

Evaluation of Road Design through Driving Simulation — a Technical Review

Matevž Jekovec*, Gregor Kralj†, Klemen Trontelj‡, Klemen Novak‡ and Sašo Tomažič§

*University of Ljubljana / Faculty of Computer and Information Science, Ljubljana, Slovenia, Email: matevz.jekovec@fri.uni-lj.si

†PNZ d.o.o., Ljubljana, Slovenia, Email: gregor.kralj@pnz.si

‡NERVteh d.o.o., Ljubljana, Slovenia, Email: {klemen.trontelj,klemen.novak}@nerv-teh.com

§University of Ljubljana / Faculty of Electrical Engineering, Ljubljana, Slovenia, Email: saso.tomazic@fe.uni-lj.si

Abstract—Road design is a complex task and despite a rich set of skills and road regulations, a number of design questions remain unanswered until after the road has been constructed. Our aim is to offer civil engineers an evaluation of a new road segment on a driving simulator early in the design phase. In this paper we determine the practical workflow for exporting the road design produced by civil engineers to a driving simulator. While industrial grade software focuses on a feasible, cost-effective road design from the construction point of view, a simulator requires, among others, an optimized 3D model for performance, compelling visual details and semantic road information for autonomous traffic. In the workflow we explicitly use open source or freely available, cross platform, and scriptable tools and formats. Finally, we perform a showcase where test subjects evaluated the new road segment using the state of the art 4 DOF hardware simulator, an eye tracker, and biometric sensors to determine the driver's stress level and response time.

Index Terms—Road design, 3D modeling, 3D file formats, Semantic Road, OpenSceneGraph

I. INTRODUCTION

Road design is a complex task involving numerous teams with different focuses. Despite a rich set of skills and road design regulations (e.g. the Road Act of Slovenia [1]), there is a number of questions and details which remain unsorted until the road has actually been constructed. We focus on filling this gap by offering architects and civil engineers an evaluation of a new road segment on a driving simulator early in the design phase. As a result, we could determine any non-obvious safety flaws a road segment might hide and reduce potential costs in the future [2, pp. 4].

In the last decade, architects and civil engineers are more and more acknowledging the BIM methodology. BIM stands for Building Information Modeling and is a digital representation of physical and functional characteristics of a facility. A BIM is a shared knowledge resource for information about a facility forming a reliable basis for decisions during its life-cycle; defined as existing from earliest conception to demolition [3]. For example, an outer wall of a residential building can be viewed from at least three perspectives: the designer's, structural engineer's, and energy engineer's view [4]. In practice all three actors use their own software to design a wall. If a software is BIM ready, it supports interchangeable file formats and collaborative tools for sharing the 3D model (3D BIM), project time plan (4D BIM), cost dimension (5D

BIM) and finally, building lifecycle management (6D BIM) [5]. Originally, BIM was developed for designing buildings. Nowadays, it is used for diverse physical infrastructures including water, electricity and communications, roads, bridges, ports etc.

The goal of this survey is to determine a practical workflow for exporting the road design produced by civil engineers in modern, BIM ready software to a driving simulator. Industrial grade software for civil engineers focuses on specific tasks regarding the road design. A driving simulator on the other hand requires an optimized, yet visually compelling 3D model. On top of that, 3D model requires semantic information on exact lane location, direction, speed limit, motorway exits and crossings, traffic lights and other details required by the autonomous traffic.

The structure of this paper is the following. In the second section, we introduce the software and file formats used in our survey. In the third section, we present our approach step-by-step and point to limitations and issues of the current software and file formats. In the fourth section, we report issues noticed by test subjects during the showcase of the new road segment. We conclude in the fifth section.

II. 3D MODELING AND FILE FORMATS

Civil engineers typically use proprietary 3D modeling software of their choice. Tables I and II describe 3D software and file formats used in our survey respectively.

While determining optimal workflow, we tried to explicitly use open source, cross platform, and scriptable tools and formats. This allows deployment of the workflow to numerous environments, e.g. to an automated-testing cloud running arbitrary operating system. In one case however, no adequate open source alternative could read 3D models stored inside .DWG format, so we used a cross-platform freeware software Draftsight.

⁷by using osgexport plugin <https://github.com/cedricpinson/osgexport>

⁸by using OSG Max exporter <https://sourceforge.net/projects/osgmaxexp/>

¹<https://www.autodesk.com/products/civil-3d>

²<http://www.cgs-labs.com/Software/Plateia.aspx>

³<https://www.3ds.com/products-services/draftsight-cad-software/>

⁴<https://www.blender.org/>

⁵<https://inkscape.org>

⁶<http://www.oktal.fr/en/automotive/range-of-simulators/software>

Software	Used for	License	Supported platforms
Autodesk Civil3D 2017 ¹	Civil engineering design	Commercial	Windows
CGS Labs Plateia 2017 ²	Roadway design & reconstruction as a Civil3D extension	Commercial	Windows
Dassault Software Draftsight 2017 ³	2D CAD software with basic 3D support	Freeware or Commercial	Windows, macOS, Linux
Blender ⁴	3D modeling	Open Source	Windows, macOS, Linux
Inkscape ⁵	2D illustration	Open Source	Windows, macOS, Linux
Oktal ScanerDT ⁶	Driving simulation	Commercial	Windows

TABLE I
SOFTWARE USED IN OUR SURVEY.

Format	Used for storing	Import/Export by	Specification/Encoding
.DWG	Native Autodesk AutoCAD and Civil3D models excl. textures	Autodesk AutoCAD and Civil3D, Draftsight	Proprietary/binary
.DXF	2D/3D models excl. textures	Autodesk software, Blender, and other CAD and 3D modeling software	Open/text or binary
.FBX	2D/3D models incl. textures	Autodesk software, Blender	Proprietary/text or binary
.SVG	Vector 2D drawing	Inkscape, Adobe software, Blender, web browsers	Open standard/XML
.OBJ	2D/3D models excl. textures	Autodesk 3D Studio MAX, Blender, and other 3D modeling software	Open/text
.MTL	Materials for .OBJ format	Autodesk 3D Studio MAX, Blender and other 3D modeling software	Open/text
.blend	Native Blender models, optionally incl. textures	Blender	Open/binary
.OSGT	OpenSceneGraph primitives excl. textures	Blender ⁷ , Autodesk 3D Studio MAX ⁸	Open/text
.OSGB	OpenSceneGraph primitives excl. textures	Blender ⁷ , Autodesk 3D Studio MAX ⁸	Open/binary
.IVE	OpenSceneGraph primitives optionally incl. textures	Blender ⁷ , Autodesk 3D Studio MAX ⁸	Open/binary
.RND	Semantic road information	Oktal ScanerDT	Open/RoadXML

TABLE II
FILE FORMATS USED IN OUR SURVEY.

III. EVALUATION

We designed a road model of a motorway segment near exit “Novo Mesto — Vzhod” in Eastern Slovenia. The model consisted of a 5 km long motorway with an exit in the middle for each direction. Exits join the main road to the south-east. The 3 km long main road includes 2 smaller roundabouts, one leveled roundabout, and ≈ 150 m long bridge. Figure 1 shows a top view of the road situation.

In the next 4 subsections, we describe steps made to design and prepare a new road segment for the driving simulator. The first step was performed by civil engineers, in our case using Civil3D software with Plateia extension. In the second step we imported models to Blender, performed various optimizations and added details important for the driving simulator. In the third step, we imported the 3D model to ScanerDT simulator and we added semantic road information. In the fourth step, a scenario for the showcase was prepared.

A. Step 1: Road design (Civil Engineers)

The 3D model of a road was designed in Civil3D. Initially, for the earth’s surface, we imported 4 km² of LIDAR points in 1 m \times 1 m resolution from the Surveying and Mapping Authority of the Republic of Slovenia (GURS) publicly available from GURS website. The point cloud was imported to Civil3D



Fig. 1. Top view of a road segment (in orange) for evaluation in the simulator. Points S1 and S2 denote starting positions and E the end point of the driving scenario.

in .XYZ ASCII format and a mesh was generated (see Figure 2). On top of this mesh proprietary aerial photos were mapped.

With the aid of Plateia for Civil3D we designed the road in the following steps:

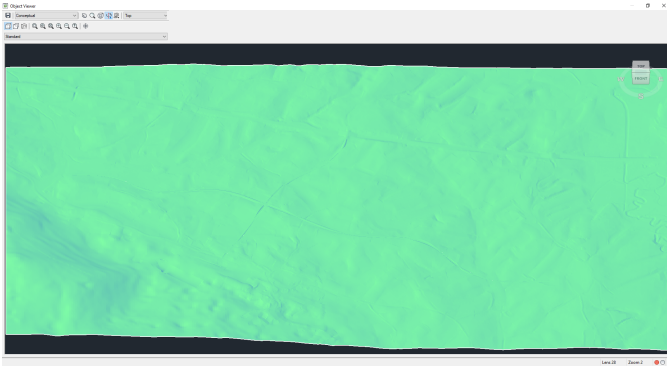


Fig. 2. 3D model of existing terrain obtained by LIDAR.

- 1) 2D alignment to the terrain situation,
- 2) projection of the road to the 3D terrain,
- 3) longitudinal profile,
- 4) cross section and superelevation (see Figure 3), and
- 5) cuts and embankments (see Figure 4).

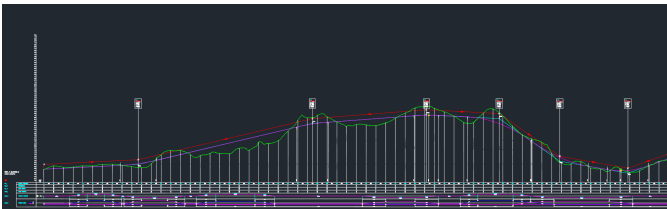


Fig. 3. Longitudinal profile of a road.

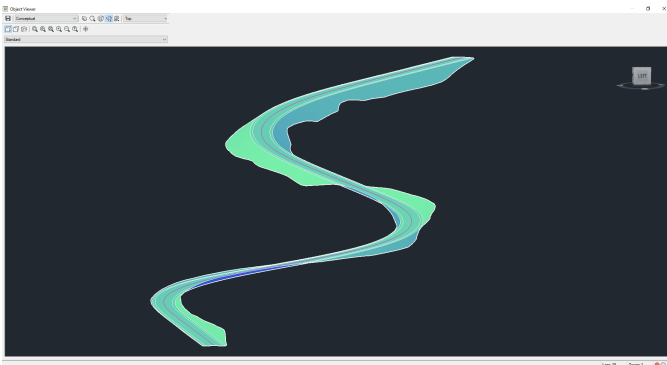


Fig. 4. 3D model of the road surface including cuts and embankments.

Finally, vertical and horizontal road markings were designed using predefined line widths and styles according to the Slovenian legislation. Road markings were placed in multiple phases over a period of six months and a number of people was involved, each person working on its own road segment.

We tried to export the 3D road model to .FBX, but Civil3D took several hours to export and unfortunately crashed. Then, we only exported 3D polylines of road edges, shoulders, and banks to interchangeable .DXF format. Export worked fine, but

since .DXF does not preserve Bezier curves, the resolution was too coarse for our case and we weren't able to set the .DXF granularity in any of the settings panes. Finally, we exported 3D polylines to native .DWG format. We also exported vertical and horizontal road markings to separate .DWG files. The earth's surface with textures was successfully exported to an .FBX format. The export of the terrain surface from Civil3D to .FBX took roughly 4 hours.

B. Step 2: Road modeling in Blender

In the second step we first imported the terrain in .FBX format to Blender. The import took about an hour for ≈ 240 MB large file excluding textures. In Blender, the terrain surface was optimized using the Decimate modifier keeping roughly 1% of the vertices. The original texture resolution was preserved to maintain the sense of speed while driving.

In order to import the 3D model of a road, we opened the .DWG file containing 3D polylines by using freely available DraftSight software. In DraftSight, we exported the drawing to ASCII .DXF format 2013 and imported it to Blender. Resolution of curves was improved by enlarging the drawing by a factor 100. In Blender, we manually recreated the road surface, cuts and embankments using the 3D polylines as base.

Vertical road markings stored in .DWG format were also opened in DraftSight. Each road sign was exported separately to vector .SVG format. Road signs were then opened in Inkscape and exported to a .PNG image to be used as a texture.

We also used DraftSight to open a file with horizontal road markings and we exported them to a single 2D .SVG file. Although Blender supports importing .SVG images directly, it does not support different brush styles and line widths. Therefore, we first imported markings to Inkscape and converted strokes to paths. At this point, we noticed a number of garbled markings (see Figure 5). We suspect this was a consequence of multiple people working on same road marks over a longer period of time using different approaches and tools.

When the horizontal markings were cleaned, we extruded the markings to form a 3D mesh and intersected it with the road surface (see Figure 6). During this process, it turned out that Boolean modifier in Blender works best when intersecting two 3D meshes i.e. objects having volume. Therefore, we also extruded the road surface beforehand to obtain volume. This was also needed to correctly compute mesh normals. Finally, from the resulting intersected mesh, only faces facing up were kept and were lifted for 5 cm.

For the motorway, we added concrete guardrails separating opposite driving lanes and either steel guardrails or poles with reflectors following the outer path of a road surface with 1 m offset. Also, five steel crumple zones were added. A game fence was added around the motorway. The fence was designed simply as a plane assigned a transparent grid-like texture. We used a set of freely available textures for asphalt, concrete, grass and sand.

Finally, we cut all continuous road objects (road surface, horizontal markings, guardrails) on ≈ 500 m long segments.

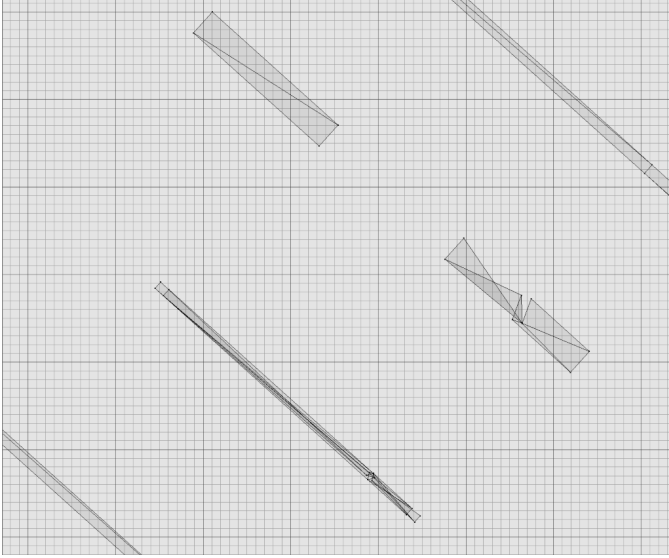


Fig. 5. Garbled road markings — a consequence of a rough projection to a 3D surface in Civil3D. Some markings (e.g. the one on top) were projected fine, others were garbled (e.g. the remaining two markings).



Fig. 6. Optimized model in Blender: decimated terrain surface, added road markings, guardrails with reflectors and supporters, deer fences, and crumple zones.

This enabled occlusion of objects in driving simulator and improved performance.

We exported the whole 3D model to .OSGT format using Blender with osgexport plugin and then to .IVE format with embedded textures by using `osgconv -O includeImageFileInIVEFile` switch. Initially we had issues that the resulting .IVE file wasn't readable by ScanerDT. This was due to a version mismatch of OpenSceneGraph library and tools installed on originating and target computers. The .IVE file was 292 MiB large including textures and had roughly 1.3 M triangles and 1 M faces.

C. Step 3: Road import to ScanerDT

In the third step, we imported the model into ScanerDT simulator. By using the Terrain tab of ScanerDT framework, we manually applied the logical lane connections so that traffic can drive over the provided road. Applying road connections was a daunting and consuming task, since we needed to precisely align the lanes with the road markings. The road description included start and end track coordinates with altitude and slope, crossings, number and type of lanes, traffic regulations (speed, allowance), and placed signs direction.

Finally, we exported the RoadXML .RND file containing the logical description of the road network.

D. Step 4: Scenario preparation in ScanerDT

In the last step, we combined the 3D model in .IVE format and semantic road information in .RND format and designed the following combination of situations resulting in 8 different scenarios:

- starting point S1 or S2 as depicted on Figure 1,
- light traffic or heavy traffic,
- sunny day or a fog.

Each driver was asked to leave the motorway and travel towards direction “Maline”. Ideally, when a driver left the motorway, he needed to make either one or two correct turns and needed to take two or three correct exits in roundabouts starting in S1 or S2 respectively. At point E on Figure 1 the scenario ended.

IV. PROOF OF CONCEPT SHOWCASE

As a showcase we used NERVteH simulator⁹ already well established in numerous surveys regarding driver responses and computer-human interactions [6]–[10]. The simulator consisted of 4 DOF motion seat, a steering wheel including the handles, three pedals, gear shifter and a handbrake. For visualization we used three TV screens. Our evaluation was based on data collected from Oktal ScanerDT and devices used to collect biometry data: Tobii Pro Glasses 2 eye tracker¹⁰ for obtaining the driver's gaze and Empatica wristband E4¹¹ for measuring heart rate, galvanic skin response, skin temperature, and hand movement.

A dozen of drivers tested the new road segment in a random scenario designed in the previous section. During the test, the worst case frame rate of the simulation was ≥ 40 FPS. All telemetry data was correctly received. For each frame,

⁹<https://www.nerv-teh.com>

¹⁰<https://www.tobii.com/product-listing/tobii-pro-glasses-2/>

¹¹<https://www.empatica.com/research/e4/>

we mapped driver's gaze to the 3D scene (see Figure 7) and computed which object was a driver focused on and for how long.



Fig. 7. Driver's gaze during the evaluation of a road segment.

During the showcase, drivers and audience mainly consisting of civil engineers reported the following smaller caveats:

- One driver starting from S1 made a wrong turn in a fog after he left the motorway. He claimed he had not noticed a sign above the highway in time because the canopy was too low.
- Audience noticed that the emergency lane was missing on a motorway in one direction. This bug appeared during the recreation of a road surface from 3D polylines since our team was not aware which polyline corresponded to a driving lane and which polyline to the emergency lane.
- Audience also noticed that the length of the broken line between lanes is too small. This bug appeared during the horizontal road markings design by civil engineers. On a motorway this distance is larger and they accidentally took broken line style of the regional road.

Otherwise, no major drawbacks regarding the simulation were observed and test subjects were eager to evaluate more road segments in the future.

V. CONCLUSIONS

In this paper we determined a practical workflow for exporting a 3D road model from industry-grade CAD software to a driving simulator. We provided an overview of file formats and either open source or freely available cross-platform tools used in the process. During the survey, we successfully resolved issues mainly originating from the large scene. Autodesk software used by civil engineers proved, despite being BIM ready, to offer unusable slow export of curved surfaces to de-facto interchangeable .FBX format. Therefore we only exported 3D polylines of the road and recreated them in Blender afterwards.

In the future, we should further simplify and optimize the workflow. A task of recreating road surfaces, cuts and embankments should be done solely using a software recognized by civil engineers, and successfully exported to an

interchangeable 3D format. Recreating them in Blender added additional cost to the project and even introduced the missing emergency lane bug described in the previous section. Finally, during the road design phase, exact lane position and direction should be determined and exported to one of the semantic road formats (e.g. RoadXML or OpenDrive) separately from the 3D model. Perhaps, this information could even be part of BIM. Eventually, this would significantly reduce time needed to prepare the model for driving evaluation.

REFERENCES

- [1] Republic of Slovenia, "Zakon o cestah (ZCes-1)," 2011. [Online]. Available: <http://www.pisrs.si/Pis.web/pregledPredpisa?id=ZAKO5788>
- [2] The Construction Users Roundtable, "Collaboration, Integrated Information, and the Project Lifecycle in Building Design, Construction and Operation," CURT, Tech. Rep., 2004. [Online]. Available: <https://kcuc.org/wp-content/uploads/2013/11/Collaboration-Integrated-Information-and-the-Project-Lifecycle.pdf>
- [3] National Institute of Building Sciences, "What is a BIM? Frequently Asked Questions About the National BIM Standard-United States™," 2014. [Online]. Available: <https://web.archive.org/web/20141016190503/http://www.nationalbimstandard.org/faq.php#\#faq1>
- [4] G. A. van Nederveen and F. P. Tolman, "Modelling multiple views on buildings," *Automation in Construction*, vol. 1, no. 3, pp. 215–224, 1992. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S002658059290014B>
- [5] B. Succar, "Episode 22: The Wedge and the S-Curve," 2015. [Online]. Available: <http://www.bimthinkspace.com/2015/02/episode-22-the-wedge-and-the-s-curve.html>
- [6] T. Čegovnik, K. Stojmenova, G. Jakus, and J. Sodnik, "An analysis of the suitability of a low-cost eye tracker for assessing the cognitive load of drivers," *Applied Ergonomics*, vol. 68, pp. 1–11, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0003687017302326>
- [7] M. Vengust, B. Kaluža, K. Stojmenova, and J. Sodnik, "NERVtch Compact Motion Based Driving Simulator," in *Proceedings of the 9th International Conference on Automotive User Interfaces and Interactive Vehicular Applications Adjunct*, ser. AutomotiveUI '17. New York, NY, USA: ACM, 2017, pp. 242–243. [Online]. Available: <http://doi.acm.org/10.1145/3131726.3132047>
- [8] K. Trontelj, T. Čegovnik, E. Dovgan, and J. Sodnik, "Evaluating safe driving behavior in a driving simulator," in *ICIST 2017*, M. Zdravković, Z. Konjović, and M. Trajanović, Eds., vol. 1, 2017, pp. 299–302.
- [9] D. Grogna, K. Stojmenova, G. Jakus, M. Barreda-Ángeles, J. G. Verly, and J. Sodnik, "The impact of drowsiness on in-vehicle human-machine interaction with head-up and head-down displays," *Multimedia Tools and Applications*, apr 2018. [Online]. Available: <https://doi.org/10.1007/s11042-018-5966-9>
- [10] K. Stojmenova, G. Jakus, and J. Sodnik, "Sensitivity evaluation of the visual, tactile, and auditory detection response task method while driving," *Traffic Injury Prevention*, vol. 18, no. 4, pp. 431–436, 2017. [Online]. Available: <https://doi.org/10.1080/15389588.2016.1214868>