

TOWARDS WIND-SIMULATION OF VIRTUAL 3D CITY MODELS IN A COLLABORATIVE VR ENVIRONMENT

R. Piepereit^{1*}, A. Beuster², M. von der Gruen¹, U. Voß¹, M. Pries², U. Wagner²

¹ Hochschule für Technik Stuttgart - University of Applied Sciences, Faculty C, Schellingstraße 24, 70174 Stuttgart, Germany - (raul.piepereit, ursula.voss, maximilian.vondergruen)@hft-stuttgart.de

² Beuth Hochschule für Technik Berlin - University of Applied Sciences, Department II, Luxemburger Straße 10, 13353 Berlin, Germany - (angelika.beuster, pries, ute.wagner)@beuth-hochschule.de

Commission VI, WG VI/10

KEY WORDS: 3D City models, CityGML, Healing, Automated Processing, CFD, Virtual Reality

ABSTRACT:

Virtual reality (VR) technologies are used more and more in product development processes and are upcoming in urban planning systems as well. They help to visualize big amounts of data in self-explanatory way and improve people's interpretation of results. In this paper we demonstrate the process of visualizing a city model together with wind simulation results in a collaborative VR system. In order to make this kind of visualization possible a considerable amount of preliminary work is necessary: healing and simplification of building models, conversion of these data into an appropriate CAD-format and numerical simulation of wind flow around the buildings. The data obtained from these procedures are visualized in a collaborative VR-System. In our approach CityGML models in the LoD (Level of Detail) 1, 2 and 3 can be used as an input. They are converted into the STEP format, commonly used in CAD for simulation and representation. For this publication we use an exemplary LoD1 model from the district Stöckach-Stuttgart. After preprocessing the model, the results are combined with those of an air flow simulation and afterwards depicted in a VR system with a HTC Vive as well as in a CAVE and a Powerwall. This provides researchers, city planners and technicians with the means to flexibly and interactively exchange simulation results in a virtual environment.

1. INTRODUCTION

The use of Virtual Reality (VR) techniques in product development processes in the automotive and aviation industry, but also in urban planning, has increased. The aim is to realistically visualize product, tool and environmental models, as well as to accurately simulate product functionalities and production processes. Achieving this goal becomes more and more feasible, as costs have decreased and processing speeds, as well as data storage capabilities of computers have increased.

The presented project is concerned with the process of visualizing a 3D city model together with wind simulation results in a collaborative VR system. This involves not only the visualization in a VR environment, but also the simplification of the necessary preprocessing of virtual buildings for a numerical simulation, as well as the simulation itself.

Virtual building models are generated mainly by aerial laser scanning and dense image matching. For geo-information systems (GIS) several data formats are in use, e.g. 3D-shapefile, dxf and CityGML. CityGML is an explicit XML-based exchange format for 3D city models as defined by the Open Geospatial Consortium (OGC). It supports not only geometric descriptions of 3D city model components, but also semantic and topological information. As this information needs to be consistent, the CityGML standard defines additional requirements, which can be checked and in some cases automatically modified accordingly, by systems such as cityDoctor, if possible (Wagner et al. 2015).

The application of virtual 3D city models is very versatile. It ranges from simple visualizations to complex numerical simulations (Biljecki et al., 2015). Different applications make different demands on 3D city models. Some of them, such as visualization and planning, do not have high demands on the geometry, whereas others, such as numerical simulation methods, have specific requirements on data formats and geometric parameters.

The numerical simulation of fluids, for example, is state-of-the-art in engineering disciplines and a number of commercial tools for computational fluid dynamics (CFD) already exist. However, mainly designed for engineering applications, commercial CFD tools require computer-aided design (CAD) data formats which are used in these disciplines. Hence, a conversion from the GIS data format to a CAD data format is necessary first. It is not sufficient to transfer only points. Additional geometrical and topological information, e.g. surface groups and solids representation, is necessary as well. This process makes additional demands on the 3D city model, e.g. they have to be watertight and without self-intersection.

In most cases 3D city models still cannot be used in CFD simulations, even if they are transferred to a CAD format and are without errors. One reason for this can be short edges or small angles in a building's geometry, which may lead to an unsatisfactory CFD mesh, resulting in non-converging solutions or unacceptably high computing time. Therefore, in addition to the healing process, city models may need to be further simplified. In order to reduce the high manual effort for the

* Corresponding author

simplification we are working on automated simplification algorithms.

For the joint representation of the model and simulation results, the collaborative VR system VENTUS (cf. VENTUS, 2019) is used for the first draft as described in this paper. The original data model was loaded into this system together with the wind simulation results generated in ANSYS Fluent.

The implementation of VR technology leads to a superior and stimulating representation of error detection, degree of reality and work environment (cf. Israel et al. 2009; Spiegel, 2016; Knöpfle, 2003). Using the VENTUS system, a method for visualizing the results of wind simulations within virtual city models in a VR environment was developed, in order to investigate and discuss the added value of a stereoscopic visualization of these results.

2. RELATED WORKS

The realistic representation of product concepts or tools, for example, is of great importance for development and planning processes in several branches of industry. It helps the customers to grasp an idea, but also the developers to achieve the desired results. Some of the currently most influential factors on these processes are an increasing complexity of products and services, interdisciplinary and location-independent teams, the involvement of customers in development processes, constantly increasing demands on flexibility and ever shorter development cycles (Schmucker et al. 2015).



Figure 1. Urban planning use case (© Berlin Partner für Wirtschaft und Technologie GmbH, VENTUS).

With these circumstances in mind, VENTUS was developed as a joint project of the Hochschule für Technik und Wirtschaft Berlin and the Beuth Hochschule für Technik Berlin and with the involvement of partners from industry. It provides an infrastructure that enables heterogeneous user groups to exchange models and simultaneously view and edit them in a virtual environment. Within the scope of the VENTUS project cases of application from two different areas are examined – on the one hand CAD modelling within a product life cycle, on the other hand the use of virtual city models available on the market in VR applications. VENTUS enables collaborative work with different immersive technologies (e.g. head-mounted displays, CAVEs or powerwalls) as well as conventional monitors or smartphone displays (see figure 1). In addition to the direct transfer of model data via application specific standard formats

(e.g. STEP for CAD models or CityGML for virtual city models) another focus is to design a cost-effective and data secure solution. (cf. Tag et al., 2018)

3. HEALING OF VIRTUAL CITY MODELS

In order to use a 3D city model, healing is usually necessary to fulfill data quality requirements, e.g. the CityGML standard. A typical quality deficit are gaps between neighboring faces. Depending on the precision of visualization, these gaps are difficult to locate and often only traceable with the help of diagnostic programs. Other examples are faces or curve segments that are too small, overlapping geometry elements, constrictions or bending of faces.

Figure 2 shows a non-valid building model with such quality deficits.

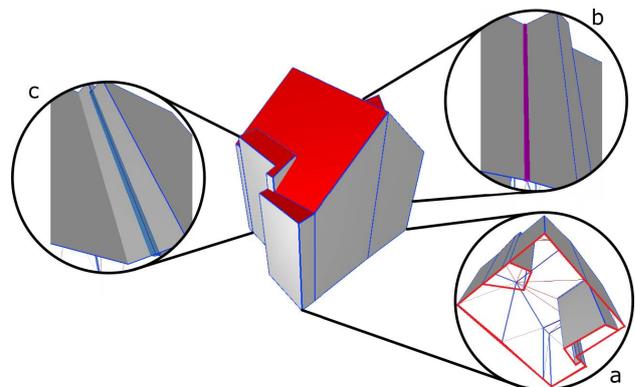


Figure 2. Example of a building model with quality deficits a) missing ground surface, internal polygons b) multiply used edges (an edge is shared by more than two faces) c) self-intersection

To detect these various errors, the city model has to be properly analyzed and run through a validation process, which includes two steps. First a schema validation has to be performed, followed by a geometric and semantic validation of the data set. Some of the geometry criteria are: watertight volume, manifoldness, planar faces, duplicated vertices etc.

Many projects such as CityDoctor (cf. Wagner et al. 2015) or CityGML Quality Interoperability Experiment (cf. Coors, Wagner 2015) concentrate on creating automated procedures which validate and identify sources of errors in city models. Simple errors can be corrected automatically using these tools. For the model of this paper, for example, missing polygons of the ground surfaces were automatically created. Other errors still need manual inspection and correction.

The healing process ensures the correct functionality of the algorithms necessary for preprocessing and lays the foundation for achieving the highest level of automation possible.

4. SIMPLIFICATION OF VIRTUAL CITY MODELS FOR AIR-FLOW SIMULATIONS

A model may formally fulfill all the quality criteria required by the GML standard, however, certain geometry details or properties may prevent further processing for CFD simulations.

In addition to the required healing of the model, application-specific preparation is needed.

For CFD, short edges or sharp angles in a building's geometry lead to an unsatisfactory mesh, because they lead to small and/or distorted mesh cells. These in turn require a small time step, which leads to high computing time or may even result in non-converging solutions.

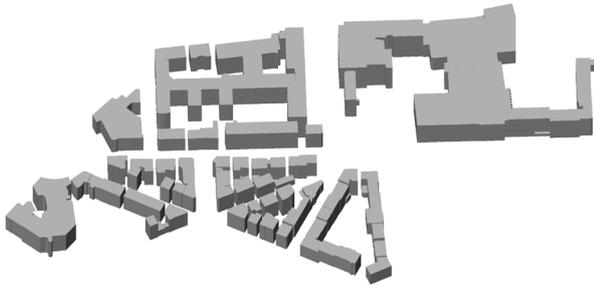


Figure 3. Virtual 3D city model of a building block in Stuttgart (source: Stadtmessungsamt Stuttgart).

Listed below are outlines of the four algorithms developed for and applied to the model described in this paper (see figure 3):

Merge Edges and Faces:

An easy but very important way to simplify building models is to merge the edges and faces respectively, which are connected to each other and are, within a certain threshold, coplanar.

Boolean Operations:

The basic operations of Boolean algebra are conjunction, disjunction and negation. For our purposes we only need conjunction. CGAL's union-operator can be used to merge buildings and building parts that are connected. As a result, other algorithms can be applied to more than one single building at a time, e.g. several buildings of a whole city block. Combing the Boolean union-operator with e.g. Minkowski operations or offset surfaces makes it possible to close gaps and holes within buildings. The Minkowski sums, like the Boolean operations, are implemented in CGAL and are well documented by Hachenberger, Kettner (2019) and Hachenberger (2019).

Sweep-plane Algorithm:

The sweep-plane algorithm (Pieperit et al., 2018) iteratively eliminates edges with a length below a predetermined threshold by moving corresponding faces along their normal direction into or out of the building. This algorithm allows for the specific elimination of small elements as they occur in offsets, bulges or corner offsets, for example. Building features, such as doors, windows or chimneys can be erased in this manner as well.

Coons Algorithm:

Round surfaces of buildings in CityGML are approximated by a great number of narrow polygons. These polygons are used as a basis for the mesh resulting in mesh cells that are at least as narrow as the polygons. Working with freeform surfaces instead enables the meshing tool to adjust the size of the mesh elements more freely. An approach to use the method of Coons in order

to replace these polygons with freeform surfaces, such as B-Splines, is described in Pieperit et al. (2016).

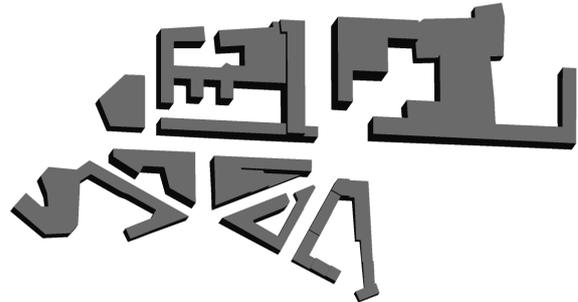


Figure 4. Healed and simplified city model of a building block in Stuttgart.

For all these algorithms to work, the CityGML model has to be converted into a CAD BRep data structure. Figure 4 shows the healed and simplified model that is used for the simulation, as described in the following section. Even though the algorithms already provide good results, it has not yet been possible to fully automate the simplification process at this point in time. For this reason, the model needed further manual preprocessing.

5. WIND SIMULATION

For this publication we simulate the wind field around some building blocks in Stuttgart, which are depicted in figure 4. As this area is part of hilly ground (the difference between the highest and lowest point within the computational domain is 140 meters), the landscape is also considered in the geometry in order to improve the quality and the significance of the result. Following the recommendations in Franke et al. (2007) we have chosen a computational domain with a height of $30 H_{max}$, a width of $55 H_{max}$ and a length of $60 H_{max}$ (H_{max} denotes the height of the highest building in the domain of interest), resulting in a computational domain of $2.4 \times 2.2 \times 1.2 \text{ km}^3$ (length \times width \times height). The long distance between the inlet and the buildings allows the air flow to develop over the complex terrain. The domain is aligned with the main wind direction, hence the inlet plane is perpendicular to the inflow velocity.

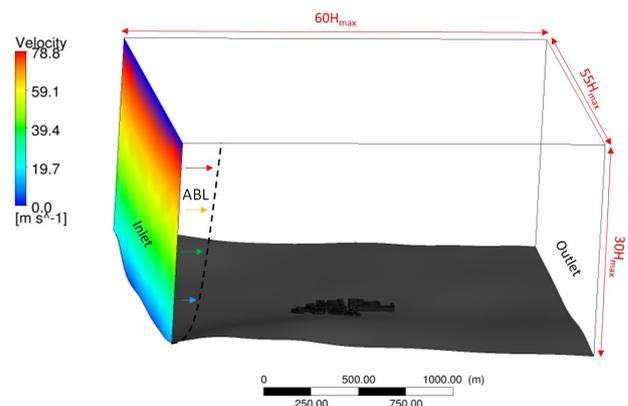


Figure 5. CFD simulation domain. Velocity on vertical line: profile of the atmospheric boundary layer at inlet.

Modelling the Atmospheric Boundary Layer (ABL) at the inlet, the inflow velocity is now specified as a velocity power law. It uses a reference velocity of 3 m/s at the height of 10 m, obtained from measured wind data in Stuttgart (City of Stuttgart, 2018). Symmetry boundary conditions are chosen on the right and left face of the domain as well as on the top, in order to reduce the influence on the flow inside the domain. For the outlet plane a pressure-outlet boundary condition with atmospheric pressure is chosen. The domain with boundary conditions and a schematic sketch of the ABL-profile is shown in figure 5.

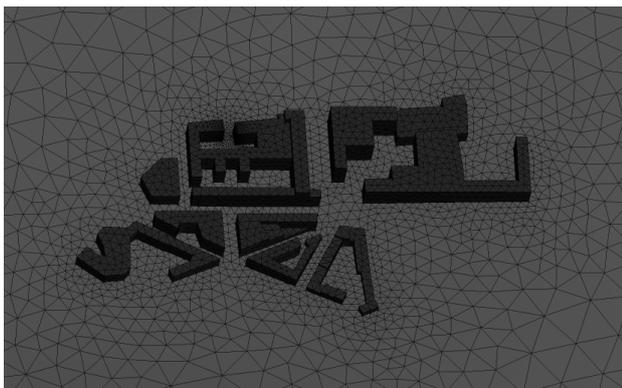


Figure 6. Boundary mesh of buildings.

Next, a suitable mesh has to be generated. For this application, we opted for a minimum cell length of one meter. In order to archive this with at least 2 cells per edge, the preprocessing algorithms were applied to generate a model with a minimum edge length of at least two meters. Figure 6 shows the surface mesh, which is the basis for the volume mesh required for CFD calculations. The mesh is an unstructured mesh, generated by the ANSYS Meshing tool. Since the buildings' surroundings are of special interest for the VR application, the mesh is locally refined along the buildings' surfaces and edges, leading to an even greater amount of mesh cells. As a result, the mesh has a total number of 139076 tetrahedral cells. To capture the boundary-air interactions and to resolve the ABL close to the ground, we use a layer of special hexahedral cells, the so-called inflation layer, at the terrain and the buildings' surfaces.

On this mesh the steady state 3-D Reynolds-Averaged Navier Stokes (RANS) equations are solved using ANSYS FLUENT with the k- ω SST turbulence model.

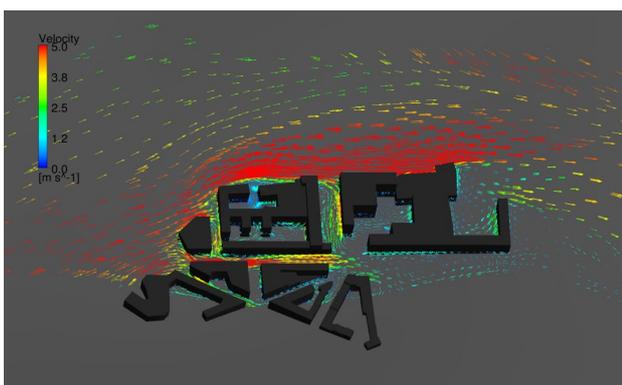


Figure 7. Example of wind velocity at pedestrian level for a building block in Stuttgart. ABL: 3 m/s at 10 m height.

Figure 7 shows the wind speeds at pedestrian level for the processed building blocks. The arrows represent the wind direction and their color the velocity magnitude. The red arrows represent high velocities, which lead to uncomfortable paths for pedestrians. Especially in street canyons, which are oriented towards the main wind direction, the flow is accelerated. Compared to that, in streets oriented perpendicular to the main wind direction the wind speeds are much lower. As can be seen, the low wind velocity or the lack thereof is very comfortable for pedestrians and ideal for outside events inside the courtyards.

6. VISUALIZATION OF THE RESULTS IN A VIRTUAL ENVIRONMENT

The VR system VENTUS (cf. VENTUS, 2019) is used to display the building models and the simulation results in a virtual environment. Thus it is possible to review and assess a simulation's result from different points of view in a 3D representation. Regions of critical wind velocities can be identified more easily and marked using a built-in 3D sketch tool or by attaching annotations. If further processing of the data is needed outside VENTUS, the 3D information of the sketched lines and annotations can be exported as a XML file. The collaboration feature of VENTUS makes it possible for various people, who may be located in different parts of the world, to view the results simultaneously and discuss them.

As VENTUS is based on the game engine Unity3D (Tag et al., 2018), city models can be added to the scene before starting the system or loaded dynamically as identifiable objects during the session. The simulation results of the CFD calculation are depicted as colored 3D arrows. They are loaded into the scene separately, for example street by street, and can be arranged into groups, thereby offering the advantage to label or delete them, and giving users the opportunity to temporarily move sets of arrows aside to have a better view off certain regions in need for inspection.

Further work needs to be done on the settings regarding the management of the amount of arrows, where they will be depicted and at which point in time, since the simultaneous display of the thousands of arrows, that translate into the simulation's result, slow down the VR system too much.

The import of 3D objects into VENTUS is realized by a STEP read-in interface allowing to overtake the assembly structure of CAD data models which can be applied to city models as well.



Figure 8. Wind velocity depicted with Rhino.

The simulation results created by ANSYS are exported as a csv file, which contains point and direction coordinates with corresponding wind velocity values. Using this file, the colored 3D arrows are created as CAD parts and afterwards stored in a STEP file. We use the Rhino C++ API for implementing a plugin of the CAD software Rhinoceros3D (see figure 8). Every arrow is capsuled in a Rhino block to conform to the required assembly structure.

The portrayal of simulation results of a three-dimensional space is often limited to a single plane and can only be viewed from a limited perspective (see figure 7). The representation in virtual reality, however, enables a clear visualization of several of these planes at the same time (see figure 9). This advantage is evident especially with turbulences that are not limited to one plane.

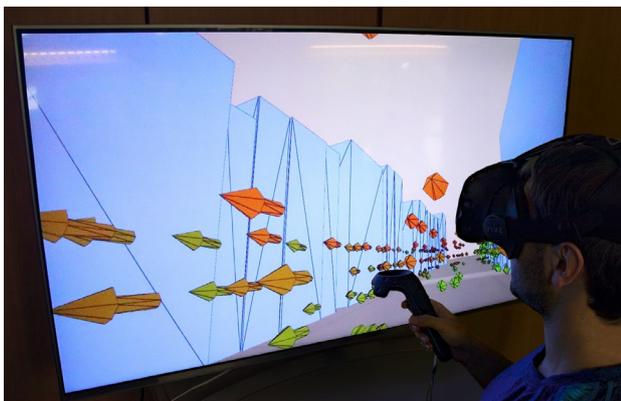


Figure 9. Wind velocity depicted in VENTUS.

7. CONCLUSION

In this paper we demonstrate the process of visualizing a virtual 3D city model together with wind simulation results in a collaborative VR system. CityGML models are converted into a CAD BRep format and afterwards healed and optimized for a numerical wind simulation. Algorithms for automated healing and simplification are used in the preprocessing procedure in order to drastically reduce the manual effort.

However, it should be noted that the sole application of the presented algorithms, without further manual editing, does not lead to a fully processed 3D model ready for simulation purposes as described.

After preprocessing, the results of a wind simulation are generated using ANSYS FLUENT. Together with the initial city model, these then are imported into the VR system VENTUS and depicted in the virtual environment. The use of stereoscopic representations allows for a better understanding of the simulation results, as they are depicted on several planes at the same time and can easily and intuitively be analyzed from different perspectives. Especially in case of complicated turbulences, which do not extend along one plane only, a representation of the simulation results in a virtual environment is of advantage.

In the long run an integration into planning systems is possible and of general interest, as methods of virtual reality find more and more application in this field as well.

ACKNOWLEDGEMENTS

This work has been developed in the project iCity. The project iCity (Funding number: 03FH9I01IA) is supported by the German Federal Ministry of Education and Research (BMBF). The authors are responsible for the content of this publication. They thank Prof. Dr. Volker Coors for the support and the valuable input during the writing process of this publication. The authors further gratefully acknowledge the Stadtmessungsamt Stuttgart for the data of 3D citymodel of Stuttgart.

REFERENCES

- Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., Çöltekin, A., 2015: Applications of 3D city models: State of the art review. *ISPRS International Journal of Geo-Information* 4(4), 2842–2889.
- City of Stuttgart, 2018. Stadtklima: Messstation Stuttgart-Mitte: https://www.stadtklima-stuttgart.de/index.php?klima_messdaten_station_snmz.
- Coors, V., Wagner, D., 2015: CityGML Quality Interoperability Experiment des OGC. *DGPF Tagungsband. Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation eV* 24, 288-295.
- Franke, J., Hellsten, A., Schlünzen, H., Carissimo, B., 2007: Best Practice Guideline for the CFD Simulation of Flows in the Urban Environment: COST Action 732 Quality Assurance and Improvement of Microscale Meteorological Models.
- Hachenberger, P., 2019. CGAL 5.0 - 3D Minkowski Sum of Polyhedra: User Manual, https://doc.cgal.org/master/Minkowski_sum_3/index.html (28. June 2019)
- Hachenberger, P., Kettner L., 2019. CGAL 5.0 - 3D Boolean Operations on Nef Polyhedra: User Manual, https://doc.cgal.org/master/Nef_3/index.html (26. May 2019)
- Israel, J. H., Wiese, E., Mateescu, M., Stark, R., 2009: Investigating three-dimensional sketching for early conceptual design – Results from expert discussions and user studies. *Comput. Graph.*, vol. 33, no. 4, 462–473.
- Knöpfle, C. 2003: Intuitive und Immersive Interaktion für Virtuelle Umgebungen am Beispiel von VR Design Review. Thesis, Technische Universität Darmstadt, Darmstadt.
- Pieperit, R., Schilling, A., Alam, N., Wewetzer, M., Pries, M., Coors, V., 2016: Towards automatic processing of virtual city models for simulations. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences. IV-2/W1*, 39-45, <https://doi.org/10.5194/isprs-annals-IV-2-W1-39-2016>, 2016
- Pieperit, R., Deininger, M., Kada, M., Pries, M., Voß, U., 2018: A Sweep-plane Algorithm for the Simplification of 3D Building Models in the Application Scenario of Wind Simulation. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences. XLII-4/W10*, 151-156, <https://doi.org/10.5194/isprs-archives-XLII-4-W10-151-2018>, 2018

Schmucker, U., Haase, T., Schumann, M., 2015: Digital Engineering and Operation. In *Produktion und Logistik mit Zukunft*, Berlin, Heidelberg: Springer Berlin Heidelberg, 283–375.

Spiegel, R. 2016: Virtual Reality: The New Prototyping Is Digital. *DesignReview*, vol. 71, no. 6, 28–32.

Tag, I., Pries, M., Wagner, U., Israel, J. H., Jung, T., 2018: Geometrische und softwaretechnische Aspekte bei der Implementierung eines VR-Kollaborationssystems. *Proc Workshop 3D-NordOst 2018*, Berlin, Germany, 149-158

VENTUS, 2019. Virtual Environment for Teamwork and ad-hoc Collaboration between Companies and heterogeneous User Groups. <http://ventus3d.com> (25. April 2019).

Wagner, D., Alam, N., Wewetzer, M., Pries, M., Coors, V., 2015: Methods for Geometric Data Validation of 3D City Models. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences* 40. XL-1/W5, 729-735, <https://doi.org/10.5194/isprsarchives-XL-1-W5-729-2015>, 2015