



# Road2Simulation- Leitfaden

Leitfaden zur Erhebung von  
Straßendaten für Simulation  
und Entwicklung



## Dokumenteigenschaften

Titel	Road2Simulation-Leitfaden
Betreff	Leitfaden zur Erhebung von Straßendaten für Simulation und Entwicklung
Institut	Verkehrssystemtechnik
Erstellt von	Andreas Richter, Michael Scholz
Beteiligte	
Geprüft von	Michael Scholz, Laura Fischer, Hartmut Friedl
Freigabe von	Michael Scholz
Datum	6. September 2019
Version	1.2.1
Dateipfad	



# Inhaltsverzeichnis

<b>Dokumenteigenschaften . . . . .</b>	<b>1</b>
<b>1. Motivation . . . . .</b>	<b>4</b>
1.1. Lizenz . . . . .	4
1.2. Onlineversion . . . . .	4
1.3. Englischsprachige Übersetzung . . . . .	4
<b>2. Generelle Definitionen . . . . .</b>	<b>5</b>
2.1. Austauschformat . . . . .	5
2.1.1. Straßenbeschreibung . . . . .	5
2.1.2. Oberflächenbeschreibung . . . . .	5
2.2. Koordinatensystem und Raumbezug . . . . .	7
2.3. Geometrietypen . . . . .	7
2.4. Topologische Integrität . . . . .	8
2.5. Zahlentypen . . . . .	8
2.6. Datenquellenbeschreibung . . . . .	9
2.6.1. Raumbezug . . . . .	9
2.6.2. Fehler . . . . .	9
2.6.3. Zeitstempel . . . . .	10
2.6.4. Qualität . . . . .	10
2.6.4.1. Quelle . . . . .	10
2.6.4.2. Nachverarbeitung . . . . .	10
2.7. Abbildungsumfang . . . . .	10
<b>3. Straßenbeschreibung . . . . .</b>	<b>11</b>
3.1. Überblick . . . . .	12
3.2. Straßenverlauf . . . . .	13
3.2.1. StandardLine . . . . .	14
3.2.2. ConnectionLine . . . . .	16
3.3. Kreuzungen . . . . .	19
3.3.1. Kreisverkehre . . . . .	20
3.3.2. Autobahnauf- und -abfahrten . . . . .	22
3.4. Fahrbahnstreifen . . . . .	23

3.5. Markierungen . . . . .	32
3.6. Objekte . . . . .	39
3.6.1. Punktueller Objekte . . . . .	40
3.6.2. Lineare Objekte . . . . .	48
3.7. Bauliche Trennungen . . . . .	51
3.7.1. Verkehrsinsel . . . . .	52
3.7.2. Mittelstreifen . . . . .	54
<b>4. Oberflächenbeschreibung . . . . .</b>	<b>56</b>
4.1. Straßenachsenreferenzierung . . . . .	56
4.2. Oberflächengenauigkeit . . . . .	56
<b>5. Anmerkungen . . . . .</b>	<b>58</b>
<b>6. Änderungshistorie. . . . .</b>	<b>59</b>
<b>Abkürzungsverzeichnis . . . . .</b>	<b>61</b>
<b>Abbildungsverzeichnis. . . . .</b>	<b>62</b>
<b>Tabellenverzeichnis . . . . .</b>	<b>63</b>

# 1. Motivation

Digitale und hoch genaue Kartendaten sind für die Entwicklung und Bewertung moderner Fahrerassistenz- und Automationssysteme nicht mehr wegzudenken. Sie werden entweder als Input in der Simulation für Akzeptanz- und Sicherheitstest genutzt, als auch als Zusatzwissen an Bord der Fahrzeuge. In der Simulation werden spezielle Beschreibungsformate eingesetzt, die eine sehr genaue Modellierung der Umgebung ermöglichen. Die Erhebung von Realdaten und Konvertierung in die Simulationsformate kann jedoch schnell sehr aufwändig werden. Bereits erhobene, digitale Katasterdaten enthalten in der Regel wenige Daten, die eine Konvertierung in die Simulatorformate vereinfachen.

Um den Aufwand der Vermessung von Strecken und Konvertierung von Katasterdaten in Simulationsformate zu reduzieren, wurde der nachfolgende Leitfaden für die standardisierte Erhebung von Straßen- und Oberflächendaten erstellt. Der Leitfaden wurde an das Straßenbeschreibungsformat OpenDRIVE<sup>®1</sup> und das Oberflächenbeschreibungsformat OpenCRG<sup>®2</sup> angelehnt, setzt aber kein Wissen über die Beschreibungsformate voraus. Ein Blick in die entsprechenden Spezifikationen schadet in keinem Fall und entsprechende Dokumente sind auf den jeweiligen Internetseiten frei verfügbar.

## 1.1. Lizenz

Der Road2Simulation-Leitfaden und das Datenmodell sind unter den Bedingungen der Creative Commons Attribution 4.0 International (CC BY 4.0)<sup>3</sup> lizenziert. Die Bildrechte der für die Bildbeispiele genutzten Luftbilder gehören der Stadt Braunschweig, Fachbereich Stadtplanung und Umweltschutz<sup>4</sup>, bzw. der Stadt Stuttgart, Stadtmessungsamt<sup>5</sup>.

## 1.2. Onlineversion

Die Onlineversion dieses Leitfadens ist zu finden unter <https://doi.org/10.5281/zenodo.3375550>.

## 1.3. Englischsprachige Übersetzung

Eine englischsprachige Übersetzung dieses Leitfadens ist zu finden unter <https://doi.org/10.5281/zenodo.3375525>.

---

<sup>1</sup><http://www.opendrive.org>

<sup>2</sup><http://www.opencrg.org>

<sup>3</sup><https://creativecommons.org/licenses/by/4.0/>

<sup>4</sup><http://www.braunschweig.de/geoinformation>

<sup>5</sup><http://www.stuttgart.de/stadtmessungsamt>

## 2. Generelle Definitionen

### 2.1. Austauschformat

#### 2.1.1. Straßenbeschreibung

Die vermessenden Daten sollten in einem offenen Format gespeichert werden, welches eine Geometrirepräsentierung als OGC Simple Features unterstützt (siehe Abschnitt 2.3). Als Zeichenkodierung ist dabei UTF-8 zu verwenden. Empfohlen werden beispielsweise die folgenden Austauschformate:

- Einfacher Datenbank-Dump aus Geodatenbanken wie zum Beispiel PostGIS, Oracle Spatial oder SpatiaLite.
- OGC GeoPackage<sup>6</sup> als SpatiaLite-Container. SpatiaLite dient dabei als räumliche Erweiterung zu SQLite. SQLite als dateibasierte Datenbank eignet sich optimal für den Datenaustausch und bietet weit verbreitete Datenbankfunktionalität. Mehrere räumliche Tabellen können auf diese Weise in einer Datei hinterlegt werden, wobei Beziehungen untereinander abgebildet werden können. Durch frei verfügbare Werkzeuge ist die Migration zwischen SpatiaLite und anderen Geodatenbanksystemen gewährleistet. GeoPackage wird von vielen Geoinformationssystemen unterstützt und erlaubt die Definition häufig verwendeter Koordinatenreferenzsysteme. Weiterhin können in einem GeoPackage neben Vektordaten auch Rasterdaten hinterlegt werden.
- Das ESRI Shapefile eignet sich als Austauschformat nur bedingt, da es Einschränkungen bei Feldnamen (auf zehn Zeichen beschränkt) und den Wertebereichen von Datentypen aufweist.

Es bietet sich generell an, das zugrundeliegende Höhenmodell der 3D-Simple-Features ebenfalls auszuliefern, da bei Arbeitsschritten, die die Geometrien nachträglich verändern (bspw. Interpolation, maschinelle Korrektur) Höhenangaben ungültig werden können und entsprechend neu abgegriffen werden müssen. Höhenmodelle (Raster/Vektor) können beispielsweise im selben GeoPackage mit den Road2Simulation-Objekten verpackt werden.

Abbildung 1 zeigt die Struktur des Datenmodells, wie es in einer Datenbank abgebildet werden kann. Kapitel 3 beschreibt detailliert die Vorgehensweise zur Erfassung der Straßenelemente und wie diese abzulegen sind. Die attribuierten Geometriedaten werden um Metainformationen ergänzt, wie in Abschnitt 2.6 erläutert.

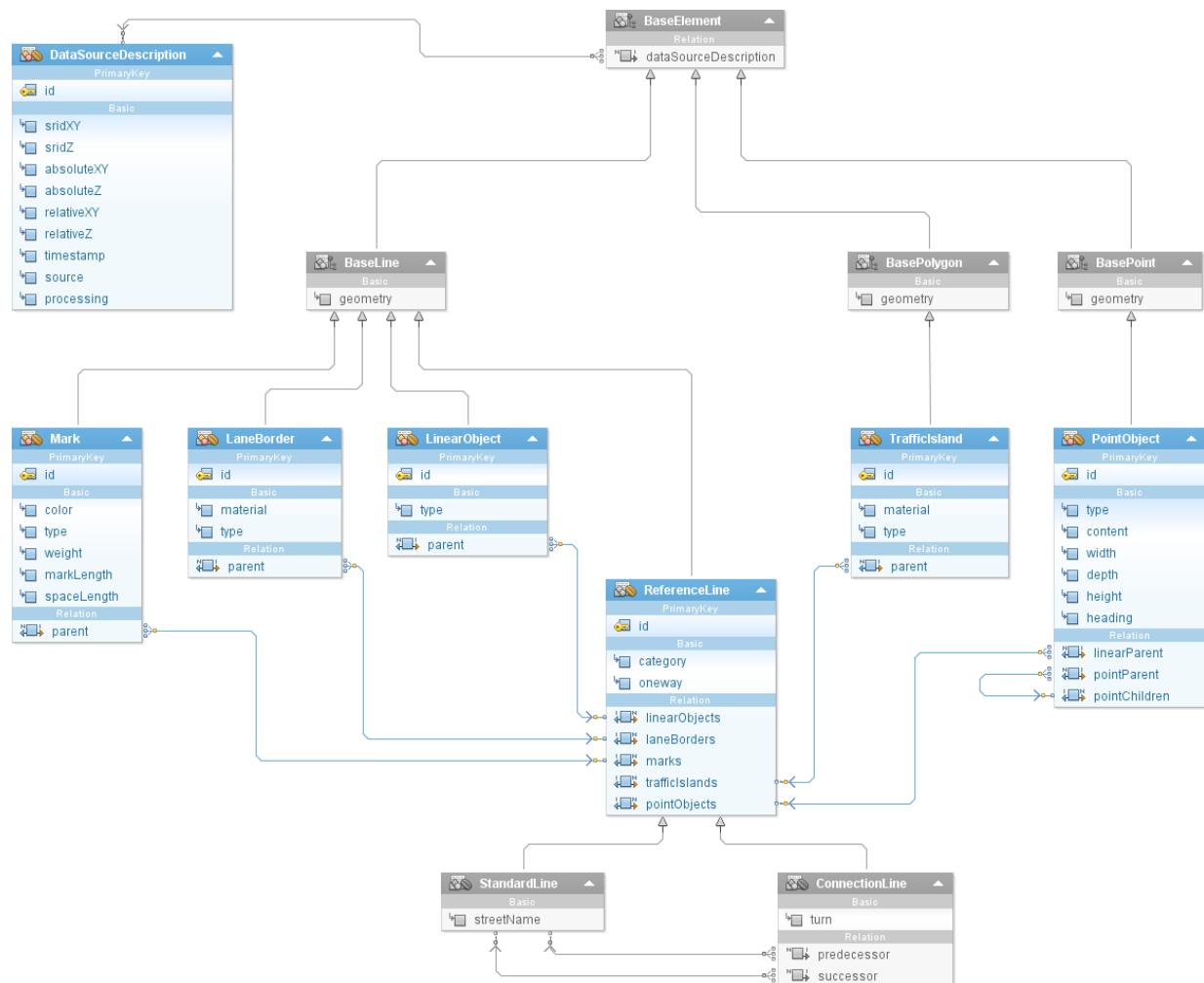
#### 2.1.2. Oberflächenbeschreibung

Die vermessenden Daten sollen binär im OpenCRG-Format abgespeichert werden, da eine komplexere Datentransformation wie für den Straßenverlauf in das OpenDRIVE-Format nicht notwendig ist. Alle weiteren wesentlichen Randbedingungen sind in Kapitel 4 beschrieben.

---

<sup>6</sup><http://www.geopackage.org/>





**Abbildung 1:** Struktur des Datenmodells. Persistente Tabellen sind als farbige Klassen dargestellt wohingegen graue Elemente der hierarchischen Gliederung dienen.

## 2.2. Koordinatensystem und Raumbezug

Alle geometrischen Messpunkte tragen eine dreidimensionale Lageinformation ( $x$ ,  $y$ ,  $z$ ). Die Koordinaten können beispielsweise in einem *projizierten* Koordinatenreferenzsystem angegeben werden und setzen sich wie folgt zusammen:

- $x$  in Metern als Rechtswert in der jeweiligen Projektion,
- $y$  in Metern als Hochwert in der jeweiligen Projektion,
- $z$  in Metern als Höhe über dem zugrundeliegenden Referenzellipsoid oder als Höhe in einem abweichenden Höhenreferenzsystem.

Die Wahl des räumlichen Koordinatenreferenzsystems auf Basis von projizierten Koordinaten orientiert sich an amtlichen Vorgaben und fällt im Bereich Mitteldeutschland beispielsweise auf ETRS89 / UTM Zone 32N (EPSG:25832<sup>7</sup>). Wird die Höhe abweichend vom Referenzellipsoid angegeben, so ist zusätzlich das verwendete Höhenreferenzsystem anzugeben, beispielsweise eine Höhe über NHN im DHHN92 (EPSG:5783<sup>8</sup>).

Liegen die Koordinaten in einem *geografischen* Referenzsystem vor, so setzt sich die Lageinformation wie folgt zusammen:

- $x$  in Grad als geografische Länge (longitude),
- $y$  in Grad als geografische Breite (latitude),
- $z$  in Metern als Höhe über dem zugrundeliegenden Referenzellipsoid.

Die Wahl des geografischen Koordinatenreferenzsystems orientiert sich an amtlichen Vorgaben und fällt im europäischen Raum üblicherweise auf ETRS89 (EPSG:4258<sup>9</sup>) oder WGS84 (EPSG:4326<sup>10</sup>).

Es ist generell zu empfehlen, die Daten in einem *projizierten* Koordinatenreferenzsystem zu referenzieren. Die Bezeichnung des gewählten Referenzsystems wird explizit als Teil der Datenquellenbeschreibung angegeben, wie in Abschnitt 2.6.1 erläutert.

Rotationen werden nur geographisch absolut angegeben, es wird also ein Linkshandsystem genutzt (im Uhrzeigersinn positiv). Entsprechend betragen die Winkel  $N = 0^\circ$ ,  $O = 90^\circ$ ,  $S = 180^\circ$ ,  $W = 270^\circ$ .

## 2.3. Geometrietypen

Geometrische Datentypen sollen *dreidimensional* im festgelegten Bezugssystem als Simple Features im Format well-known binary (WKB) oder well-known text (WKT) nach OGC Simple Feature Access – Part 1: Common Architecture, Version 1.2.1<sup>11</sup> angegeben werden.

Beispiel für ein punktförmiges 3D-Objekt als WKT:

- Point Z (605044.419819 5791949.77898 128.12)

Beispiel für ein linienförmiges 3D-Objekt mit zwei Stützpunkten als WKT:

- LineString Z (

<sup>7</sup><http://epsg.io/25832>

<sup>8</sup><http://epsg.io/5783>

<sup>9</sup><http://epsg.io/4258>

<sup>10</sup><http://epsg.io/4326>

<sup>11</sup><http://www.opengeospatial.org/standards/sfa>

```
605059.7409 5791956.6 128.12,
605014.3896 5791935.49 128.22)
```

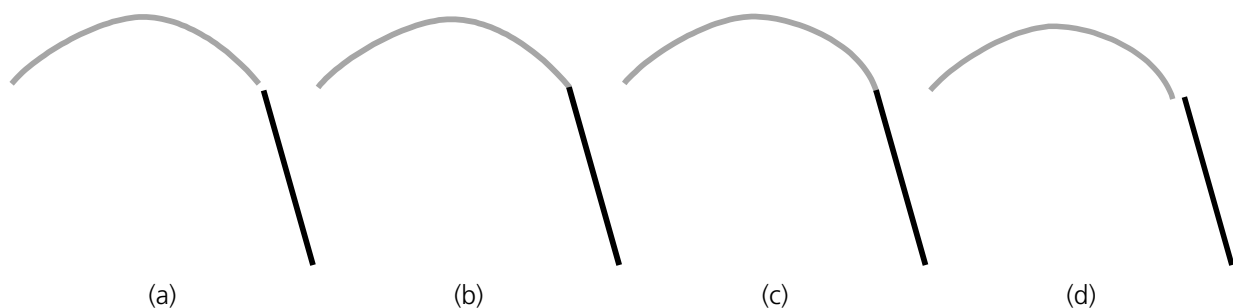
Beispiel für ein flächenförmiges 3D-Objekt mit vier Stützpunkten als WKT:

```
• 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. Topologische Integrität

Für die erleichterte Nachverarbeitung der erhobenen Daten ist darauf zu achten, dass topologisch in Beziehung stehende Objekte geometrisch konsistent modelliert werden. Sich überlagernde Objekte teilen sich an den Überlagerungspunkten identische geometrischen Informationen. Punktdaten, wie am selben Mast befestigte Schilder, unterscheiden sich geometrisch dabei beispielsweise nur in der z-Komponente ihrer Lageinformation oder in ihrer Ausrichtung.

Topologisch aufeinanderfolgende Linienzüge müssen die Punktstetigkeit ( $C^0$ -Stetigkeit) erfüllen, sich beim Übergang also denselben geometrischen Punkt teilen. Die Modellierung von Linienzügen soll nach Möglichkeit ebenfalls die Tangentenstetigkeit ( $C^1$ -Stetigkeit) erfüllen. In späteren Beispielen wird auf aufeinanderfolgende, parallel versetzte Linienzüge eingegangen, die nach Möglichkeit an ihren topologischen Verbindungsstellen ebenfalls dieselbe Tangentensteigung aufzuweisen haben. Abbildung 2 verdeutlicht geometrische Unterschiede dieser Stetigkeitsbedingungen.



**Abbildung 2:** Geometrischer Übergang zwischen aufeinanderfolgenden Linienzügen: unstetig (a),  $C^0$ -stetig (b),  $C^1$ -stetig (c), paralleler Versatz mit derselben Tangentensteigung (d).

## 2.5. Zahlentypen

Ganze Zahlen (Integer) sollten in der Regel als `uint64_t` angegeben werden, also nur positive Zahlen von 0 bis  $2^{64}-1 = \text{max}$ . Gleitkommazahlen (Float), insbesondere bei Koordinatenangaben, sollten in der Regel als `Double` angegeben werden. In der Regel sind die Zahlen positiv und bewegen sich innerhalb der Grenzen von 0.0 bis  $(2-2^{-52}) \cdot 2^{1023} = \text{max}$ .

## 2.6. Datenquellenbeschreibung

Zur vereinfachten Interpretation und zur Qualitätsbeschreibung des erfassten Datenbestands ist dieser um die im Folgenden beschriebenen Metainformationen zu ergänzen. Bei einer Datenlieferung erfolgt dies durch die Zuordnung eines entsprechenden `DataSourceDescription`-Objekts (im Verlaufe des Textes als `dSD` abgekürzt) zu jedem Geometrieobjekt. Tabelle 1 gibt Beispiele für mögliche Metainformationen, die Einträge werden in den nachfolgenden Abschnitten beschrieben.

**Tabelle 1:** Beispiel für Metainformationen von Geometriedatensätzen.

DataSourceDescription									
id	sridXY	sridZ	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. Raumbezug

Wie in Abschnitt 2.2 beschrieben, wird für erfasste Geometriedaten der Raumbezug eindeutig definiert. Die European Petroleum Survey Group (EPSG) beschreibt häufig verwendete Referenzsysteme (geografisch, projiziert, vertikal) über eindeutige Bezeichner, sogenannte Spatial Reference Identifier (SRID). Für die Koordinatenkomponenten `xy` und `z` sind die sie beschreibenden EPSG-SRID entsprechend getrennt anzugeben.

- `sridXY`
- `sridZ`

Dabei ist `sridZ` optional und nur anzugeben, sofern Höhenangaben abweichend von dem bereits über `sridXY` definierten Ellipsoiden beschrieben werden.

### 2.6.2. Fehler

Der relative und absolute Fehler der erfassten `xy`- sowie `z`-Komponente ist als `Double` anzugeben. Die Einheit hängt dabei von dem jeweilig verwendeten Koordinatenreferenzsystem ab (siehe Abschnitte 2.2 und 2.6.1) und ist in der Regel SI-Meter (m) oder SI-Grad (°).

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

Der relative Fehler entspricht dabei der Präzision, der absolute Fehler der Richtigkeit. Sind beide Fehler gering, ist die Genauigkeit hoch<sup>12</sup>.

<sup>12</sup><https://de.wikipedia.org/wiki/Genauigkeit>



### 2.6.3. Zeitstempel

Der Erfassungszeitpunkt (`timestamp`) der Datenlieferung ist als ISO 8601-Datum (`YYYY-MM-DDT HH:MM:SS`) anzugeben. Die Uhrzeitangabe `T` ist optional.

### 2.6.4. Qualität

Mit der Quellenangabe soll eine natürlichsprachliche Information über die Herkunft und Verarbeitung der Daten mitgegeben werden. Es wird eine Quelle (`source`) und die Art der Nachverarbeitung (`processing`) angegeben.

#### 2.6.4.1. Quelle

Die Angabe kann beispielsweise aus mehreren Quellenangaben für unterschiedliche Sensoren mit unterschiedlichen Firmware-Versionen bestehen, oder aus unterschiedlichen Katastern stammen. Angabe entsprechend der Kategorie ergänzt mit Bezeichnung/Name als `String`.

- Sensorname und Firmware-Version (kann auch „mobile mapping“ subsumiert werden) = `sensor`
- Katasterdaten = `cadaster`
- generierte oder abdigitalisierte Daten ohne Nutzung von Sensorik = `custom`

#### 2.6.4.2. Nachverarbeitung

In dieser Angabe wird definiert, ob und in welcher Weise die Daten nachbearbeitet wurden. Die Angabe erfolgt in folgenden Kategorien als `String`:

- direkt aus dem Sensor = `raw`
- Ausreißer und Fehlstellen bereinigt = `cleaned`
- Werte sind bereits geclustert, gemittelt, transformiert worden = `processed`
- Werte kommen aus unterschiedlichen Datenquellen = `fused`

## 2.7. Abbildungsumfang

Dieser Leitfaden unterteilt sich in einen Standard- und einen erweiterten Abbildungsumfang. Dies wird mit einer grauen Hinterlegung für Attribute des erweiterten Abbildungsumfangs dargestellt:

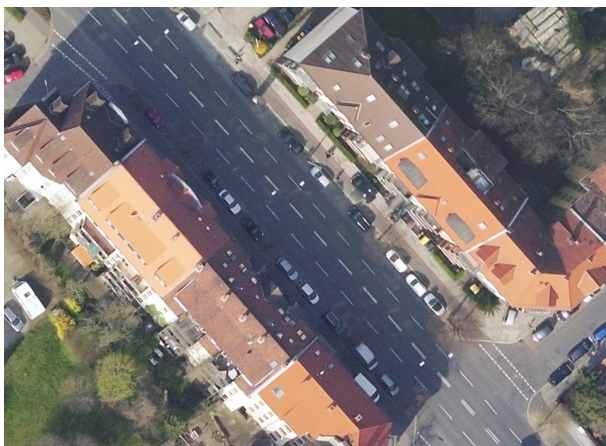
- Standardabbildungsumfang
- erweiterter Abbildungsumfang

### 3. Straßenbeschreibung

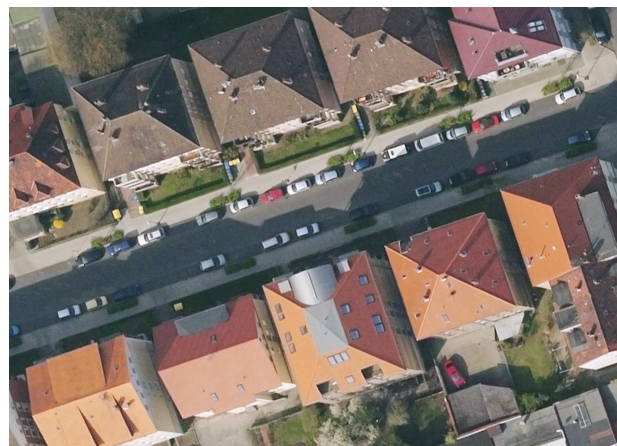
Die Straßenbeschreibung besteht in der Regel aus topologischen und topographischen Informationen. Die topologischen Daten spannen ein Knoten-Kanten-Modell auf, in dem alle Elemente miteinander mittels Vorgänger- und Nachfolgerbeziehung verknüpft sind. Darin sind zum Teil auch Informationen enthalten, die Einfluss auf die Nutzung der Straßen haben (bspw. Schilder). Die topographischen Daten beschreiben das Aussehen der Straße, in dem bspw. Borde, Art der Fahrbahnmarkierung u.v.a.m. modelliert sind.

Die im weiteren Verlauf dieses Leitfadens beschriebenen Elemente werden jeweils einleitend in sechs Überblicksgrafiken gezeigt. In Tabelle 2 sind die dafür genutzten Straßensituationen ohne modellierte Elemente dargestellt. Diese in Tabelle 2 dargestellten Situationen können als Vergleich für die folgenden Übersichten genutzt werden, in denen dann die entsprechenden Elemente eingezeichnet sind.

**Tabelle 2:** Unterschiedliche Straßensituationen ohne modellierte Elemente.



einbahnig, mehrere Streifen pro Richtung



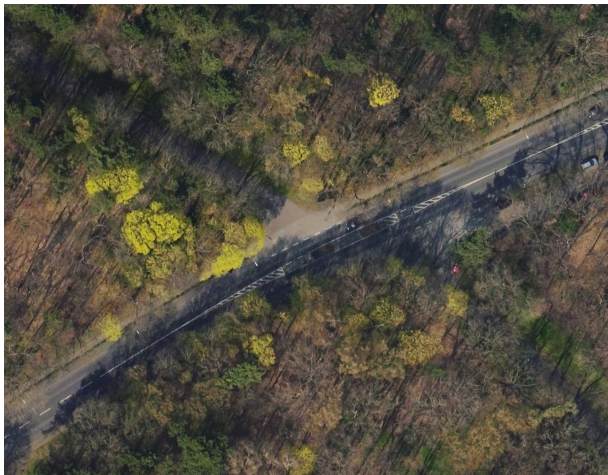
einbahnig, ein Streifen pro Richtung



zweibahnig, mehrere Streifen pro Richtung



einbahnig, ein Streifen in eine Richtung



einbahnig, ein Streifen pro Richtung mit Trennung



mehrbahnig, unterschiedliche Streifen pro Richtung

### 3.1. Überblick

Alle wesentlichen Elemente der Straßentopographie werden als Linienzüge (bspw. Straßenverlauf, Bordsteinkanten, Fahrbahnmarkierungen) oder Einzelkoordinaten (bspw. Straßenbeleuchtung, Beschilderung, Lichtsignalanlagen) modelliert. Mittelinseln stellen eine Ausnahme dar, denn ihre Bauweise entsteht durch das Zusammentreffen mehrerer Straßenverläufe. Daher werden Mittelinseln als Flächen modelliert.

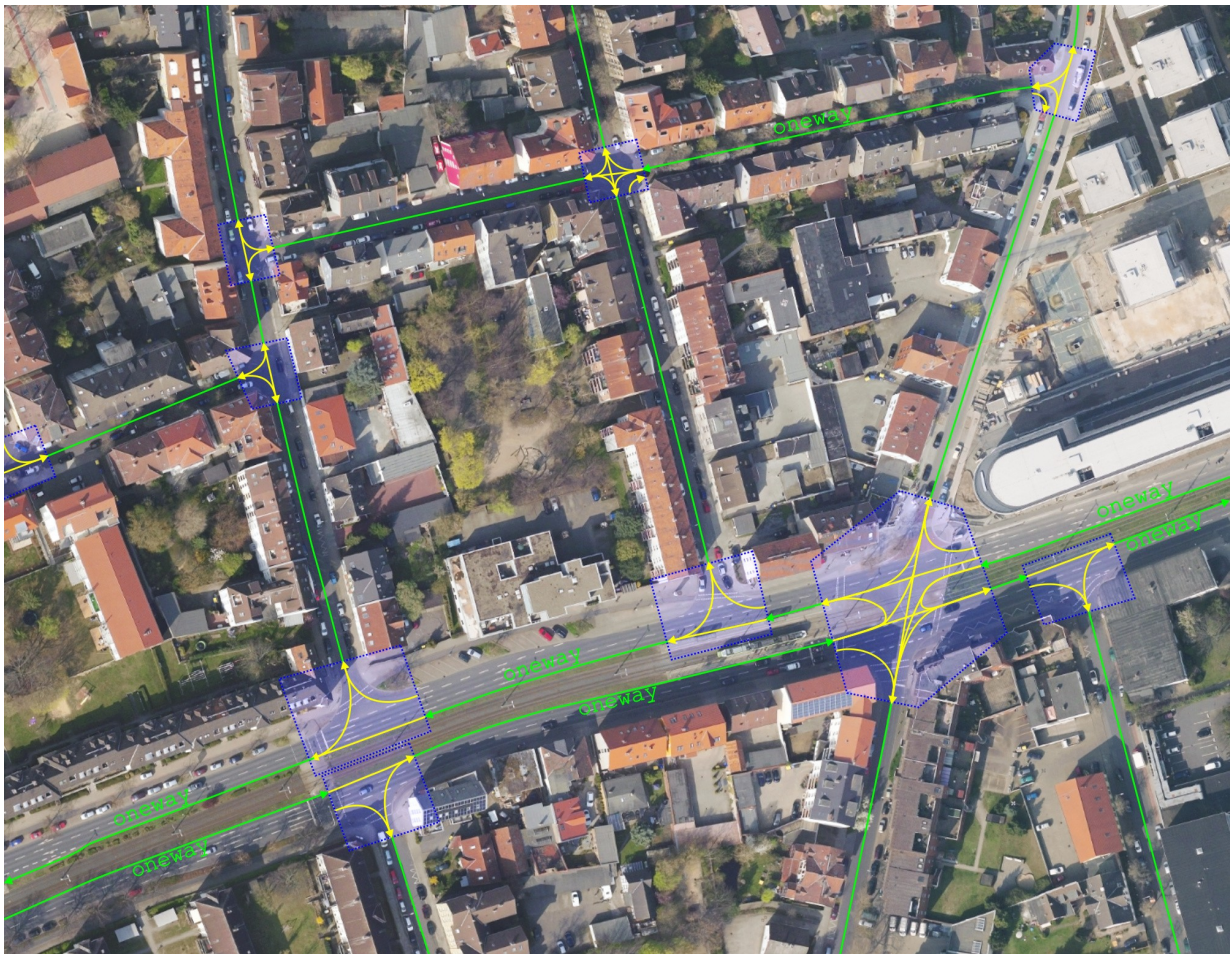
Die Linienzüge (LineStrings als Simple Features<sup>13</sup>) können einfach erhoben und gespeichert werden. Für eine genaue und effiziente Modellierung sollten die Stützstellen eines Linienzugs im Krümmungsbereich möglichst dicht sein, in geraden Abschnitten möglichst groß. Die Richtung der Linienzüge wird implizit durch die Reihenfolge der Stützstellen definiert. Die absolute Abweichung der Straßentopographie sollte für Katasteranwendungen 100 mm in der zweidimensionalen Lage nicht überschreiten. Für Anwendung in der Simulation sollte der zweidimensionale Lagefehler maximal zwischen 20 und 50 mm liegen. Der Höhenfehler sollte davon nicht zu stark abweichen. Somit sind die Hauptklassen `StandardLine`, `ConnectionLine` und `LaneBorder` im Grunde ähnlich und unterscheiden sich nur durch spezifische Attribute.

Wesentlich strukturierende Elemente sind Straßenzüge und Kreuzungen. Die Elemente leiten sich aus der OpenDRIVE-Spezifikation ab. Straßenzüge führen von Kreuzung zu Kreuzung, es kann kein Straßenzug direkt an einen weiteren Straßenzug anschließen. Der Straßenzug beschreibt alle Eigenschaften des Abschnitts, beinhaltet somit Informationen über die Fahrbahn und Infrastruktur. Straßenzüge werden mittels `StandardLines` modelliert.

Die Kreuzung ist in diesem Leitfaden als logischer Container für das Zusammenfassen aller Verknüpfungen zu verstehen. Innerhalb einer Kreuzung werden die eingehenden Straßenzüge mittels Verknüpfungen verbunden, die sich ähnlich wie Straßenzüge aber mit einigen Einschränkungen verhalten. Diese Verknüpfungen werden als `ConnectionLines` modelliert. Abbildung 3 zeigt ei-

<sup>13</sup><http://www.opengeospatial.org/standards/sfa>





**Abbildung 3:** Übersicht über Straßenzüge (grün), Kreuzungen (blau) und darin enthaltene Verknüpfungen (gelb).

ne Übersicht mit mehreren Kreuzungen, die von Straßenzügen verbunden werden. Innerhalb der Kreuzungen sind alle notwendigen Verbindungen mittels Verknüpfungen modelliert.

### 3.2. Straßenverlauf

Der Straßenverlauf in Lage und Höhe wird mittels der sogenannten *StandardLine* modelliert. Eine *StandardLine* wird bei Straßen ohne bauliche Trennung in die Mitte der Straße gelegt, dort, wo sich die Fahrstreifen beider Fahrtrichtungen berühren. Bei baulicher Trennung (bspw. Mittelstreifen) oder Einbahnstraßen wird die *StandardLine* an die in Fahrtrichtung linke Begrenzung (Außenkante) der Fahrstreifen gelegt. Innerhalb von Kreuzungen werden die *StandardLines* als *ConnectionLines* bezeichnet. Sie modellieren die möglichen Abbiegebeziehungen einer Straße. *StandardLines* und *ConnectionLines* stellen die Kernelemente des gesamten Datenmodells dar und dienen als *Referenzobjekte* für die logische Zuordnung anderer abzubildender Objekte. *StandardLine* und *ConnectionLine* nutzen im Datenmodell (siehe Abbildung 1) denselben ID-Bereich. Es kann daher keine *StandardLine* und *ConnectionLine* mit derselben ID existie-



ren.

Jede `StandardLine` und jede `ConnectionLine` hat eine Richtung. Tabelle 4 zeigt Modellierungsbeispiele mit in Grün dargestellter `StandardLines` und gelben `ConnectionLines`. Pfeilspitzen geben eine explizite Richtung an. Ohne Pfeilspitze sind beide Richtungen möglich. Zudem gehören `StandardLines` und `ConnectionLines` einer Straßenkategorie an, aus welcher weitere Eigenschaften bei weiterer maschineller Verarbeitung implizit abgeleitet werden können.

### 3.2.1. StandardLine

Die `StandardLine` modelliert wie in Abschnitt 3.2 beschrieben den Straßenverlauf zwischen Kreuzungen. Daher wird der `StandardLine` eine Straßenkategorie zugeordnet. Diese kann folgende Ausprägungen haben:

- nicht näher definiert = `unknown`
- Landstraße = `rural`
- Autobahn = `motorway`
- Stadt = `town`
- 30er-Zone = `low speed`
- verkehrsberuhigter Bereich = `pedestrian`
- Fahrradstraße = `bicycle`

Des Weiteren wird mit dem `oneway`-Attribut festgelegt, ob der Straßenabschnitt nur in Modellierungsrichtung befahren werden kann (`oneway = true`), oder in beide Richtungen (`oneway = false`). Zusätzlich kann einem entsprechenden Straßenverlauf auch ein Name („Bahnhofstraße“) oder eine Bezeichnung („L118“) zugeordnet werden. Eine `StandardLine` ist somit durch die in Tabelle 3 aufgeführten Attribute beschrieben. Tabelle 5 zeigt ein Modellierungsbeispiel für `StandardLines` mit Tabellenauszug.

**Tabelle 3:** Eigenschaften von `StandardLines`.

Name	Anzahl	Typ	Wert	Beschreibung
<b>id</b>	1	unit64_t	[0, max]	eindeutige Kennung
<b>dataSource-Description</b>	1	unit64_t	[0, max]	ID des beschreibenden <code>DataSourceDescription</code> -Objekts
<b>geometry</b>	1	LineString Z	siehe Beispiel in Abschnitt 2.3	Linienzug als WKT oder WKB
<b>category</b>	1	string	unknown, rural, motorway, town, low speed, pedestrian, bicycle	Straßenkategorie
<b>oneway</b>	1	boolean	true, false	true, wenn <code>StandardLine</code> eine Einbahnstraße ist, ansonsten false
<b>streetName</b>	0..1	string		Straßenname/-bezeichnung

**Tabelle 4:** Modellierung der StandardLines (grün) und ConnectionLines (gelb) in unterschiedlichen Situationen.



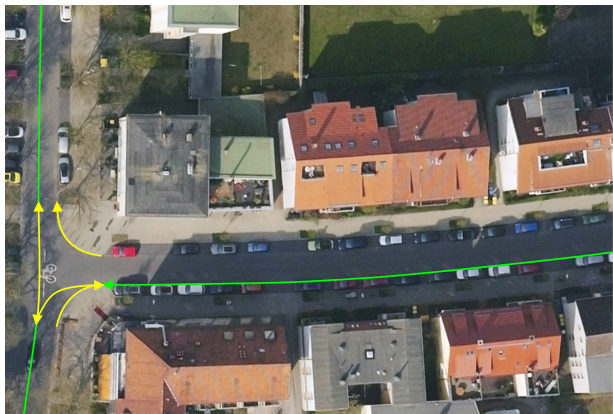
einbahnig, mehrere Streifen pro Richtung



einbahnig, ein Streifen pro Richtung



zweibahnig, mehrere Streifen pro Richtung



einbahnig, ein Streifen in eine Richtung



einbahnig, ein Streifen pro Richtung mit Trennung



mehrbahnig, unterschiedliche Streifen pro Richtung



**Tabelle 5:** Beispiel zweier `StandardLines` (grün) mit Tabellenauszug.

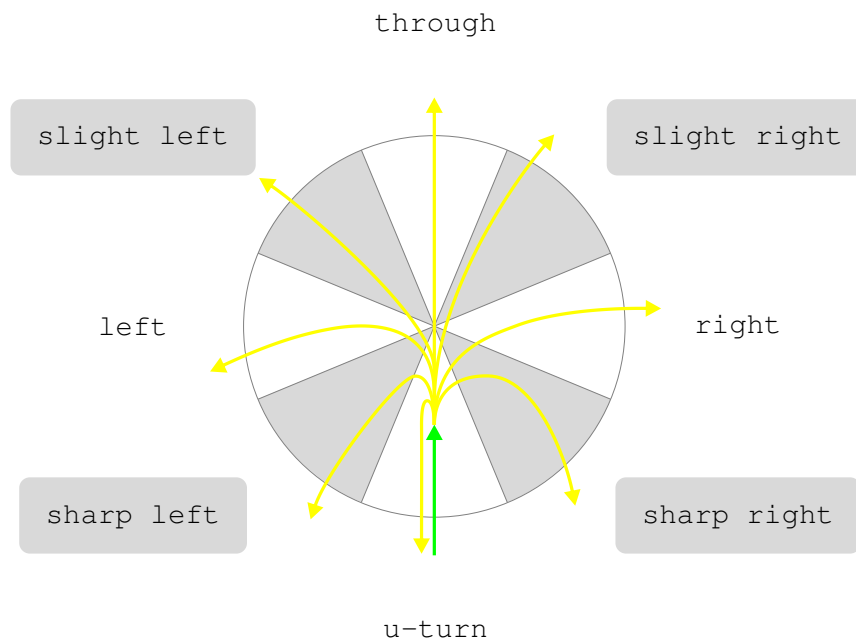
ReferenceLine (StandardLine)					
id	dSD	geometry	category	oneway	streetname
100	0	WKT/WKB	low speed	false	Broitzemer Str.
101	0	WKT/WKB	low speed	true	Virchowstr.

### 3.2.2. ConnectionLine

Die `ConnectionLines` modellieren die Abbiegebeziehungen auf Kreuzungen. Daher gibt es eine Angabe über die Art der Abbiegebeziehung und über etwaige Vorgänger und Nachfolger. Vorgänger und Nachfolger werden im Bezug zur geometrischen Linienrichtung, also ihrer Punktreihenfolge, definiert. Im topologischen Modell stellt eine `ConnectionLine` eine logische Verknüpfung zwischen zwei `StandardLines` her und hat in solch einem Fall immer genau *einen* Vorgänger und *einen* Nachfolger.

Kann eine entgegengerichtete Abbiegebeziehung durch *dieselbe* `ConnectionLine` modelliert werden, wird über das Attribut `oneway = false` eine bidirektionale Verknüpfung modelliert. Auf diese Weise können redundante Geometrie vermieden werden.

Gibt es keine logische Beziehung zu anknüpfenden `StandardLines`, so kann eine `ConnectionLine` auch lediglich als Referenz für die geometrische Modellierung weiterer `LaneBorders` im Kreuzungsbereich verwendet werden (Behelfs-`ConnectionLine`). Ein Beispiel für solche Situationen ist in Tabelle 9 unten links bzw. unten rechts abgebildet.



**Abbildung 4:** Abbiegebeziehung.

Analog zur `StandardLine` wird auch für die `ConnectionLine` eine Befahrriichtung und Straßenkategorie angegeben. Hinzu kommt die Definition der Art der Abbiegebeziehung `turn` als weiteres Attribut:

- geradeaus = `through`
- links = `left`
- rechts = `right`
- wenden = `u-turn`
- Behelfs-`ConnectionLine` = `auxiliary`

- leicht links = `slight left`
- scharf links = `sharp left`
- leicht rechts = `slight right`
- scharf rechts = `sharp right`

Die Zuordnung der Abbiegebeziehung richtet sich nach einem in acht Segmente unterteilten Vollkreis-Schema, wie in Abbildung 4 dargestellt. Die Zuordnung zu den Segmenten kann allerdings je nach Situation unschärfer betrachtet werden. Eine `ConnectionLine` ist somit durch die in Tabelle 6 aufgeführten Attribute beschrieben. Tabelle 7 zeigt ein Modellierungsbeispiel einer `ConnectionLine` mit Tabellenauszug.



**Tabelle 6:** Eigenschaften von ConnectionLines.

Name	Anzahl	Typ	Wert	Beschreibung
<b>id</b>	1	unit64_t	[0, max]	eindeutige Kennung
<b>dataSource-Description</b>	1	unit64_t	[0, max]	ID des beschreibenden DataSourceDescription-Objekts
<b>geometry</b>	1	LineString Z	siehe Beispiel in Abschnitt 2.3	Linienzug als WKT oder WKB
<b>category</b>	1	string	unknown, rural, motorway, town, low speed, pedestrian, bicycle	Straßenkategorie wie bei StandardLine
<b>oneway</b>	1	boolean	true, false	false, wenn ConnectionLine eine bidirektionale Verknüpfung darstellt, ansonsten true
<b>turn</b>	1	string	through, u-turn, left, slight left, sharp left, right, slight right, sharp right, auxiliary	
<b>predecessor</b>	1	unit64_t	[0, max]	ID des vorhergehenden Elements (eingehende StandardLine)
<b>successor</b>	1	unit64_t	[0, max]	ID des nachfolgenden Elements (ausgehende StandardLine)

**Tabelle 7:** Beispiel einer ConnectionLine (gelb) mit Tabellenauszug.


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

### 3.3. Kreuzungen

Kreuzungen sind überall dort notwendig, wo sich die Anzahl der zu verknüpfenden *StandardLines* ändert. Das kann eine klassische Kreuzungssituation sein (Straßenverlauf geht nach links, rechts, geradeaus), aber auch das Einfädeln auf die Autobahn (siehe Tabelle 4).

Alle *ConnectionLines* werden ebenfalls an der linken Seite des äußerst linken Fahrstreifens modelliert. Entsprechend beginnen und enden nicht alle Verknüpfungen an der geometrisch selben Position wie die ein- bzw. ausgehende *StandardLine* beginnt bzw. endet, sondern können auch orthogonal verschoben modelliert werden. Somit beginnen und enden *ConnectionLines* *auf gleicher Höhe* (maximal longitudinale Ausdehnung) relativ zur *StandardLine* (siehe auch Abschnitt 3.4). Schließt eine *ConnectionLine* direkt an eine *StandardLine* an, muss der Übergang im Rahmen des Möglichen  $C^1$ -stetig sein, wie in Abschnitt 2.4 erläutert.

Bei der Modellierung der *ConnectionLines* ist darauf zu achten, dass diese dem natürlichen Fahrverhalten der Menschen entsprechen, wenn es nicht bspw. durch Fahrbahnmarkierungen bereits Anhaltspunkte zur Modellierung gibt. Tabelle 8 zeigt Modellierungsbeispiele von *Connec-*

tionLines im Kreuzungsbereich.

Ein Spezialfall sind Wendemöglichkeiten. Diese sollen bei Kreuzungen, in denen bauliche Trennungen vorhanden sind, immer modelliert werden, auch wenn sie ggf. technisch nicht fahrbar oder regelwidrig wären. Eine `ConnectionLine` vom Typ `u-turn` wird zur Modellierung und Verortung von `LaneBorders` oder `TrafficIslands` benötigt.

**Tabelle 8:** `ConnectionLines` im Kreuzungsbereich.



`ConnectionLines` von Norden ausgehend



`ConnectionLines` von Osten ausgehend



`ConnectionLines` von Süden ausgehend



`ConnectionLines` von Westen ausgehend

Komplexe Kreuzungen müssen in kleinere Einheiten unterteilt werden, um die Modellierungskomplexität gering zu halten. Dies gelingt immer dort gut, wo wiederum bauliche Trennungen auftauchen, bspw. mit Verkehrsinseln insbesondere bei Kreisverkehren. Abbildung 5 zeigt eine beispielhaft aufgeteilte Kreuzung, Abschnitt 3.3.1 beschreibt Kreisverkehre detaillierter.

### 3.3.1. Kreisverkehre

Um eine Unterteilung eines Kreisverkehrs durchzuführen, sollte jeweils ein Segment auf dem Kreis, das gegenüber einer Einmündung liegt, als kurze `StandardLine` definiert werden. Somit lassen sich alle Abbiegebeziehungen einfach modellieren. Tabelle 9 veranschaulicht die Unterteilung eines Kreisverkehrs.





**Abbildung 5:** Komplexe Kreuzung aufgeteilt in mehrere einfachere Einzelkreuzungen.

**Tabelle 9:** ConnectionLines im unterteilten Kreisverkehr.



ConnectionLines vom Zentrum ausgehend

ConnectionLines von Nordosten ausgehend



ConnectionLines von Südosten ausgehend



ConnectionLines von Westen ausgehend



ConnectionLines von Norden ausgehend



ConnectionLine von Südwesten ausgehend

Die letzten beiden Grafiken in Tabelle 9 für die Richtungen Nord und Südwest stellen dabei bereits einen Sonderfall für die spätere Definition der Fahrbahnstreifen (siehe Abschnitt 3.4) dar: Da unter normalen Umständen `ConnectionLines` nicht entgegen der Fahrtrichtung von Einbahnstraßen definiert werden, müssen zur Modellierung etwaiger Randstreifen (Bürgersteig, Fahrradweg, o.ä.) Behelfs-`ConnectionLines` erstellt werden, um diese Randstreifen trotzdem logisch zuordnen zu können.

### 3.3.2. Autobahnauf- und -abfahrten

Im Fall von Autobahnauf- und -abfahrten treffen sich wie bei einer Kreuzung zwei (gerichtete) `StandardLines`, die daher auch mittels gerichteter `ConnectionLines` zu verknüpfen sind, wie in Abbildung 6 dargestellt. Diese `ConnectionLines` werden als through-Abbiegebeziehung modelliert, der Fahrstreifenwechsel in der Simulationsanwendung findet auf den Fahrstreifen der nachfolgenden bzw. vorhergehenden `StandardLines` statt.





**Abbildung 6:** StandardLines und ConnectionLines bei Autobahnauf- und -abfahrten.

### 3.4. Fahrbahnstreifen

Das Aussehen der Straße wird durch die Fahrbahn mittels ihrer Streifen und deren Breite definiert. Fahrbahnstreifen können Fahrstreifen (Taxi- und Busfahrstreifen), Parkstreifen und Seitenstreifen, Sperrflächen, Fahrradwege, Gehwege, Grünstreifen, Straßenbahntrassen und Verkehrsinseln sein. Dabei sind die (baulichen) Grenzen die wesentlichen Informationen, die wieder als Linienzug aufgenommen werden. Diese Linien werden `LaneBorder` genannt. `StandardLines` haben immer mindestens eine `LaneBorder` wohingegen `ConnectionLines` auch ohne `LaneBorder` auskommen können. In der Regel ist hier aber auch eine Modellierung von `LaneBorders` hilfreich.

An Kreuzungen werden die `StandardLines` durch `ConnectionLines` weitergeführt. Teilweise laufen bei großen Kreuzungen einige `LaneBorders` allerdings über das Ende einer `StandardLine` hinaus und gehören dann zu einer `ConnectionLine`. In diesem Fall muss die `LaneBorder` auf gleicher Höhe enden (Schnittpunkt der Senkrechten der `StandardLine`, parallel verschoben zur `StandardLine` und nicht über die Ausdehnung dieser hinaus) und eine neue `LaneBorder` mit gleichen Eigenschaften beginnen. Diese Trennung auf gleicher Höhe ist in den folgenden

Bildbeispielen mit einer roten Linie dargestellt. Aufeinander folgende Linien müssen im Rahmen des Möglichen  $C^1$ -stetig sein, wie in Abschnitt 2.4 erläutert. Zu Beginn der Modellierung von LaneBorders sollte sichergestellt werden, dass alle Anforderungen der StandardLines und ConnectionLines (bspw. Stetigkeiten) erfüllt sind, da bei später notwendigen Anpassungen dieser Referenzlinien erhebliche Anpassungen bei vielen LaneBorders notwendig werden können.

LaneBorders sind Linienzüge, die auch geometrisch auf anderen bereits existierenden Linienzügen (StandardLine, ConnectionLine) liegen können. Dies kommt besonders bei baulicher Trennung zum Tragen, siehe Abschnitt 3.7.

Tabelle 10 zeigt Modellierungsbeispiele mit in Cyan dargestellten LaneBorders (definiert durch Fahrbahnmarkierungen, Bordsteine oder anderen baulichen Unterschieden) sowie in Rot eingezeichneten StandardLine-Enden. Diese Begrenzungslinien verdeutlichen die maximal longitudinale Ausdehnung einer LaneBorder entlang ihrer jeweilig zugeordneten StandardLine bzw. ConnectionLine. In der Grafik unten links der Tabelle 10 wird eine LaneBorder nicht ganz so glatt wie die anderen modelliert. Damit soll verdeutlicht werden, dass prinzipiell auch detailliert eine Fahrstreifengrenze mit entsprechend hoher Punktdichte modelliert werden kann. Eine mathematische Abbildung wird dadurch jedoch komplizierter.

**Tabelle 10:** LaneBorders (cyan) und StandardLine-Enden (rot) als maximale longitudinale Ausdehnung einer LaneBorder entlang ihrer jeweilig zugeordneten StandardLine bzw. ConnectionLine.

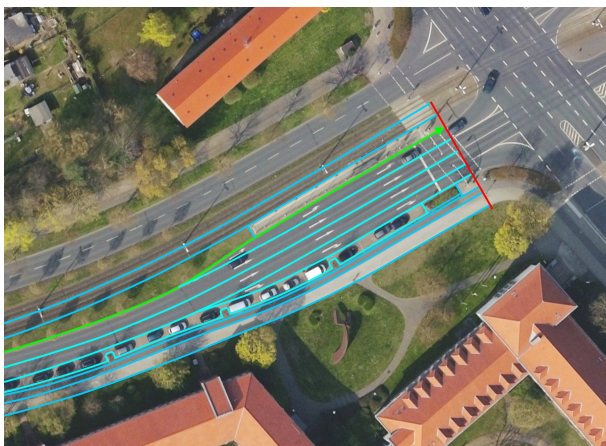


einbahnig, mehrere Streifen pro Richtung

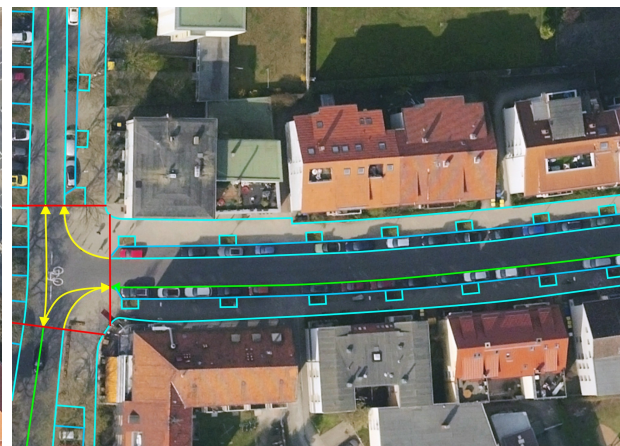


einbahnig, ein Streifen pro Richtung

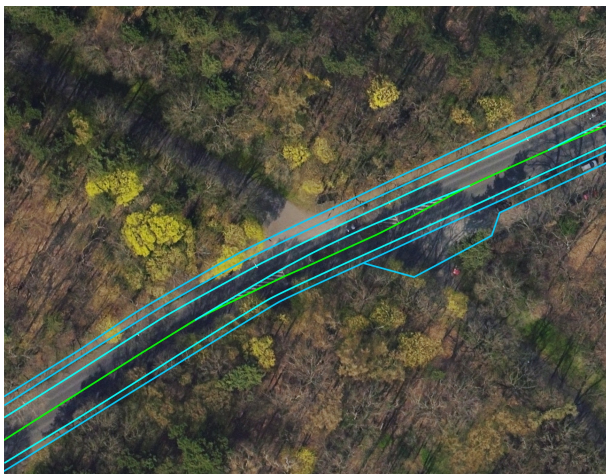




zweibahnig, mehrere Streifen pro Richtung



einbahnig, ein Streifen in eine Richtung



einbahnig, ein Streifen pro Richtung mit Trennung



mehrbahnig, unterschiedliche Streifen pro Richtung

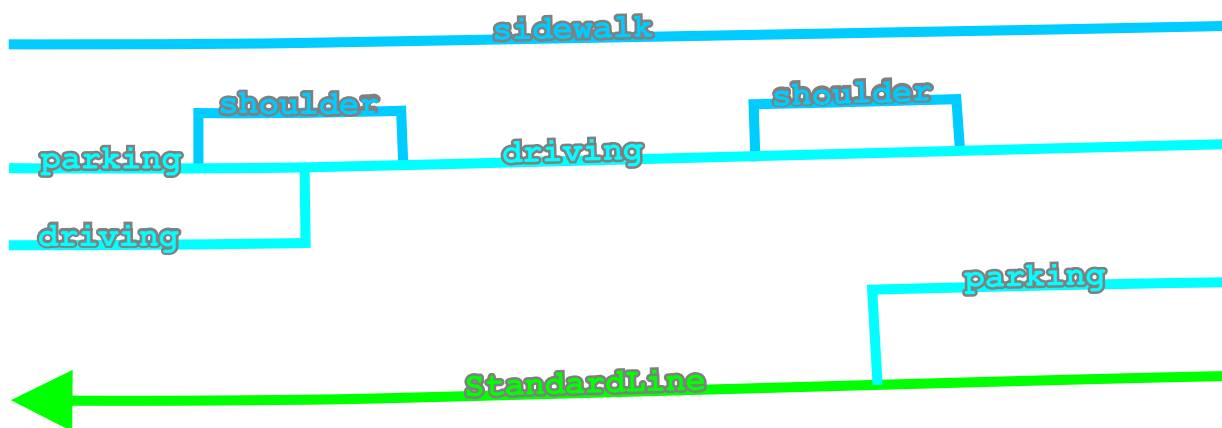
Die `LaneBorder` beschreibt den Nutzungstyp der Fahrbahnstreifen, die durch die `LaneBorder` begrenzt wird. Entsprechend wird der Typ in Richtung der zugehörigen `StandardLine` bzw. `ConnectionLine` angegeben. Mögliche Nutzungstypen sind:

- Fahrstreifen = `driving`
  - Parkstreifen = `parking`
  - Sperrfläche = `restricted`
  - undefiniert = `none`
- 
- Fußgängerweg = `sidewalk`
  - Radweg = `biking`
  - Grün- bzw. Mittelstreifen = `shoulder`
  - Straßenbahntrasse = `tram`

Bei der Modellierung der `LaneBorders` ist bei der Konstruktion zu beachten, dass keine `LaneBorder` vom `parent` aus betrachtet entlang einer imaginären, senkrechten Linie doppelt interpretiert wird. Abbildung 7 zeigt links die richtige Modellierung beispielhaft an einer Bauminself



**Abbildung 7:** Richtige (links) und falsche (rechts) Modellierungsreihenfolge von LaneBorders.



**Abbildung 8:** Modellierung von abrupten Fahrstreifenenden.

schen zwei Parktaschen. Abrupt endende Fahrstreifen (bspw. Parkstreifen auf der Fahrbahn ohne Parktaschen) sollten rechtwinklig abgeschlossen werden. Abbildung 8 zeigt beispielhafte Einbahnstraßensituation.

In einigen Situationen muss ggf. eine Hilfskonstruktion genutzt werden. Dafür gibt es den zusätzlichen Nutzungstyp `none` welcher bspw. genutzt wird, um einen Abschnitt einer Verkehrsinsel zu definieren. Die einzelnen Abschnitte können dann über die `TrafficIslands` näher definiert werden (siehe Abschnitt 3.7.1).

Generell werden `LaneBorders` zu beiden Seiten einer `StandardLine` oder `ConnectionLine` modelliert. In diesem Fall ist das `oneway`-Attribut dieser Linienzüge auf `false` zu setzen. Dies ist häufig bei `StandardLines` der Fall. Werden `LaneBorders` nur zur einer Seite einer `StandardLine` oder `ConnectionLine` modelliert, kann das `oneway`-Attribute dieser Linienzüge *auch* auf `false` gesetzt werden. Somit können die Fahrstreifen in beide Richtungen befahren werden. Dies ist häufig bei `ConnectionLines` der Fall.

Die `LaneBorder` beschreibt auch das mehrheitlich vorherrschende Material, aus dem der durch die `LaneBorder` begrenzte Streifen erstellt ist. Entsprechend wird auch hier das Material in Richtung der zugehörigen `StandardLine` bzw. `ConnectionLine` angegeben. Mögliche Materialien sind:

- Asphalt = asphalt
- Beton = concrete
- Kleinpflaster = pavement
- Kopfsteinpflaster = cobble
- Begrünung (Gras, Sträucher, usw.) = vegetation
- Schotter = gravel
- Erde = soil

Eine `LaneBorder` ist entsprechend durch die in Tabelle 11 aufgeführten Attribute beschrieben. Tabelle 12 zeigt ein Modellierungsbeispiel von `LaneBorders` mit Tabellenauszug.

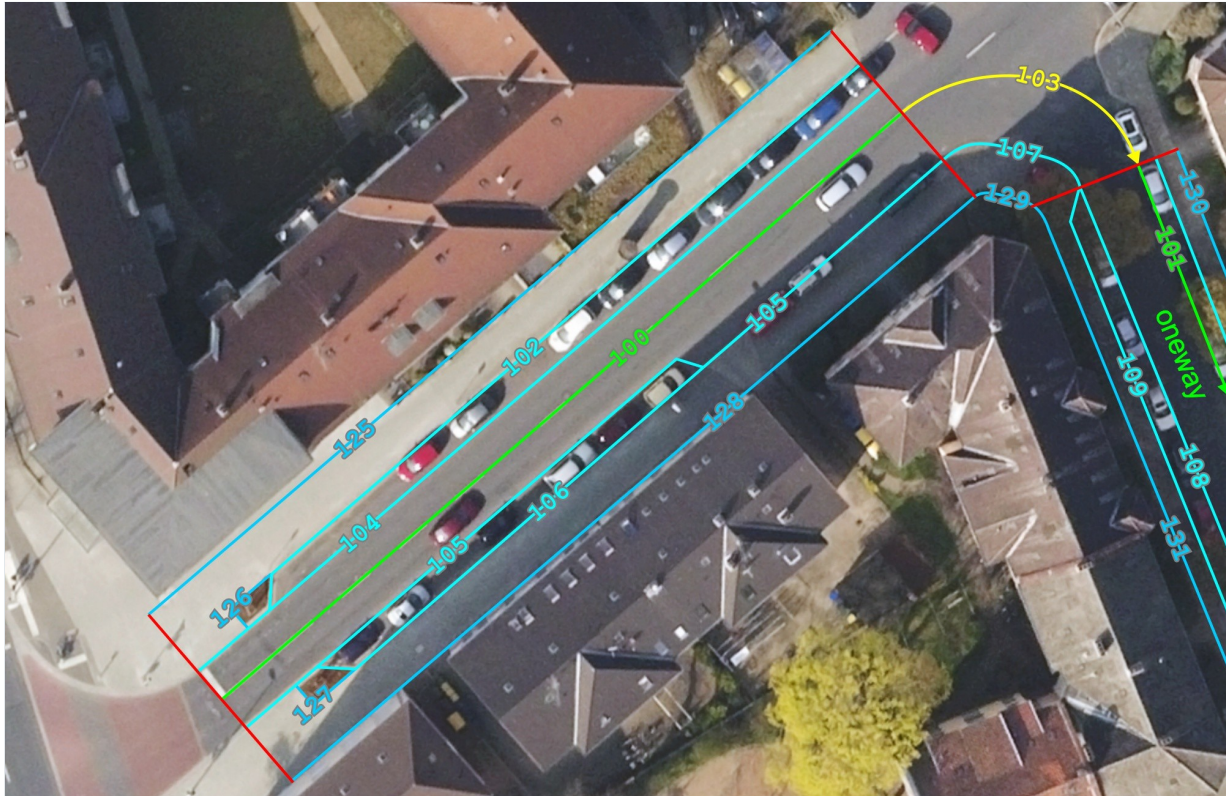
Im Kreuzungsbereich werden auch Fahrstreifenbegrenzungsinformationen benötigt. Dabei werden Fahrstreifenkanten modelliert, die bspw. durch Fahrbahnmarkierungen abgebildet werden. Sind aber bspw. auf kleinen Kreuzungen keine Markierungen vorhanden, muss keine `LaneBorder` modelliert werden. Zur Unterstützung der Datenkonvertierung können aber auch „imaginäre“ Fahrstreifenbegrenzungen modelliert werden, um einen natürlichen Verlauf zu Erzeugen. Diese `LaneBorders` basieren somit nicht zwangsweise auf baulichen Gegebenheiten. Tabelle 13 zeigt eine typische Kreuzungssituation. Ein Kreisverkehr wird analog zu Kreuzungen modelliert, somit wird jede Einfahrt zu einer kleinen Kreuzung, siehe Tabelle 14.

**Tabelle 11:** Eigenschaften von `LaneBorders`.

Name	Anzahl	Typ	Wert	Beschreibung
<b>id</b>	1	unit64_t	[0, max]	eindeutige Kennung
<b>dataSource-Description</b>	1	unit64_t	[0, max]	ID des beschreibenden <code>DataSourceDescription</code> -Objekts
<b>geometry</b>	1	LineString Z	siehe Beispiel in Abschnitt 2.3	Linienzug als WKT oder WKB
<b>material</b>	1	string	asphalt, concrete, pavement, cobble, vegetation, soil, gravel	Material des durch die <code>LaneBorder</code> eingegrenzten Streifens
<b>type</b>	1	string	driving, parking, restricted, none, sidewalk, biking, shoulder, tram	Nutzungstyp des durch die <code>LaneBorder</code> eingegrenzten Streifens
<b>parent</b>	1	unit64_t	[0, max]	ID einer <code>StandardLine</code> oder <code>ConnectionLine</code> zu der die <code>LaneBorder</code> gehört



**Tabelle 12:** Beispiel mehrerer LaneBorders (cyan) mit Tabellenauszug. Rote Begrenzungslinien verdeutlichen die maximal longitudinale Ausdehnung einer LaneBorder entlang ihrer jeweilig zugeordneten StandardLine bzw. ConnectionLine.

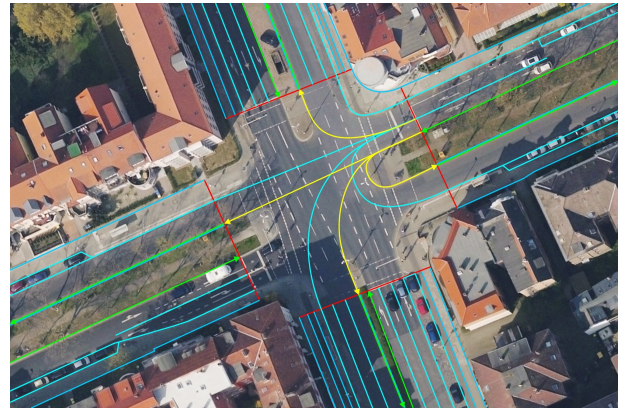


LaneBorder					
id	dSD	geometry	material	type	parent
102	0	WKT/WKB	asphalt	parking	100
104	0	WKT/WKB	asphalt	driving	100
105	0	WKT/WKB	asphalt	driving	100
106	0	WKT/WKB	asphalt	driving	100
107	0	WKT/WKB	asphalt	driving	103
108	0	WKT/WKB	asphalt	driving	101
109	0	WKT/WKB	pavement	parking	101
110	0	WKT/WKB	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

**Tabelle 13:** LaneBorders (cyan) im Kreuzungsbereich. Rote Begrenzungslinien verdeutlichen die maximal longitudinale Ausdehnung einer LaneBorder entlang ihrer jeweilig zugeordneten StandardLine bzw. ConnectionLine. Die Fahrradstreifen auf der Fahrbahn (im Beispiel aus westlicher und östlicher Richtung) sind hier nicht mit extra LaneBorders modelliert.



LaneBorders von Norden ausgehend



LaneBorders von Osten ausgehend



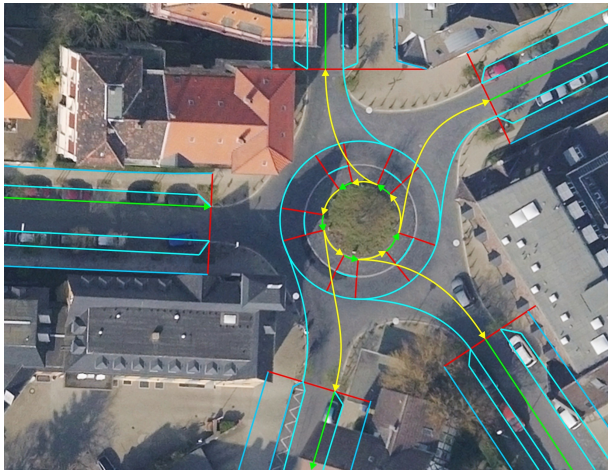
LaneBorders von Süden ausgehend



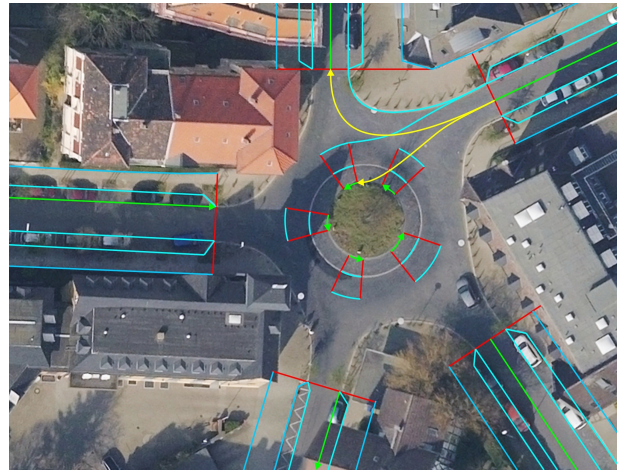
LaneBorders von Westen ausgehend



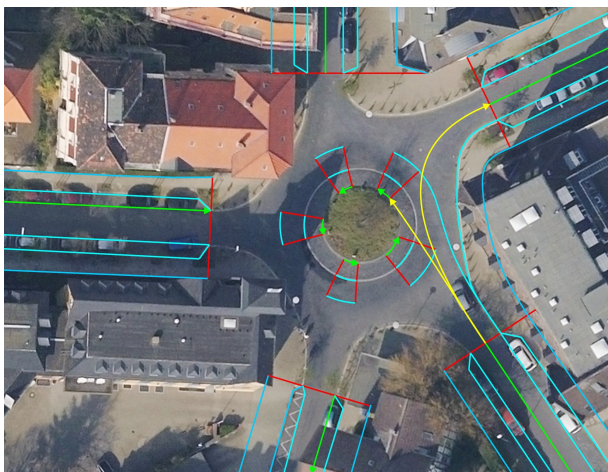
**Tabelle 14:** LaneBorders (cyan) im unterteilten Kreisverkehr. Rote Begrenzungslinien verdeutlichen die maximal longitudinale Ausdehnung einer LaneBorder entlang ihrer jeweilig zugeordneten StandardLine bzw. ConnectionLine.



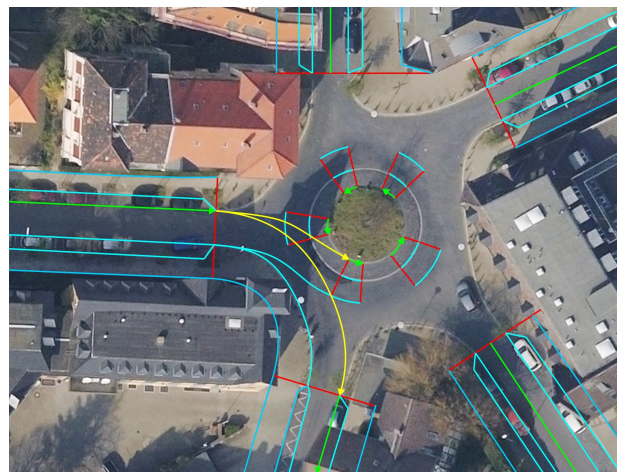
LaneBorders vom Zentrum ausgehend



LaneBorders von Nordosten ausgehend



LaneBorders von Südosten ausgehend



LaneBorders von Westen ausgehend

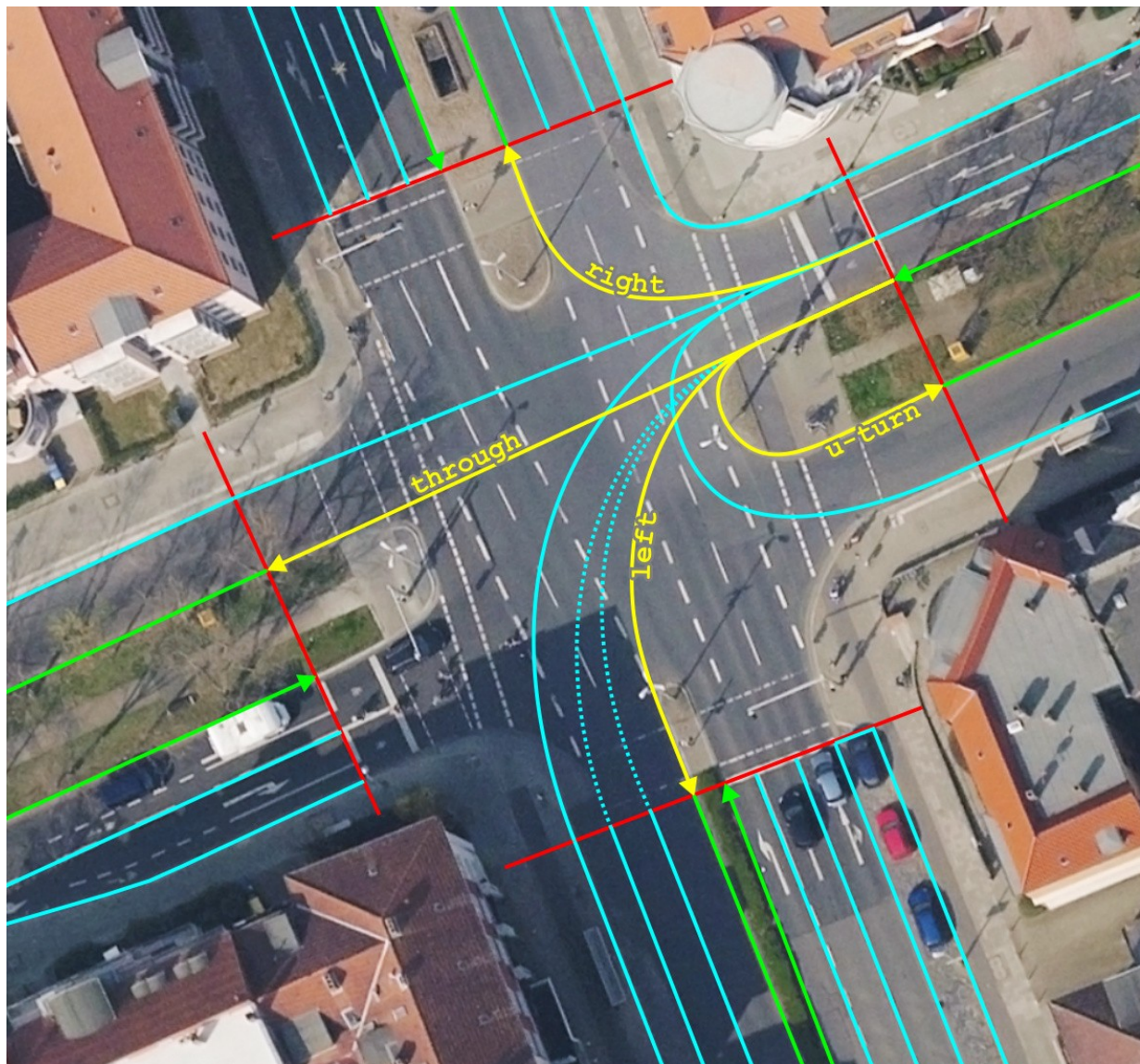


LaneBorders von Norden ausgehend



LaneBorder von Südwesten ausgehend

Die letzten beiden Grafiken in Tabelle 14 für die Richtungen Nord und Südwest stellen wie in Abschnitt 3.3.1 beschrieben einen Sonderfall dar. Damit in der Einbahnstraßensituation die Randstreifen (Bürgersteig, Fahrradweg, o.ä.) logisch zugeordnet werden können, kommen nun die `Behelfs-ConnectionLines` zum Tragen.



**Abbildung 9:** Notwendige (durchgezogen) und obligatorische (gepunktet) `LaneBorders`.

Abbildung 9 soll verdeutlichen, wann das Modellieren von `LaneBorders` sinnvoll und wann notwendig ist:

- `ConnectionLine right` benötigt mindestens eine `LaneBorder`, da sonst die Begrenzung der Fahrbahn nicht modelliert werden kann.
- `ConnectionLine through` benötigt prinzipiell keine `LaneBorder`, da hier die Fahrstreifengrenze zwischen Vorgänger und Nachfolger interpoliert werden kann.



- `ConnectionLine left` benötigt prinzipiell ebenfalls keine `LaneBorder`. Hier bietet sich die Modellierung aber an, insbesondere beim Hinzufügen der weiteren Fahrstreifen, da diese erst ab einer bestimmten Position beginnen sollten.
- `ConnectionLine u-turn` benötigt prinzipiell ebenfalls keine `LaneBorder`, da auch hier die Fahrstreifenbreite aus Vorgänger und Nachfolger interpoliert werden kann. Allerdings bietet sich auch hier eine explizite Modellierung an, damit bei linearer Breiteninterpolation die Breitenzunahme nicht in andere Fahrstreifen hineinragt.

### 3.5. Markierungen

Weitere wesentlich prägende Elemente sind die Fahrbahnmarkierungen, die `Mark` genannt werden. Diese sind oft mit den Begrenzungen der Fahrstreifen (`LaneBorder`) gleichzusetzen, bzw. decken nur teilweise Abschnitte ab und sind in Längsrichtung zu `LaneBorders` angeordnet (bspw. Fahrradschutzstreifen). Manchmal sind Fahrbahnmarkierungen auch nur innerhalb einer Kreuzungsfläche vorhanden. Einige Markierungen liegen aber auch quer zu den Fahrstreifen (Haltelinien, Fußgängerfurten). Die Modellierung von `Marks` orientiert sich an folgenden Punkten:

- Markierungen werden ebenfalls in Form eines Linienzugs als `Mark` mit jeweils einer Typangabe gespeichert. Ändert sich der Typ der Markierung, wird eine neue `Mark` erzeugt. Aufeinanderfolgende Markierungen müssen im Rahmen des Möglichen  $C^1$ -stetig sein, wie in Abschnitt 2.4 erläutert.
- `Marks` sind Linienzüge, die geometrisch auf anderen, bereits existierenden linienartigen Geometrien (`StandardLine`, `ConnectionLine`, `LaneBorder`) liegen können. Sie können aber auch leicht versetzt aufgebracht sein (siehe Beispiel in Tabelle 17).
- Ein `Mark`-Objekt ist in mehrere Objekte zu zerlegen, falls die Markierung längsseits/parallel zu einem `parent`-Objekt (`StandardLine`, `ConnectionLine`) verläuft und dieses an einem seiner Enden überragt. Ansonsten ist eine eindeutige Zuordnung der `Mark` nicht möglich.
- Eine Doppellinie wird als *eine* `Mark` mit entsprechendem Typ modelliert (siehe Abbildung 10). Sie wird entsprechend in der Mitte der beiden Linien modelliert.
- `Marks` beginnen und enden *immer* auf der tatsächlichen Markierung, um bspw. Strich-Lücke-Verhältnis möglichst genau abbilden zu können.

Tabelle 15 zeigt Modellierungsbeispiele mit Fahrbahnmarkierungen (magenta), sowie Abschnittstrenner bei Typwechsel und Strich-Lücke-Längenangabe.



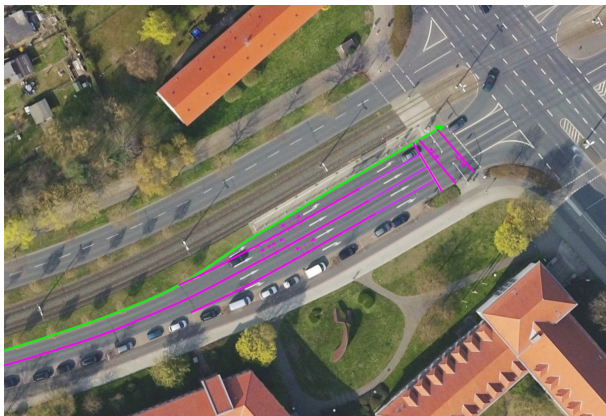
**Tabelle 15:** Fahrbahnmarkierungen (magenta) und Lücke-Strich-Angaben.



einbahnig, mehrere Streifen pro Richtung



einbahnig, ein Streifen pro Richtung



zweibahnig, mehrere Streifen pro Richtung



einbahnig, ein Streifen in eine Richtung



einbahnig, ein Streifen pro Richtung mit Trennung



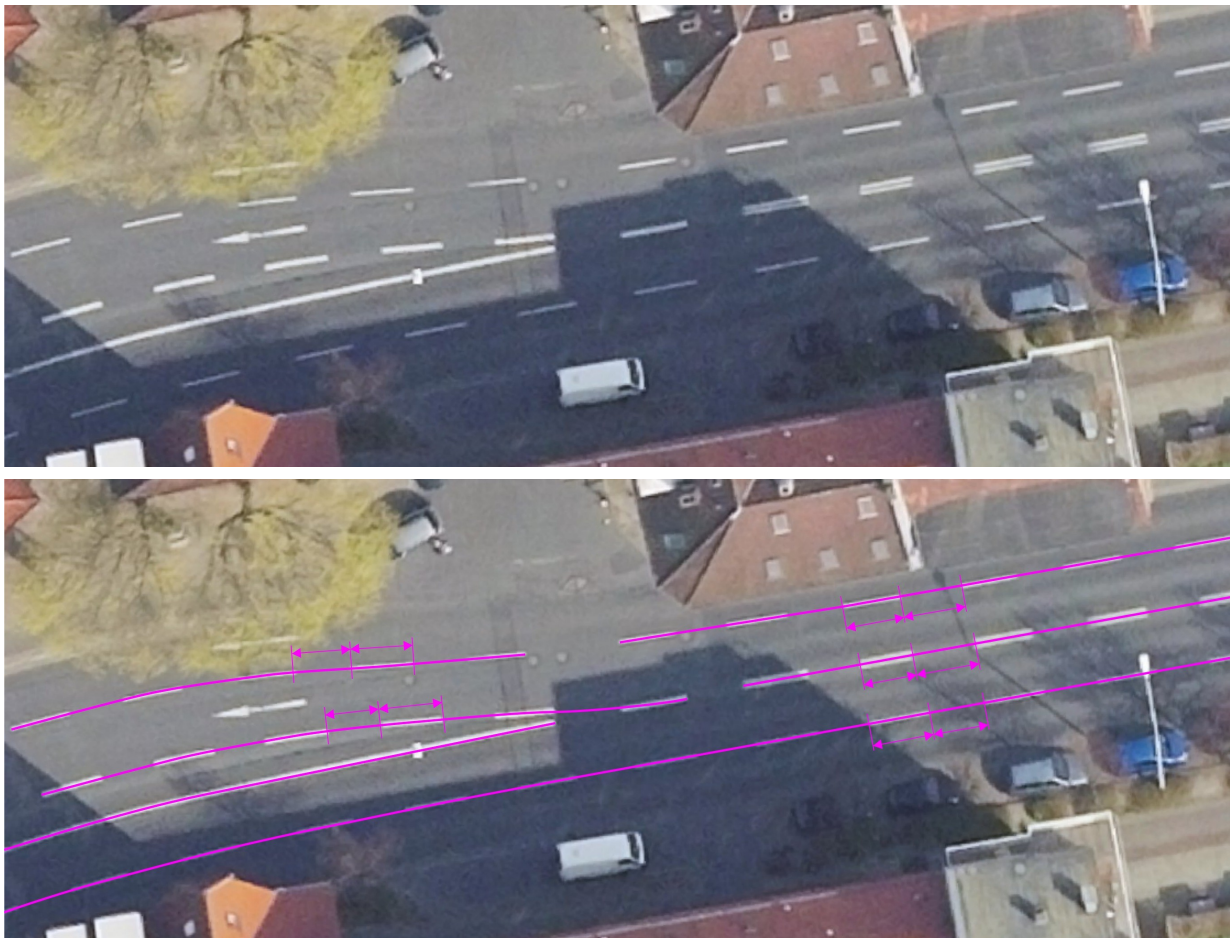
mehrbahnig, unterschiedliche Streifen pro Richtung

Als Fahrbahnmarkierung `Mark` werden zum einen Fahrstreifengrenzen und Fahrstreifenränder modelliert (Längsmarkierungen), zum anderen aber auch Halte- und Wartelinien (Quermarkierungen). Längsseits der Fahrstreifen verlaufende `Marks` können folgende Typen aufweisen:



- durchgezogen = `solid`
- unterbrochen = `broken`
- doppelt durchgezogen = `solid solid`
- doppelt unterbrochen = `broken broken`
- durchgezogen und unterbrochen = `solid broken`
- unterbrochen und durchgezogen = `broken solid`
- metallene Markierungen = `botts dots`
- sonstige Kante = `curb`

Laufen doppelte Markierungen auseinander (oder einzelne zusammen), sind jeweils unterschiedliche Marks zu modellieren. Abbildung 10 zeigt ein entsprechendes Beispiel.

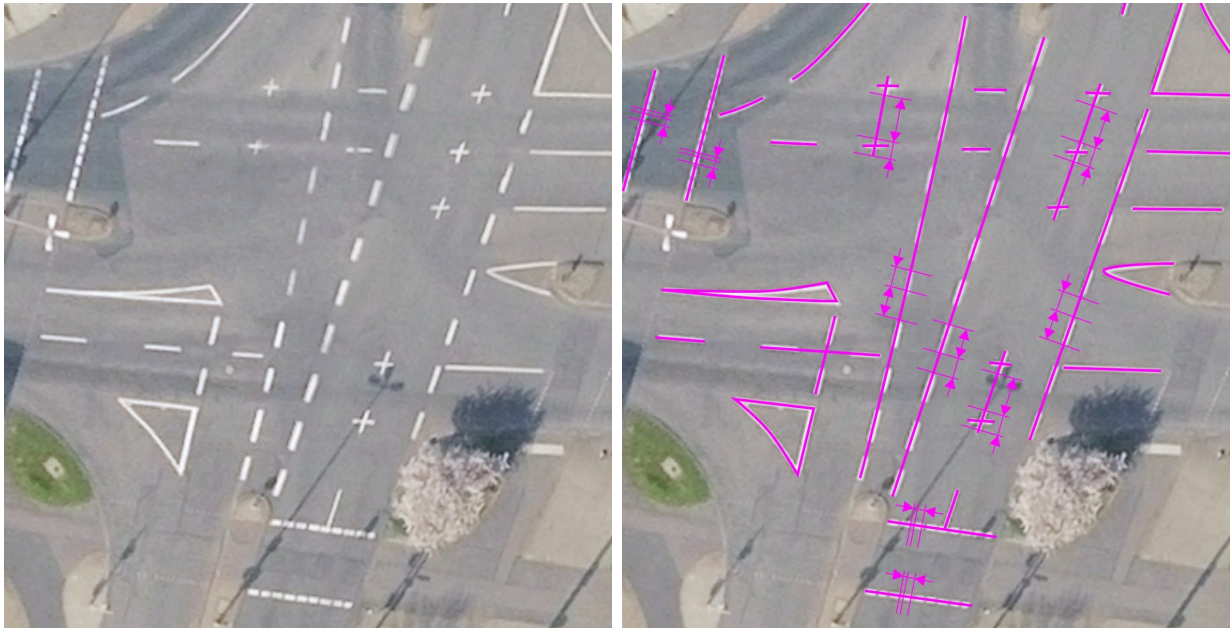


**Abbildung 10:** Unterschiedliche Markierung eines Straßenabschnitts.

Quermarkierungen und sonstiges Marks können folgende Typen aufweisen:

- Fußgängerfurt, auch Radwegefurt = `crossing`
- Haltelinie = `stop line`
- Wartelinie = `waiting line`
- Halte-/Parkverbot (Zick-Zack-Linie) = `stopping restriction`
- Zebrastreifen = `zebra crossing`

Bei unterbrochenen Marks sind die Längen der Striche bzw. der Lücken von Interesse. Diese werden als (gemittelte) Längenangabe modelliert. Kurze Marks vom Typ `broken`, die nur einen einzelnen Markierungstrich modellieren, sind mit einem Strich-Lücke-Verhältnis von `markLength =` Gesamtlänge der Mark und `spaceLength = 0` zu modellieren. Auf diese Weise könnte man auch eine ganze Abfolge von Strichen einer unterbrochenen Markierungsline einzeln modellieren. Abbildung 11 zeigt unterschiedliche Markierungen im Kreuzungsbereich.



**Abbildung 11:** Unterschiedliche Markierungen auf einer Kreuzungsfläche.

Das Gewicht einer Fahrbahnmarkierung gibt ihre Dicke an. Dabei werden nur zwei Typen unterschieden:

- Schmalstrich = `standard`
- Breitstrich = `bold`

In Deutschland entspricht das den Breiten Autobahn: 0,15 m und sonstige Straßen: 0,12 m für Schmalstrich und Autobahn: 0,30 m, sonstige Straßen: 0,25 m für Breitstrich.

Die Farbe der Fahrbahnmarkierung wird mit einem einfachen Bezeichner angegeben. Dabei werden folgende Möglichkeiten unterschieden:

- weiß = `standard`
- gelb = `yellow`
- orange = `orange`
- rot = `red`
- grün = `green`
- blau = `blue`

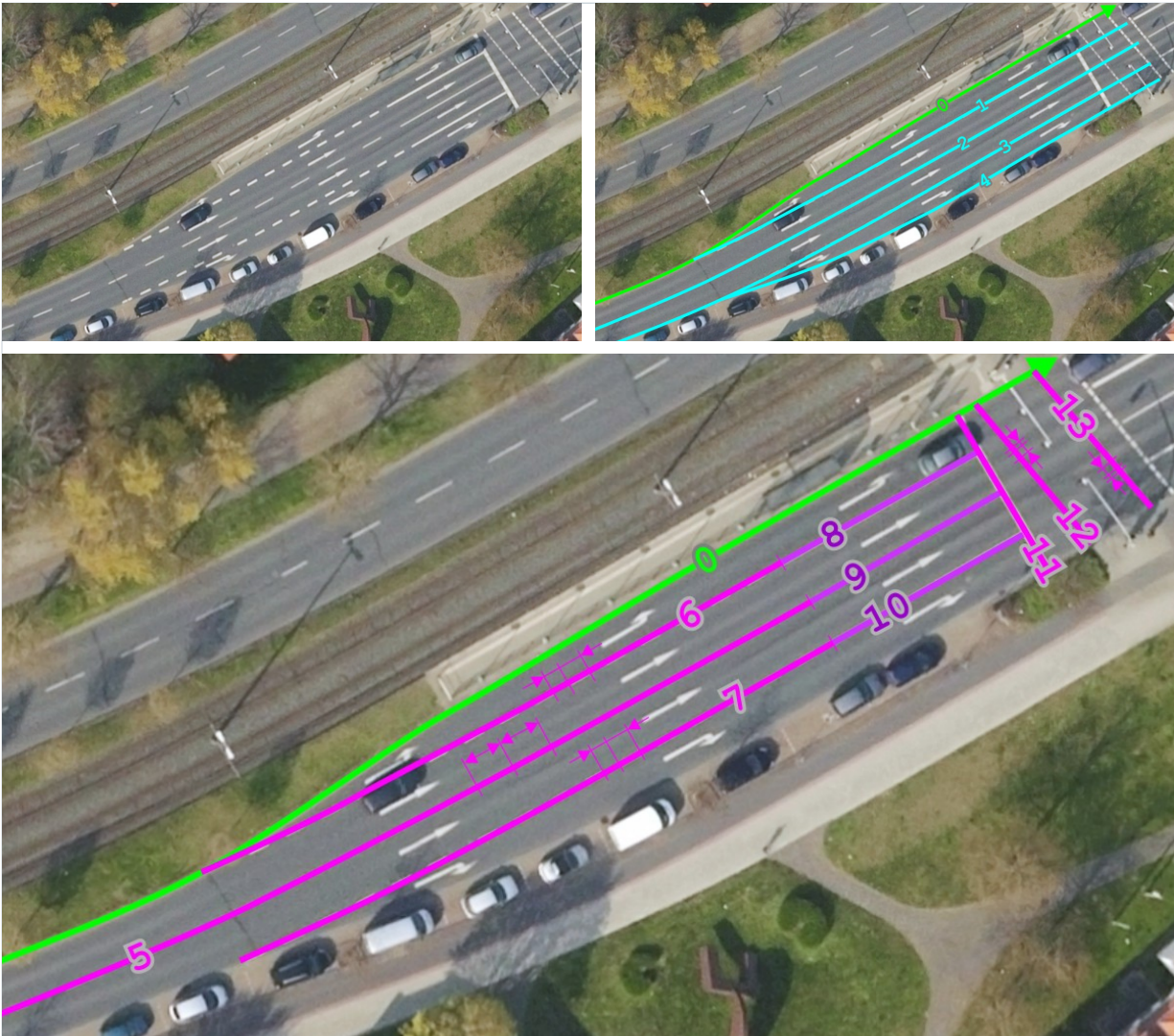
Markierungen erhalten dabei die in Tabelle 16 beschriebenen Attribute. Die Tabellen 17 und 18 zeigen Beispiele zur Digitalisierung von Markierungen im Datenmodell.

**Tabelle 16:** Eigenschaften von Marks.

Name	Anzahl	Typ	Wert	Beschreibung
<b>id</b>	1	unit64_t	[0, max]	eindeutige Kennung
<b>dataSource-Description</b>	1	unit64_t	[0, max]	ID des beschreibenden DataSourceDescription-Objekts
<b>geometry</b>	1	LineString Z	siehe Beispiel in Abschnitt 2.3	Linienzug als WKT oder WKB
<b>color</b>	1	string	standard, yellow, orange, red, green, blue	Farbe der Fahrbahnmarkierung (in der Regel weiß = standard)
<b>type</b>	1	string	solid, broken, solid solid, broken broken, solid broken, broken solid, botts dots, curb, zebra crossing, crossing, stop line, waiting line, stopping restriction	Typ der Fahrbahnmarkierung
<b>weight</b>	1	string	standard, bold	Dicke/Gewicht der Fahrbahnmarkierung
<b>markLength</b>	0..1	double	[0.0, max]	Länge des Striches [m] bei einer unterbrochenen Fahrbahnmarkierung
<b>spaceLength</b>	0..1	double	[0.0, max]	Länge der Lücke [m] bei einer unterbrochenen Fahrbahnmarkierung
<b>parent</b>	1	unit64_t	[0, max]	ID der logisch zugehörigen StandardLine bzw. Connection-Line



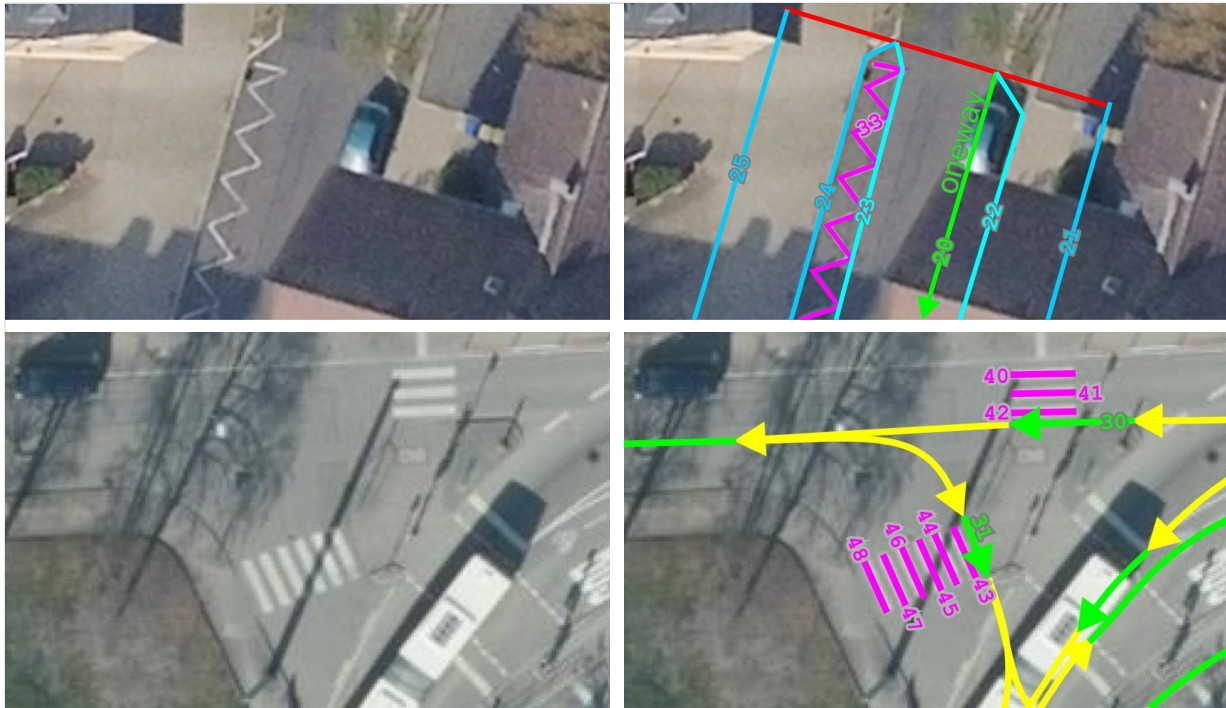
**Tabelle 17:** Beispiel mehrerer parallel und quer verlaufender Markierungen/Marks (magenta) mit Tabellenauszug.



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



**Tabelle 18:** Beispiel für Sperrzonen (oben) und Zebrastreifen (unten) als Markierungen/Marks (magenta) mit Tabellenauszug.



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. Objekte

Objekte wie Straßenschilder, Schilderbrücken und Lichtsignalanlagen sind wichtige Informationen für die Straßenbeschreibung. Aber auch Masten für Straßenbeleuchtung, Leitplanken, Gitter und Schallschutzwände sind wesentliche Elemente, die das Erscheinungsbild der Straße prägen.

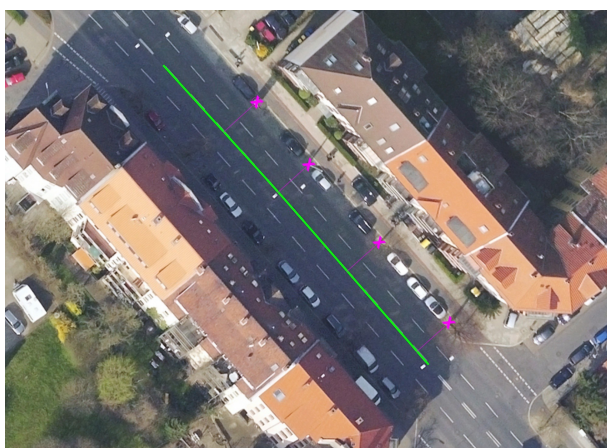
Straßenschilder werden einmal als Schild mit entsprechendem Typ angegeben und an derselben Geokoordinate auch als Schild- oder Ampelmast modelliert.

Lichtsignalanlagen (Ampeln) werden in Mast, Ausleger und Signalgeber untergliedert. Jedes Objekt bekommt entsprechend einen Eintrag. Gleiches gilt für Schilderbrücken, deren Querträger bei zwei Masten nur einem von beiden zugeordnet wird, aber über die gesamte Länge geht.

Punktuelle Objekte werden durch eine Standortkoordinate, den absoluten Ausrichtungswinkel sowie weiteren Attributdaten wie der Zugehörigkeit zu einer `StandardLine`, `ConnectionLine` oder zu einem anderen Objekt definiert. Die Ausrichtung wird *geografisch absolut* angegeben, es wird also ein Linkshandsystem genutzt mit positiver Drehung im Uhrzeigersinn. Entsprechend betragen die Winkel  $N = 0^\circ$ ,  $O = 90^\circ$ ,  $S = 180^\circ$ ,  $W = 270^\circ$ .

Lineare Objekte werden wie Markierungen durch Linienzüge mit Typangabe modelliert und ebenfalls zu einer `StandardLine`, `ConnectionLine` zugeordnet. Tabelle 19 zeigt beispielhaft lineare und punktuelle Objekte.

**Tabelle 19:** Lineare und punktuelle Straßenobjekte (magenta).



einbahnig, mehrere Streifen pro Richtung



einbahnig, ein Streifen pro Richtung





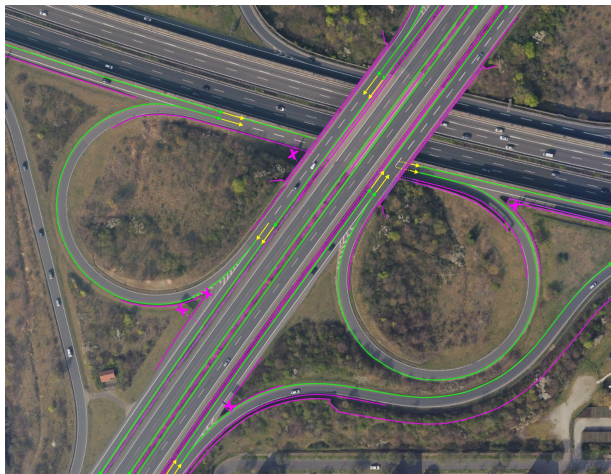
zweibahnig, mehrere Streifen pro Richtung



einbahnig, ein Streifen in eine Richtung



einbahnig, ein Streifen pro Richtung mit Trennung



mehrbahnig, unterschiedliche Streifen pro Richtung

### 3.6.1. Punktuelle Objekte

Relevante punktuelle Objekte und deren Ausrichtungsseite sind in Tabelle 20 aufgelistet. Signalgebertypen werden in Tabelle 21 unterschieden. Kann ein Signalgeber aus den Daten nicht definiert werden, wird der Default-Wert 1.000.000 angegeben.

Auch Piktogramme und Sonderzeichen, die direkt auf die Fahrbahn aufgebracht werden, werden als punktuelle Objekte modelliert. Dabei werden die in Tabelle 22 dargestellten Typen unterschieden.

**Tabelle 20:** Typen punktförmiger Objekte.

Objekttyp	Bezeichner	Front für Ausrichtung	Punkt für Positionierung
Ampelmast	traffic light post	-/-	Mastmitte auf Bodenfläche
Ampelmastausleger	traffic light boom	entlang des Auslegers	Montagepunkt am Mast
Peitschenmast	curved traffic light post	entlang der Peitsche	Mastmitte auf Bodenfläche
























Objektyp	Bezeichner	Front für Ausrichtung	Punkt für Positionierung
Signalgeber	siehe Tabelle 21 der Signalgebertypen	Frontansicht	Mitte der Frontseite
Straßenschild	entsprechende Nummer der Bildtafel der Verkehrszeichen <sup>14</sup>	Frontansicht	Mitte der Frontseite
Wechselverkehrszeichen	entsprechend der ISO 14823 <sup>15</sup>	Frontansicht	Mitte der Frontseite
Schildmast (auch genereller Mast)	pole	-/-	Mastmitte auf Bodenfläche
Schilderbrückenmast	overhead gantry mast	-/-	Mastmitte auf Bodenfläche
Schilderbrücke	overhead gantry	Entlang der Brücke	Montagepunkte am Mast
Leitpfosten	reflector post	-/-	Pfostenmitte auf Bodenfläche
Stadtbeleuchtungsmast	street lamp	Richtung Ausleger oder quer bei zwei Auslegern	Mastmitte auf Bodenfläche
Stadtbeleuchtungspeitschenmast	curved street lamp	entlang der Peitsche	Mastmitte auf Bodenfläche
Briefkasten	postbox	Briefkastenfront	Bodenflächenmitte
Brunnen	well	erkennbare Front oder -/-	Bodenflächenmitte
Bus- und Tramwarte- häuschen	shelter	längere offene Seite	Bodenflächenmitte
Denkmal und Skulptur	monument	erkennbare Front oder -/-	Bodenflächenmitte
Fahnenmast und Fahne	flagpole	-/-	Mastmitte auf Bodenfläche
Hydrant	hydrant	ein Abgang zur Front oder zwei Abgänge zur Seite	Hydrantmitte auf Bodenfläche
Litfaßsäule	advertising column	-/-	Bodenflächenmitte
Mülleimer	waste container	längere Seite oder -/-	Bodenflächenmitte
Notrufsäule	emergency phone	Telefonfront	Bodenflächenmitte
Piktogramm	siehe Tabelle 22 der Bodenmarkierungstypen	Front in Blickrichtung	Piktogrammmitte
Poller	bollard	-/-	Pollermitte auf Bodenfläche
Sitzbank	bench	Sitzflächenseite	Bodenflächenmitte
Telefonzelle oder -stele	phone box, phone stele	Tür- bzw. Stelenfront	Bodenflächenmitte
Verteilerkasten	distribution box	Frontansicht (längere Seite)	Bodenflächenmitte
Werbetafel	advertising panel	Frontansicht (längere Seite)	Mitte der Frontseite

<sup>14</sup>siehe bspw. <http://www.vzkat.de/2017/VzKat.htm> oder [https://de.wikipedia.org/wiki/Bildtafel\\_der\\_Verkehrszeichen\\_in\\_der\\_Bundesrepublik\\_Deutschland\\_seit\\_2013](https://de.wikipedia.org/wiki/Bildtafel_der_Verkehrszeichen_in_der_Bundesrepublik_Deutschland_seit_2013)

<sup>15</sup>Intelligent transport systems – Graphic data dictionary



**Tabelle 21:** Signalgebertypen.

Signal	Bezeichner	Signal	Bezeichner	Signal	Bezeichner
	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	Bezeichner	Signal	Bezeichner	Signal	Bezeichner
	1.000.015		1.000.010.20		1.000.013.10
	F 0		W 0, auch 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		

Tabelle 22: Bodenmarkierungstypen.

Symbol	Bezeichner	Symbol	Bezeichner	Symbol	Bezeichner
<b>TAXI</b> <b>BUS</b>	taxi horizontal, bus horizontal	<b>30,</b> <b>50,</b> <b>...</b>	horizontal 30, horizontal 50, ...		pedestrian
<b>T</b> <b>A</b> <b>X</b> <b>I</b>	taxi vertical, bus vertical		stopping restriction		handicap
	297.1		297.2		cyclist
	297 through		297 through left		297 through right




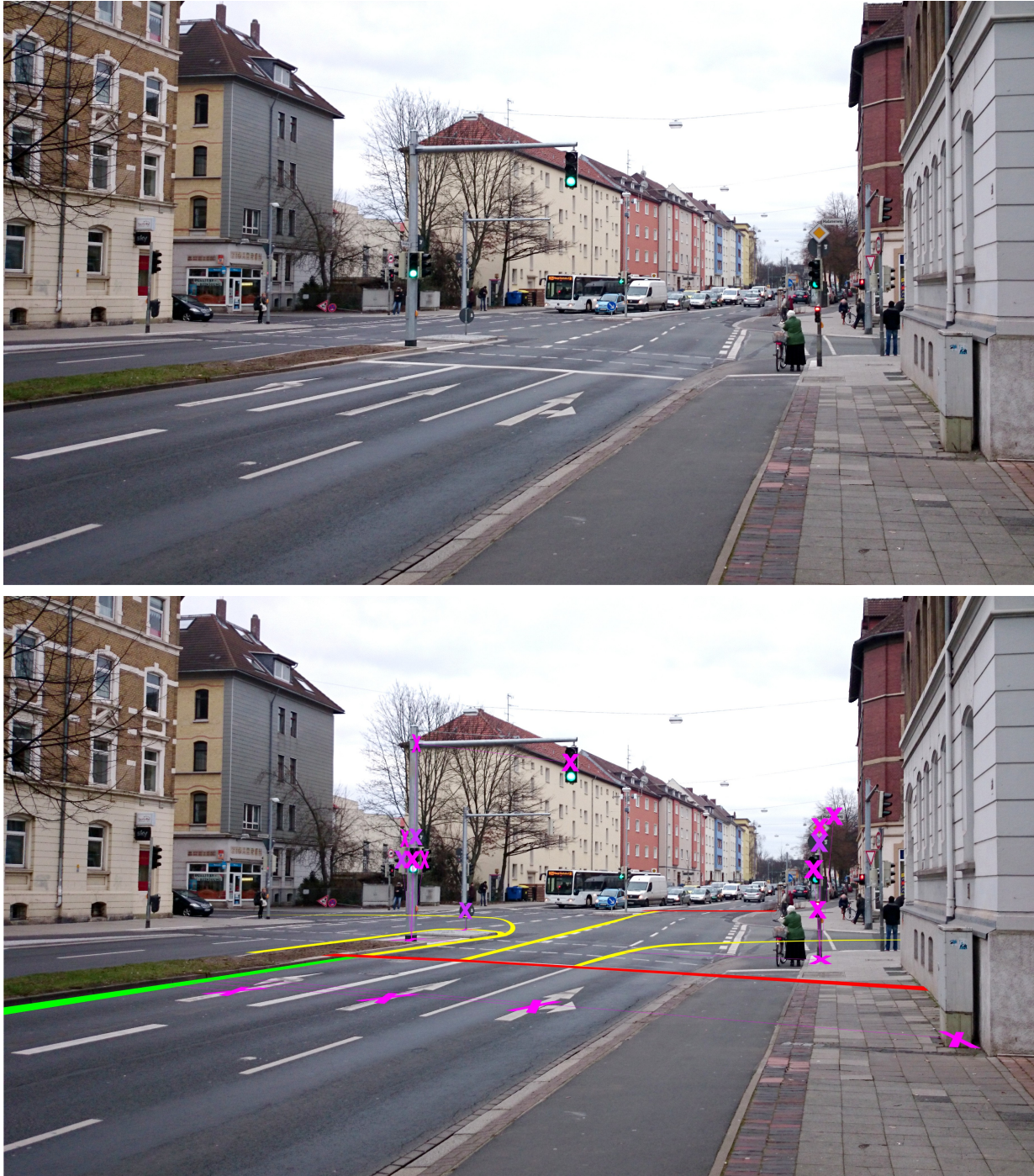
Symbol	Bezeichner	Symbol	Bezeichner	Symbol	Bezeichner
	297 left right		297 left		297 right

Abbildung 12 zeigt eine beispielhafte Situation. Die eingehende `StandardLine` sowie die nachfolgenden `ConnectionLines` und deren Trennung sind modelliert. Alle in Fahrtrichtung befindlichen punktförmigen Objekte sind markiert und mit einer Zuordnung zur `StandardLine`, `ConnectionLine` oder zu einem weiteren Objekt markiert.





**Abbildung 12:** Kreuzung, mehrere Fahrstreifen mit baulicher Trennung sowie auf StandardLine (grün) und ConnectionLine (gelb) verortete punktuelle Objekte (magenta).

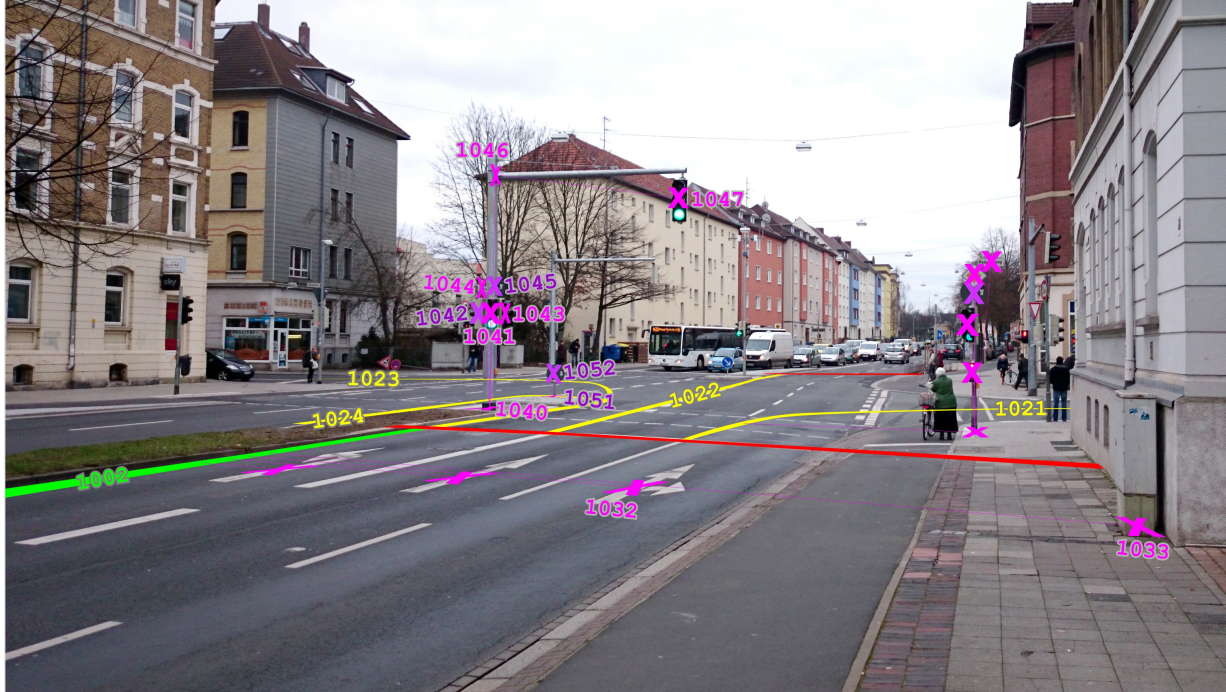
Ein punktuelltes Objekt trägt Information über seine Größe/Ausdehnung. Es erhält die in Tabelle 23 gelisteten Attribute. Tabelle 24 zeigt ein vollständiges Modellierungsbeispiel von `PointObjects` dieser Kreuzungssituation.

**Tabelle 23:** Eigenschaften von `PointObjects`.

Name	Anzahl	Typ	Wert	Beschreibung
<b>id</b>	1	unit64_t	[0, max]	eindeutige Kennung
<b>dataSource-Description</b>	1	unit64_t	[0, max]	ID des beschreibenden <code>DataSourceDescription</code> -Objekts
<b>geometry</b>	1	Point Z	siehe Beispiel in Abschnitt 2.3	Punkt als WKT oder WKB
<b>type</b>	1	string	siehe Tabelle 20, Tabelle 21 und Tabelle 22 sowie Bildtafelnummer	Typ des Objekts
<b>content</b>	0..1	string		Text auf dem Straßennamenschild oder einfachen Wegweiser
<b>heading</b>	0..1	double	[0.0, 360.0[	Ausrichtung des Objekts als Winkel in Grad
<b>width</b>	0..1	double	[0.0, max]	Breite des Objekts bei Frontansicht in Meter
<b>depth</b>	0..1	double	[0.0, max]	Tiefe des Objekts bei Frontansicht in Meter
<b>height</b>	0..1	double	[0.0, max]	Höhe Objekts bei Frontansicht in Meter
<b>pointParent</b>	0..1	unit64_t	[0, max]	ID des zugehörigen punktuellen Objekts
<b>linearParent</b>	0..1	unit64_t	[0, max]	ID der zugehörigen <code>StandardLine</code> bzw. <code>ConnectionLine</code>



**Tabelle 24:** Beispielhafte Kreuzung mit verorteten punktuellen Objekten (magenta) und WKT-Geometrien im Tabellenauszug. Straßenverlauf führt Richtung Süden  $S = 180^\circ$ . Die DataSource-Description (dSD) aller Objekte in diesem Beispiel ist 1.



PointObject									
id	geometry	type	content	heading	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	content	heading	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. Lineare Objekte

Lineare Objekte benötigen keine Ausrichtung und werden daher wie die restlichen Linientypen behandelt. Die Linienzüge werden auf der Oberkante der linearen Objekte modelliert, somit wird implizit auch die Höhe modelliert. Lineare Objekte werden bspw. auch *auf gleicher Höhe* der StandardLine-Enden getrennt (maximale longitudinale Ausdehnung) und jeweils einer StandardLine oder ConnectionLine zugeordnet (vgl. Modellierung von Fahrstreifen in Abschnitt 3.4). Ansonsten ist eine eindeutige Zuordnung der linearen Objekte nicht möglich.

Streckenabschnitte mit Brücken oder Tunneln werden als lineare Objekte modelliert, deren Geometrie die jeweilige parent-Geometrie (StandardLine oder ConnectionLine) für den Abschnitt der Gültigkeit überlagert. Relevante lineare Objekte sind:

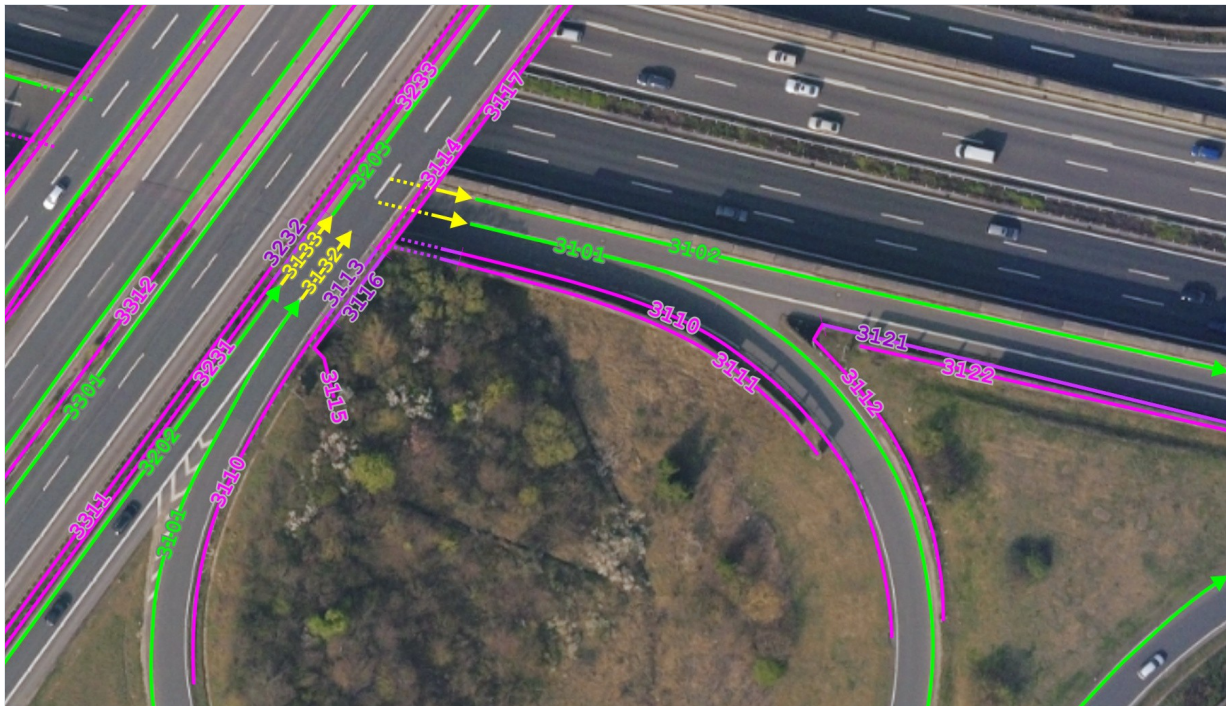
- Schutzplanken = guardrail
- Brücke = bridge
- Tunnel = tunnel

- Schutzplanken mit Blendschutz = `guardrail anti-glare`
- Schutzplanke mit Verstärkung = `superrail`
- Betonleitwand = `jersey barrier`
- Kabelbarriere = `cable barrier`
- Lärmschutzwände = `noise barrier`
- Zäune = `fence`
- Absperrgitter = `barrier`

Ein lineares Objekt enthält dabei die in Tabelle 25 aufgeführten Attribute. Die Tabellen 26 und 27 zeigen Modellierungsbeispiele mit Tabellenauszügen.

**Tabelle 25:** Eigenschaften von `LinearObjects`.

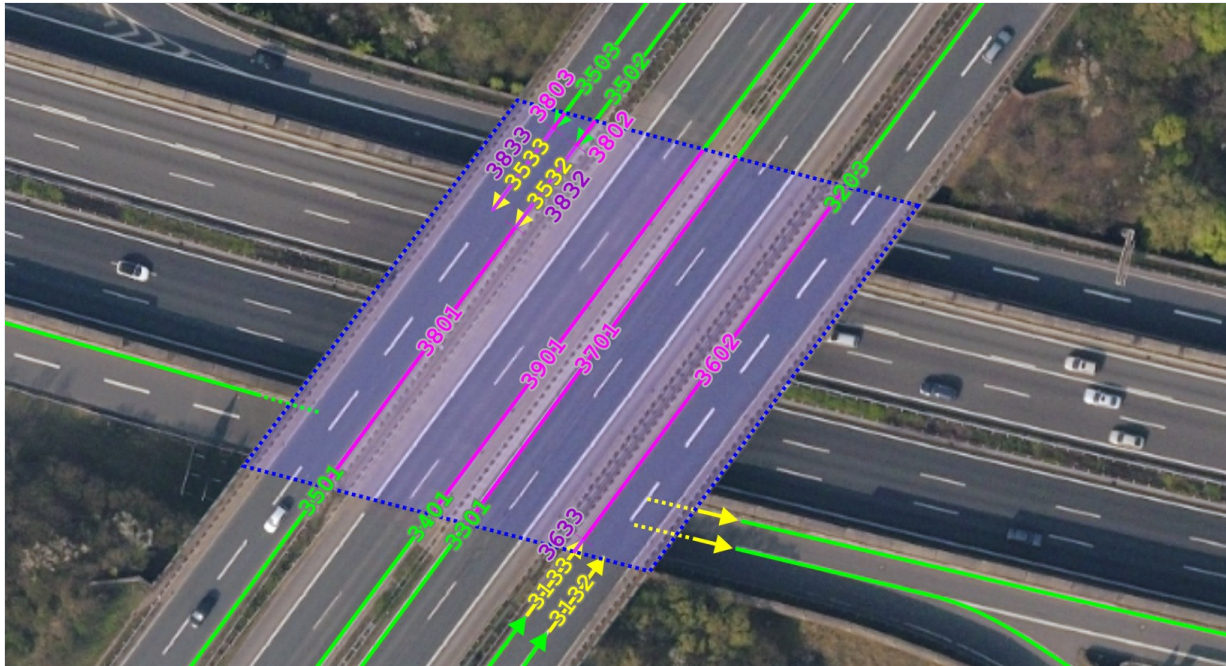
Name	Anzahl	Typ	Wert	Beschreibung
<b>id</b>	1	unit64_t	[0, max]	eindeutige Kennung
<b>dataSource-Description</b>	1	unit64_t	[0, max]	ID des beschreibenden <code>DataSourceDescription</code> -Objekts
<b>geometry</b>	1	LineString Z	siehe Beispiel in Abschnitt 2.3	Linienzug als WKT oder WKB
<b>type</b>	1	string	<code>guardrail</code> , <code>bridge</code> , <code>tunnel</code> , <code>guardrail anti-glare</code> , <code>superrail</code> , <code>jersey barrier</code> , <code>cable barrier</code> , <code>noise barrier</code> , <code>fence</code> , <code>barrier</code>	Typ des Objekts
<b>parent</b>	1	unit64_t	[0, max]	ID der zugehörigen <code>StandardLine</code> oder <code>ConnectionLine</code>

**Tabelle 26:** Beispiel für lineare Objekte (ohne Brücken-Objekt) mit Tabellenauszug.

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



**Tabelle 27:** Beispiel einer Brücke (blau) als `LinearObject` (ohne weitere lineare Objekte) mit Tabellenauszug.



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. Bauliche Trennungen

Im Fall von baulichen Trennungen zwischen Fahrstreifen wird bei der Modellierung unterschieden, wie sich der Nutzungstyp der entstehenden Zwischenräume relativ zum Verlauf eines Straßenverlaufs (`StandardLine` oder `ConnectionLine`) verhält. Ändert sich der Nutzungstyp *quer* zum Verlauf eines Straßenverlaufs, werden die Zwischenräume als flächenhafte Verkehrsinseln (`TrafficIslands`) modelliert (siehe Abschnitt 3.7.1). Dies ist zum Beispiel der Fall bei Mittelinseln mit querenden Fußgängerfurten oder Fahrradwegen. Ändert sich der Nutzungstyp hingegen *in Längsrichtung* des Verlaufs, so erfolgt eine Modellierung als lineare Fahrstreifenbegrenzung (`LaneBorder`) eines Mittelstreifens (siehe Abschnitt 3.4). Dies ist zum Beispiel der Fall bei einfachen, parallel

verlaufenden Grünstreifen zwischen zwei `StandardLines`. Im Folgenden werden beide Situationen beschrieben.

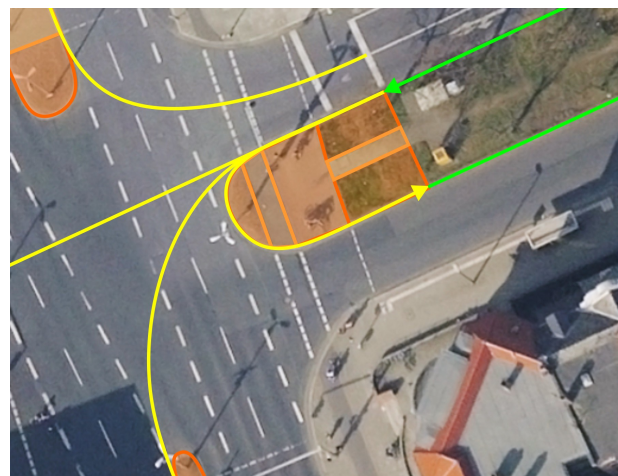
### 3.7.1. Verkehrsinsel

Verkehrsinseln (auch Mittelinseln genannt) werden als Flächen (`TrafficIsland`) modelliert, da sie besonders bei Kreuzungen nicht eindeutig entlang eines Straßenverlaufs konstruiert wurden. Eine Verkehrsinsel besteht dabei aus einer oder mehrerer `TrafficIsland`-Flächen. Jede dieser Flächen ist genau einer `StandardLine` oder `ConnectionLine` zugeordnet. Der Nutzungstyp ändert sich häufig *quer* zum Verlauf dieser Referenzlinie. Tabelle 28 zeigt zwei beispielhafte Verkehrsinseln.

**Tabelle 28:** Mittelinseln/`TrafficIslands` (orange).



einbahnig (Verkehrsinsel)



zweibahnig (Kreuzungsmittelinsel)

Die Modellierung von `TrafficIslands` orientiert sich an folgenden Punkten:

- Der Umriss einer `TrafficIsland`-Fläche wird wie auch eine Mark auf der Mitte der Markierung oder Bordsteinkante modelliert.
- Benachbarte `TrafficIsland`-Flächen liegen mit ihren Kanten topologisch aufeinander.
- Ändert sich der Typ der `TrafficIsland`, wird eine neue `TrafficIsland` erzeugt.
- `TrafficIslands`, die über eine Markierung definiert werden, benötigen keine weitere umlaufende Mark, die den Rand (oder Umriss) modellieren würde.

Die Art der von der Fläche bedeckten Stelle wird als Nutzungstyp und Material definiert. Dabei kommen dieselben Nutzungstypen und Materialien wie bei `LaneBorders` (siehe Abschnitt 3.4) zum Einsatz:

- Fahrstreifen = `driving`
- Parkstreifen = `parking`
- Sperrfläche = `restricted`
- undefiniert = `none`

- Fußgängerweg = `sidewalk`
- Radweg = `biking`
- Grünstreifen = `shoulder`
- Straßenbahntrasse = `tram`

Mögliche Materialien sind:

- Asphalt = `asphalt`
- Beton = `concrete`
- Kleinpflaster = `pavement`
- Kopfsteinpflaster = `cobble`
- Begrünung (Gras, Sträucher, usw.) = `vegetation`
- Schotter = `gravel`
- Erde = `soil`

Eine `TrafficIsland` erhält dabei die in Tabelle 29 aufgeführten Attribute. Tabelle 30 führt ein Modellierungsbeispiel mit Tabellenauszug auf.

**Tabelle 29:** Eigenschaften von `TrafficIslands`.

Name	Anzahl	Typ	Wert	Beschreibung
<b>id</b>	1	unit64_t	[0, max]	eindeutige Kennung
<b>dataSource-Description</b>	1	unit64_t	[0, max]	ID des beschreibenden <code>DataSourceDescription</code> -Objekts
<b>geometry</b>	1	Polygon Z	siehe Beispiel in Abschnitt 2.3	Polygon als WKT oder WKB
<b>material</b>	1	string	<code>asphalt</code> , <code>concrete</code> , <code>pavement</code> , <code>cobble</code> , <code>vegetation</code> , <code>soil</code> , <code>gravel</code>	Material der durch die <code>TrafficIsland</code> definierten Fläche
<b>type</b>	1	string	<code>driving</code> , <code>parking</code> , <code>restricted</code> , <code>none</code> , <code>sidewalk</code> , <code>biking</code> , <code>shoulder</code> , <code>tram</code>	Nutzungstyp der durch die <code>TrafficIsland</code> definierten Fläche
<b>parent</b>	1	unit64_t	[0, max]	ID einer <code>StandardLine</code> oder <code>ConnectionLine</code> zu der die <code>TrafficIsland</code> gehört



**Tabelle 30:** Beispielhafte Verkehrsinsel mit Flächenteilen mit Tabellenauszug.

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. Mittelstreifen

Sind Fahrstreifen baulich voneinander getrennt und ändert sich der Nutzungstyp des entstehenden Zwischenraums (Mittelstreifen) *in Längsrichtung* zum Verlauf einer angrenzenden *StandardLine* oder *ConnectionLine*, wird solch ein Mittelstreifen als *LaneBorder* einer der angrenzenden Referenzlinien modelliert. Das Vorgehen ist dabei analog zu der allgemeinen in Abschnitt 3.4 beschriebenen Fahrstreifendefinition. In diesem Fall kann die Geometrie einer Mittelstreifenbegrenzung mit der gegenüberliegenden Referenzlinie übereinstimmen oder diese teilweise überdecken.

Prinzipiell kann *eine* StandardLine auch in der Mitte eines Mittelstreifens liegen und die Fläche bis zum jeweils ersten Fahrstreifen wird mit einem Fahrstreifen vom Nutzungstyp *shoulder* modelliert. Um mögliche Konstruktionsprobleme bei Kreuzungen zu vermeiden, sollten aber möglichst *zwei* StandardLines bei baulicher Trennung genutzt werden. Bei parallel laufenden StandardLines sollten diese für eine einfachere Modellierung *auf gleicher Höhe enden*. Tabelle 31 zeigt ein entsprechendes Modellierungsbeispiel.

**Tabelle 31:** Grünstreifen zwischen zwei StandardLines (grün) modelliert als LaneBorder 1338. Die Geometrie dieser LaneBorder ist in dem Fall identisch mit StandardLine 1339 und der StandardLine 1337 zugeordnet. Alternativ kann die LaneBorder 1338 auch auf der StandardLine 1337 liegen und der StandardLine 1339 zugeordnet sein.



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. Oberflächenbeschreibung

Die Beschreibung einer Straßenoberfläche ist besonders für Reifen- und Fahrwerkssimulation in der Entwicklung als auch in der Bewertung (Fahrwerksabstimmung) wichtig. Dabei sind in der Regel möglichst genaue Daten notwendig, um exakt Unebenheiten wie Spurrillen, Schlaglöcher, Kanaldeckel usw. abbilden zu können. Messpunkte werden dafür in einem regelmäßigen Gitter modelliert und diese Oberflächenprofile dann mit einer Straßenbeschreibung verknüpft, sodass sie diese entsprechend ergänzen können. Die Modellierung der Daten ist recht einfach, die Komplexität liegt in der Erhebung der Daten mit adäquaten Sensoren selbst.

### 4.1. Straßenachsenreferenzierung

Die Referenzlinie zur Erfassung der Oberflächeninformationen muss mit der Referenzlinie für die Straßenbeschreibung (`StandardLine` und `ConnectionLine`) im Längs- und Höhenprofil übereinstimmen. Die Linie soll einen kontinuierlichen, ggf. geglätteten Verlauf haben.

Wird die Strecke in Teilstücken aufgenommen, darf zwischen den Teilstücken keine Lücke entstehen. Die Übergänge müssen sowohl in absoluter, als auch relativer Höhe stetig sein.

Auch im Querprofil muss die Oberflächenbeschreibung stetig sein. Diese sollte die gesamte Breite der Straßenbeschreibung bis zum Fahrstreifenrand abdecken (bzw. alle relevanten Fahrstreifen wie ggf. auch Fußgängerweg). Da die Oberflächenbeschreibung in OpenCRG in fester Breite erfolgt, die Straße in der Breite aber variieren kann, sollten der Datensatz außerhalb des Fahrstreifenbereichs mit NaN aufgefüllt werden. „Slope“ und „Banking“ sollten im Datensatz separiert sein.

Für eine spätere Kombination der Oberflächendaten mit der Straßenbeschreibung muss die finale Straßenachse endgültig festgelegt sein. Das Road2Simulation-Format sieht vor, dass die Straßenbeschreibungsdaten durch mathematische Transformation bspw. ins OpenDRIVE-Format überführt werden. Diese dann als Kurve modellierte `StandardLine` muss dann als Referenzlinie für die Oberflächeninformationen dienen.

Für die Zusammenführung sollte in OpenDRIVE der `attached`-Mode genutzt werden, in dem die `StandardLine` der Oberflächeninformation durch die `StandardLine` der Straßenbeschreibung ersetzt wird. Die Höheninformation der Oberflächenbeschreibung wird auf das Profil der Straßenbeschreibung addiert. Somit bleiben die originale Position, Krümmung, Höhenangabe und Überhöhung in OpenCRG unberücksichtigt und die Oberflächeninformation folgt der Straßenbeschreibung.

### 4.2. Oberflächengenauigkeit

Die Oberflächenbeschreibung soll mit einer Rasterweite von 5x5 mm erzeugt werden. Die Oberflächenabschnitte (in Bezug zur Straßenreferenzlinie) sollten in 5x5 m Flächen erzeugt werden.

Die relative zweidimensionale Lagegenauigkeit und die Höhengengenauigkeit muss mindestens 1 mm

betragen. Die absolute Lagereferenzierung und Höhenreferenzierung muss mindestens 1 m oder besser betragen. Die Abweichung der Oberflächendaten darf an markanten Punkten (bspw. Schlaglöcher, Kanaldeckel), nicht größer als 5 bis 25 mm in der zweidimensionalen Lage und 1 bis 5 mm in der Höhe sein. Durch Messstörungen generierte Ausreißer dürfen nicht enthalten sein.

Die Angaben sollten in UTM (bspw. UTM 32N) mit dem zugrundeliegenden, amtlichen Lagebezugssystem ETRS89 (also EPSG:25832<sup>16</sup>) mit entsprechend passendem Höhenreferenzsystem (bspw. DH-HN92 (EPSG:5783<sup>17</sup>)) erfolgen, siehe Kapitel 2.2. Werden Straßenverlauf *und* Oberfläche vermessen, sollten beide Datensätze mit gleicher Projektion modelliert werden.

In OpenCRG sind die vorgesehenen Variablen `reference_line_offset_x`, `reference_line_offset_y` und `reference_line_offset_z` im Block `$ROAD_CRG` für die Dokumentation der Georeferenzierung zu benutzen. Die Projektion wird im Block `$ROAD_CRG_MPRO` über die Variable `proj_nm` angegeben (bspw. `proj_nm = 'UTM_32T'`).

---

<sup>16</sup><http://epsg.io/25832>

<sup>17</sup><http://epsg.io/5783>



## 5. Anmerkungen

Die Arbeiten für den Road2Simulation-Leitfaden und das zugehörige Datenmodell basieren auf dem DLR-Forschungsprojekt „Virtuelle Welt<sup>18</sup>“ und der vom DLR betriebenen „Anwendungsplattform Intelligente Mobilität“ (AIM<sup>19</sup>). Die Erstellung des Leitfadens sowie des Datenmodells wurde durch Audi, BMW, Daimler, Porsche, Volkswagen sowie dem DLR ermöglicht.

---

<sup>18</sup><http://www.dlr.de/ts/mittendrinn/virtuellewelt>

<sup>19</sup><http://www.dlr.de/ts/aim>

## 6. Änderungshistorie

### v1.2 zu v1.2.1

- Überführung des gesamten Dokuments nach  $\text{\LaTeX}$  mit daraus resultierenden zahlreichen Formatierungsänderungen und -anpassungen
- Offizielle Veröffentlichung des Leitfadens auf [zenodo.org](https://zenodo.org) (siehe Abschnitt 1.2)
- Veröffentlichung einer englischsprachigen Übersetzung (siehe Abschnitt 1.3)
- Dokumenttitel der deutschsprachigen Version geändert von „Road2Simulation Guidelines“ zu „Road2Simulation-Leitfaden“
- Lizenz geändert zu CC BY 4.0
- Separates Beispielverzeichnis entfällt aufgrund der Überführung der vorher eigenständigen „Beispiel“-Kategorie zu normaler Tabellenkategorie
- Explizite Querverweise auf Beispieltabellen im Fließtext hinzugefügt
- Einträge im Abbildungs- und Tabellenverzeichnis leserlicher gestaltet und verkürzt
- Viele kleine Korrekturen im Text
- Kapitel „Notizen“ entfällt
- Ungültiger Website-Link aus Fußnote 12 in Abschnitt 2.6.2 gegen Wikipedia-Link ersetzt
- Abbildungen 12 und 13 zusammengefasst zu Abbildung 12

### v1.1 zu v1.2

- Viele Verbesserungen/Klarstellungen im Text und den Beispielen
- Lizenz hinzugefügt
- Änderungshistorie hinzugefügt
- OpenCRG: Details zur Straßenachsenreferenzierung und Attributverwendung hinzugefügt
- Hinweis hinzugefügt: Erhobenes Geländemodell sollte zusammen mit den Vektor-Daten ausgeliefert werden
- Beispiele hinzugefügt: Detaillierte Modellierung von `LaneBorders`
- Beispiele hinzugefügt: Detaillierte Modellierung von `Marks`
- Datenmodell: `BaseReferenceLine`: Umbenannt zu `ReferenceLine`
- Datenmodell: `ReferenceLine`: Attribut `category` hinzugefügt
- Datenmodell: Persistente Tabellen `StandardLine` und `ConnectionLine` wurden gruppiert zu `ReferenceLine`
- Datenmodell: `ConnectionLine`: Attribut `auxiliary` hinzugefügt
- Datenmodell: `LaneBorder`: Material `grass` umbenannt zu `vegetation`
- Datenmodell: `TrafficIsland`: Material `grass` umbenannt zu `vegetation`
- Datenmodell: `Mark`: Attribut `length` umbenannt zu `markLength`
- Datenmodell: `Mark`: Attribut `space` umbenannt zu `spaceLength`
- Datenmodell: `Mark`: Typ `pedestrian walk` umbenannt zu `crossing`
- Datenmodell: `Mark`: Typ `cross walk` umbenannt zu `zebra crossing`

- Datenmodell: Mark: Farbe orange hinzugefügt
- Datenmodell: PunctualObject: Umbenannt zu PointObject
- Datenmodell: PointObject: Neue Attribute width, depth, height
- Datenmodell: PointObject: Bidirektionale one-to-many-self-relationship hinzugefügt
- Datenmodell: PointObject: Typ curved traffic light post hinzugefügt
- Datenmodell: PointObject: Typ overhead gantry mast hinzugefügt
- Datenmodell: PointObject: Typ overhead gantry hinzugefügt
- Datenmodell: PointObject: Typ reflector post hinzugefügt
- Datenmodell: PointObject: Typ curved street lamp hinzugefügt
- Datenmodell: PointObject: Typ boom umbenannt zu traffic light boom
- Datenmodell: PointObject: Typ emergencyphone umbenannt zu emergency phone
- Datenmodell: PointObject: Typ phonebox umbenannt zu phone box
- Datenmodell: PointObject: Typ phonestele umbenannt zu phone stele
- Datenmodell: PointObject: Doppelter Signaltyp 1.000.008.10 umbenannt zu 1.000.008.20
- Datenmodell: PointObject: Standardsignaltyp 1.000.000 hinzugefügt
- Datenmodell: PointObject: Signaltyp F 3 hinzugefügt
- Datenmodell: PointObject: Signaltyp F 4 hinzugefügt
- Datenmodell: PointObject: Signaltyp F 5 hinzugefügt
- Datenmodell: PointObject: Signaltyp A 1 hinzugefügt
- Datenmodell: PointObject: Signaltyp A x hinzugefügt
- Datenmodell: PointObject: Signaltyp A 2b hinzugefügt
- Datenmodell: PointObject: Signaltyp W 3 hinzugefügt
- Datenmodell: PointObject: Signaltyp W 11 hinzugefügt
- Datenmodell: PointObject: Signaltyp W 12 hinzugefügt
- Datenmodell: PointObject: Signaltyp W 13 hinzugefügt
- Datenmodell: PointObject: Signaltyp 1.000.021 umbenannt zu F 0
- Datenmodell: PointObject: Signaltyp 1.000.022 umbenannt zu F 1
- Datenmodell: PointObject: Signaltyp 1.000.023 umbenannt zu F 2
- Datenmodell: PointObject: Signaltyp 1.000.024 umbenannt zu F 6 und W 0
- Datenmodell: PointObject: Signaltyp 1.000.025.10 umbenannt zu W 1
- Datenmodell: PointObject: Signaltyp 1.000.025.20 umbenannt zu W 2
- Datenmodell: PointObject: Signaltyp 1.000.026 umbenannt zu W 14
- Datenmodell: PointObject: Piktogrammtyp 30 umbenannt zu horizontal 30
- Datenmodell: PointObject: Piktogrammtyp 50 umbenannt zu horizontal 50
- Datenmodell: PointObject: Schreibweise des Piktogrammtyps stopping restriciton korrigiert zu stopping restriction
- Datenmodell: LinearObject: Typ tunnel hinzugefügt
- Datenmodell: LinearObject: Typ bridge hinzugefügt



## Abkürzungsverzeichnis

**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

**NHN** Normalhöhennull

**OGC** Open Geospatial Consortium

**OpenCRG** open curved regular grid

**OpenDRIVE** open digital road information for virtual environments

**SI** Système international d'unités

**SRID** Spatial Reference Identifier

**UTM** Universal Transverse Mercator coordinate system

**WKB** well-known binary

**WKT** well-known text

# Abbildungsverzeichnis

1 Struktur des Datenmodells . . . . .	6
2 Geometrischer Übergang zwischen aufeinanderfolgenden Linienzügen . . . . .	8
3 Übersicht über Straßenzüge, Kreuzungen und darin enthaltene Verknüpfungen . . . . .	13
4 Abbiegebeziehung . . . . .	17
5 Komplexe Kreuzung aufgeteilt in mehrere einfachere Einzelkreuzungen . . . . .	21
6 StandardLines und ConnectionLines bei Autobahnauf- und -abfahrten . . . . .	23
7 Richtige und falsche Modellierungsreihenfolge von LaneBorders. . . . .	26
8 Modellierung von abrupten Fahrstreifenenden. . . . .	26
9 Notwendige und obligatorische LaneBorders . . . . .	31
10 Unterschiedliche Markierung eines Straßenabschnitts. . . . .	34
11 Unterschiedliche Markierungen auf einer Kreuzungsfläche. . . . .	35
12 Kreuzung, mehrere Fahrstreifen mit baulicher Trennung sowie auf StandardLine und ConnectionLine verortete punktuelle Objekte . . . . .	45

## Tabellenverzeichnis

1 Beispiel für Metainformationen von Geometriedatensätzen . . . . .	9
2 Unterschiedliche Straßensituationen ohne modellierte Elemente . . . . .	11
3 Eigenschaften von <code>StandardLines</code> . . . . .	14
4 Modellierung der <code>StandardLines</code> und <code>ConnectionLines</code> . . . . .	15
5 Beispiel zweier <code>StandardLines</code> mit Tabellenauszug . . . . .	16
6 Eigenschaften von <code>ConnectionLines</code> . . . . .	18
7 Beispiel einer <code>ConnectionLine</code> mit Tabellenauszug . . . . .	19
8 <code>ConnectionLines</code> im Kreuzungsbereich . . . . .	20
9 <code>ConnectionLines</code> im unterteilten Kreisverkehr . . . . .	21
10 <code>LaneBorders</code> und <code>StandardLine</code> -Enden. . . . .	24
11 Eigenschaften von <code>LaneBorders</code> . . . . .	27
12 Beispiel mehrerer <code>LaneBorders</code> mit Tabellenauszug . . . . .	28
13 <code>LaneBorders</code> im Kreuzungsbereich. . . . .	29
14 <code>LaneBorders</code> im unterteilten Kreisverkehr . . . . .	30
15 Fahrbahnmarkierungen und Lücke-Strich-Angaben . . . . .	33
16 Eigenschaften von <code>Marks</code> . . . . .	36
17 Beispiel mehrerer parallel und quer verlaufender Markierungen/ <code>Marks</code> mit Tabellenauszug . . . . .	37
18 Beispiel für Sperrzonen und Zebrastreifen als Markierungen/ <code>Marks</code> mit Tabellenauszug . . . . .	38
19 Lineare und punktuelle Straßenobjekte . . . . .	39
20 Typen punktförmiger Objekte. . . . .	40
21 Signalgebertypen . . . . .	42
22 Bodenmarkierungstypen . . . . .	43
23 Eigenschaften von <code>PointObjects</code> . . . . .	46
24 Beispielhafte Kreuzung mit verorteten punktuellen Objekten mit Tabellenauszug . . . . .	47
25 Eigenschaften von <code>LinearObjects</code> . . . . .	49
26 Beispiel für lineare Objekte (ohne Brücken-Objekt) mit Tabellenauszug . . . . .	50
27 Beispiel einer Brücke als <code>LinearObject</code> mit Tabellenauszug . . . . .	51
28 Mittelinseln/ <code>TrafficIslands</code> . . . . .	52
29 Eigenschaften von <code>TrafficIslands</code> . . . . .	53
30 Beispielhafte Verkehrsinsel mit Flächenteilen mit Tabellenauszug . . . . .	54
31 Grünstreifen zwischen zwei <code>StandardLines</code> modelliert als <code>LaneBorder</code> . . . . .	55