Painted papers category

Road2Simulation Guidelines

Guidelines for acquisition of road data for simulation and development

DLR

Document properties

Title	Road2Simulation Guidelines
Subject	Guidelines for acquisition of road data for simulation and development
Institute	Transportation Systems
Compiled by	Andreas Richter, Michael Scholz
Participants	
Checked by	Michael Scholz, Laura Fischer, Hartmut Friedl
Release by	Michael Scholz
Date	September 6, 2019
Version	1.2.1
File Path	

Contents

1. Motivation

Digital and highly accurate road map data represent an indispensable prerequisite for the development and assessment of modern advanced driver assistance and automation systems. They are utilized as input for simulation in acceptance and safety tests, and as complementary knowledge on board the vehicles. For simulation, specific description formats are used, which allow for a very precise modelling of the environment. However, the acquisition of real data and their conversion into simulation formats may quickly become very time-consuming and cost-intensive. The digital cadastre data already gathered usually contains only few data that facilitate conversion into the required simulation formats.

To reduce the efforts for surveying routes and converting cadastre data into simulation formats, the present Guidelines have been prepared for a standardized surveying of road and surface data. The Guidelines follow the road description format OpenDRIVE®[1] and the surface description format OpenCRG®[2], but do not require any knowledge of the description formats. A look into the corresponding specifications might be useful – the documents are freely available on the related websites.

1.1. Licence

The Road2Simulation Guidelines and the data model are licenced under Creative Commons Attribution 4.0 International (CC BY 4.0)[3]. The aerial photos used in examples are owned by the City of Braunschweig, Fachbereich Stadtplanung und Umwelschutz (Department for Urban Planning and Environmental Protection)[4], and the City of Stuttgart, Stadtmessungsamt (City Surveying Department)[5].

1.2. Online version

The online version of these Guidelines is available at
`https://doi.org/10.5281/zenodo.3375525`.

1.3. German translation

A German translation of these Guidelines is available at
`https://doi.org/10.5281/zenodo.3375550`.

[1] http://www.opendrive.org
[2] http://www.opencrg.org
[3] https://creativecommons.org/licenses/by/4.0/
[4] http://www.braunschweig.de/geoinformation
[5] http://www.stuttgart.de/stadtmessungsamt

2. General definitions

2.1. Exchange format

2.1.1. Road description

The surveyed data should be stored in an open format that supports geometry representation as OGC Simple Features (see chapter 2.3). UTF-8 is to be used as encoding. For instance, the following exchange formats are recommended:

- Simple database dump from spatial databases, such as PostGIS, Oracle Spatial or SpatiaLite.
- OGC GeoPackage[6] as SpatiaLite container. SpatiaLite serves as spatial expansion of SQLite. SQLite as file-based database is optimally suited for data exchange and offers a widely spread database functionality. Several spatial tables can be stored in one file in this way, and their interrelations can be represented. The migration between SpatiaLite and other spatial database systems is ensured by freely available tools. GeoPackage is supported by many geographical information systems and allows the definition of frequently used coordinate reference systems. Moreover, in addition to vector data also raster data can be recorded in a GeoPackage.
- ESRI Shapefile is an exchange format suitable only with some reservations, since it has restrictions with regard to the field names (limited to ten characters) and the value ranges of data types.

It is generally useful to deliver the underlying elevation model of the 3D Simple Features as well, since in work steps that change the geometries subsequently (e.g., interpolation, computational correction) the elevation data may become invalid and have to be newly interpolated. It is possible, for example, to include elevation models (grid/vector) in the same GeoPackage as the Road2Simulation objects.

Figure 1 shows the structure of the data model as it can be depicted in a database. Chapter 3 describes in detail the procedure applied for acquisition of the road elements and how these are to be stored. The attributed geometry data are supplemented by meta information, which is explained in section 2.6.

2.1.2. Surface description

The data measured shall be binary-stored in OpenCRG format, since a more complex data transformation as the one needed for the course of the road into OpenDRIVE format is not required. All further important boundary conditions are described in chapter 4.

[6] http://www.geopackage.org/

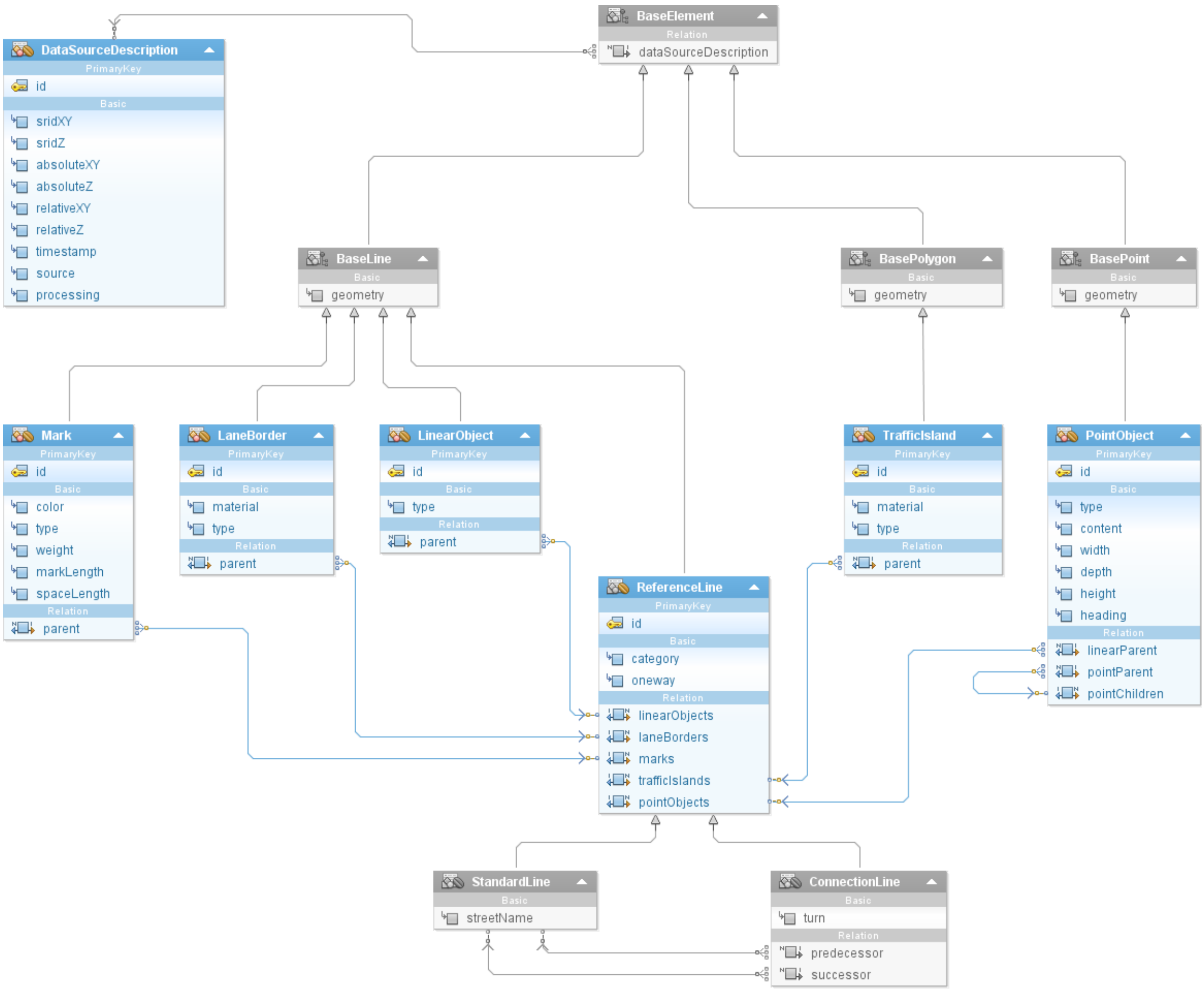


Figure 1: Structure of the data model. Persistent tables are represented as coloured classes, whereas grey elements serve for hierarchical structuring.

2.2. Coordinate system and spatial reference

All geometric measuring points contain three-dimensional position information (`x`, `y`, `z`). For example, the coordinates can be indicated in a *projected* coordinate reference system and would be composed as follows:

- `x` in metres, as easting in the related projection,
- `y` in metres, as northing in the related projection,
- `z` in metres, as elevation above the underlying reference ellipsoid, or as elevation in a deviating elevation reference system.

The selection of the spatial coordinate reference system based on projected coordinates is guided by official data; for Central Germany, it is opted for ETRS89 / UTM Zone 32N (EPSG:25832[7]). If the elevation is indicated deviating from the reference ellipsoid, the elevation reference system used is to be additionally indicated, for example an elevation above sea level in DHHN92 (EPSG:5783[8]).
If the coordinates are indicated in a *geographic* reference system, the position information is composed as follows:

- `x` in degrees as longitude,
- `y` in degrees as latitude,
- `z` in metres as elevation above the underlying reference ellipsoid.

The selection of the geographic coordinate reference system is guided by official data; for the European region, it is usually opted for ETRS89 (EPSG:4258[9]) or WGS84 (EPSG:4326[10]).

It is generally recommended referencing the data in a *projected* coordinate reference system. The designation of the selected reference system is explicitly specified as part of the data source description, as explained in section 2.6.1.

Rotations are only indicated as geographically absolute values, i.e. a left-hand system is used (clockwise positive). This means, the angles are N = 0°, O = 90°, S = 180°, W = 270°.

2.3. Geometry data types

Geometric data types are to be indicated *three-dimensionally* in the defined reference system as simple features in well-known binary (WKB) or well-known text (WKT) format, in accordance with OGCSimple Feature Access – Part 1: Common Architecture, Version 1.2.1[11].

Example for a point-shaped 3D object as WKT:

- `Point Z (605044.419819 5791949.77898 128.12)`

Example for a linear 3D object with two control points as WKT:

- `LineString Z (`
  `605059.7409 5791956.6 128.12,`

[7] `http://epsg.io/25832`
[8] `http://epsg.io/5783`
[9] `http://epsg.io/4258`
[10] `http://epsg.io/4326`
[11] `http://www.opengeospatial.org/standards/sfa`

```
605014.3896 5791935.49 128.22)
```

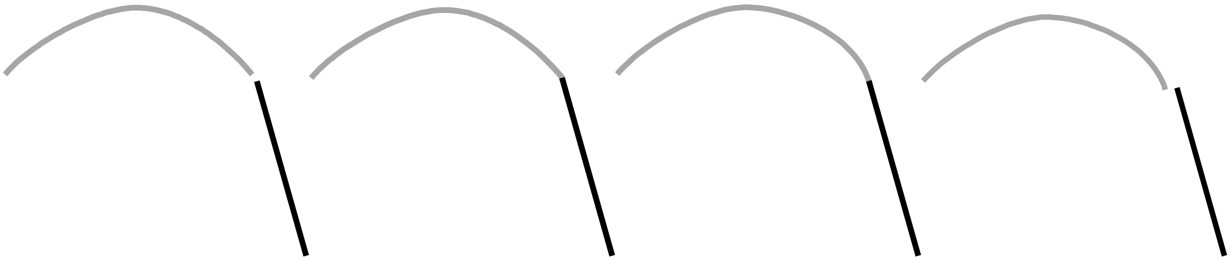
Example for an areal 3D object with four control points as WKT:

- `Polygon Z ((`
  `606556.8 5797302.2 129.7, 606563.5 5797301.6 129.7,`
  `606563.0 5797297.2 129.7, 606556.5 5797297.7 129.7,`
  `606556.8 5797302.2 129.7))`

2.4. Topological integrity

For an easier post-processing of the data collected, it has to be ensured that topologically interrelating objects are modelled geometrically consistently. Overlapping objects share identical geometric information at the overlap points. For example, point data, like signs fixed at the same post, will differ geometrically only in the `z` component of their position information or in their orientation.

Topologically successive polylines have to fulfil the point continuity (C^0 continuity), i.e. they have to share the same geometric point at the junction. The modelling of polylines shall also fulfil, if possible, the tangent continuity (C^1 continuity). Some examples presented later will deal with successive polylines that are parallel but offset and which have to provide, if possible, the same tangential gradient at their topological junctions. Figure 2 shows geometric differences of these continuity conditions.



(a) (b) (c) (d)

Figure 2: Geometric transition of successive polylines: discontinuous (a), C^0-continuous (b), C^1-continuous (c), parallel offset with the same tangential gradient (d).

2.5. Types of numbers

Integers should generally be indicated as `uint64_t`, i.e. only positive numbers from `0` to $2^{64}-1$ `= max`. Floats (floating-point numbers), in particular for indication of coordinates, should generally be indicated as `Double`. The numbers are normally positive, ranging within the limits from `0.0` to $(2-2^{-52})\cdot 2^{1023}$ `= max`.

2.6. Data source description

For an easier interpretation and quality description of the data stock collected, the data are to be supplemented by meta information as described in the following. For data deliveries, this is realized by allocating a corresponding `DataSourceDescription`object (abbreviated below with `dSD`) to each geometric object. Table 1 shows examples of possible meta information – the entries will be described in the following sections.

Table 1: Example of meta information describing geometric data records.

DataSourceDescription									
id	srid XY	srid Z	absoluteXY	absoluteZ	relativeXY	relativeZ	timestamp	source	processing
0	4647		0.2	0.5	0.01	0.01	2012-12-12	mobile mapping	processed
1	4647	5783	0.5	1.0	0.25	0.25	2011-11-11	cadaster	fused

2.6.1. Spatial reference

As described in section 2.2, the spatial reference for the geometric data collected is clearly defined. The European Petroleum Survey Group (EPSG) describes frequently used reference systems (geographic, projected, vertical) by using unique identifiers, so-called Spatial Reference Identifier (SRID). For the coordinate components `xy` and `z`, the EPSG-SRID describing them are to be indicated separately:

- `sridXY`
- `sridZ`

with `sridZ` being optional – it has to be indicated only if the elevation deviates from the ellipsoid already defined by `sridXY`.

2.6.2. Errors

The relative and absolute error of the `xy` and `z` components acquired is to be indicated as `Double`. The unit depends on the respective coordinate reference system used (see sections 2.2 and 2.6.1), which is normally SI-metre (m) or SI-degree (°).

- `absoluteXY`
- `absoluteZ`
- `relativeXY`
- `relativeZ`

Here, the relative error corresponds to precision, and the absolute error to trueness. If both errors are low, the accuracy is high[12].

[12] https://en.wikipedia.org/wiki/Accuracy_and_precision

2.6.3. Timestamp

The `timestamp` of a data delivery is to be indicated as ISO 8601 date (`YYYY-MM-DDTHH:MM:SS`). The indication of the time of day `T` is optional.

2.6.4. Quality

By indicating the source, natural-language information about the origin and processing of the data shall be given. The `source` and the type of `processing` are indicated.

2.6.4.1. Source

This indication may consist, for example, of several sources for different sensors of different firmware versions, or from different cadastres. The indication according to the category is supplemented with designation/name as `String`.

- sensor name and firmware version (may also be subsumed as "mobile mapping") = `sensor`
- cadastre data = `cadaster`
- generated or digitized data without using sensors = `custom`

2.6.4.2. Post-processing

This indication is to define, whether and in which way the data were post-processed. The following categories are indicated as `String`:

- directly from the sensor = `raw`
- cleaned from outliers and imperfections = `cleaned`
- values have already been clustered, averaged and transformed = `processed`
- values come from different data sources = `fused`

2.7. Scope of mapping

These Guidelines are subdivided into a standard scope of mapping and an extended scope of mapping, with the attributes of the extended scope of mapping being highlighted in grey:

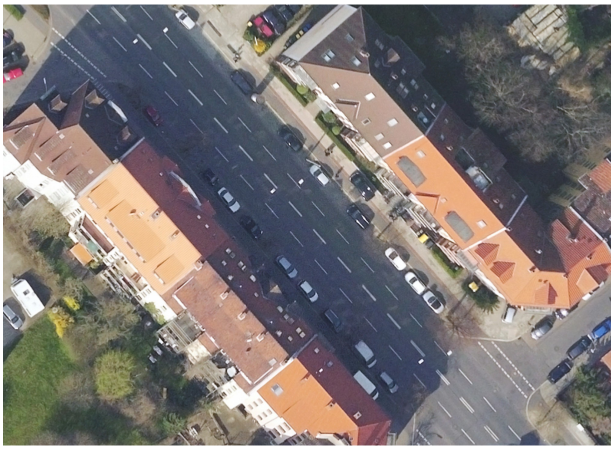
- standard scope of mapping
- extended scope of mapping

3. Road description

The road description normally comprises topological and topographical information. The topological data create a node-edge model linking all elements to each other via the predecessor-successor relationship, which partly also contains information with influence on the roads use (e.g., signs). The topographical data describe the appearance of the road, by modelling, for example, edges, the type of road marking, and much else.

The elements described in these Guidelines below will first be represented in six overview pictures. Table 2 shows the road situations used for this without modelled elements. The situations shown in Table 2 can be used as comparison for the following overviews, in which the respective elements are marked.

Table 2: Different road situations without modelled elements.

	
single roadway, several lanes per direction	single roadway, one lane per direction
	
two roadways, several lanes per direction	single roadway, one lane in one direction

single roadway, one lane per direction with separation

several roadways, different lanes per direction

3.1. Overview

All important elements of the road topography are modelled as connected polylines (e.g., course of the road, curbstones, road markings) or single coordinates (e.g., street lighting, signposts, traffic lights). Central islands are exceptions, since their construction results from several roads joining. Central islands are therefore modelled as areas/polygons.

The polylines (LineStrings as Simple Features[13]) can be easily acquired and stored. For precise and efficient modelling, the control points of a polyline should be as dense as possible in the curvature area and in straight sections as widely spread as possible. The direction of the polylines is implicitly defined by the order of the control points. The absolute deviation of the road topography should not exceed 100 mm within the northing and easting for cadastre applications. For application in simulation, the two-dimensional error should maximally range between 20 mm and 50 mm. The error of elevation should not deviate from this too strongly. This means, the main classes `StandardLine`, `ConnectionLine` and `LaneBorder` are basically similar and to be distinguished only by specific attributes.

Significantly structuring elements are streets and crossroads. The elements are derived from the OpenDRIVE specification. Streets lead from crossroad to crossroad, no street can connect directly to another street. The street describes all characteristics of the section; it thus contains information about roadway and infrastructure. Streets are modelled using `StandardLines`.

In these Guidelines, a crossroad is to be understood as logical container for summarizing all links. Within a crossroad, the arriving streets are connected via links that behave similarly to streets, but with some restrictions. These links are modelled as `ConnectionLines`. Figure 3 shows an overview with several crossroads connected by streets. Within these crossroads, all necessary connections are modelled using links.

[13] `http://www.opengeospatial.org/standards/sfa`

Figure 3: Overview of street sections (green), crossroads (blue) and therein contained links (yellow).

3.2. Course of roads

The course of a road is modelled with position and elevation by means of the so-called `StandardLine`. A `StandardLine` for roads is laid without structural separation in the middle of the road, where the two lanes of both directions of travel touch each other. In case of structural separations (e.g. a median strip) or one-way roads, the `StandardLine` is laid at the lane limitation (outer edge) to the left in the direction of travel. Within crossroads, the `StandardLines` are referred to as `ConnectionLines`, which model the possible turning options of a road.

`StandardLines` and `ConnectionLines` represent the core elements of the entire data model and serve as *reference objects* for the logical allocation of other objects to be mapped. `StandardLine` and `ConnectionLine` use the same ID range in the data model (see figure 1). Therefore, `StandardLines` and `ConnectionLines` cannot have identical IDs.

Each `StandardLine` and each `ConnectionLine` has a direction. Table 4 shows modelling examples with `StandardLines` in green and `ConnectionLines` in yellow. The arrowheads indicate the explicit direction. No arrowhead means both directions are possible. Moreover, `StandardLines` and `ConnectionLines` have a road category assigned, from which further characteristics can be implicitly deduced in further machine processing steps.

3.2.1. StandardLine

As described in section 3.2, the `StandardLine` models the course of a road between crossroads. Therefore, a road category is allocated to the `StandardLine`, which may have the following attributes:

- unspecified = `unknown`
- rural road = `rural`
- freeway/motorway = `motorway`
- city = `town`
- 20 mph zone = `low speed`
- living street = `pedestrian`
- bicycle street = `bicycle`

Moreover, the `oneway` attribute defines, whether the road section can only be driven in modelling direction (`oneway = true`), or in both directions (`oneway = false`). In addition, a name (e.g. "Bahnhofstraße") or a designation ("L118") can be allocated to the course of a road. This means, a `StandardLine` is described by the attributes listed in table 3. Table 5 show a modelling example for `StandardLines` with table outline.

Table 3: Attributes of `StandardLines`.

Name	Number	Type	Value	Description
id	1	unit64_t	`[0, max]`	unique identifier
dataSource-Description	1	unit64_t	`[0, max]`	ID of the describing `Data-SourceDescription` object
geometry	1	LineString Z	see example in section 2.3	polyline as WKT or WKB
category	1	string	`unknown, rural, motorway, town, low speed, pedestrian, bicycle`	road category
oneway	1	boolean	`true, false`	`true`, if `StandardLine` is a one-way road, otherwise `false`
streetName	0..1	string		street name/designation

Table 4: Modelling of `StandardLines` (green) and `ConnectionLines` (yellow) in different situations.

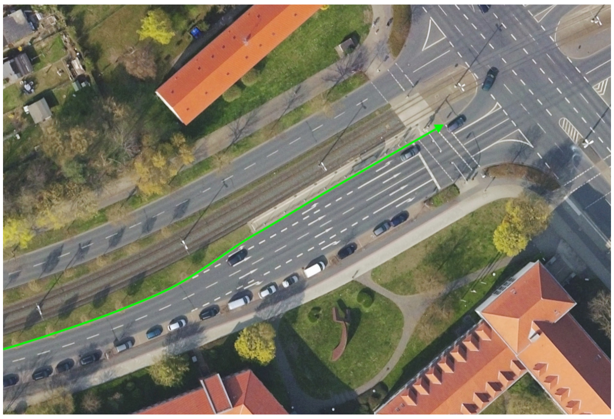
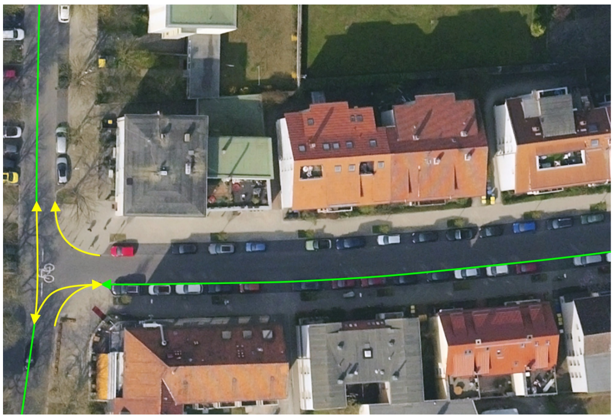
	
single roadway, several lanes per direction	single roadway, one lane per direction
	
two roadways, several lanes per direction	single roadway, one lane in one direction
	
single roadway, one lane per direction with separation	several roadways, different lanes per direction

Table 5: Example of two `StandardLines` (green) with table extract.

ReferenceLine (StandardLine)					
id	dSD	geometry	category	oneway	streetname
100	0	WKT/WKB	low speed	false	Broitzemer Str.
101	0	WKT/WKB	low speed	true	Virchowstr.

3.2.2. ConnectionLine

`ConnectionLines` model the turning options on crossroads. So there is information about the type of turning option and about possible predecessors and successors. Predecessors and successors are defined with reference to the geometric line direction, i.e. to their point sequence. In the topological model, a `ConnectionLine` creates a logical linking between two `StandardLines`, which means that in such a case, it always has exactly *one* predecessor and *one* successor.

When an opposite turning option can be modelled by *the same* `ConnectionLine`, a bidirectional link is modelled via the attribute `oneway` = `false`. In this way, redundant geometry can be avoided.

If there is no logical relation to connected `StandardLines`, a `ConnectionLine` can be simply used as reference for the geometric modelling of further `LaneBorders` in the crossroad area. An example for such situations is shown in table 9 at the bottom left and right.

Analogous to the StandardLine, the direction of travel and the road category are also indicated for

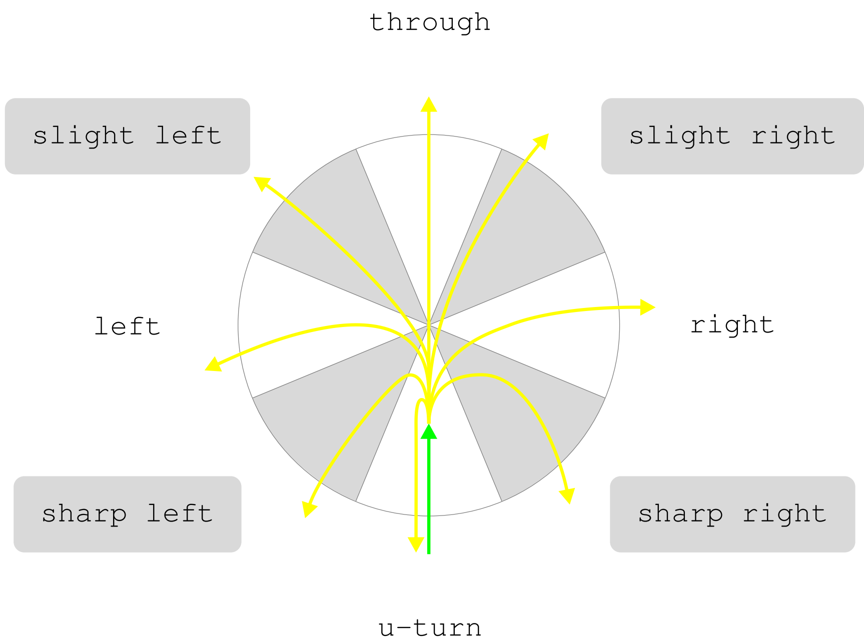


Figure 4: Turning options.

the ConnectionLine. Here, the definition of the type of turning option is added as further attribute `turn`:

- straight forward = `through`
- left = `left`
- right = `right`
- U-turn = `u-turn`
- auxiliary/helper `ConnectionLine` = `auxiliary`

- slight left = `slight left`
- sharp left = `sharp left`
- slight right = `slight right`
- sharp right = `sharp right`

The allocation of the turning option follows a full-circle scheme subdivided into eight segments, as shown in figure 4. Depending on the situation, the allocation to the segments may be considered less sharply. A `ConnectionLine` is therefore described by the attributes in table 6. Table 7 shows a modelling example of one `ConnectionLine` with table extract.

Table 6: Attributes of `ConnectionLines`.

Name	Number	Type	Value	Description
id	1	unit64_t	`[0, max]`	unique identifier
dataSource-Description	1	unit64_t	`[0, max]`	ID of the describing `Data-SourceDescription` object
geometry	1	LineString Z	see example in section 2.3	polyline as WKT or WKB
category	1	string	`unknown, rural, motorway, town, low speed, pedestrian, bicycle`	road category as for `Standard-Line`
oneway	1	boolean	`true, false`	`false`, if `ConnectionLine` represents a bidirectional link, otherwise `true`
turn	1	string	`through, u-turn, left, slight left, sharp left, right, slight right, sharp right, auxiliary`	
predecessor	1	unit64_t	`[0, max]`	ID of the preceding element (incoming `StandardLine`)
successor	1	unit64_t	`[0, max]`	ID of the succeeding element (outgoing `StandardLine`)

Table 7: Example of a `ConnectionLine` (yellow) with table extract.

ReferenceLine (ConnectionLine)							
id	**dSD**	**geometry**	**category**	**oneway**	**turn**	**predecessor**	**successor**
103	0	WKT/WKB	low speed	true	right	100	101

3.3. Crossroads

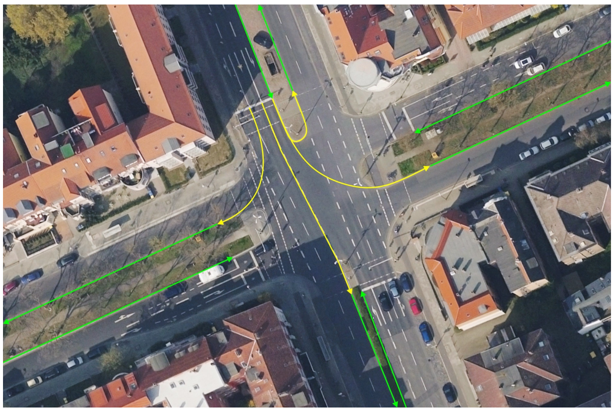
Crossroads are required everywhere, where the number of `StandardLines` to be linked changes. This may be a classical crossing situation (the course of the road continues to the left, right or straight forward), but also the joining or leaving of a motorway (see table 4).

`ConnectionLines`, too, are modelled on the left side of the left-most lane. Correspondingly, not all the links start and end at the same geometrical position as the incoming or outgoing `StandardLine` starts or ends, but they may be modelled orthogonally displaced. This means, `ConnectionLines` start and end *on the same level* (maximum longitudinal expansion/extent) relative to the `StandardLine` (see also section 3.4). If a `ConnectionLine` connects directly to a `StandardLine`, the junction has to be C^1-continuous, if possible, as explained in section 2.4.

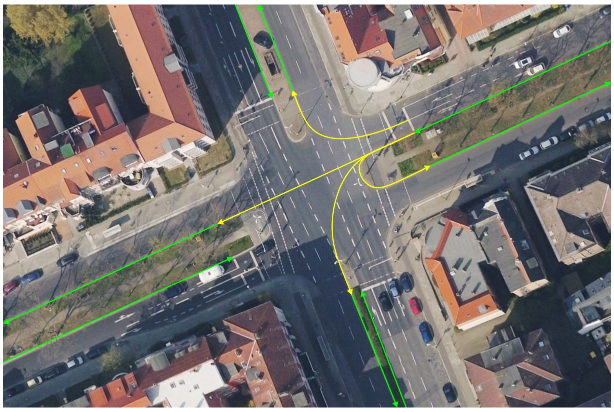
When modelling the `ConnectionLines`, attention has to be paid that they correspond to the humans' natural driving behaviour, unless there are already reference points such as road markings for modelling. Table 8 shows modelling examples of `ConnectionLines` in crossroads.

A special case are U-turn possibilities, which should always be modelled for crossroads with structural separation, even if they are possibly technically not realizable or would be against the rules. A `ConnectionLine` of the `u-turn` type is needed for modelling and localization of `LaneBorders` or `TrafficIslands`.

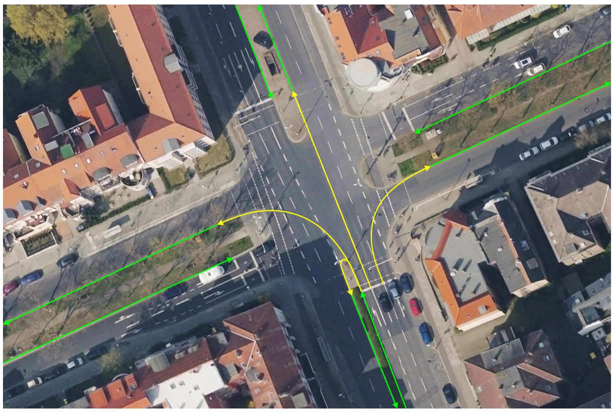
Table 8: `ConnectionLines` in the crossroad area.



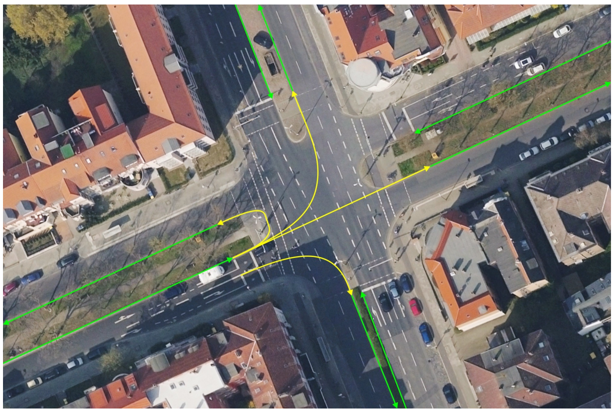
`ConnectionLines` starting from the north



`ConnectionLines` starting from the east



`ConnectionLines` starting from the south



`ConnectionLines` starting from the west

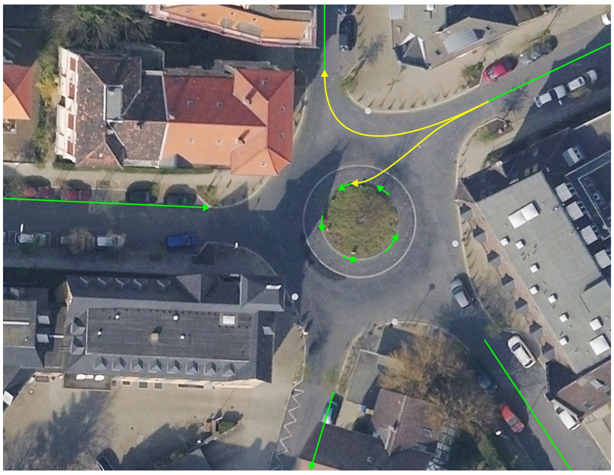
Complex crossroads have to be subdivided into smaller units to keep the modelling complexity low. This usually works well, where structural separations exist, e.g. traffic islands in traffic circles. Figure 5 shows an example for a subdivided crossroad; section 3.3.1 describes traffic circles/ roundabouts in more detail.

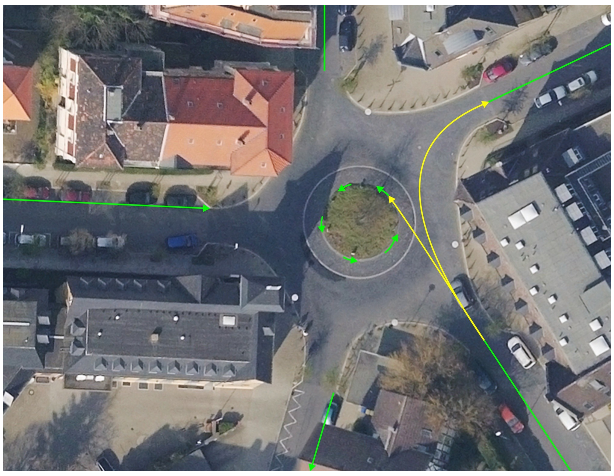
3.3.1. Roundabouts

To subdivide the roundabout, one segment each on the circle lying opposite the merging traffic should be defined as short `StandardLine`. In this way, turning options can be easily modelled. Table 9 demonstrates the subdivision of a roundabout.

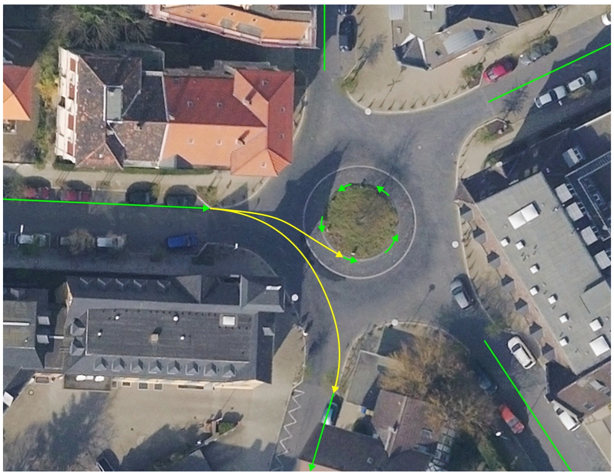
Figure 5: Complex crossroads subdivided into several simpler single crossings.

Table 9: `ConnectionLines` in the subdivided roundabout.

	
`ConnectionLines` starting from the centre	`ConnectionLines` starting from the north-east

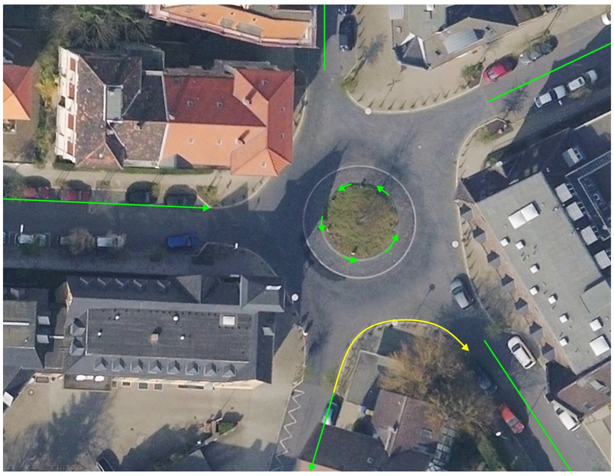


ConnectionLines starting from the south-east



ConnectionLines starting from the west

ConnectionLines starting from the north



ConnectionLine starting from the south-west

The last two pictures in table 9 for the directions north and south-west represent a special case for the later definition of the lanes (see section 3.4): Since under normal circumstances, ConnectionLines are not defined against the direction of travel of one-way roads, auxiliary ConnectionLines have to be created for modelling possible shoulders (sidewalks, bicycle lanes, or similar), in order to logically allocate these shoulders nonetheless.

3.3.2. Motorway accesses and exits

In case of motorway accesses and exits, two (directed) StandardLines join up, like at crossroads, which means they are thus to be linked using directed ConnectionLines – as shown in figure 6. These ConnectionLines are modelled as through turning options, and the change of the lane in the simulation application takes place onto the lane of the succeeding or preceding StandardLines.

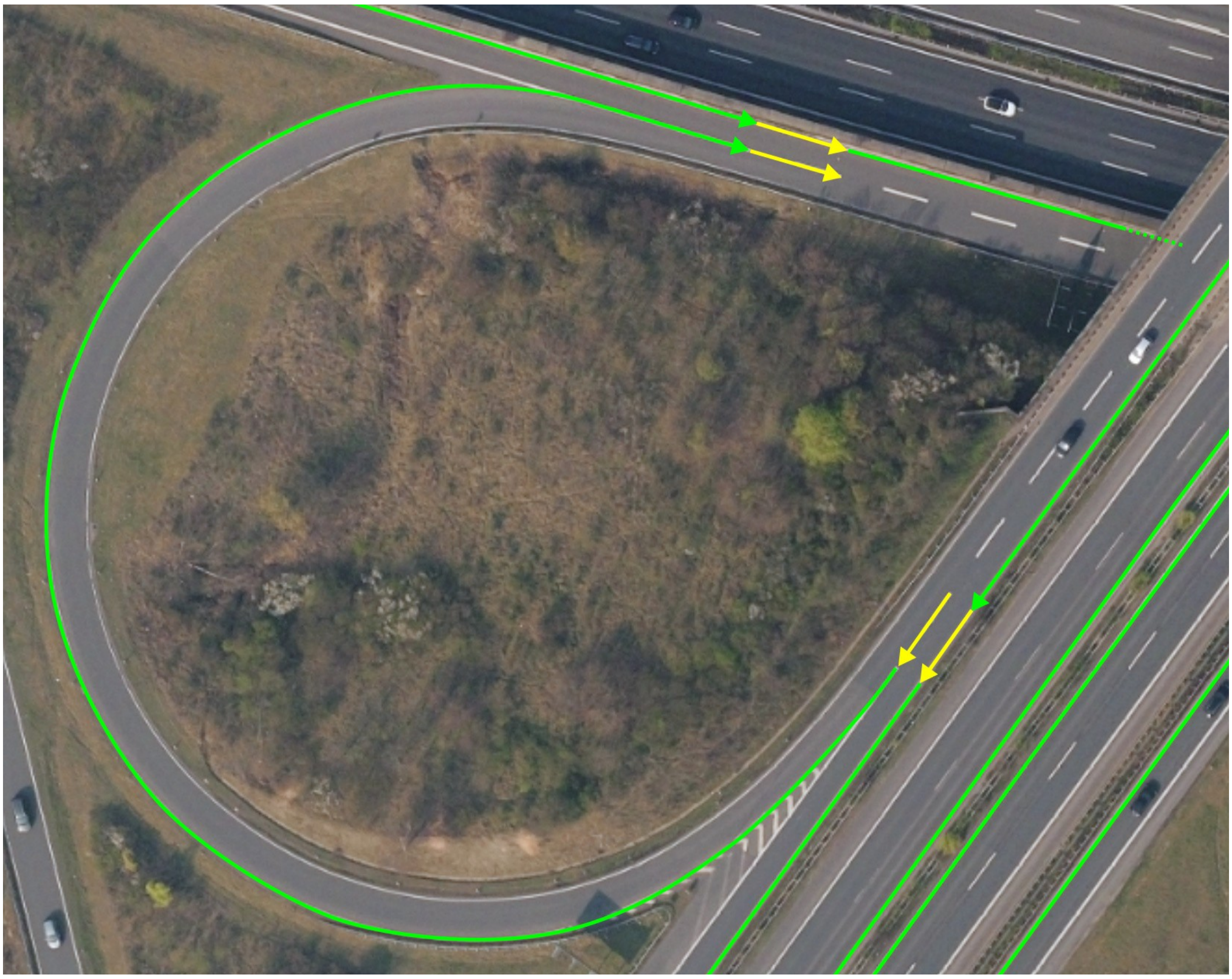


Figure 6: `StandardLines` and `ConnectionLines` for motorway accesses and exits.

3.4. Lanes of roadways

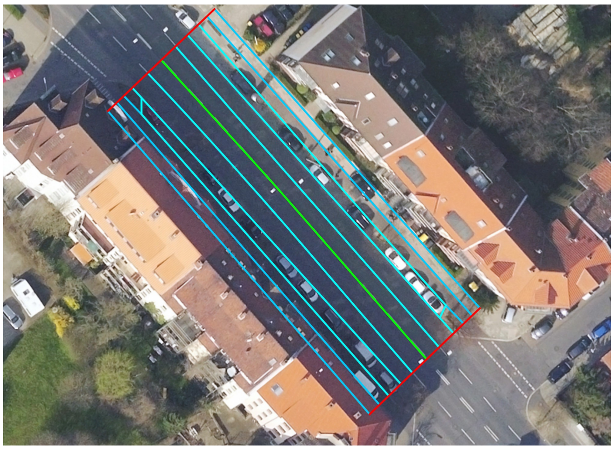
The appearance of the road is defined by the roadway with its lanes and their widths. Roadways may have lanes (taxi and bus lanes), parking lanes and shoulders, restricted areas, bicycle lanes, sidewalks, grass verges, tram routes, and traffic islands. Here, the (physical) boundaries represent the essential information that is adopted as polyline. These lines are called `LaneBorders`. `StandardLines` always have at least one `LaneBorder`, whereas `ConnectionLines` may also go without a `LaneBorder`, but normally, modelling of `LaneBorders` is equally useful here.

At crossroads, the `StandardLines` are continued as `ConnectionLines`. However, in case of large crossroads, some `LaneBorders` partly exceed the end of a `StandardLine` and then form part of a `ConnectionLine`. In such a case, the `LaneBorder` has to end *at the same level* (intersection point of the `StandardLine`'s perpendicular, offset in parallel to the `StandardLine` and not exceeding the `StandardLine`'s extent) and a new `LaneBorder` has to start with the same attributes. Successive lines have to be C^1-continuous, if possible, as explained in section 2.4. When starting `LaneBorder` modelling, it should be ensured that all requirements of the StandardLines and ConnectionLines (e.g., continuities) are fulfilled, since for later necessary adaptations of these reference lines, considerable adaptations may become necessary at many LaneBorders.

LaneBorders are polylines, which may lie geometrically on other already existing polylines (StandardLine, ConnectionLine). This becomes particularly important in case of structural separations – please refer to section 3.7.

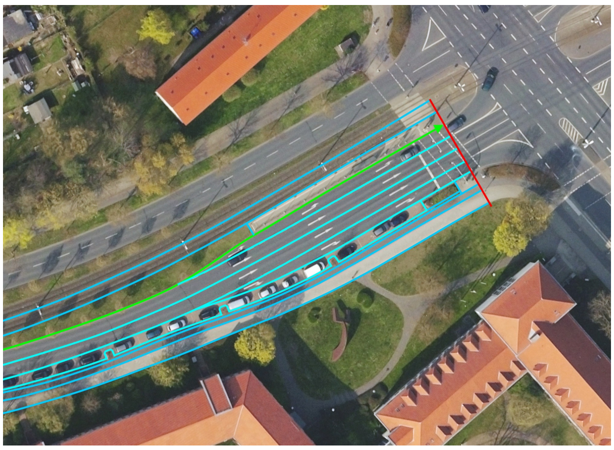
Table 10 shows modelling examples with LaneBorders in cyan (defined by road markings, curbs or other structural differences) and ends of StandardLines in red. These boundary lines demonstrate the maximum longitudinal expansion of a LaneBorder along its respectively allocated StandardLine or ConnectionLine. In the figure at the bottom left of table 10, one LaneBorder has been modelled not so evenly as the others. This shall demonstrate that in principle, a lane border can be modelled also in detail with a correspondingly high dot density. A mathematical representation, however, becomes more complicated in this way.

Table 10: LaneBorders (cyan) and StandardLine ends (red) as maximum longitudinal expansion of a LaneBorder along its respectively allocated StandardLine or ConnectionLine.

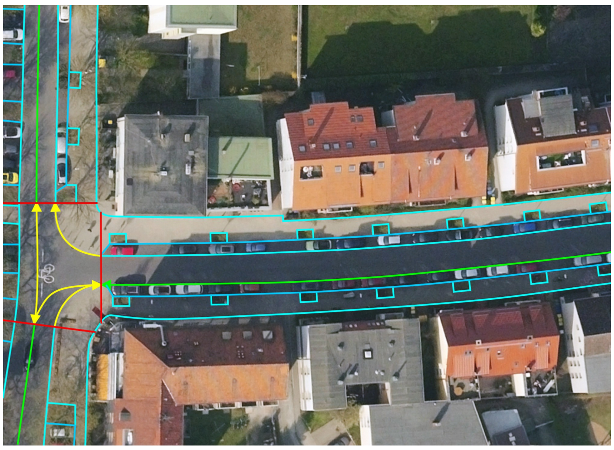


single roadway, several lanes per direction

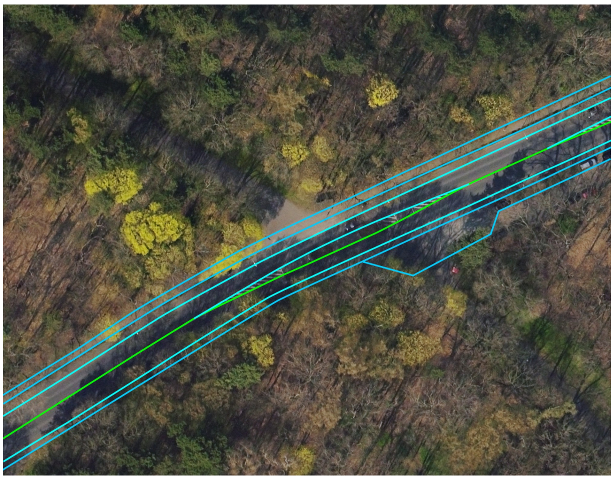
single roadway, one lane per direction

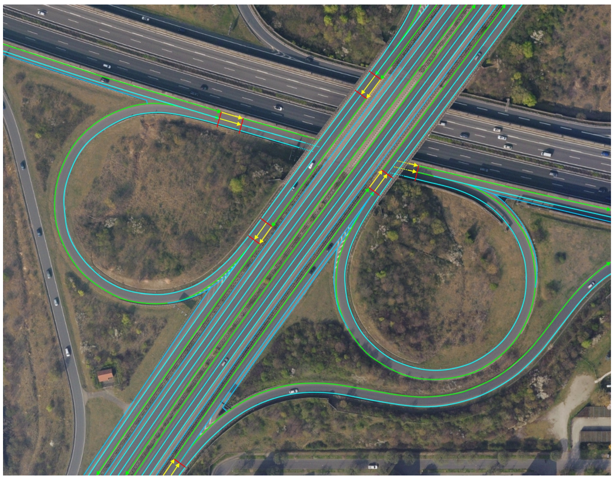


two roadways, several lanes per direction



single roadway, one lane in one direction


single roadway, one lane per direction with separation


several roadways, different lanes per direction

The `LaneBorder` describes the type of usage of the roadway lane that is limited by the `LaneBorder`. Correspondingly, the type of usage is indicated in the direction of the related `StandardLine` or `ConnectionLine`. Possible types of usage:

- driving lane = `driving`
- parking lane = `parking`
- restricted area = `restricted`
- undefined = `none`

- walkway = `sidewalk`
- bicycle lane = `biking`
- grass verges or median strip = `shoulder`
- tram lane = `tram`

When modelling `LaneBorders`, it has to be paid attention during designing that no `LaneBorder` is double-interpreted, viewed from the `parent` along an imaginary perpendicular line. Figure 7 shows the correct modelling at the left using the example of a tree island between two parking bays. Abruptly ending lanes (e.g., parking lanes on the roadway without parking bays) should be finished with a right angle. Figure 8 shows examples of a one-way road situation.

In some situation, an auxiliary structure might be useful. For this purpose, there is `none` as additional type of usage, which is used, for example, to define a section of a traffic island. The individual sections can then be defined with more details via `TrafficIslands` (see section 3.7.1).

In general `LaneBorders` are modelled at both sides of either a `StandardLine` or a `ConnectionLine`. In such cases the attribute `oneway` of these polylines has to be set to `false` which is usually the case for `StandardLines`. If `LaneBorders` are modelled to just one side of a `StandardLine` or `ConnectionLine`, the attribute `oneway` of these polylines can *equally* be set to `false`. This implies that lanes can be used in both driving directions which is often the case for `ConnectionLines`.

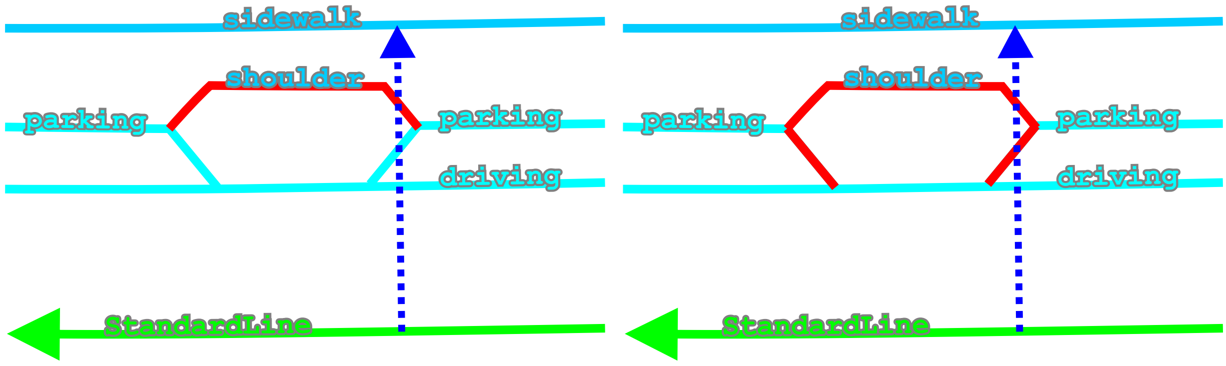


Figure 7: Correct (left) and wrong (right) modelling sequence of `LaneBorders`.

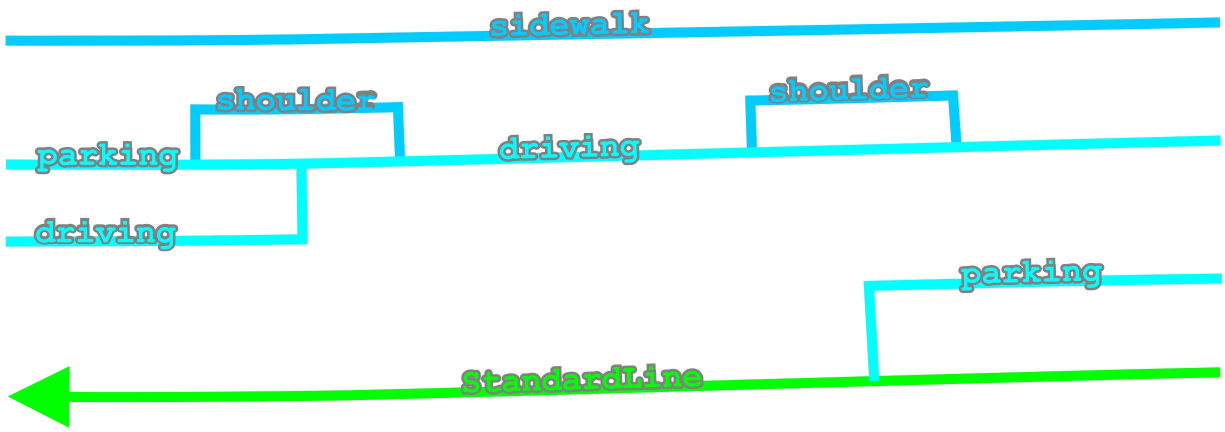


Figure 8: Modelling of abrupt lane ends.

Moreover, the `LaneBorder` describes the material mostly used, from which the lane limited by the `LaneBorder` is made. Hence, the material is indicated in the direction of the respective `StandardLine` or `ConnectionLine`. Possible materials:

- asphalt = `asphalt`
- concrete = `concrete`
- small paving material = `pavement`
- cobblestone pavement = `cobble`
- greening (grass, bushes, etc.) = `vegetation`
- gravel = `gravel`
- soil = `soil`

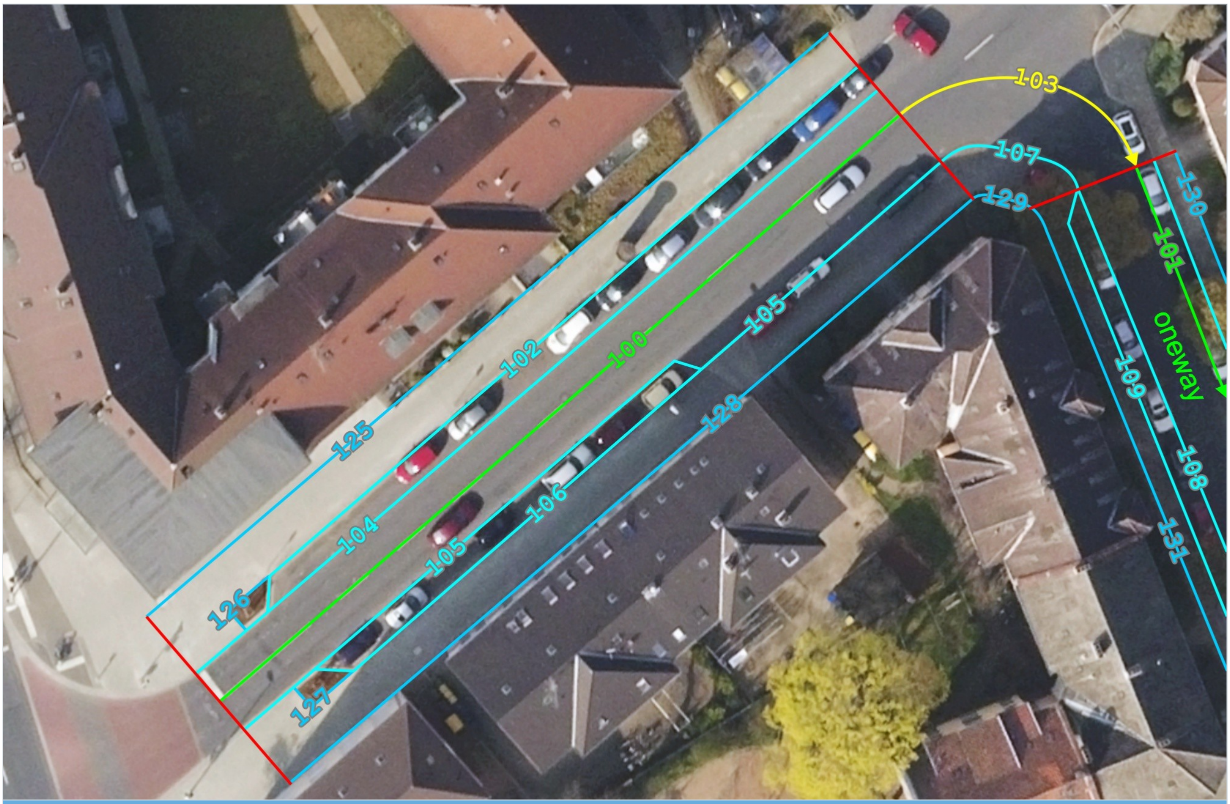
Accordingly, a `LaneBorder` is given the attributes listed in table 11. Table 12 shows a modelling example of `LaneBorders` with table extract.

In the crossroad area, also lane border information is needed. Here, lane edges are modelled that are, for example, shown via lane markings. But if there are no markings, for example in case of small crossroads, no `LaneBorder` needs to be modelled. To support data conversion, also "imaginary" lane markings can be modelled to produce a natural pathway. This means that `LaneBorders` are not necessarily based on structural conditions. Table 13 shows a typical crossroad situation. A roundabout/traffic circle is modelled analogously to crossroads, this means each entry is a small crossroad, see table 14.

Table 11: Attributes of `LaneBorders`.

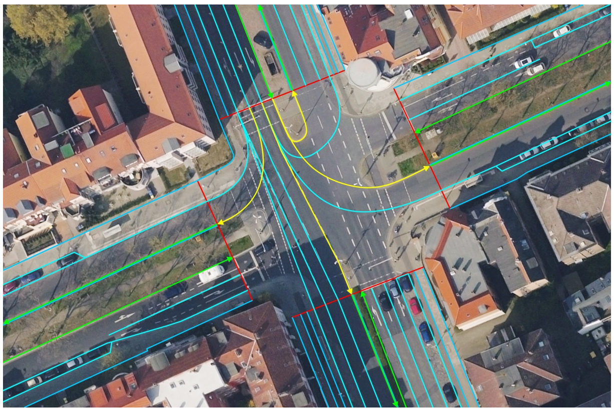
Name	Number	Type	Value	Description
id	1	unit64_t	`[0, max]`	unique identifier
dataSource-Description	1	unit64_t	`[0, max]`	ID of the describing `Data-SourceDescription` object
geometry	1	LineString Z	see example in section 2.3	polyline as WKT or WKB
material	1	string	`asphalt, concrete, pavement, cobble, vegetation, soil, gravel`	material of the lane limited by the `LaneBorder`
type	1	string	`driving, parking, restricted, none, sidewalk, biking, shoulder, tram`	type of usage of the lane limited by the `LaneBorder`
parent	1	unit64_t	`[0, max]`	ID of a `StandardLine` or `Con-nectionLine` to which the `LaneBorder` belongs

Table 12: Example of several `LaneBorders` (cyan) with table extract. Red boundary lines demonstrate the maximum longitudinal expansion of a `LaneBorder` along its respectively allocated `StandardLine` or `ConnectionLine`.

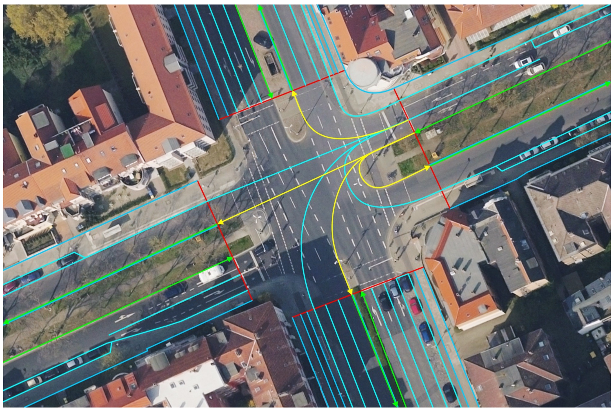


LaneBorder					
id	**dSD**	**geometry**	**material**	**type**	**parent**
102	0	WKT/WKB	asphalt	parking	100
104	0	WKT/WKB	asphalt	driving	100
105	0	WKT/WKB	asphalt	driving	100
106	0	WKT/WKB	asphalt	driving	100
107	0	WKT/WKB	asphalt	driving	103
108	0	WKT/WKB	asphalt	driving	101
109	0	WKT/WKB	pavement	parking	101
110	0	WKT/WKB	pavement	parking	101
125	0	WKT/WKB	pavement	sidewalk	100
126	0	WKT/WKB	vegetation	shoulder	100
127	0	WKT/WKB	vegetation	shoulder	100
128	0	WKT/WKB	pavement	sidewalk	100
129	0	WKT/WKB	pavement	sidewalk	103
130	0	WKT/WKB	pavement	sidewalk	101
131	0	WKT/WKB	pavement	sidewalk	101

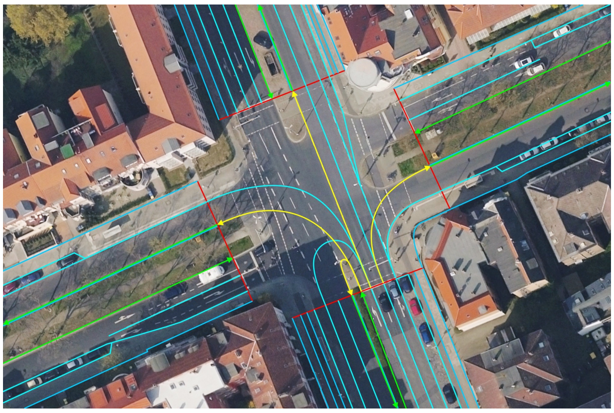
Table 13: `LaneBorders` (cyan) in the crossroads area. Red boundary lines demonstrate the maximum longitudinal expansion of a `LaneBorder` along its respectively allocated `StandardLine` or `ConnectionLine`. The bicycle lanes on the roadway (in the example coming from the west and east) have not been modeled with extra `LaneBorders`.



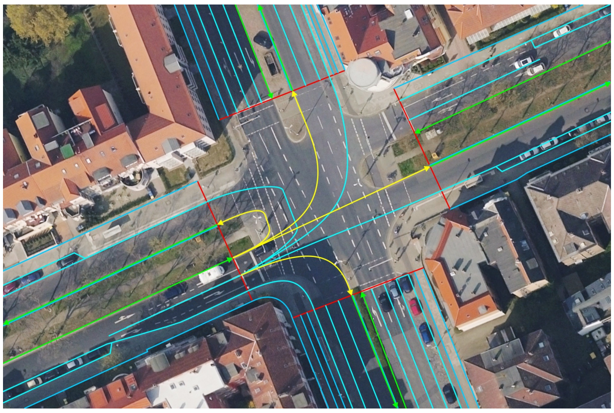
`LaneBorders` starting from the north



`LaneBorders` starting from the east

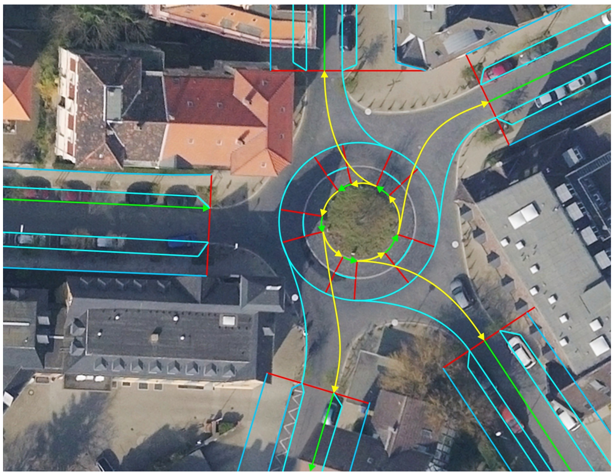


`LaneBorders` starting from the south

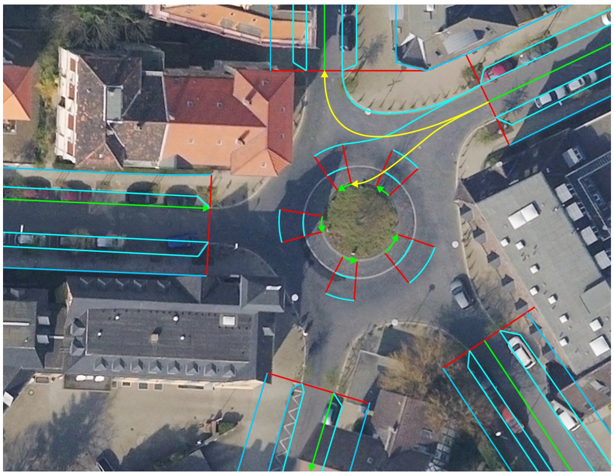


`LaneBorders` starting from the west

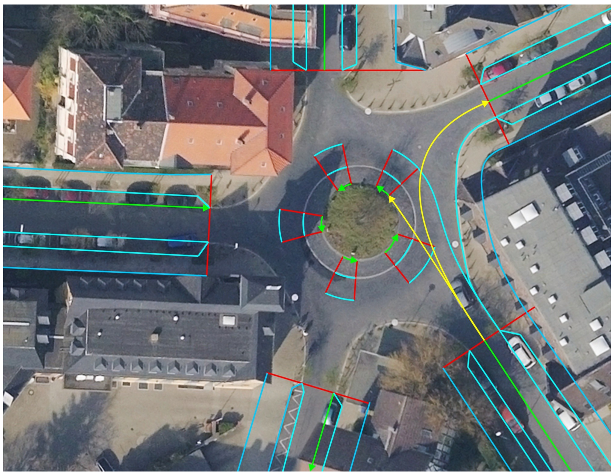
Table 14: `LaneBorders` (cyan) in a subdivided roundabout. Red boundary lines demonstrate the maximum longitudinal expansion of a `LaneBorder` along its respectively allocated `Standard-Line` or `ConnectionLine`.



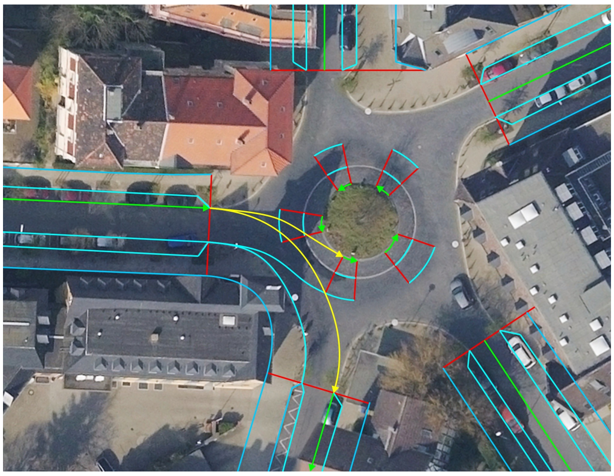
`LaneBorders` starting from the centre



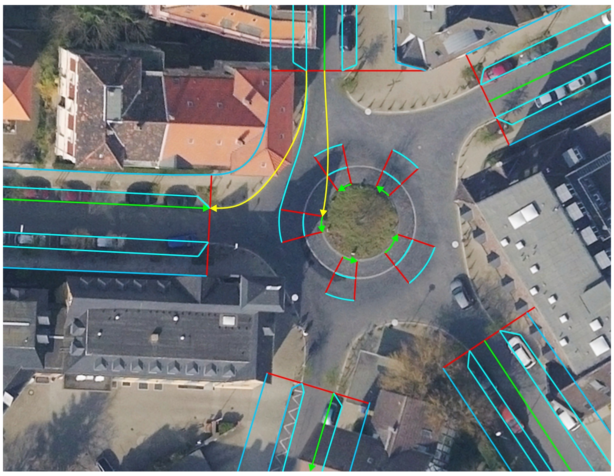
`LaneBorders` starting from the north-east



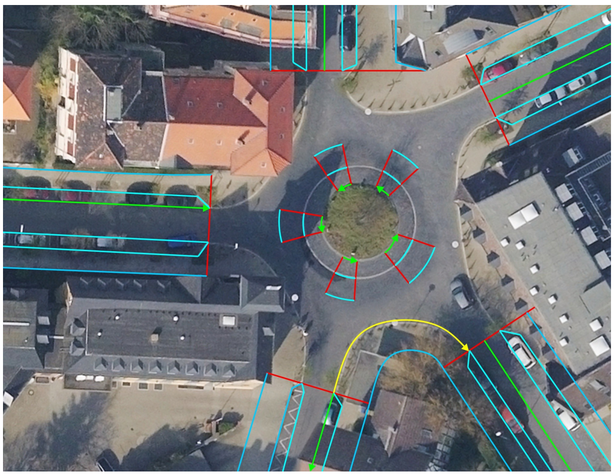
`LaneBorders` starting from the south-east



`LaneBorders` starting from the west



`LaneBorders` starting from the north



`LaneBorder` starting from the south-west

The last two figures in table 14 for the north and southwest direction represent, as described in section 3.3.1, a special case. For allocating the shoulders (sidewalk, bicycle lane, or similar) logically in a one-way road situation, auxiliary `ConnectionLines` are used here.

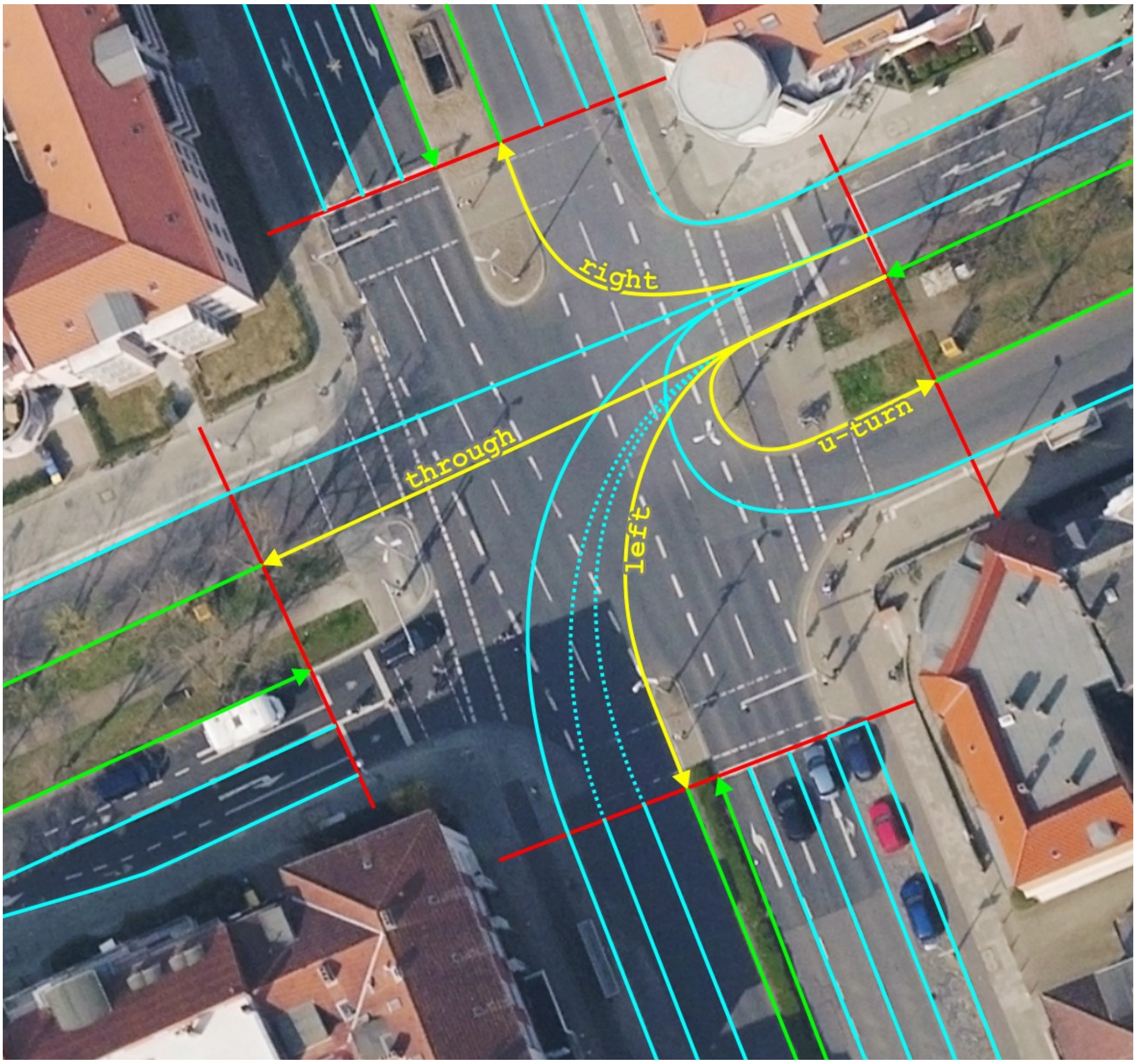


Figure 9: Necessary (continuous) and mandatory (dotted) `LaneBorders`.

Figure 9 is to show, when `LaneBorder` modelling is useful and when it is required:

- `ConnectionLine right` requires at least one `LaneBorder`, since otherwise the boundary of the lane cannot be modelled.
- `ConnectionLine through` requires no `LaneBorder` in principle, since here, the lane boundary between predecessor and successor can be interpolated.
- `ConnectionLine left`, too, requires no `LaneBorder` in principle. But here, modelling is recommended, in particular when further lanes are added, since these are intended to start

at a specific position.

- ConnectionLine u-turn requires no LaneBorder in principle, since here, too, the lane width can be interpolated from predecessor and successor. But an explicit modelling is recommended, so that with linear width interpolation, a width increase can be better controlled.

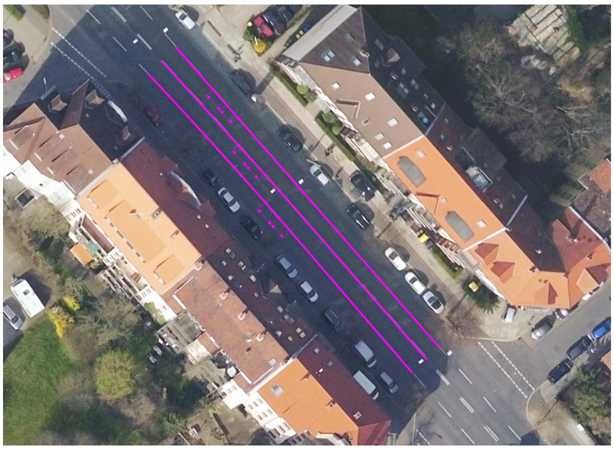
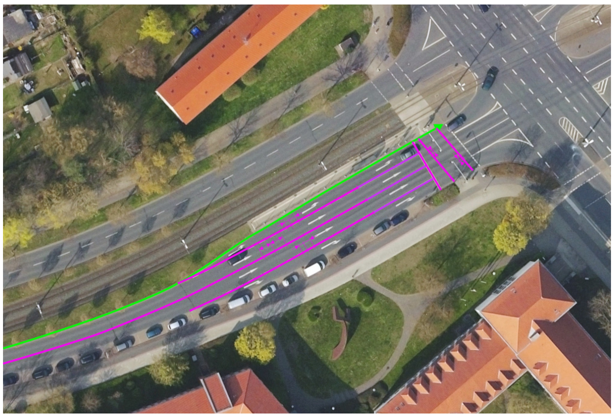
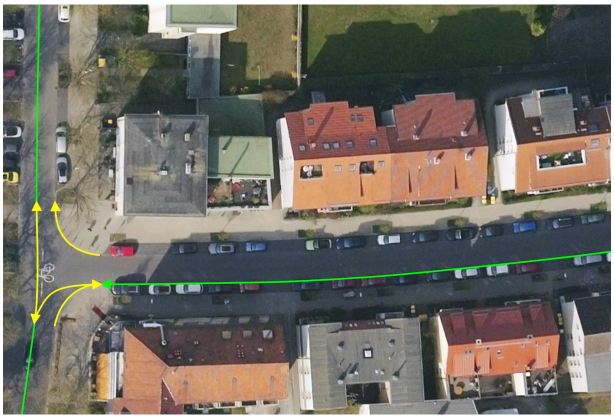
3.5. Markings

Further, major formative elements are road markings, called Marks. They are often to be equated with lane boundaries (LaneBorder), or they cover only part of the sections and are arranged in longitudinal direction to LaneBorders (e.g., bicycle lane). Some road markings exist only within a crossroad area, and other markings lie transversely to the lanes (stop lines, pedestrian crossing). For modelling Marks, there are the following characteristics for orientation:

- Markings are also stored as Mark in the form of a polyline with one type specification each. If the type of marking changes, a new Mark is generated. Successive markings have to be C^1-continuous, if possible, as explained in section 2.4.
- Marks are polylines that can geometrically lie on other, already existing polylines (StandardLine, ConnectionLine, LaneBorder). They can equally be positioned slightly offset (see example in table 17).
- A Mark object is to be subdivided into several objects, if the marking runs in parallel with/ alongside a parent object (StandardLine, ConnectionLine) and protrudes at one of its ends. Otherwise, no unambiguous assignment of the Mark is possible.
- A double line is modelled as one Mark with corresponding type (see figure 10). It is modelled in the middle of the two lines.
- Marks *always* start and end on the actual marking in order to depict, for example, a dash-space ratio as precisely as possible.

Table 15 shows modelling examples with road markings (magenta), as well as section separators for type changes and dash-space length indications.

Table 15: Road markings (magenta) and dash-space indications.

	
single roadway, several lanes per direction	single roadway, one lane per direction
	
two roadways, several lanes per direction	single roadway, one lane in one direction
	
single roadway, one lane per direction with separation	several roadways, different lanes per direction

Lane boundaries and edges are modelled as road marking `Mark` (longitudinal markings), but also stop lines and waiting lines (transverse markings). The `Marks` running alongside the lane can have the following types:

- continuous = `solid`
- interrupted = `broken`
- double-continuous = `solid solid`
- double-interrupted = `broken broken`
- continuous and interrupted = `solid broken`
- interrupted and continuous = `broken solid`
- metal marking = `botts dots`
- other edge = `curb`

If double markings diverge (or single markings merge), different `Marks` are to be modelled in each case. Figure 10 shows an example accordingly.

Figure 10: Different markings of a road section.

Transverse markings and other `Marks` can have the following types:

- pedestrian crossing, also bicycle lane crossing = `crossing`
- stop line = `stop line`
- waiting line = `waiting line`
- no-stopping/no-parking (zigzag line) = `stopping restriction`
- zebra crossing = `zebra crossing`

In case of interrupted `Marks`, the length of the dashes and the spaces is of interest. These are modelled as (averaged) lengths. Short `Marks` of the `broken` type, which model only one single marking dash, are to be modelled with a dash-space ratio of `markLength` = total length of the `Mark` and `spaceLength = 0`. In this way, it would also be possible to individually model a sequence of dashes of an interrupted marking line. Figure 11 shows different markings on a crossroads area.

Figure 11: Different markings on a crossroads area.

The weight of road marking indicates its thickness. Here, only two types are distinguished:

- slim line = `standard`
- bold line = `bold`

In Germany, the corresponding slim line widths are 0.15 m for motorways and 0.12 m for other roads, and the bold line widths are 0.30 m for motorways and 0.25 m for other roads.

The colour of the road marking is indicated with a simple identifier, with the following possibilities:

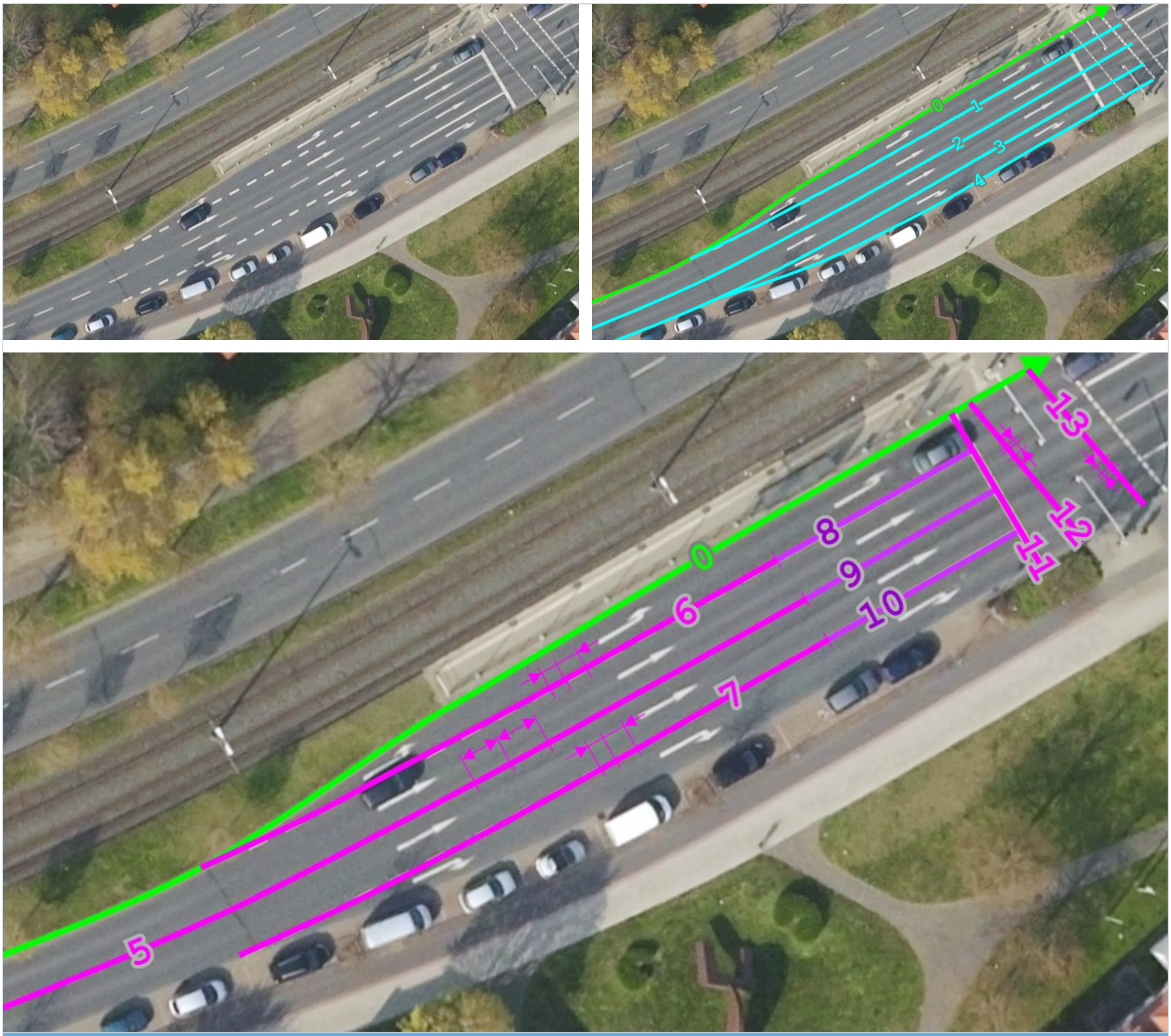
- white = `standard`
- yellow = `yellow`
- orange = `orange`
- red = `red`
- green = `green`
- blue = `blue`

Markings are given the attributes listed in table 16. Tables 17 and 18 show modelling examples for digitalization of markings within the data model.

Table 16: Attributes of `Marks`.

Name	Number	Type	Value	Description
id	1	unit64_t	`[0, max]`	unique identifier
dataSource-Description	1	unit64_t	`[0, max]`	ID of the describing `DataSource-Description` object
geometry	1	LineString Z	see example in section 2.3	polyline as WKT or WKB
color	1	string	`standard, yellow, orange, red, green, blue`	colour of road marking (normally, white = `standard`)
type	1	string	`solid, broken, solid solid, broken broken, solid broken, broken solid, botts dots, curb, zebra crossing, crossing, stop line, waiting line, stopping restriction`	type of road marking
weight	1	string	`standard, bold`	thickness/weight of road marking
markLength	0..1	double	`[0.0, max]`	length of the dash [m] in an interrupted road marking
spaceLength	0..1	double	`[0.0, max]`	length of the space [m] in an interrupted road marking
parent	1	unit64_t	`[0, max]`	ID of the logically related `StandardLine` or `ConnectionLine`

Table 17: Example of several parallel and transverse markings/`Marks` (magenta) with table extract.



Mark								
id	**dSD**	**geometry**	**color**	**type**	**weight**	**mark-Length**	**space-Length**	**parent**
5	0	WKT/WKB	standard	broken	standard	3.0	3.0	0
6	0	WKT/WKB	standard	broken	bold	1.5	1.5	0
7	0	WKT/WKB	standard	broken	bold	1.5	1.5	0
8	0	WKT/WKB	standard	solid	bold			0
9	0	WKT/WKB	standard	solid	standard			0
10	0	WKT/WKB	standard	solid	bold			0
11	0	WKT/WKB	standard	stop line	standard			0
12	0	WKT/WKB	standard	crossing	standard	0.5	0.2	0
13	0	WKT/WKB	standard	crossing	bold	0.75	0.25	0

Table 18: Example for restricted areas (upper photos) and zebra crossing (lower photos) as markings/Marks (magenta) with table extract.

Mark								
id	dSD	geometry	color	type	weight	mark-Length	space-Length	parent
33	0	WKT/WKB	standard	stopping restriction	standard			20
40	0	WKT/WKB	standard	zebra crossing	standard			30
41	0	WKT/WKB	standard	zebra crossing	standard			30
42	0	WKT/WKB	standard	zebra crossing	standard			30
43	0	WKT/WKB	standard	zebra crossing	standard			31
44	0	WKT/WKB	standard	zebra crossing	standard			31
45	0	WKT/WKB	standard	zebra crossing	standard			31
46	0	WKT/WKB	standard	zebra crossing	standard			31
47	0	WKT/WKB	standard	zebra crossing	standard			31
48	0	WKT/WKB	standard	zebra crossing	standard			31

3.6. Objects

Objects like road signs, overhead gantries and traffic lights represent important information for road description, but also posts for street lighting, guardrails, grids and noise barriers are important elements that characterize the appearance of a street.

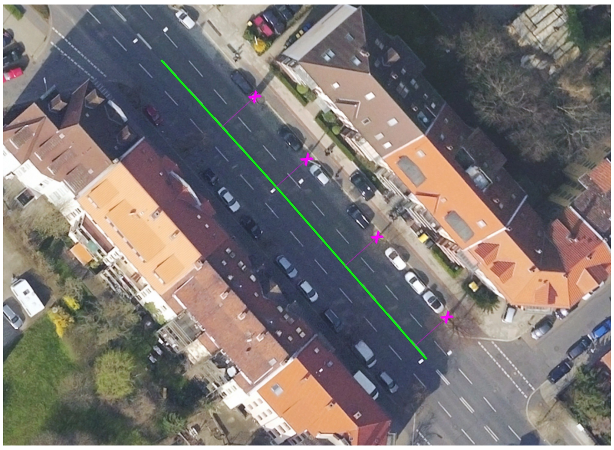
Road signs are indicated once as sign with a corresponding type and modelled at the same geographic coordinate also as signpost or traffic light post.

Traffic lights are subdivided into post, boom and signalling device. Each object is given a specific entry. The same applies to overhead gantries, where the transverse beam is allocated only to one of the two posts, but it covers the whole length.

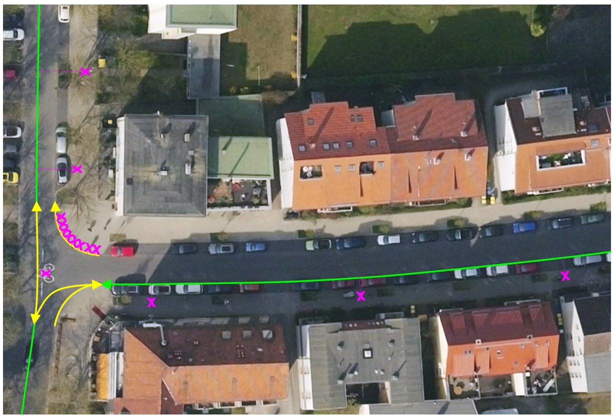
Point objects are defined by a position coordinate, the absolute orientation angle as well as further attribute data, such as the allocation to a `StandardLine`, `ConnectionLine` or to another object. The orientation is indicated *geographically absolutely*, i.e. a left-handed system is used with positive clockwise rotation. So, the angles are N = 0°, O = 90°, S = 180°, W = 270°.

Like markings, linear objects are modelled by polylines with type indication and also allocated to a `StandardLine` or `ConnectionLine`. Table 19 shows examples for linear and point-shaped objects.

Table 19: Linear and point-shaped street objects (magenta).

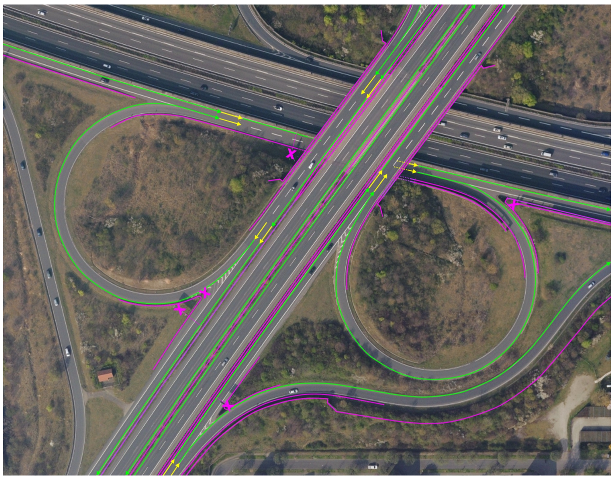
	
single roadway, several lanes per direction	single roadway, one lane per direction

two roadways, several lanes per direction



single roadway, one lane in one direction

single roadway, one lane per direction with separation



several roadways, different lanes per direction

3.6.1. Point objects

Relevant point objects and their orientation front are listed in table 20. Signalling devices are differentiated in table 21. If a signalling device cannot be defined based on the data, `1.000.000` is indicated as default value.

Pictograms and special signs, too, which are applied onto the roadway, are modelled as point objects. Table 22 shows the distinction between different types.

Table 20: Types of point objects.

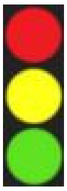
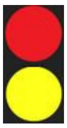
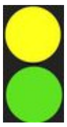
Objekt type	Identifier	Front for alignment	Anchor for positioning
traffic light post	`traffic light post`	-/-	post centre on ground
boom of traffic light post	`traffic light boom`	along the sidearm/boom	mounting point at post
whip-type post	`curved traffic light post`	along the bent post	post centre on ground

Objekt type	Identifier	Front for alignment	Anchor for positioning
signaling device	see table 21 of signaling device types	front view	centre of the front
road sign	corresponding number of the traffic sign plate[14]	front view	centre of the front
variable message signs	according to ISO 14823[15]	front view	centre of the front
sign post (also generic post/pole)	`pole`	-/-	post centre on ground
post of overhead gantry	`overhead gantry mast`	-/-	post centre on ground
overhead gantry	`overhead gantry`	along the gantry	mounting point at post
reflector post	`reflector post`	-/-	post centre on ground
post of street lamp	`street lamp`	in the direction of the boom or transversely in case of two booms	post centre on ground
whip-type post of street lamp	`curved street lamp`	along the curve	post centre on ground
postbox	`postbox`	postbox front	centre of ground face
well	`well`	recognizable front or -/-	centre of ground face
bus and tram shelter	`shelter`	longer open side/face	centre of ground face
monument and sculpture	`monument`	recognizable front or -/-	centre of ground face
flagpole and flag	`flagpole`	-/-	post centre on ground
hydrant	`hydrant`	one outflow towards the front, or two outflows laterally	hydrant centre on ground
advertising column	`advertising column`	-/-	centre of ground face
waste container	`waste container`	longer side or -/-	centre of ground face
SOS telephone	`emergency phone`	phone front	centre of ground face
pictogram	see table 22 of pictogram types	front in direction of view	pictogram centre
bollard	`bollard`	-/-	bollard centre on ground
bench	`bench`	side of the seating surface	centre of ground face
phone box and stele	`phone box, phone stele`	door or stele front	centre of ground face
distribution box	`distribution box`	front view (longer side)	centre of ground face
advertising panel	`advertising panel`	front view (longer side)	centre of ground face

[14] see, e.g., `http://www.vzkat.de/2017/VzKat.htm` or `https://de.wikipedia.org/wiki/Bildtafel_der_Verkehrszeichen_in_der_Bundesrepublik_Deutschland_seit_2013`

[15] Intelligent transport systems – Graphic data dictionary

Table 21: Types of signalling devices.

Signal	Identifier	Signal	Identifier	Signal	Identifier
	1.000.001		1.000.008		1.000.011.10
	1.000.002		1.000.008.10		1.000.011.20
	1.000.002.10		1.000.008.20		1.000.011.30
	1.000.007		1.000.009.10		1.000.011.40
	1.000.007.10		1.000.009.20		1.000.011.50
	1.000.007.20		1.000.009.30		1.000.012.10
	1.000.007.30				1.000.012.20
	1.000.014		1.000.010.10		1.000.013

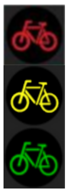
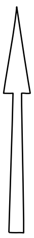
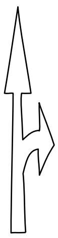
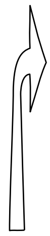
Signal	Identifier	Signal	Identifier	Signal	Identifier
	`1.000.015`		`1.000.010.20`		`1.000.013.10`
	`F 0`		`W 0`, **also** `F 6`		`W 14`
	`F 1`		`W 1`		`W 11`
	`F 2`		`W 2`		`W 12`
	`F 3`		`W 3`		`W 13`
	`F 4`		`A 1`		`A 2b`
	`F 5`		`A X`		

Table 22: Types of pictograms.

Symbol	Identifier	Symbol	Identifier	Symbol	Identifier
TAXI **BUS**	taxi horizontal, bus horizontal	**30,** **50,** **...**	horizontal 30, horizontal 50, ...		pedestrian
T B **A U** **X S** **I**	taxi vertical, bus vertical		stopping restriction		handicap
	297.1		297.2		cyclist
	297 through		297 through left		297 through right

Symbol	Identifier	Symbol	Identifier	Symbol	Identifier
	`297 left right`		`297 left`		`297 right`

In figure 12, the incoming `StandardLine` and the successive `ConnectionLines` and their separation are modelled. All point objects lying in the direction of travel are marked and provided with an allocation to a `StandardLine`, `ConnectionLine` or to another object.

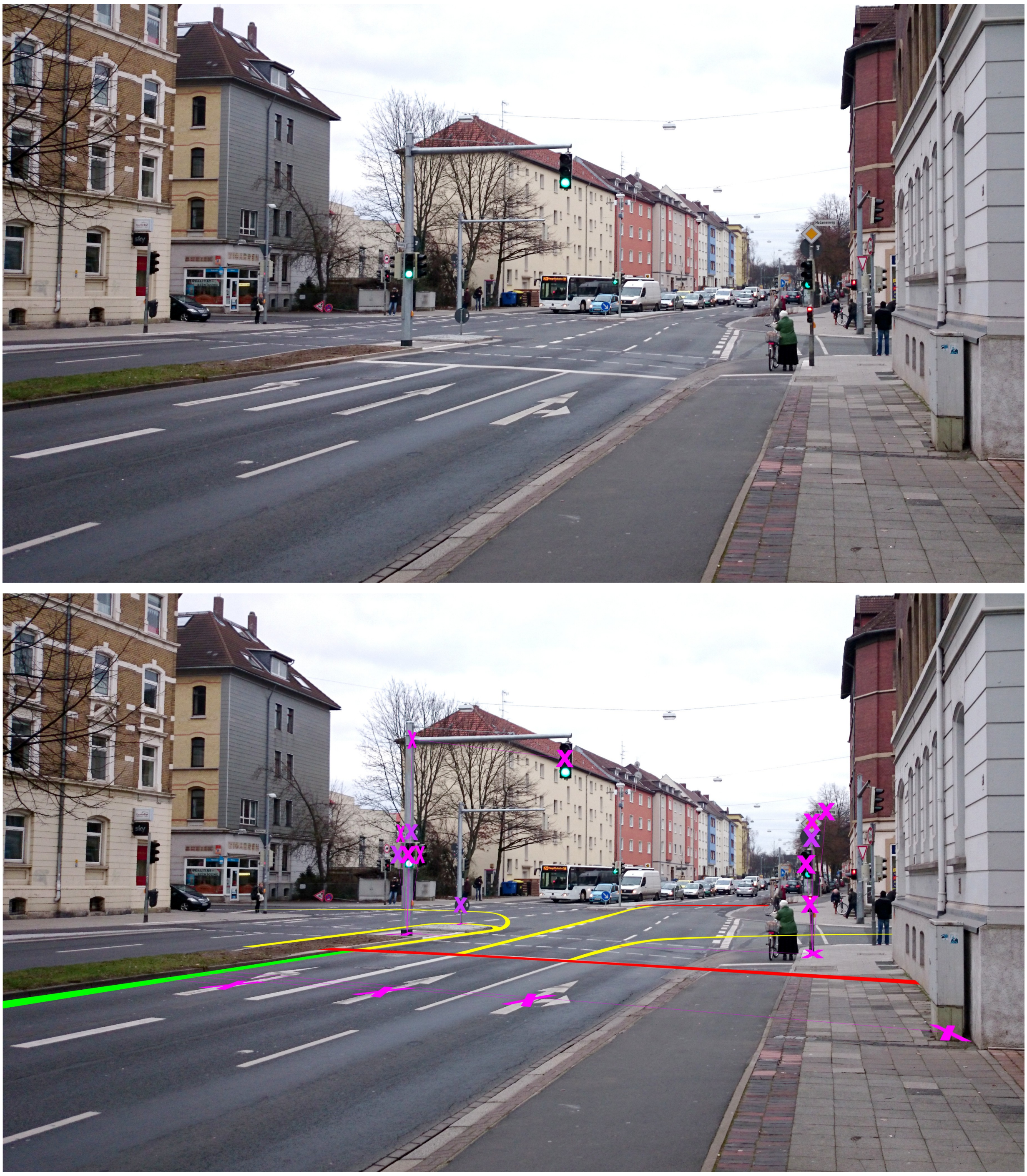


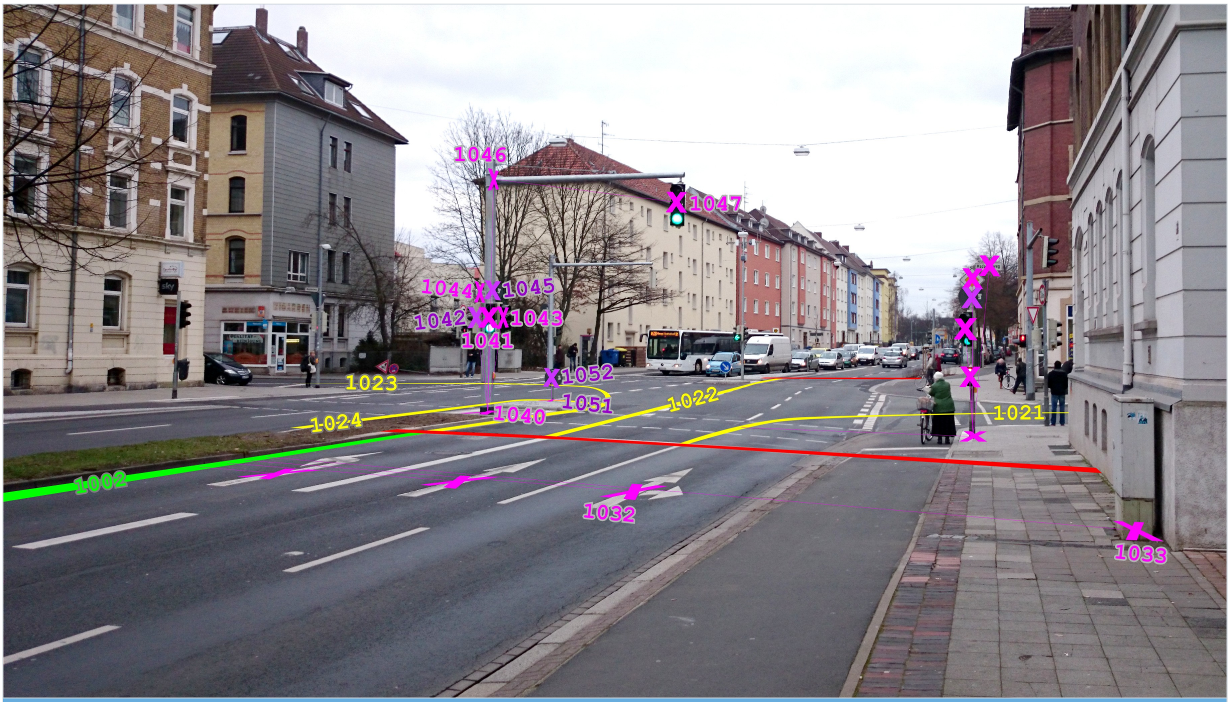
Figure 12: Crossroad, several lanes with structural separation and point objects (magenta) located on the `StandardLine` (green) and `ConnectionLine` (yellow).

A point object carries information about its size/dimension. It is given the attributes listed in table 23. Table 24 shows a complete modelling example of `PointObjects` in this crossroad situation.

Table 23: Attributes of `PointObjects`.

Name	Number	Type	Value	Description
id	1	unit64_t	`[0, max]`	unique identifier
dataSource-Description	1	unit64_t	`[0, max]`	ID of the describing `Data-SourceDescription` object
geometry	1	Point Z	see example in section 2.3	point as WKT or WKB
type	1	string	see table 20, table 21 and table 22 as well as pictogram number	type of the object
content	0..1	string		text on the street name sign or simple route sign
heading	0..1	double	`[0.0, 360.0[`	orientation of the object as angle indicated in degrees
width	0..1	double	`[0.0, max]`	width of the object when looking at its front side in metre
depth	0..1	double	`[0.0, max]`	depth of the object when looking at its front side in metre
height	0..1	double	`[0.0, max]`	height of the object when looking at its front side in metre
pointParent	0..1	unit64_t	`[0, max]`	ID of the related point object
linearParent	0..1	unit64_t	`[0, max]`	ID of the related `StandardLine` or `ConnectionLine`

Table 24: Example crossroad with located point objects (magenta) and WKT geometries, with table extract. The course of the road is in south direction S = 180°. The `DataSourceDescription` (`dSD`) of all objects in this example is `1`.



PointObject									
id	**geometry**	**type**	**con-tent**	**head-ing**	**width**	**depth**	**height**	**point-Parent**	**linear-Parent**
1032	Point Z (602557.8 5791116.4 117.8)	297 through right		180	3	0.5			1002
1033	Point Z (602550.8 5791118.2 117.8)	dis-tri-bution box		90	0.75	0.31	1		1002
1040	Point Z (602569.2 5791104.3 118.1)	traf-fic light post			0.25	0.25	6.42		1024
1041	Point Z (602569.2 5791104.3 121.1)	1.000.001		315	0.265	0.347	0.79	1040	
1042	Point Z (602569.2 5791104.3 121.1)	1.000.002		90	0.265	0.347	0.53	1040	

id	geometry	type	con-tent	head-ing	width	depth	height	point-Parent	linear-Parent
1043	Point Z (602569.2 5791104.3 121.1)	1.000.002		270	0.265	0.347	0.53	1040	
1044	Point Z (602569.2 5791104.3 121.6)	437	Alt-stadt-ring	0	0.08	0.35		1040	
1045	Point Z (602569.2 5791104.3 121.6)	267		180	0.6	0.6		1040	
1046	Point Z (602566.2 5791103.6 124.0)	boom		270	0.12	0.12	4.84	1040	
1047	Point Z (602565.6 5791104.0 123.6)	1.000.001		0	0.35	0.53	1.09	1046	
1051	Point Z (602570.2 5791100.5 118.2)	pole			0.076	0.076			1024
1052	Point Z (602570.2 5791100.5 118.9)	222-20		180	0.6	0.6		1051	

3.6.2. Linear objects

Linear objects do not need an orientation and are therefore treated like the other line types. Their polylines are modelled on the upper edge of linear objects in order to implicitly model elevation information. For example, they are also separated *at the same level* at the `StandardLine`-ends (maximum longitudinal expansion/extent) and allocated to a `StandardLine` or a `ConnectionLine` (compare modelling of lane borders in section 3.4). Otherwise, a clear assignment of linear objects is not possible.

Route sections including bridges or tunnels are modelled as linear objects, the geometry of which overlies the respective `parent` geometry (`StandardLine` or `ConnectionLine`) for the section of validity. There are the following relevant linear objects:

- guardrails = `guardrail`
- bridges = `bridge`
- tunnels = `tunnel`

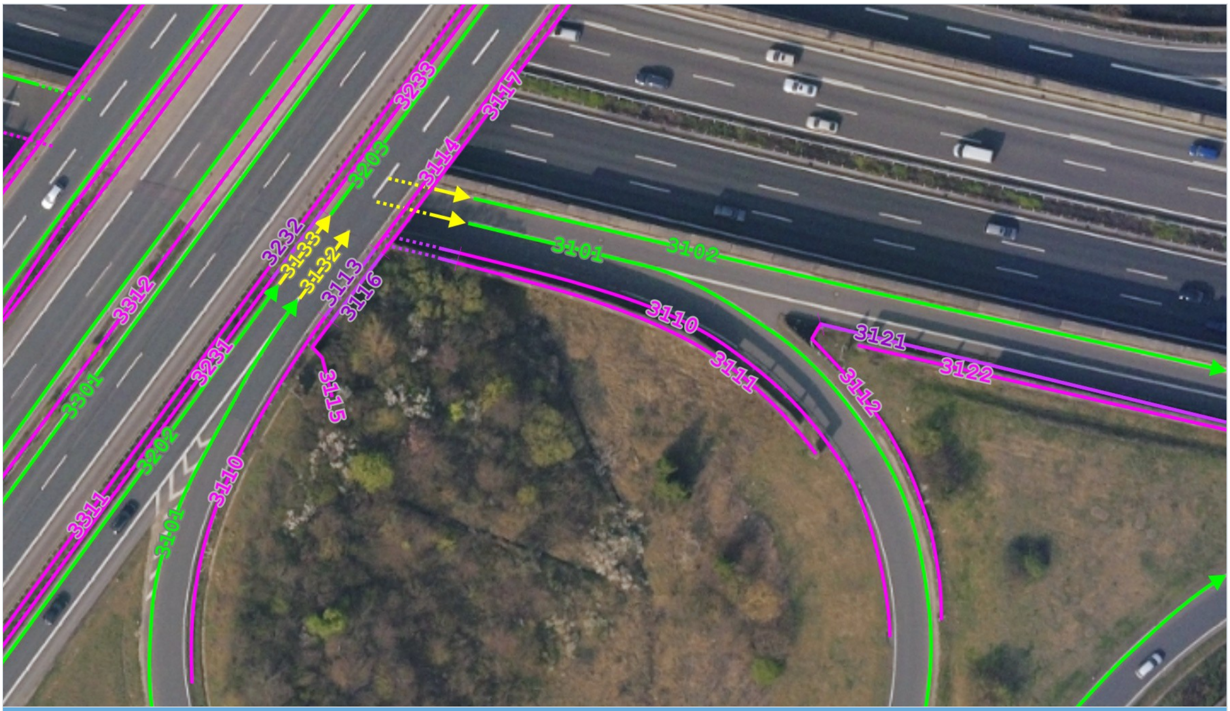
- guardrails with anti-glare shield = `guardrail anti-glare`
- guardrails with reinforcement = `superrail`
- concrete guide wall = `jersey barrier`
- cable barrier = `cable barrier`
- noise barrier = `noise barrier`
- fences = `fence`
- barriers = `barrier`

A `LinearObject` is described by the attributes listed in table 25. Tables 26 and 27 show modelling examples with data table extracts.

Table 25: Attributes of `LinearObjects`.

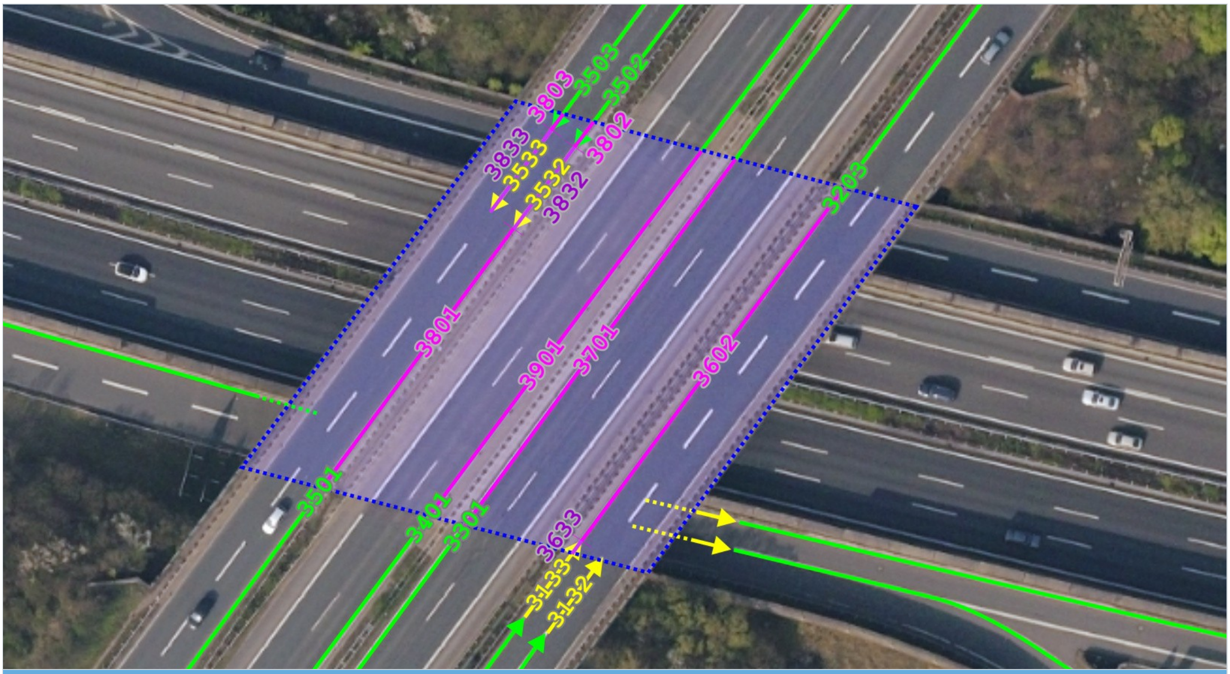
Name	Number	Type	Value	Description
id	1	unit64_t	`[0, max]`	unique identifier
dataSource-Description	1	unit64_t	`[0, max]`	ID of the describing `Data-SourceDescription` object
geometry	1	LineString Z	see example in section 2.3	polyline as WKT or WKB
type	1	string	`guardrail, bridge, tunnel, guardrail anti-glare, superrail, jersey barrier, cable barrier, noise barrier, fence, barrier`	type of the object
parent	1	unit64_t	`[0, max]`	ID of the related `StandardLine` or `ConnectionLine`

Table 26: Example for linear objects (without bridge object) with table extract.



LinearObject				
id	**dSD**	**geometry**	**type**	**parent**
3110	0	WKT/WKB	guardrail	3101
3111	0	WKT/WKB	noise barrier	3101
3112	0	WKT/WKB	guardrail	3101
3113	0	WKT/WKB	guardrail	3132
3114	0	WKT/WKB	guardrail	3203
3115	0	WKT/WKB	fence	3101
3116	0	WKT/WKB	fence	3132
3117	0	WKT/WKB	fence	3203
3121	0	WKT/WKB	guardrail	3102
3122	0	WKT/WKB	noise barrier	3102
3231	0	WKT/WKB	guardrail	3202
3232	0	WKT/WKB	guardrail	3133
3233	0	WKT/WKB	guardrail	3203
3311	0	WKT/WKB	guardrail	3301
3312	0	WKT/WKB	guardrail	3301

Table 27: Example of a bridge (blue) as `LinearObject` (without further linear objects) with table extract.



LinearObject				
id	**dSD**	**geometry**	**type**	**parent**
3602	0	WKT/WKB	bridge	3203
3633	0	WKT/WKB	bridge	3133
3701	0	WKT/WKB	bridge	3301
3801	0	WKT/WKB	bridge	3501
3802	0	WKT/WKB	bridge	3502
3803	0	WKT/WKB	bridge	3503
3832	0	WKT/WKB	bridge	3532
3833	0	WKT/WKB	bridge	3533
3901	0	WKT/WKB	bridge	3401

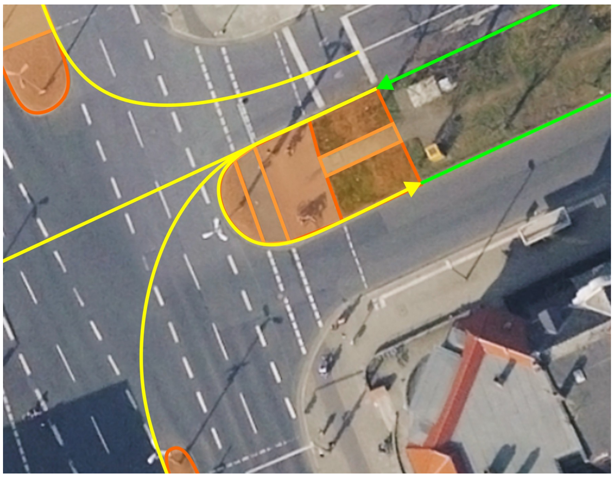
3.7. Structural separations

If there are structural separations between lanes, it is distinguished for modelling, how the type of usage of the spaces in-between that are created develops relatively to the course of a road (`StandardLine` or `ConnectionLine`). If the type of usage changes *transversely* to the course of the road, the spaces are modelled as areal traffic island (`TrafficIslands`, see section 3.7.1). This is, for example, the case with central islands including pedestrian crossings or bicycle lanes crossing. But if the type of usage changes in *longitudinal* direction of the course of the road, this is modelled as linear lane border (`LaneBorder`) of a middle lane (see section 3.4). Examples for this are simple, parallel-running grass verges between two `StandardLines`. Both situations are described in the following.

3.7.1. Traffic islands

Traffic islands (also called central islands) are modelled as areas (`TrafficIsland`), since they were not designed clearly along the course of a road, particularly in case of crossroads. A traffic island consists of one or several `TrafficIsland` areas, with each of these areas being allocated to exactly one `StandardLine` or `ConnectionLine`. The type of usage often changes *transversely* to the course of this reference line. Table 28 shows two examples of traffic islands.

Table 28: Central islands/`TrafficIslands` (orange).

	
single roadway (traffic island)	two roadways (central island in a crossroad)

`TrafficIsland` modelling is guided by the following points:

- The outline of a `TrafficIsland` area is modelled, like a `Mark`, in the middle of the marking or curbstone.
- Neighboring `TrafficIsland` areas lie with their edges topologically one on top of the other.
- If the type of `TrafficIsland` changes, a new `TrafficIsland` is generated.
- `TrafficIslands` that are defined via a marking, do not need any further circumferential `Mark` that would model the edges (or outline).

The type of the place covered by the area is defined as type of usage and material. Here, the same types of usage and materials are used as in case of `LaneBorders` (see section 3.4):

- lane = `driving`
- parking lane = `parking`
- restricted area = `restricted`
- undefined = `none`

- pedestrian ways = `sidewalk`
- bicycle lane = `biking`

- grass verge = `shoulder`
- tram route = `tram`

Possible materials are:

- asphalt = `asphalt`
- concrete = `concrete`
- small paving material = `pavement`
- cobblestone pavement = `cobble`
- greening (grass, bushes, etc.) = `vegetation`
- gravel = `gravel`
- soil = `soil`

A `TrafficIsland` is given the attributes listed in table 29. Table 30 gives a modelling example with data table extract.

Table 29: Attributes of `TrafficIslands`.

Name	Number	Type	Value	Description
id	1	unit64_t	`[0, max]`	unique identifier
dataSource-Description	1	unit64_t	`[0, max]`	ID of the describing `Data-SourceDescription` object
geometry	1	Polygon Z	see example in section 2.3	polygon as WKT or WKB
material	1	string	`asphalt, concrete, pavement, cobble, vegetation, soil, gravel`	material of the area defined by the `TrafficIsland`
type	1	string	`driving, parking, restricted, none, sidewalk, biking, shoulder, tram`	type of usage of the area defined by the `TrafficIsland`
parent	1	unit64_t	`[0, max]`	ID of the related `StandardLine` or `ConnectionLine` to which the `TrafficIsland` belongs

Table 30: Example of a traffic island with area sections, with table extract.

TrafficIsland					
id	dSD	geometry	material	type	parent
2301	0	WKT/WKB	asphalt	restricted	2001
2302	0	WKT/WKB	asphalt	driving	2001
2303	0	WKT/WKB	vegetation	shoulder	2001
2304	0	WKT/WKB	asphalt	sidewalk	2001
2305	0	WKT/WKB	vegetation	shoulder	2001
2306	0	WKT/WKB	asphalt	restricted	2001
2307	0	WKT/WKB	asphalt	driving	2001
2308	0	WKT/WKB	asphalt	restricted	2001

3.7.2. Median strips

If lanes are structurally separated from each other and if the type of usage of the space in-between that is created (median strip) changes in *longitudinal direction* to the course of the adjacent `StandardLine` or `ConnectionLine`, such a median strip is modelled as `LaneBorder` for *one* of the adjacent reference lines. The procedure is analogue to the general lane definition described in section 3.4. In this case, the geometry of a median strip boundary may correspond with the opposite reference line or overlap this in part.

In principle, *one* StandardLine may also lie in the middle of a median strip, and the area up to the first lane is modelled with a lane with shoulder as type of usage. To avoid potential design problems in case of crossroads, however, *two* StandardLines should be used, if possible, in case of a structural separation. In case of parallel StandardLines, these should end *at the same level*, for simpler modelling. Table 31 shows a modelling example for this particular situation.

Table 31: Grass verges between two StandardLines (green) modelled as LaneBorder 1338. In this case, the geometry of this LaneBorder is identical with StandardLine 1339 and allocated to StandardLine 1337. Alternatively, LaneBorder 1338 can equally lie on StandardLine 1337 and be allocated to StandardLine 1339.

LaneBorder					
id	**dSD**	**geometry**	**material**	**type**	**parent**
1334	0	WKT/WKB	pavement	sidewalk	1337
1335	0	WKT/WKB	asphalt	driving	1337
1336	0	WKT/WKB	asphalt	driving	1337
1338	0	WKT/WKB	vegetation	shoulder	1337
1340	0	WKT/WKB	asphalt	driving	1339
1341	0	WKT/WKB	asphalt	parking	1339
1342	0	WKT/WKB	pavement	sidewalk	1339

4. Surface description

The description of the road surface is of particular importance for tire and chassis simulation in development work as well as for assessments (chassis tuning). Generally, data with the highest possible accuracy are needed to image surface irregularities like wheel ruts, potholes, sewer covers, etc. For this purpose, measuring points are modelled in a regular grid and these surface profiles are then linked to a road description, so that they can supplement the description correspondingly. The modelling of the data is quite simple, the complexity lies in the acquisition of the data by means of adequate sensors.

4.1. Road axis referencing

The reference line for acquisition of the surface information has to match the reference line for the road description (`StandardLine` and `ConnectionLine`) in the longitudinal and elevation profile. The line shall have a continuous and, if required, smoothened course.

If the route is surveyed in sections, there must not be any gap between the sections. The transitions have to be continuous in absolute and also relative elevation.

In the transverse profile, too, the surface description has to be continuous. It should cover the entire width of the road description up to the lane edge (or all relevant lanes, including also pedestrian ways, if required). Since in OpenCRG the surface description is made with a fixed width, but the road width may vary, the data record should be filled outside the lane region with `NaN`. Slope and banking should be separated in the data record.

The final road axis has to be finally defined for a later combination of the surface data with the road description. The Road2Simulation format provides for transmitting the road description data by mathematical transformation into, for example, the OpenDRIVE format. This `StandardLine` then modelled as curve has to serve as reference line for the surface information.

For merging, the `attached` mode should be used in OpenDRIVE, where the `StandardLine` of the surface information is replaced by the `StandardLine` of the road description. The elevation information of the surface description is added to the profile of the road description. In this way, the original position, curvature, elevation and superelevation remain unconsidered in OpenCRG, and the surface information follows the road description.

4.2. Surface accuracy

The surface description should be generated with a cell width of 5x5 mm. The surface sections (referred to the road reference line) should be generated in 5x5 m areas.

The relative two-dimensional position accuracy and elevation accuracy has to be at least 1 mm, whereas the absolute two-dimensional position referencing and elevation referencing has to be at least 1 m or better. At distinctive points (e.g., potholes, sewer covers), the deviation of the surface data must not be larger than 5 to 25 mm within the northing and easting, and 1 to 5 mm in the elevation. Outliers due to measurement faults must not be contained.

The data should be indicated in UTM (e.g., UTM 32N) with the underlying, official two-dimensional spatial reference system ETRS89 (i.e., EPSG:25832[16]) with the correspondingly matching elevation reference system (e.g., DHHN92 (EPSG:5783[17])) – please refer to section 2.2. If the course of the road *and* its surface are surveyed, both data records should be modelled with the same projection.

In OpenCRG, the variable `reference_line_offset_x`, `reference_line_offset_y`, and `reference_line_offset_z` are to be used in the block `$ROAD_CRG` for documenting the geometric referencing. The projection is indicated in the block `$ROAD_CRG_MPRO` by using the variable `proj_nm` (e.g., `proj_nm = 'UTM_32T'`).

[16] http://epsg.io/25832
[17] http://epsg.io/5783

5. Remarks

The work for these Road2Simulation Guidelines and the related data model are based on the DLR research project "Virtual World[18]" and on the DLR-operated "Application Platform for Intelligent Mobility" (AIM[19]). The preparation of these Guidelines was enabled by Audi, BMW, Daimler, Porsche, Volkswagen, and DLR. This English translation of the German guidelines document was realized by Schomäcker Datenverarbeitung GmbH, Braunschweig.

[18] http://www.dlr.de/ts/mittendrin/virtuellewelt
[19] http://www.dlr.de/ts/aim

6. Changelog

v1.2 to v1.2.1

- Conversion of the whole document to LaTeXwith resulting changes in formatting
- Official publication of these Guidelines at `zenodo.org` (refer to section 1.1)
- Publication of this English translation
- Document title in German version changed from “Road2Simulation Guidelines” to “Road2Simulation-Leitfaden”
- Licence changed to CC BY 4.0
- Separate list of examples removed due to conversion of former example category into regular tables
- Added explicit cross-references to example tables in the text
- Entries in list of figures and list of tables shortened and improved regarding readability
- Many small corrections in the text
- Chapter “Notes” removed
- Invalid website link from footnote 12 in section 2.6.2 changed to a Wikipedia link
- Figure 12 and 13 merged into figure 12

v1.1 to v1.2

- Many improvements/clarifications in text and examples
- Added licence
- Added changelog
- OpenCRG: Added details on road referencing and attribute usage
- Added hint: Surveyed elevation model should be delivered together with the Simple Feature data
- Examples added: Detailed modelling of `LaneBorders`
- Examples added: Detailed modelling of `Marks`
- Data model: `BaseReferenceLine`: Renamed to `ReferenceLine`
- Data model: `ReferenceLine`: Added attribute `category`
- Data model: Persistent tables `StandardLine` and `ConnectionLine` have been grouped into `ReferenceLine`
- Data model: `ConnectionLine`: Added turn type `auxiliary`
- Data model: `LaneBorder`: Renamed material `grass` to generic `vegetation`
- Data model: `TrafficIsland`: Renamed material `grass` to generic `vegetation`
- Data model: `Mark`: Renamed attribute `length` to `markLength`
- Data model: `Mark`: Renamed attribute `space` to `spaceLength`
- Data model: `Mark`: Renamed type `pedestrian walk` to `crossing`
- Data model: `Mark`: Renamed type `cross walk` to `zebra crossing`
- Data model: `Mark`: Added color `orange`

- Data model: `PunctualObject`: Renamed to `PointObject`
- Data model: `PointObject`: New attributes `width`, `depth`, `height`
- Data model: `PointObject`: Added bi-directional one-to-many-self-relationship
- Data model: `PointObject`: Added type `curved traffic light post`
- Data model: `PointObject`: Added type `overhead gantry mast`
- Data model: `PointObject`: Added type `overhead gantry`
- Data model: `PointObject`: Added type `reflector post`
- Data model: `PointObject`: Added type `curved street lamp`
- Data model: `PointObject`: Renamed type `boom` to `traffic light boom`
- Data model: `PointObject`: Renamed type `emergencyphone` to `emergency phone`
- Data model: `PointObject`: Renamed type `phonebox` to `phone box`
- Data model: `PointObject`: Renamed type `phonestele` to `phone stele`
- Data model: `PointObject`: Renamed duplicate signal type `1.000.008.10` to `1.000.008.20`
- Data model: `PointObject`: Added default signal type `1.000.000`
- Data model: `PointObject`: Added signal type `F 3`
- Data model: `PointObject`: Added signal type `F 4`
- Data model: `PointObject`: Added signal type `F 5`
- Data model: `PointObject`: Added signal type `A 1`
- Data model: `PointObject`: Added signal type `A X`
- Data model: `PointObject`: Added signal type `A 2b`
- Data model: `PointObject`: Added signal type `W 3`
- Data model: `PointObject`: Added signal type `W 11`
- Data model: `PointObject`: Added signal type `W 12`
- Data model: `PointObject`: Added signal type `W 13`
- Data model: `PointObject`: Renamed signal type `1.000.021` to `F 0`
- Data model: `PointObject`: Renamed signal type `1.000.022` to `F 1`
- Data model: `PointObject`: Renamed signal type `1.000.023` to `F 2`
- Data model: `PointObject`: Renamed signal type `1.000.024` to `F 6` and `W 0`
- Data model: `PointObject`: Renamed signal type `1.000.025.10` to `W 1`
- Data model: `PointObject`: Renamed signal type `1.000.025.20` to `W 2`
- Data model: `PointObject`: Renamed signal type `1.000.026` to `W 14`
- Data model: `PointObject`: Renamed pictogram type `30` to `horizontal 30`
- Data model: `PointObject`: Renamed pictogram type `50` to `horizontal 50`
- Data model: `PointObject`: Corrected spelling of pictogram type `stopping restriciton` to `stopping restriction`
- Data model: `LinearObject`: Added type `tunnel`
- Data model: `LinearObject`: Added type `bridge`

List of abbreviations

DHHN92 Deutsches Haupthöhennetz 1992
dSD dataSourceDescription

EPSG European Petroleum Survey Group
ESRI Environmental Systems Research Institute
ETRS89 European Terrestrial Reference System 1989

ISO Internationale Organisation für Normung

OGC Open Geospatial Consortium
OpenCRG open curved regular grid
OpenDRIVE open digital road information for virtual environmens

SI Système international d'unités
SRID Spatial Reference Identifier

UTM Universal Transverse Mercator coordinate system

WKB well-known binary
WKT well-known text

List of figures

List of tables