outgoing edges are not unambiguously defined. Thus, we utilize the *turn:lane* tag of ways to deduce the outgoing lanes of each incoming lane, as shown in Fig. 4. *Turn:lane* tags define the direction of each of its lanes by tags, such as *through*, *right*, or *slight_right*. Analogously, links between lanes can be ruled out using turn-restriction relations. If neither *turn:lane* tags nor turn restrictions are available in the OSM dataset, we assume that each incoming lane is connected to one outgoing lane and that the outermost lanes are turning to the left and right, respectively.
3) The final CommonRoad map is generated by creating a lanelet for each lane: The lateral boundaries of lanelets are computed and their lateral offset is optimized to align through-going lanes when the number of lanes changes at a node. *turn:lane* tags are used to determine the lanes that fork or merge (cf. Fig. 5). Intersection elements are generated for intersecting and merging/forking lanelets and adjacency relations are added to each lanelet. Fig. 9d shows that the resulting map provides realistic road geometries.

Due to the missing level of detail in OSM data, the deduction of lane-level topologies in step 2 often reaches its limit at complex intersections with more than four to five crossing ways. For such cases, automatically proposed connections can be corrected manually using our GUI.

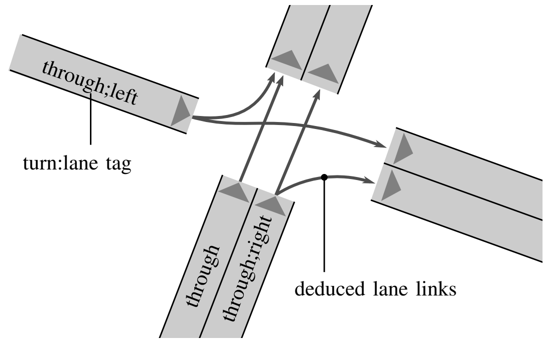


Fig. 4: Deducing links of incoming and outgoing lanes at intersections using *turn:lane* tags.

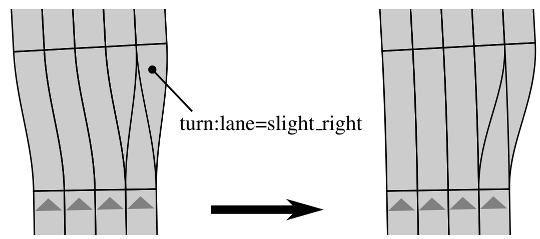


Fig. 5: Aligning of through-going lanes: using *turn:lane* tags, the swerving lane is detected and we assume the other lanes are continuing straight.

Traffic signs: After generating the lanelets, traffic lights and user-selected traffic signs are added to the map using traffic sign elements or other tags of OSM:

- Traffic lights: Positions of traffic lights can be provided in OSM. However, traffic lights for individual lanes or traffic cycles are usually not specified. Thus, we create default traffic light cycles that can be manually adjusted if desired. Alternatively, traffic light cycles can be generated using SUMO when simulating traffic scenarios.
- Speed limits: Speed limits are either explicitly given as tags or deduced from the road type.
- Further considered traffic signs: stop, give way, no overtaking, and turn restrictions.

Since traffic signs are not reliably tagged in OSM, we integrate an interface to the Mapillary[7] API. Through access to this database, locations of more than 60 types of traffic signs can be integrated optionally into the map conversion and complement the data from OSM.

In the end, the geometric quality and amount of information in the converted maps strongly depend on the data in the underlying databases. The availability of additional data, such as traffic signs or *turn:lane* tags, usually correlates with the population density of the mapped area. Automatically assessing the integrity of the maps remains to be future work.

[7] `https://www.mapillary.com`

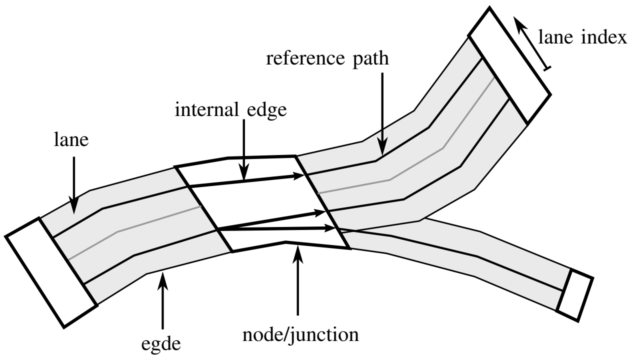


Fig. 6: Schematic visualization of the SUMO map representation [24].

E. CommonRoad to SUMO and Vice Versa

Traffic simulators are often used for creating scenarios based on provided maps and testing motion-planning algorithms during interaction with other traffic participants. One commonly used traffic simulator is SUMO, which can simulate traffic on large and complex road networks. An interface to CommonRoad enables a closed-loop simulation with a motion planner [24].

As an extension to this approach, we provide a map conversion from the CommonRoad format to the SUMO map representation. SUMO uses nodes, edges, and junctions to define road networks, where a node connects either two edges or embodies a junction. An edge connects two nodes and can represent multiple adjacent lanes, where a lane is defined by its width and a reference path is represented as a polyline. The order of lanes is specified by ascending indices from right to left. Internal edges specify the lane geometry within junctions (cf. Fig. 6).

We generate junctions and edges from the lanelet network for the conversion from CommonRoad to SUMO. They are provided to the SUMO tool *netconvert*, which generates the final SUMO network. Additionally, we are able to convert regulatory elements, such as stop lines, traffic lights, and selected traffic signs to SUMO, although SUMO does not support the latter directly. We support three traffic sign categories:

1) *Speed limit:* Speed limit signs are directly assigned to the corresponding SUMO lane.
2) *Vehicle ban:* A ban for a vehicle type is considered by removing the corresponding vehicle type from the list of the allowed vehicle types of an edge.
3) *Priority:* We assign priority values to edges for selected priority traffic signs. For example, edges related to a yield sign receive a lower priority than edges related to a right of way sign. The traffic simulation uses the assigned priorities to decide the traversal order of traffic participants within the junction.

We detect edges relevant to the traffic sign encoding using a breath-first search on succeeding lanes of the edge the sign corresponds to. We terminate the search when a traffic sign-specific stopping criterion is met, e.g., another speed limit or vehicle ban ends. Stop lines and traffic lights are directly converted to their SUMO representation, where either the CommonRoad traffic light cycles can be used or SUMO can simulate traffic light cycles for a junction, which are converted back to CommonRoad. The latter method can also be used by the other converters to define a traffic light cycle when the map does not provide the information.

Our framework can also convert a SUMO map to the CommonRoad format. Therefore, two steps are necessary: 1) the SUMO tool *netconvert* converts the SUMO map into the OpenDRIVE format; 2) our OpenDRIVE converter transforms the map into the CommonRoad format.

IV. Graphical User Interface

The central way of interacting with the CommonRoad Scenario Designer is a GUI to convert, create, and edit maps and scenarios. The GUI is visualized in Fig. 7 and provides several toolboxes, which can easily be extended:

1) *Map Conversion:* The converters presented in Sec. III can also be accessed from the GUI, where the conversion parameters can easily be changed. For the OSM conversion, one can directly download an OSM map based on a selected area and edit the road topology, i.e., a graph representation of the OSM map, before the conversion to the CommonRoad format starts.
2) *Road Network:* The GUI provides functionalities to add, remove, and update lanelets, traffic signs, traffic lights, and intersections. It provides parameterizable templates for adding lanelets and adding three-way and four-way intersections with or without traffic signs and lights. Additionally, the GUI provides functions for rotating, translating, and appending lanelets, and automatically forming a connection between lanelets, among others.
3) *Obstacle:* This toolbox allows users to create new obstacles manually, edit the type, width, and length of obstacles and to remove obstacles. Users can also inspect states of a selected obstacle. Additionally, traffic can be simulated and added to a map by executing a SUMO simulation.

The GUI also provides functionalities to save the rendered figures and videos for publications and presentations. It also allows editing scenario meta-data, such as planning problems for motion planning, location, time, or weather conditions. For simple interaction with the GUI, the user can click on visualized scenario elements to automatically select them for editing in the corresponding toolboxes.

V. Selected Results

Subsequently, we show conversion results for maps given in the OSM, SUMO, and CommonRoad format and manually generated maps using customizable elements from the GUI. Figs. 7 and 8 show manually-created intersection scenarios with traffic signs and traffic lights, respectively, at different time steps in the SUMO simulation [24]. The SUMO interface also simulates vehicle signals, e.g., turn indicators or braking lights. Fig. 9 shows the results from an OSM conversion of a complex intersection. We provide examples

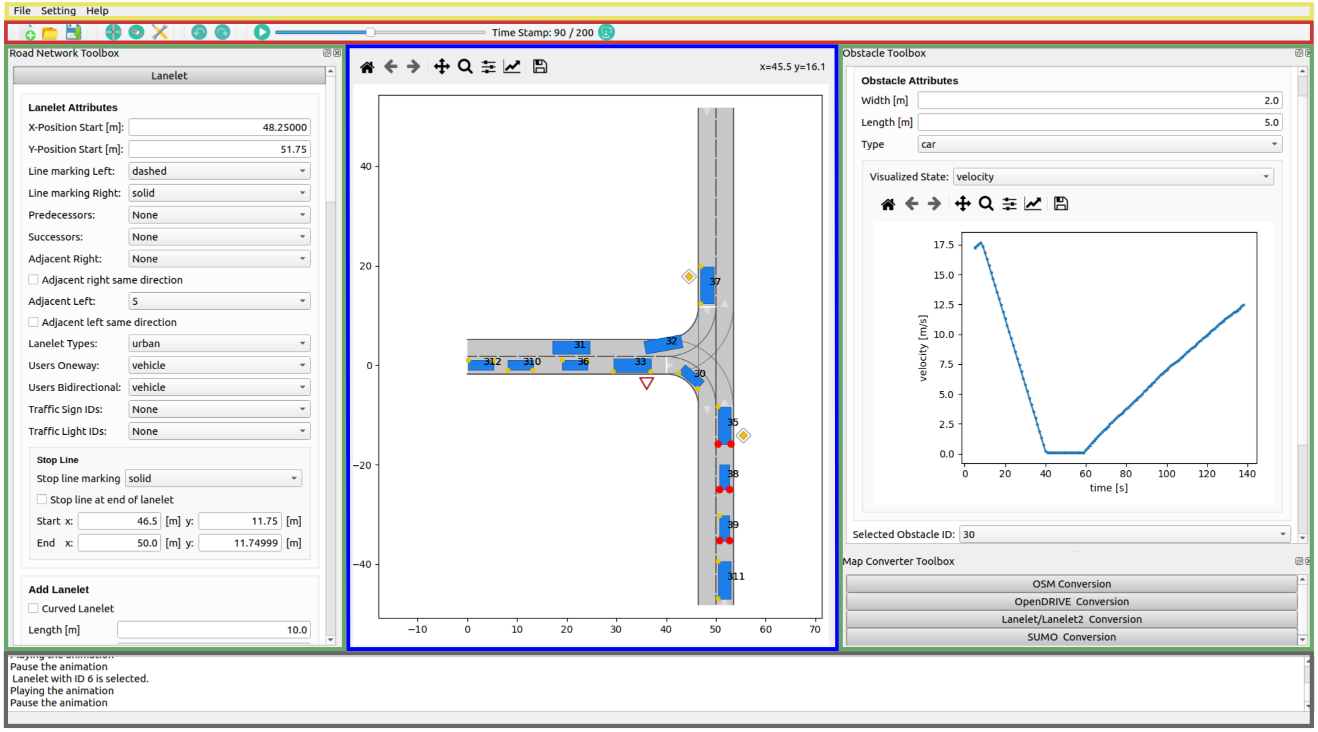


Fig. 7: The CommonRoad Scenario Designer GUI. The differently colored rectangle borders indicate the toolboxes (green), user information window (gray), visualized scenario (blue), toolbar (red), and menu bar (yellow).

of OSM conversion limitations in the code documentation. Fig. 10 shows the results from a conversion of a simple intersection from CommonRoad to SUMO and back to CommonRoad. The latter conversion also employs our presented OpenDRIVE converter.

Note that all figures showing CommonRoad scenarios are taken directly (only axes are removed) from the CommonRoad Scenario Designer. This shows the simple usage of the tool and allows researchers to quickly create figures for their publications. The supplementary video demonstrates the functionality of the GUI in more detail.

VI. Conclusions

We have presented a framework for converting and creating maps and scenarios for autonomous driving. The framework can convert different map formats into a common representation. It can also provide several interfaces for easy and fast usage. The toolbox is available under `https://commonroad.in.tum.de/`. In the future, we will add further conversions from the CommonRoad format to other formats, such as OpenDRIVE, so that CommonRoad scenarios can also be used in driving simulators like Carla. The OSM conversion can be improved by considering aerial images and integrating further road types. Additionally, an offline map validation can be integrated to capture errors in the original data, manually-created maps, or during map conversion.

Acknowledgment

We thank Benjamin Orthen, Fabian Höltke, Maximilian Frühauf, Maximilian Rieger, and Stefan Urban for their contribution to the different map converters and the graphical user interface. Furthermore, the authors gratefully acknowledge partial financial support by the BMW Group within the CAR@TUM project and the Central Innovation Programme of the German Federal Government under grant no. ZF4086007BZ8.

References

[1] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highD dataset: A drone dataset of naturalistic vehicle trajectories on German highways for validation of highly automated driving systems," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2018, pp. 2118–2125.

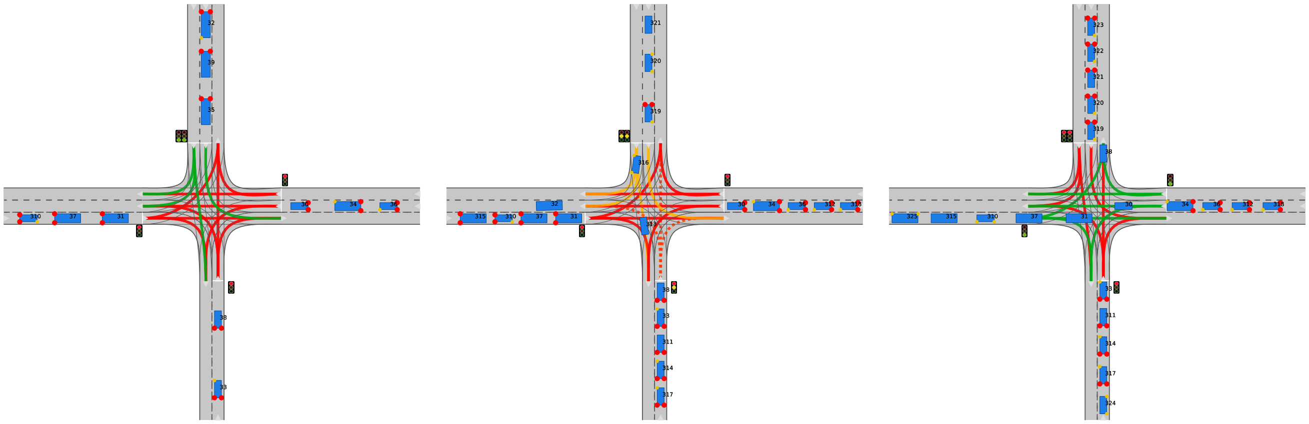
[2] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater, and L. Eckstein, "The inD dataset: A drone dataset of naturalistic road user trajectories at german intersections," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2020, pp. 1929–1934.

[3] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kümmerle, H. Königshof, C. Stiller, A. de La Fortelle, and M. Tomizuka, "INTERACTION Dataset: An INTERnational, Adversarial and Cooperative moTION Dataset in Interactive driving scenario with semantic maps," *arXiv preprint arXiv:1910.03088*, 2019.

[4] E. Barmpounakis, G. M. Sauvin, and N. Geroliminis, "Lane detection and lane-changing identification with high-resolution data from a swarm of drones," *Transportation Research Record*, vol. 2674, no. 7, pp. 1–15, 2020.

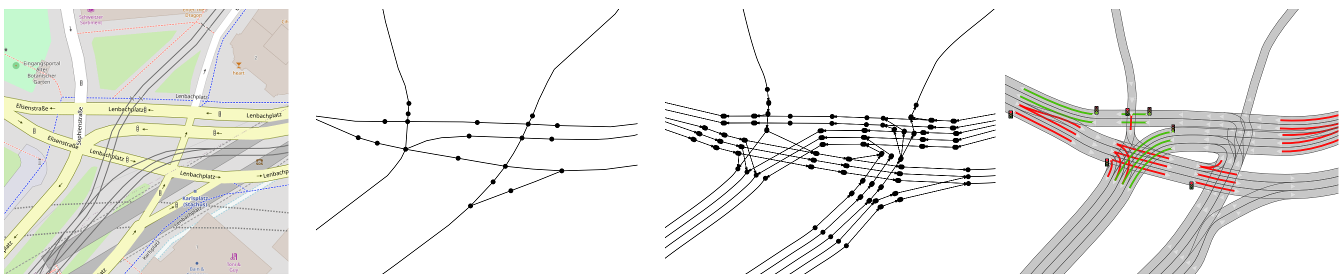
[5] A. Breuer, J. A. Termhlen, S. Homoceanu, and T. Fingscheidt, "openDD: A large-scale roundabout drone dataset," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2020, pp. 1–6.

[6] V. Kovvali, V. Alexiadis, L. Zhang, and P. Length, "Video-based vehicle trajectory data collection," in *Proc. of the Transportation Research Board 86th Annual Meeting*, 2007, pp. 1–18.



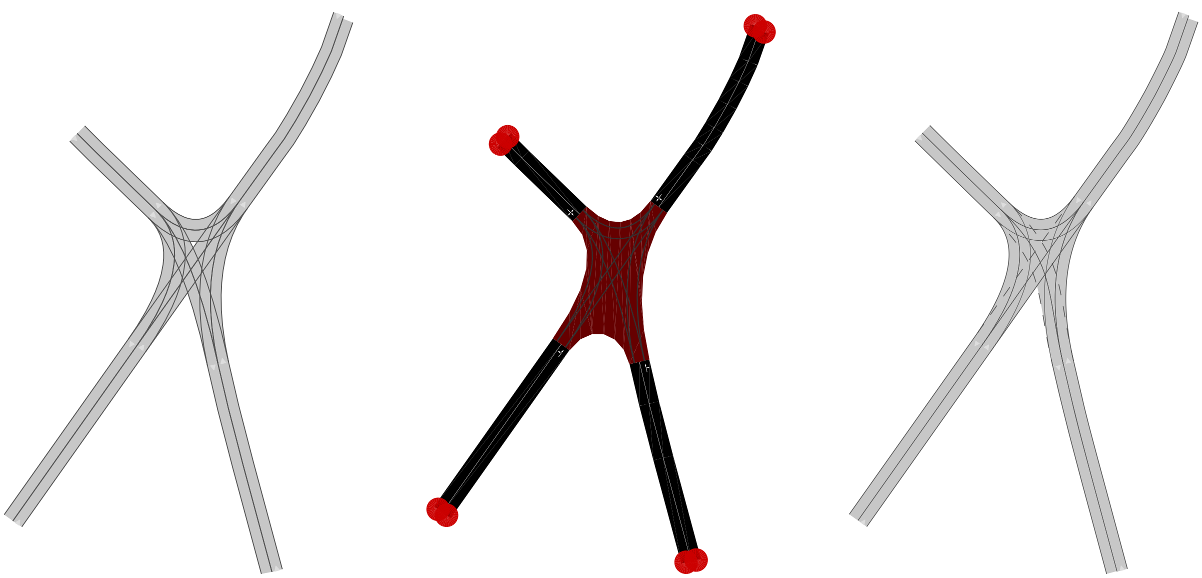
(a) Time $t = 0.0\,\mathrm{s}$. (b) Time $t = 11.0\,\mathrm{s}$. (c) Time $t = 26.5\,\mathrm{s}$.

Fig. 8: Manually-created four-way intersection with a customized traffic light cycle.



(a) Original map in OSM. (b) Road network topology given by OSM ways. (c) Linked lanes. (d) Converted CommonRoad map with traffic lights (red and green lines).

Fig. 9: Exemplary OSM conversion of a map snippet in Munich, Germany.



(a) Original CommonRoad map. (b) Converted SUMO map visualized with the SUMO tool *netedit*. (c) Converted CommonRoad map.

Fig. 10: Conversion of a CommonRoad map to SUMO and back to CommonRoad. Note that during the conversion the line marking type and the shape of lanelets are slightly updated (CommonRoad benchmark ID using this map: BEL_Putte-1_3_T-1).

[7] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2443–2451.

[8] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wiessner, "Microscopic traffic simulation using SUMO," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2018, pp. 2575–2582.

[9] G. Rong, B. H. Shin, H. Tabatabaee, Q. Lu, S. Lemke, M. Možeiko, E. Boise, G. Uhm, M. Gerow, S. Mehta, *et al.*, "LGSVL simulator: A high fidelity simulator for autonomous driving," *arXiv preprint arXiv:2005.03778*, 2020.

[10] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[11] T. Stahl and J. Betz, "An open-source scenario architect for autonomous vehicles," in *Proc. of the Int. Conf. on Ecological Vehicles and Renewable Energies*, 2020, pp. 1–8.

[12] K. Tong, Z. Ajanovic, and G. Stettinger, "Overview of Tools Supporting Planning for Automated Driving," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2020, pp. 2950–2957.

[13] F. Poggenhans, J. H. Pauls, J. Janosovits, S. Orf, M. Naumann, F. Kuhnt, and M. Mayr, "Lanelet2: A high-definition map framework for the future of automated driving," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2018, pp. 1672–1679.

[14] M. Althoff, M. Koschi, and S. Manzinger, "CommonRoad: Composable benchmarks for motion planning on roads," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2017, pp. 719–726.

[15] S. M. Azimi, P. Fischer, M. Krner, and P. Reinartz, "Aerial lanenet: Lane-marking semantic segmentation in aerial imagery using wavelet-enhanced cost-sensitive symmetric fully convolutional neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 5, pp. 2920–2938, 2019.

[16] L. Zheng, B. Li, B. Yang, H. Song, and Z. Lu, "Lane-level road network generation techniques for lane-level maps of autonomous vehicles: A survey," *Sustainability*, vol. 11, no. 16, 2019.

[17] M. Haklay and P. Weber, "OpenStreetMap: User-generated street maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, 2008.

[18] M. A. Brubaker, A. Geiger, and R. Urtasun, "Map-based probabilistic visual self-localization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 4, pp. 652–665, apr 2016.

[19] K. Wong, E. Javanmardi, M. Javanmardi, Y. Gu, and S. Kamijo, "Evaluating the Capability of OpenStreetMap for Estimating Vehicle Localization Error," in *Proc. of the IEEE Intelligent Transportation Systems Conf.*, 2019, pp. 142–149.

[20] J. M. Álvarez, A. M. López, T. Gevers, and F. Lumbreras, "Combining Priors, Appearance, and Context for Road Detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 3, 2014.

[21] J. Godoy, A. Artuñedo, and J. Villagra, "Self-generated OSM-based driving corridors," *IEEE Access*, vol. 7, pp. 20 113–20 125, 2019.

[22] D. Krajzewicz, G. Hertkorn, and J. Ringel, "Preparation of digital maps for traffic simulation; Part 1: Approach and algorithms," in *Proc. of the Industrial Simulation Conference*, 2005, pp. 285–290.

[23] M. Althoff, S. Urban, and M. Koschi, "Automatic conversion of road networks from opendrive to lanelets," in *Proc. of the IEEE Int. Conf. on Service Operations and Logistics, and Informatics*, 2018, pp. 157–162.

[24] M. Klischat, O. Dragoi, M. Eissa, and M. Althoff, "Coupling SUMO with a motion planning framework for automated vehicles," in *SUMO: Simulating Connected Urban Mobility*, 2019, pp. 1–9.

[25] M. Klischat, E. I. Liu, F. Höltke, and M. Althoff, "Scenario factory: Creating safety-critical traffic scenarios for automated vehicles," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2020, pp. 2964–2970.

[26] P. Bender, J. Ziegler, and C. Stiller, "Lanelets: Efficient map representation for autonomous driving," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2014, pp. 420–425.