

## 8 3D visualization for microscopic traffic data sources

*Andreas Wenger;*

*Xenoage Software, Germany*

*Matthew Fullerton, Mathias Baur, Silja Hoffmann, Jonas Lüßmann;*

*Chair of Traffic Engineering and Control, Technische Universität München, Germany*

### 8.1 Abstract

The typical approach to realistic 3D visualization of microscopic traffic data is usually location-specific and hence requires a lot of effort from the user to generate an eye-pleasing model. We present a simulation/location/data-neutral 3D visualization software module that allows the 3D display of an XML-coded road network, vehicles and communication involving vehicles. These objects are automatically modeled from source data with empty space filled in an intelligent way. The simulation can be freely explored in time and space. Vehicles can be color-coded, viewed from above or from the driver's perspective, or alternatively in a traditional 2D form.

Keywords: Microscopic traffic simulation, data visualization, 3D visualization

### 8.2 Introduction

Microscopic traffic simulation has become an established tool for the examination of microscopic level "what if" questions in traffic engineering [1]. These can include proposed infrastructure changes, changes in traffic regulation, and devices in traffic providing communication between vehicles for purposes of improved traffic safety or efficiency. Although the critical outcomes of simulation are usually 'hard' numbers (such as average speeds, travel times, traffic flows), visualization of some kind is essential for purposes of visual validation and debugging. In both science and industry, 3D visualizations have become more popular: driver and traffic behavior is easier to judge, the viewer is more engaged in the presentation and the display invites exploration and questions. Hence such visualizations are useful both for demonstration and teaching.

For a traffic simulation, we usually only configure roads, junctions, traffic lights etc. There may also be objects in the simulation that correspond to real world objects, but are somewhat abstracted (for example, a network link with reduced speed might actually correspond to an overhead motorway control sign). Overall, the normal simulation objects are not enough to generate a realistic and attractive 3D visualization where viewers probably expect other real-world items such as safety barriers, vegetation and buildings. In addition, with the trend towards simulating communication amongst vehicles and infrastructure [2], we have data on communication that we also want to see in a simulation visualization. As tools for simulating communication are usually in the form of add-on modules that interface with the simulation tool (e.g. VSimRTI [3], iTETRIS [2]), the communication is not usually displayed in the simulation user interface. This abstraction also motivates the choice for the visualization to

also be abstracted and separated from the simulation tool: modern questions of traffic engineering, in particular vehicular communication, may require a greater diversity of simulation tools, if only for verification in what is still a relatively young field [4]. We know of no popular microscopic traffic simulator that visualizes the communication between vehicles as part of the normal visualization. SUMO [5] in particular lacks any kind of 3D visualization, either online during the simulation or offline based on the simulation log file.

Furthermore, in modern traffic engineering, we also have access to other data with a similar level of detail as microscopic simulation results, such as Floating Car Data (FCD) or trajectories from video analysis. Given visualization software that only requires a network description and the trajectories of vehicles, there is no reason why these data can also not be viewed in the same animated, 3D format.

To resolve these issues, we have developed a module for the vtSim data and simulation management framework [4], vtSim.VIEW that visualizes any road network, set of vehicle trajectories upon it and communication between entities. The data source for the visualization is a simple XML: the main vtSim software is not necessarily required, acting only as a bridge between simulation-native data formats and vtSim.VIEW. Based on this road network, the module automatically generates much of the road environment and surroundings that are not defined in the simulation network: no 3D modeling is required on the part of the user. Although the visualization is offline in the sense that the simulation is performed independent of the visualization (only the data describing the network and vehicle trajectories are required), it is performed in real time based on this data, hence not requiring any advance video rendering before viewing. It should be stressed that the visualization is in no way a traffic simulator and is fully independent of whatever vehicle or communication models were used in the simulation. This also has the advantage that for slower-than-real-time simulations (as is often the case when communication between vehicles is included), the visualization is not delayed and can be viewed and reviewed from any viewpoint at the preferred speed.

The paper is structured as follows: first details of the implementation are covered, including a description of the data format and how the source data are interpreted and displayed. This is followed by a section on how the program can be used, including a number of examples that showcase the module features. We conclude with an outlook towards future work.

## 8.3 Implementation

### 8.3.1 Connection to other software modules

The data necessary for vtSim.VIEW are first a set of XML SUMO-compatible network source files (primarily edge and connection descriptions) and second a description (also in XML) of the vehicle data. These can either be created by the user (e.g. through conversion scripts) or by processing the network and results with the main vtSim tool. Here it is important to preserve the network, regardless of simulation tool used, as-simulated, which differs from the normal vtSim use case of importing networks and standardizing them into the SUMO format. Hence the network actually used for simulation is treated as a result of the simulation itself. Processing of the simulation output allows vehicle trajectories and communication data to be output for visualization. This workflow is summarized in Figure 8-1.

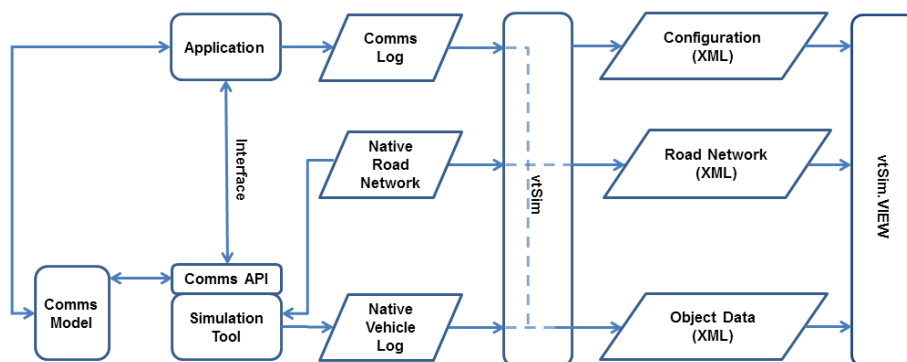


Figure 8-1: Visualization workflow. The network used for simulation along with the simulation vehicle trajectories log and communications log from the simulated application are prepared as XML files for the vtSim.VIEW visualization tool. This can either be done by the user as part of their results-handling process or achieved (currently for VISSIM) using vtSim.

### 8.3.2 Data format

The road network is coded in XML, broadly in accordance with the SUMO XML specification [6]. This specification is well documented and acts as a standard for the current vtSim software. Three XML files are required; .nod.xml representing road network nodes, .edg.xml representing road network edges (links) and .con.xml describing the connections between links. As we use both VISSIM [7] and SUMO [5] in our work, some parameters were added for preserving the geometry of edge connections (or in VISSIM terminology, connectors between links), which, unlike in SUMO, can have non-negligible lengths, geometries and must not necessarily begin or end at the start- or end-point of an edge. However, the data format used was intended to provide for SUMO compatibility. Vehicle data is structured as shown below.

The structure of an object data file for vtSim.VIEW. The minimum information to see moving vehicles are vehicle definitions with timestep entries describing their locations. In addition communication messages and vehicle statuses can be provided, referenced to relevant timestep entries. LKW represents truck traffic and PKW normal vehicles.

```
<vehicles>
  <vehicle id="29" type="LKW">
    <timestep LateralPosRel="0.5" LinkPos="927.10" LinkPosRel="0.97" X="25.62"
      Y="-89.61" acceleration="1.42" lane="2" link="200733801"
      speed="24.67" time="300.0" timeGap="80.22">
      <warning distanceToEvent="9.7" type="congestion" />
      <status type="warned" />
    </timestep>
    ... further timesteps ...
  </vehicle>
  ... further vehicles ...
</vehicles>
```

Another scenario xml file states which files are required and which camera positions will be available. This file can be created manually or (as in vtSim) automatically written along with the data files to be visualized.

### 8.3.3 Language choice and libraries used

The module is written in Java, ensuring runtime compatibility with all desktop platforms. The fact that most popular traffic simulations are not written in Java is not a problem because of the modular structure of vtSim and vtSim.VIEW where data-transfer is achieved via files. SUMO in particular already takes an extensible approach to simulation whereby the core software is written in C++ but additional tools provided are written in Java or Python, and an online control interface for external applications is provided [5]. For 3D rendering, the open-source jMonkeyEngine [8] is used, which comes with some free models including the car object used. It also allows the display of all kinds of 3D models compatible with the free modeling software Blender [9], such as other objects that were acquired from the commercial model library TurboSquid [10].

### 8.3.4 Visualization

The conversion from the network representation to a (currently motorway-specific) 3D graphic is performed by first building a wireframe model based on the network description. Connections to other links and the number of lanes are taken into account. Figure 8-2 shows the construction of a road section schematically.

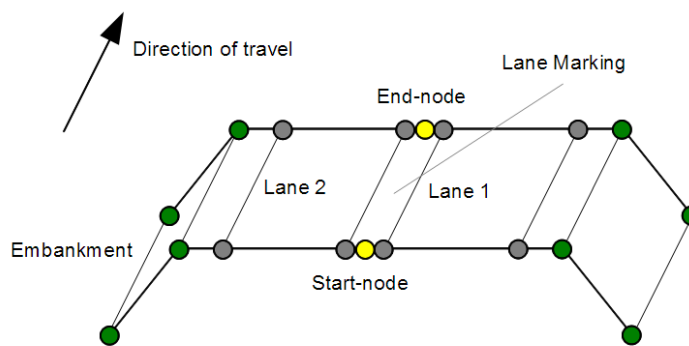


Figure 8-2: Schematic diagram of road link construction.

The edges of roads are augmented with a hard shoulder and curbing. Empty land is shown as grass featuring undulating hills and the sky is textured. Vehicles are shown with detailed vehicle models. The size of the network displayed by vtSim.VIEW is only limited by system memory. Efficient culling techniques in the jMonkeyEngine compute only the geometry which is actually visible on the screen.

In addition to the road geometry, vehicular communication and vehicle status can also be shown. A table of features currently available is shown below.

Table 8-1: Entities that can be visualized with the software.

Entity	Creation mechanism/relevant XML
Road	<edge> elements in edge definition XML
Road boundaries	Automated
Lane Markings	Automated based on road information
Vehicles	<vehicle> elements in vehicles definition XML; positions described by <timestep> elements per time step
Landscape	Automated grass & hills
Sky	Automated
Communication	<warning> element within a <timestep>
Vehicle Status	<status> element within a <timestep>

## 8.4 Usage

### 8.4.1 Control

Currently, navigation is accomplished with keyboard controls. A number of (starting) camera positions can be stored and switched between using numbered keys or an on-screen control (see Figure 8-3, below). This can include the 2D view. Other on-screen controls are also shown in Figure 8-3.



Figure 8-3: Controls for the visualization: the current time (shown on extreme left) can be freely selected using the slider on the left. Speed can be altered using the speed slider. The simulation can be paused at any time using the pause button, and the pre-defined camera positions can be used by clicking "Switch Camera".

### 8.4.2 Use cases

So far, the software has been used for simulations with and without vehicular communication in the research projects simTD [11] and SOCIONICAL [12], using source data from the VISSIM [7] simulator and data derived from video recordings. In simTD in particular the focus was communicating to the viewer what traffic situation required the communication application, and what the driver of the vehicle would see in such a situation. This can be seen in the driver viewpoint, in Figure 8-4 below. Visualization of communication data for a vehicular communication application can be seen in Figure 8-5. Visualization of data gathered from video is shown in Figure 8-6.



Figure 8-4: Driver view while approaching a work zone.



Figure 8-5: Vehicles receiving warnings about a traffic jam; communication shown.

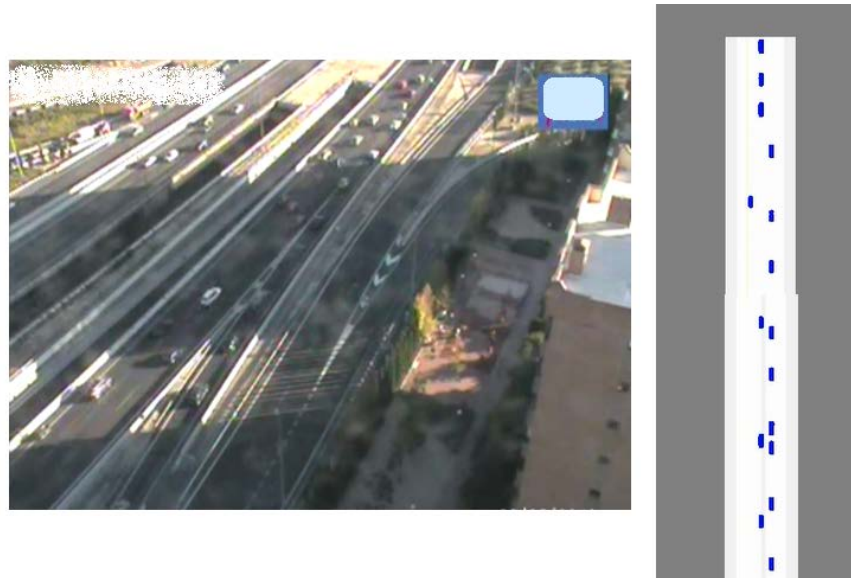


Figure 8-6: Data taken from video (merging and right-hand-lane vehicles only, left frame) animated in the 2D view (right frame).

## 8.5 Conclusions and Outlook

The creation of a tool for generic, realistic 3D visualization based only the simulation geometry, vehicle trajectories and (optionally) vehicular communication data has rapidly accelerated the process of creating life-like 3D material for exploration and display. The software is aimed at those in research and industry who require realistic visualizations that do not need to exactly reflect the exact geography of the modeled location. Likewise the software can add realism to generic simulations (e.g. vehicular communication system prototyping) that are not based on any real road network. As described in section 8.4, we have already used the software to great effect. By using a well-described, XML-based data format, the software is not specific to any one data source, simulated or otherwise.

We are currently extending the software to handle urban road networks and scenery, traffic lights, pedestrians, cyclists and roads that cross over one another (e.g. motorway junctions). To this end we will consider using information beyond edges and their shapes (which are

normally enough for a motorway network) such as the shapes of junctions in the simulation. As not all necessary information can always be gleaned from the simulation data, an editor will be created where necessary information for the visualization can be annotated and stored.

## 8.6 References

- [1] Barceló, J.: Fundamentals of Traffic Simulation. Springer, Heidelberg (2010)
- [2] Härrä, J., Cataldi, P., Krajzewicz, D., Blokpoel, R. J., Lopez, Y., Leguay, J., & Bieker, L.: Modeling and simulating ITS applications with iTETRIS. In: Proceedings of the 6th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks, pp. 33--40. ACM, New York (2011)
- [3] Rieck, D., Schünemann, B., Radusch, I., Meinel, C.: Efficient traffic simulator coupling in a distributed V2X simulation environment. In: Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques, Article No.72. ICST, Brussels (2010)
- [4] Baur, M., Fullerton, M., Busch, F.: Realizing an Effective and Flexible ITS Evaluation Strategy Through Modular and Multi-Scaled Traffic Simulation. Intelligent Transportation Systems Magazine. 2, 34--42 (2010)
- [5] Krajzewicz, D.: Traffic Simulation with SUMO - Simulation of Urban Mobility. In: Barceló, J. (ed.) Fundamentals of Traffic Simulation. International Series in Operations Research and Management Science, pp. 269--294. Springer, Heidelberg (2010)
- [6] SUMO User Documentation - Networks/Building Networks from own XML-descriptions, [http://sumo.sourceforge.net/doc/current/docs/userdoc/Networks/Building\\_Networks\\_from\\_own\\_XML-descriptions.html](http://sumo.sourceforge.net/doc/current/docs/userdoc/Networks/Building_Networks_from_own_XML-descriptions.html)
- [7] Fellendorf, M., Vortisch, P: Microscopic Traffic Flow Simulator VISSIM. In: Barceló, J. (ed.) Fundamentals of Traffic Simulation. International Series in Operations Research and Management Science, pp. 63--93. Springer, Heidelberg (2010)
- [8] jMonkeyEngine, <http://www.jmonkeyengine.com>
- [9] Blender, <http://www.blender.org>
- [10] TurboSquid, <http://www.turbosquid.com>
- [11] simTD, - Safe and Intelligent Mobility Test Field Germany, <http://www.simtd.de/index.dhtml/-/enEN>

- [12] Complex socio-technical system in ambient intelligence (SOCIONICAL),  
[http://cordis.europa.eu/search/index.cfm?fuseaction=proj.document&PJ\\_RCN=1039312](http://cordis.europa.eu/search/index.cfm?fuseaction=proj.document&PJ_RCN=1039312)  
8