Supporting the Implementation of Driving Simulator Environments Through Established GIS Approaches by Extending the Geospatial Data Abstraction Library (GDAL) with OpenDRIVE

Michael Scholz[1], Ana María Orozco Idrobo[2]

(1) German Aerospace Center (DLR), Institute of Transportation Systems, Lilienthalplatz 7, 38108 Braunschweig, Germany, e-mail: michael.scholz@dlr.de
(2) E-mail: ana.orozco.net@gmail.com

Abstract - *Extensive and highly detailed real-world road networks obtained through mobile mapping and spatial data processing build a basis for development and evaluation of advanced driver assistant and automation systems nowadays. Such road networks, as used in driving/traffic simulation and test vehicles, are provided in specialised description formats – one of which being OpenDRIVE. Sparse tool support makes generation, processing and validation of OpenDRIVE cumbersome. The GIS domain provides well-established and convenient tools for spatial data processing, but does not yet offer support for OpenDRIVE data. This paper describes an extension of the free and open-source Geospatial Data Abstraction Library (GDAL) with OpenDRIVE as missing link between the domains of driving simulation and geographic information systems. By bringing both domains closer together we hope to stimulate promising development of scenario generation and synthesis of reality-based road networks for driving simulator applications.*

Keywords: *Driving Simulation, OpenDRIVE, GDAL/OGR, Simple Features, GIS*

Fuelling the Domain of Driving Simulation with GIS

For the development and evaluation of advanced driver assistant and automation systems different driving simulators and test vehicles are used by the DLR-Institute of Transportation Systems, OEMs and various other research facilities. In this context OpenDRIVE [Dup15] evolved as a widely-used road description format for geometrical and logical representation of complex road networks. The increasing demand for generation of OpenDRIVE – based on real-world scenarios [Ric15] – requires an easy integration of OpenDRIVE in GIS (Geographic Information System) applications for combination with additional geo-referenced data, for example with road infrastructure, aerial imagery or GNSS traces. Prior research by Orozco Idrobo revealed that the few available OpenDRIVE tools and editors are mostly commercial and offer insufficient support for common geodata, or even none at all [Idr15]. The work by Orozco Idrobo serves as basis for this paper and describes one possible approach to transform OpenDRIVE datasets into common GIS data formats, thus offering the possibility to use well-established GIS tools ad hoc in analysis, processing and visualisation of OpenDRIVE. This paper focusses on the data format characteristics, the implementation and the publishing of our developed GDAL extension as open source, making it available to a broad audience.

Bringing the domains of driving simulation and GIS closer together is the first step in stimulating promising development of scenario generation and synthesis of reality-based road networks for driving simulator applications. A case study shows interfacing OpenDRIVE data with GIS tools for automatic matching of cadastral data such as road infrastructure to a real-world road network in OpenDRIVE using conventional geo-processing tools.

Extending GDAL/OGR with OpenDRIVE

Spatial data commonly used in the GIS domain by, for instance, road infrastructure operators or public authorities can be categorised into raster and vector formats. Pixel-based rasters mostly represent continuous data such as imagery and elevation, whereas vector data is used for discrete geographic features. Those vector formats typically offer a flat hierarchy only, focussing on the representation of distinct features through discrete coordinates and an arbitrary number of attributes per feature.

OpenDRIVE

OpenDRIVE [Dup15], as an XML-based format for detailed description of road networks, is predestined to fit into such a GIS vector model. However, it offers a much deeper hierarchy of different elements interacting with each other compared to standard vector formats and thus cannot be transformed simply without ambiguity. Considering OGC's Simple Features [Ope11] most of OpenDRIVE's elements could be represented as discrete `Points`, as `LineStrings` or as `Polygons`. Junctions as logical elements, for example, do not have any geometric properties in OpenDRIVE at all. Regardless, a visual representation of junctions can often be useful and could be implemented through the vector model as well.

Starting with version 1.4 OpenDRIVE offers the possibility to specify a spatial reference system as PROJ.4-string for its contained data (see section 5.2.1 of the OpenDRIVE specification), which makes it easy to set road networks into relation with other spatial data. But unlike other GIS formats OpenDRIVE does not strictly follow the convention of solely one coordinate reference system per dataset.

Just the 2D starting point coordinates of road reference geometries are given in an *inertial system*, which complies with the aforementioned spatial reference specified on document-level.

These road reference geometries itself are then described by a series of mathematical functions (lines, arcs, Euler spirals, 3rd order polynomials) in a linear *local system* relative to the inertial position and heading, as shown in Listing 1 as an example.

Road-related geometric elements are described by coordinates in yet another linear *track system* relative to their parents, such as traffic signs assigned to certain roads. This track system depicts linear referencing along the parent's reference geometries. Transformation between these three systems is necessary when transferring OpenDRIVE into the GIS vector model.

Listing 1: Road reference geometry definition in OpenDRIVE showing an `arc` segment followed by a `line` with `s` being the linear measure along a road.

```
<planView>
  <geometry s="0.0" x="604944.1037"
            y="5792860.1272"
            hdg="3.5148"
            length="9.7589">
    <arc curvature="9.0884E-4"/>
  </geometry>
  <geometry s="9.7589" x="604935.03"
            y="5792856.5285"
            hdg="3.5237"
            length="12.0">
    <line/>
  </geometry>
</planView>
```

GDAL/OGR

In the GIS domain the modular open-source Geospatial Data Abstraction Library (GDAL) [Ope17b] serves as a standard interface for conversion between heterogeneous geo-formats. Whereas GDAL itself focusses on *raster* formats, its integrated Simple Feature Library (OGR) deals with processing of *vector* formats. Extending OGR to natively support OpenDRIVE will close the gap between the domains of driving simulation and GIS to offer new approaches of OpenDRIVE creation and processing to the driving simulation community. The objective is to extend OGR by a driver providing read-support for OpenDRIVE by converting its mathematical geometry representation to OGC Simple Features [Ope11] and exposing them through the OGR-specific interface. This step involves harmonising OpenDRIVE's different internal coordinate systems through appropriate transformations. The open-source geometry engine GEOS [Ope17a] as component of GDAL provides helpful processing functionality such as for linear referencing, spatial overlays or coordinate snapping.

Implementation

OGR as part of GDAL already offers drivers for different other XML-based geographic data formats. Its heterogeneous implementation was examined to find the best solution in regard of a new OpenDRIVE driver (see [Idr15]). For the desired extension we generate a C++ class structure automatically, resembling the OpenDRIVE schema in version 1.4 through common XML data binding techniques. The used open-source framework CodeSynthesis XSD [Cod17] offers straightforward un-marshalling and marshalling of OpenDRIVE XML files and covers the general need of road network parsing. It unfolds complex road networks as easily processable object structures in C++.

As implementation of our novel OpenDRIVE driver these objects are converted into OGC Simple Features, which is the standardised interface for all vector-based data formats in OGR, and enriched with OpenDRIVE-relevant attributes. OGC Simple Features basically encompass `Points`, `LineStrings` and `Polygons`. We suggest a mapping of OpenDRIVE elements to these feature types as depicted in Table 1. Once converted into this representation, such features are easily accessible through common GIS frameworks such as QGIS, GRASS, ArcGIS, FME, OpenLayers or Google Earth. Due to their mathematical, continuous representation in OpenDRIVE, linear geometries have to be discretised into point sequences during conversion into Simple Features. The desired point sample distance can be specified as layer creation option of the OGR driver implementation.

Table 1: Suggested mapping of OpenDRIVE elements to OGC Simple Feature types in OGR.

Simple Feature type	OpenDRIVE element
`Point`	punctual object (e.g. signal, sign)
`LineString`	road reference line, driving lane boundary, road mark, linear object (e.g. guardrail, barrier)
`Polygon`	driving lane, parking space

At the time of writing only road reference lines have been mapped to discrete `LineStrings`. The remaining linear elements depicted in Table 1 are yet to come. Further, GDAL/OGR offers to load *one data source* as *multiple geodata layers*. This functionality could be used to expose `Point` and `Polygon` types of OpenDRIVE elements as individual layers additionally.

Future Tasks

Right now we just considered plain 2D geometries. In OpenDRIVE a road can be attributed with different kinds of elevation information. At first, it can have assigned a longitudinal elevation profile in form of 3rd order polynomials (see Listing 2). Additionally, the road's crossfall and superelevation describe its lateral profile. Integrating these properties into the Simple Feature representation of geometries is desirable for the future but not necessarily required for basic GIS applications such as visualisation and spatial overlay operations.

Listing 2: Linearly referenced elevation information of a road in OpenDRIVE with coefficients of 3rd order polynomials.

```
<elevationProfile>
  <elevation s="0.0"
    a="118.0237" b="-0.0019"
    c="-0.0109"  d="0.0021"/>
  <elevation s="3.5028"
    a="117.9712" b="-0.0028"
    c="0.0012"   d="-1.0364E-4"/>
</elevationProfile>
```

The current driver implementation has been developed around OpenDRIVE schema of version 1.4. Backwards compatibility can be considered to be supported implicitly because previous versions roughly just added new elements to the OpenDRIVE standard. Experimental loading of road networks in version 1.2 and 1.3 was successful but full compatibility is not guaranteed and schema validation is unsupported. Full legacy support would require either the driver being implemented individually for each preceding schema version or an automated version recognition with independent handling.

Free and Open Source

Orozco Idrobo's first driver implementation has been refined and can now be distributed as decoupled and easy-to-use plug-in for GDAL. It offers cross-platform compatibility due to the nature of C++ and the used dependencies. The library extension was successfully tested both under Linux and Windows on 32- and 64-bit architectures. GDAL has been forked on GitHub to be extended with this new driver [Ger17] which again is published as free and open source under the X/MIT licence [Mas88]. In the future our OpenDRIVE driver may be moved into GDAL as core library part.

Results and Discussion

The functionality of the extension was tested in OGR's provided utility programs and the established open-source GIS tool QGIS which uses GDAL/OGR as data management backend. At the time of writing the implementation of the GDAL OpenDRIVE driver already offers the possibility to convert basic OpenDRIVE elements through built-in utilities into different standardised geodata formats such as ESRI Shapefile, GeoJSON, GML, KML, PostGIS or Oracle Spatial (Figure 1). First tests in QGIS showed that OpenDRIVE XML files in version 1.4 can be successfully loaded ad hoc as additional map layers with correct spatial reference information (Figure 2).

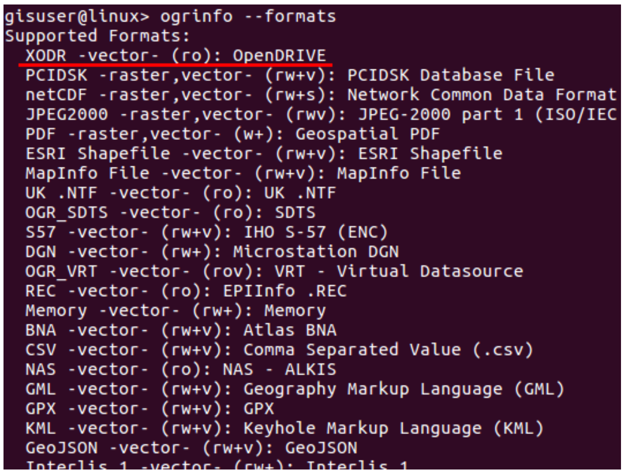

Figure 1: Output of OGR's utility program `ogrinfo` listing all supported data formats including OpenDRIVE.

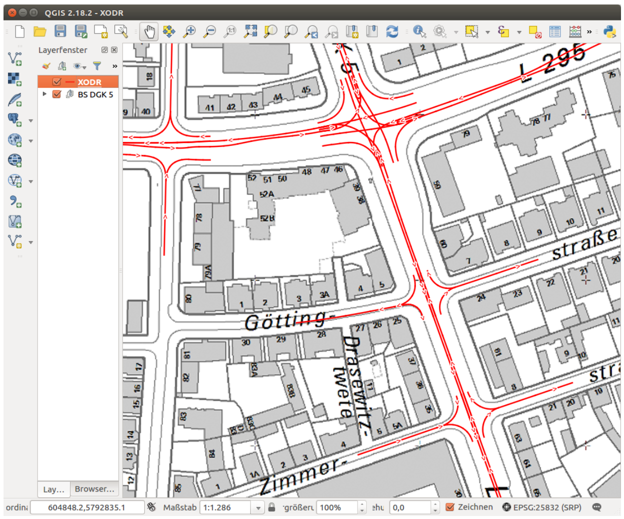

Figure 2: QGIS map with digital city plan of Brunswick and overlaid road reference lines of the surveyed inner city ring in OpenDRIVE 1.4 (red).

In our current research one important use case for the newly added support is the geometric and attributive update of OpenDRIVE road networks. As specialisation of this, the enrichment with punctual road infrastructure obtained from cadastral geodata such as traffic signals or road sign information is considered. Infrastructure databases may change more frequently than their corresponding road networks which makes a decoupled handling necessary. Based on our novel straightforward-processing capability of OpenDRIVE a first tool chain was developed to accumulate plain road networks (surveyed or digitised from cadastral plans) with third-party geodata using common GIS functionalities. For this, reading of OpenDRIVE was performed as described above. Once having OpenDRIVE elements queryable as Simple Features in memory the association with point geometries can be conducted through GIS methods and with the help of basic computational geometry algorithms. This allows linking of traffic signs to potential parental road elements, for example. Such roads are "enriched" by adding the according `<object>` or `<signal>` elements directly into the XML data tree. Writing the enriched XML back to the file system must not touch any of its original elements, thus keeping the continuous road geometries as they are. Discretisation takes only place temporarily for processing simplification. However, the accumulated network can not be written back through our GDAL driver yet. This last step has been realised by a different in-house implementation within the scope of the project Virtual World [Ric16b].

For now only *importing* of OpenDRIVE and exposing some of its elements as OGC Simple Features through OGR has been outlined. As the power of GDAL/OGR lies in conversion between different formats and as our aforedescribed use case pointed out, the demand of *exporting* other road-related geodata into the OpenDRIVE format will emerge consequentially. This, however, can not yet be accomplished straightforwardly through our approach because of OpenDRIVE's complexity compared to common "flatter" geodata. Its strong focus on topological information representation, internal cross-reference between elements and detailed attribution makes it difficult to adopt arbitrary geodata. Just plain geometries could be transformed ad hoc into OpenDRIVE but even then it would be unclear whether a certain `LineString` identified a road reference geometry, a lane border or a linear object, for example. A previous manual classification and attribution, adapted to each possible input format, would be necessary to achieve the desired automatic mapping. Therefore, our OGR driver implementation currently lacks support for automatic exporting into OpenDRIVE.

Simple map visualisation, data comparison and, for example, spatial overlay of OpenDRIVE road networks with aerial imagery for quick data validation lies at hand when realised through the GIS tools mentioned here. Also, the spatial analysis of GNSS trajectories obtained in real-world driving tests combined with high-precision OpenDRIVE maps is a benefit and easily possible. Our approach offers broad access to new analysis techniques of OpenDRIVE for interdisciplinary applications independent from commercial/proprietary software frameworks.

Conclusion and Outlook

We showed that the complex road description format OpenDRIVE can be integrated into common GIS workflows by extending the established Geospatial Data Abstraction Library (GDAL). On the one hand, with the help of GIS new analysis and processing functionality is directly available to be used in driving simulation applications. We already use the described transformation of OpenDRIVE elements into OGC Simple Features for infrastructure updates of Brunswick's road networks in the project Virtual World [Ric16b]. On the other hand, the GIS community can benefit of additional geodata made available through OpenDRIVE as yet another provider format. Certainly the GIS community may use better-suited geodata formats for its purposes but the rising availability of high-precision street data obtained through mobile mapping in the automotive industry nowadays discloses pretty databases not to be scoffed at.

The modularity and maturity of GDAL/OGR offers the possibility to implement support for different other domain-specific road description formats such as the Navigation Data Standard (NDS), RoadXML or ROAD5 in the future. The well-established and standardised interface of Simple Features is also used as simplified data model in the project Road2Simulation [Ric16a]. It could further suit as an intermediate data format for the translation between those different road description formats, although conversion forth and back between continuous and discrete geometry representations will yield numeric challenges. Free availability of our GDAL extension will allow easy access and contribution to future development for research facilities as well as industrial partners in this field of application.

References

Code Synthesis Tools CC, CodeSynthesis XSD, `http://www.codesynthesis.com/products/xsd`, 2017, accessed: 2017-07-07.

M. Dupuis, **OpenDRIVE Format Specification, Rev. 1.4, Issue H**, VIRES Simulationstechnologie GmbH, 2015, accessed: 2017-04-21.

German Aerospace Center (DLR), Institute of Transportation Systems, gdal, `https://github.com/DLR-TS/gdal`, 2017, accessed: 2017-07-07.

A. M. O. Idrobo, **Extension of the Geospatial Data Abstraction Library (GDAL/OGR) for OpenDRIVE Support in GIS Applications for Visualisation and Data Accumulation for Driving Simulators**, Master's thesis, Technische Universität München, 2015.

Massachusetts Institute of Technology, MIT License, `https://opensource.org/licenses/mit-license.php`, 1988, accessed: 2017-07-07.

Open Geospatial Consortium Inc., **OpenGIS Implementation Standard for Geographic information – Simple feature access – Part 1: Common architecture**, Tech. rep., Open Geospatial Consortium Inc., 2011.

Open Source Geospatial Foundation, Geometry Engine, Open Source, `https://trac.osgeo.org/geos`, 2017, accessed: 2017-07-07.

Open Source Geospatial Foundation, Geospatial Data Abstraction Library, `http://www.gdal.org`, 2017, accessed: 2017-07-07.

A. Richter, M. Fischer, T. Frankiewicz, L. Schnieder and F. Köster, **Reducing the gap between simulated and real life environments by introducing high-precision data**, in Driving Simulator Conference 2015 Europe, 227–228, Max Planck Institute for Biological Cybernetics, Tübingen, 2015.

A. Richter and M. Scholz, **"The Surveyor's Guide to the Driving Simulation": Developing and Evaluating of Guidelines for Straightforward Road Surveying for Driving Simulator Databases for Development and Test of Driver Assistance and Automation Systems**, in Proceedings of the Driving Simulator Conference 2016 Europe VR, 119–122, Driving Simulator Association, 2016.

A. Richter, M. Scholz, H. Friedl, T. Ruppert and F. Köster, **Challenges and experiences in using heterogeneous, geo-referenced data for automatic creation of driving simulator environments**, *Simulation-Transactions of the Society for Modeling and Simulation International*, vol. 92(5): 437–446, 2016.