Dreiländertagung der DGPF, der OVG und der SGPF in Bern, Schweiz – Publikationen der DGPF, Band 25, 2016

Semantic 3D City Models Serving as Information Hub for 3D Field Based Simulations

BRUNO WILLENBORG[1], MAXIMILIAN SINDRAM[1] & THOMAS H. KOLBE[1]

Abstract: A major challenge in the context of global urbanization is to make cities "smarter" in order to satisfy their increasing requirements to safety, environmental conditions and quality of living. This requires both the environmental and social processes taking place in cities to be perspicuous and predictable. Today, simulations from various disciplines like wind field or flood simulations, noise propagation, pollutant dispersion or explosion simulations have become essential tools for decision making in urban planning and analytics. A common simulation approach is the "Finite Volume"-method, where the observation area is partitioned into a regular grid of cells to solve partial differential equations numerically. 3D city models serve as reliable representation of the real world objects.

In this study we present an approach for the integration of such field-based simulations with object-based, semantic 3D city models. It allows for automatically deducing a finite volume-based representation of arbitrary objects in 3D city models. The city model is used as an information hub between the systems. It serves both as the data source for the geometry representation of the simulation and its result storage. For this, the corresponding region is divided into equal cubes (voxels) and each voxel is tested against intersection with the city objects. Thus, each voxel inherits the semantics of its intersecting city object, which ensures that a) the voxel holds the characteristics of its counterpart and b) the results of the simulation runs can be aggregated per object and stored back in the city model.

Visualization and result analysis is of crucial importance, since highly tailored simulation tools do not offer interactive access to the simulation results. In contrast, the frameworks for 3d city models provide comprehensive tools for both visualization and intuitive analysis. They allow even for non-expert users to assess the information generated by the simulation tools.

The proposed approach is introduced and evaluated on the example of a blast simulation and implemented using a 3D city model according to the international standard CityGML. It is realized in an Open Source plug-in for the 3D geodatabase software 3DCityDB.

1 Introduction and problem statement

Urbanization is one of the biggest challenges of our time. While today, already more than 50% of the global population lives in urban areas, this figure is expected to increase to over 70% by the year 2050 (CHEN et al. 2012). For this reason, many disciplines deal with the question of how urban regions can be made smarter. The aim of smart cities initiatives is to assess the state of the city as a system and to understand the processes and thus develop tailored and sustainable optimization strategies. For this reason, environmental, ecological and social simulation tools support the urban planners in decision making. The tools require a representation of the simulation area that reflects the real world objects in a most realistic way. Semantic 3D city models have been successfully used in current smart city projects. They serve as reliable source

---

[1] Lehrstuhl für Geoinformatik, Technische Universität München, Arcisstr. 21, 80333 München,
E-Mail [b.willenborg, maximilian.sindram, thomas.kolbe]@tum.de

of information and provide a common information hub with the international standard CityGML as an interoperable representative of semantic 3d city models.

A major challenge is to bring the object based city model together with the field based simulation system in 3D space, as both follow different modelling paradigms. On the one hand, according to the ISO 19109 definition of geographic objects, the city model objects are decomposed following logical criteria which can be observed in the real world (ISO 19109 2005). Their shape, orientation and location in the model is derived from their real world counterparts. All spatial objects carry their inherent semantics. The simulation system on the other hand, divides the observed area into a regular grid of finite volume elements so called volume pixels (voxels) without any thematic information. Real world objects are approximated by an accumulation of these cells.

An essential advantage of semantic 3D city models that is also mandatory for field-based simulation tools is that they provide information about the characteristics of each city object and its parts. On the left side of Fig. 1 a CityGML building is shown in its object-based boundary representation. The building is semantically subdivided into two roof surfaces, 4 wall surfaces and one ground surface. Each of these thematic surfaces can be enriched by thematic information (e.g. area, material, color, …). This object based representation is overlain by its field based representation that is approximating the shape of the building by a voxel grid. Currently, the semantic information is lost during the geometric derivation of the voxel model from the city model objects.

The central research question of this paper is to find a way to relate both systems – the object based and the field based - to allow information exchange between them. This comprises three parts: (1) the derivation of geometrically and topologically sound voxel models from semantic 3d city models and (2) persisting semantics in the voxels of the field based simulation model as illustrated with the blue voxels in Fig. 1 and (3) the back referencing of simulation results to the city model. The general objective of the discussed research work is the development of an

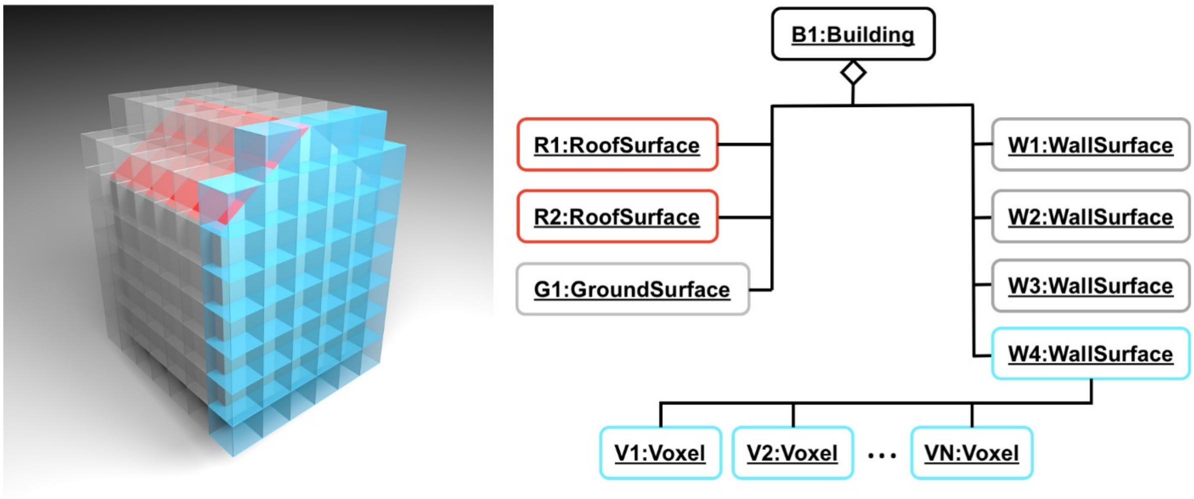


Fig. 1: Object based CityGML building model with its corresponding field based voxel representation. The schema on the right highlights the logical relation between WallSurface 4 and the voxels related to it.

automated workflow for coupling CityGML city models and field based simulation tools, enabling non-expert users to perform simulations and asses their results.

2 Theoretical basis

Today, Computational Fluid Dynamics (CFD) simulation tools support planning and decision making in various fields of applications in an urban context. Examples are given in FADL & KARADELIS (2013) describing a CFD simulation for wind comfort and safety in pedestrian areas or at HANNA et al. (2006) investigating wind flow and dispersion of pollutants in urban space. Research on the topic of 3D noise mapping has been done by STOTER et al. (2008), an example on the detailed 3D simulation of flooding for the city of Munich is presented by VARDUHN et al. (2015). In the field of explosive safety and building construction CFD simulations allow strategic and conceptual preparation for individual blast scenarios (RUTNER et al. 2008; KLOMFASS & THOMA 2012; TROMETER & MENSINGER 2014). These highly tailored applications allow for efficient simulation form different domains, but lack interactive access to simulation results. 3D city models and their frameworks offer comprehensive tools for visualization and result analysis even for non-expert users. They represent a reliable data source for both geometries and semantics in an urban context.

2.1 CityGML

The CityGML standard is an open data model for the representation and exchange of virtual 3D city and landscape models. The model includes information on geometry, appearance, semantics, and topology of urban objects organized in a multi scale approach with four Levels of Detail (LOD). It is a format based on the Extensible Markup Language (XML) and the ISO 19100 standards family. CityGML is an application schema of the Geography Markup Language version 3.1.1 (GML3) which implements the ISO 19107 standard (HERRING 2001). In 2012, its latest issue, version 2.0.0, was released as an official standard of the Open Geospatial Consortium (OGC) (OGC 2012).

The CityGML model not only represents the shape and graphical illustration of city objects. The objects' semantics, their thematic properties, taxonomies, aggregations and interrelations are taken in to account (KOLBE 2009). This enables CityGML to serve as an information model for a broad range of applications (BILJECKI et al. 2015). The rich semantic properties of CityGML's urban objects represent a potential source for input parameters for field based simulations like for example construction materials of buildings. The standard allows the creation of additional attributes and objects, which are not part of the predefined classes and thus, allows its extension for the discussed application context (KOLBE 2009). In the recent years, a framework of tools has evolved around the standard. This gives developers access to well proven and maintained software components for data storage, management and distribution, visualization and assessment tasks.

Moreover, CityGML datasets become increasingly available and have improved in both quality and coverage. For example, about two third of all buildings in Bavaria, Germany, are already available in LoD2 (LDBV 2016). Hence, CityGML models are an attractive data source for simulation tools, as time and cost intensive data acquisition is likely not to be necessary.

Especially simulations for short term response or emergency scenarios like evacuations depend upon existing data.

3 Derivation of Voxel models from CityGML geometries

The host system for the derivation of the voxel representation from CityGML geometries is the 3DCityDB (www.3dcitydb.org). The 3DCityDB maps the CityGML data model onto a relational database schema. It operates on the commercial spatial database management system Oracle Spatial and the Open Source relational database management system PostgreSQL using PostGIS as extension for spatial data types and operations (KOLBE et al. 2009; STADLER et al. 2009; POSTGIS 2015; POSTGRESQL 2016). The whole calculation process is performed on the database where all operations are encapsulated in PL/pgSQL SQL-Procedural Language database functions. This approach is beneficial, as no data but lightweight function calls have to be transferred between the database and the model generation application avoiding network traffic as a bottleneck. The model generation is performed within the spatial reference system the underlying CityGML dataset is presented in. Hence, a projected spatial reference system with e.g. a metric system is required.

Only a few input parameters are needed for the conversion process. First, the thematic layers of the CityGML city model which should be converted to a voxel representation need to be selected. Depending on the given use case, certain aspects of the city model can be ignored to reduce the amount of data that has to be processed.

Second, the simulation domain within the city model is confined by a 3D bounding box (BBox). Additionally, the edge length of the voxels needs to be specified. This parameter defines the resolution of the voxel grid and should be selected according to the quality requirements of the simulation that will be conducted. It strongly influences the performance of both the model

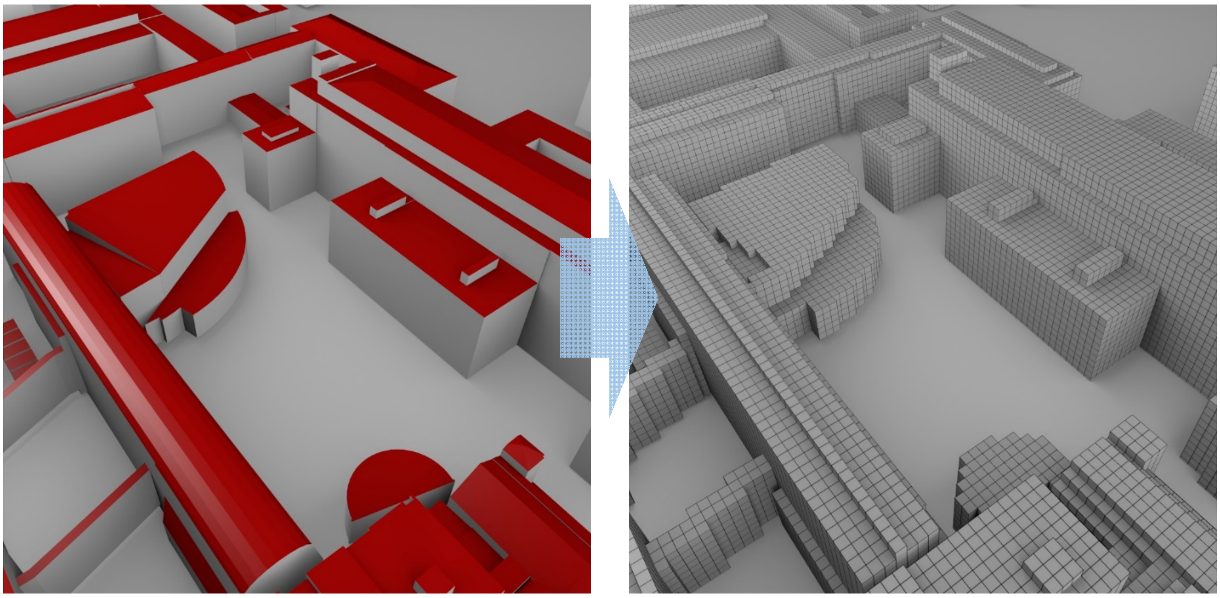


Fig. 2: CityGML buildings of the TUM campus and the derived voxel representation at a voxel edge length of two meter.

generation and the simulation run.

The derivation of the voxel model is performed in two step, the creation of the voxel geometry representation and the identification of voxels intersecting objects of the city model. For the geometry creation the number of voxels required to fill the whole simulation domain for each coordinate axis is calculated first based on the specified simulation BBox and the voxel edge length. Thus, all voxels can be organized in an integer IJK coordinate system, which is oriented as the XYZ Cartesian coordinate system of the city model, as depicted in Fig. 3. Each voxel is therefore uniquely identified by its IJK coordinates. By passing the voxels IJK coordinates, the XYZ origin of the coordinate system, which is the lower left bottom of the simulation domain, and the voxel edge length to a database function, the voxel geometry can be computed.

In the function a Well-Known-Text (WKT) representation (OGC 2011) of the voxels geometry is created which is scaled by the voxel edge length and translated to its position in the voxel grid relative to the given origin by multiplying the IJK coordinates with the voxel edge length. The voxels geometry is returned as a PostGIS PolyhedralSurface spatial object by the voxel creation function (OGC 2010, 2011).

Now, the voxel geometry objects interacting with city model objects have to be identified. They are tested for intersection with geometries of the city model using 3D intersection operations of PostGIS. Thereby, the GiST index structures provided by the database system are used (HELLERSTEIN et al. 1995). They implement an R-Tree spatial index which increases query performance using a tree data structure for bounding box comparisons (GUTTMAN & STONEBRAKER 1983). An exampe of a voxel model derived from a CityGML model of the Campus of Technical University Munich (TUM) is illustrated in Fig. 2.

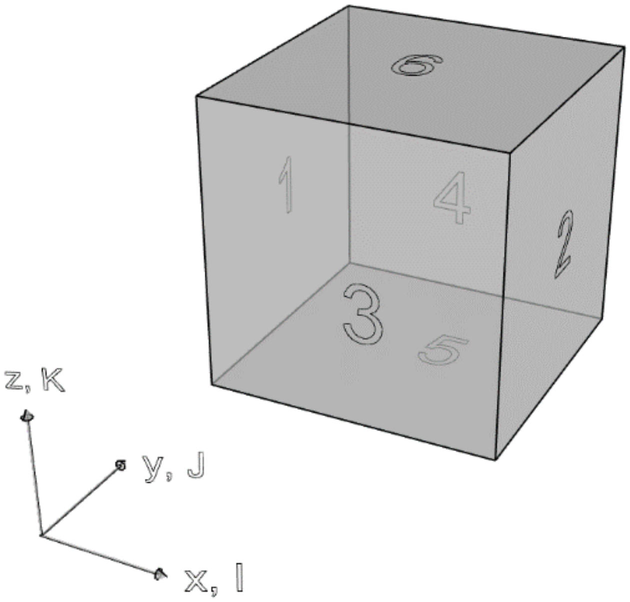


Fig. 3: Voxel with Cartesian XYZ coordinates system, integer IJK coordinate system and numbering schema for voxel faces. Face one is located on the minimum side of yz plane, face two is located on the maximum side of the yz plane. The other faces are numbered accordingly.

4 Information exchange between city and voxel model

When identifying voxels with spatial relation to the city model geometries, only voxels with a positive result for the intersection test are saved. These voxels are stored in a VOXEL database table with their IJK coordinates, their geometry and a unique identifier (UID_VOXEL). In another database table named INTERSECTION the logical link between the voxels and the city model geometries is stored as tuples of the UID_VOXEL and the SURFACE_GEOMETRY_ID, the unique identifier of the city models' geometries. With this n:n relationship it is possible to put single city model geometries or whole city objects in relation with single voxels. The numbering schema depicted in Fig. 3 even allows a unique identification and relation of individual voxel faces. Hence, the information exchange between city model entities and the voxel model discussed in section 1 is now feasible using standard database joins. The semantic information contained in the city model can be query from the database for use in a simulation application that is operating on the derived voxel representation.

For allowing interactive visualization and analysis within the city model the simulation results need to be stored in the city model objects. This can be done using CityGML's extension mechanisms. Using the above mentioned logical relation between both models, simulation results that are returned per voxel or voxel face, can be aggregated per related city object and persistently stored with it as generic attribute. The enriched city model objects are now usable for further analytic or visualization tasks based on the simulation results.

The presented semantic enrichment of the city model with simulation results poses an added value for both the city model data and simulation tool, as linking the simulation results with the already existing information in the model opens new possibilities for assessment of the data.

5 Example usage scenario: Blast simulation with the Apollo Blastsimulator

In the following section the proposed approach for the integration of CFD simulation tools and 3D city models is evaluated on the example of a blast simulation. The work described in this section is inspired by the Master Thesis of WILLENBORG (2015).

5.1 Apollo Blastsimulator

The Apollo Blastsimulator is a CFD simulation tool developed at Fraunhofer Institute for High-Speed Dynamics, Ernst-Mach-Institute (EMI), in Freiburg, Germany. According to KLOMFASS et al. (2013) and KLOMFASS & HERZOG (2010), when analyzing blast effects mostly two kinds of calculation models are available. First, there are strongly simplified approaches based on TNT equivalences and scaled distances. They work very fast, but have restrictions regarding accuracy and the spectrum of permitted applications. Second, there are general purpose CFD simulations delivering both accurate results and usability by a broad range of applications for the price of difficult handling and long setup and computation times. The Apollo Blastsimulator tries to fill the gap between both approaches and combines good usability, versatility and computational efficiency. This is achieved by tailoring the methodological concepts to the application of explosions and blast waves.

The computational mesh the application works on, needs to be computed for the observation area prior to the simulation run. The mesh consists of square finite volume elements that can be created from CityGML data with the presented approach. Every solid structure in the simulations domain needs to be approximated by a closed shell of voxels. After a simulation run, the Apollo Blastsimulator returns two types of results. Besides physical quantities a set of probability values for various damage categories is provided. For each voxel surface exposed to fluid (air) in the simulation domain a set of result values is returned.

5.2 System architecture and workflow

To provide a user friendly application the described approach for the voxel model creation has been implemented as a plugin for the 3DCityDB Importer Exporter, the management software of the 3DCityDB. Besides its data import/export capabilities for various formats, a powerful Application Programming Interface (API) is shipped, enabling developers to extend the tool with new components, implemented as plugins. The API offers a wide range of ready-to-use core functionality like database interaction, GUI elements, parallelisation strategies or logging, making plugin development easy (NAGEL & KADEN 2011). The programmer can rely on a well-tested application backbone and focus on the logic of the use-case.

The core element of the workflow is the *Apollo Blast Plugin*, the plugin extension for the 3DCityDB Importer/Exporter implemented in this study. It operates as the control and data exchange unit between simulation, city model, visualization, and result analysis.

The plugin provides a GUI allowing the user to specify a simulation extent on an interactive 2D map, the explosion location, and the explosive device. Further, the calculation approach, either engineering method (simplified approach based on TNT equivalences and scaled distances) or CFD simulation, can be selected. For the CFD simulation the computational time and result quality can be configured by adapting the resolution of the voxel mesh. This is required for multi-stage analysis, typically conducted in risk analysis. First, a big amount of scenarios is analysed with the engineering method to identify potentially critical scenarios, which are re-evaluated with the high quality CFD simulation afterwards (KLOMFASS et al. 2013). All settings are persistently stored on disk in an XML configuration file to facilitate easy reuse and exchange.

For the creation of the voxel model a parallelization strategies has been implemented. The IJK BBox of the simulation domain is split into slices along the I axis. This allows to process several slices on a given number database connections in parallel to leverage multiple cores of the database server and significantly reduce the time consumption of the conversion process.

In a typical workflow, the user first creates a scenario definition with the above settings in the Apollo Blast Plugins GUI. After that, the ASCII input files for the simulation tool, containing the computational mesh, are computed and passed to the simulation application. When the Apollo Blastsimulator has terminated, the returned ASCII results files are read and processed by the plugin and all affected CityGML objects are updated within the connected 3DCityDB.

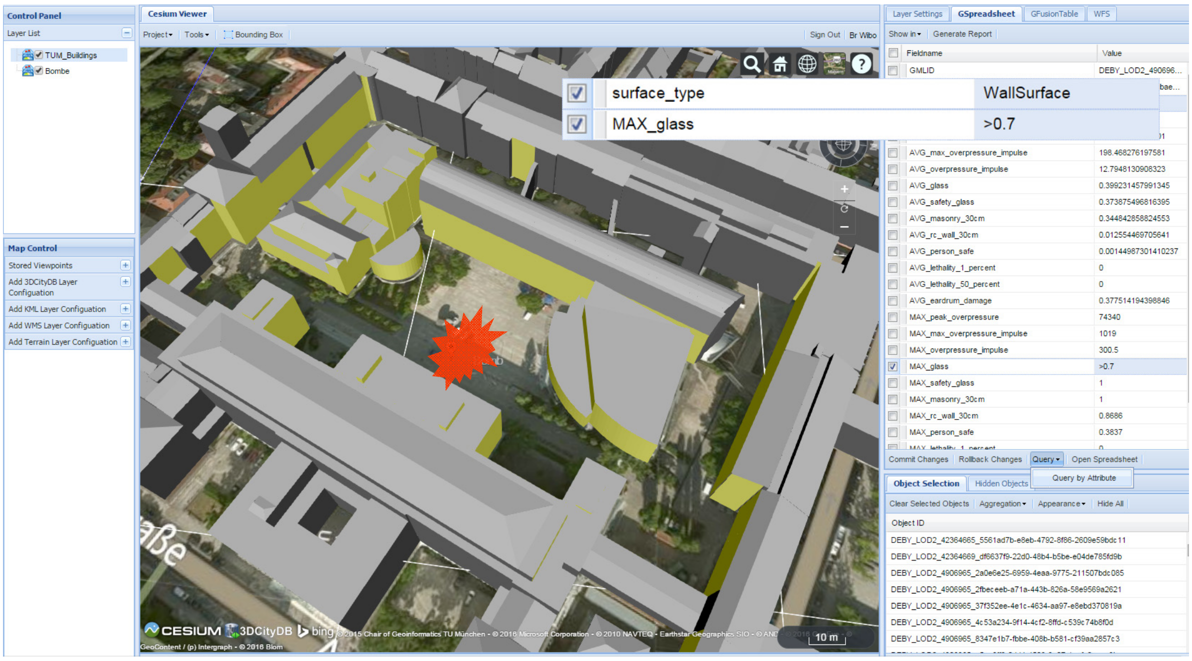
5.3 Aggregation and storage of simulation results

After a successful run, the simulation results need to be transferred back to the city model. First, the Apollo Blastsimulator ASCII results file is read by the Apollo Blast Plugin. It contains a set of results values for each voxel surface exposed to fluid (air) in the simulation domain. Now, a two level aggregation is performed. First, the maximum and average of all result parameters is aggregated per voxel. After that, all voxels related to a CityGML Thematic Surface (e.g. Wall-, Roof or GroundSurface) are aggregated based on the information stored in the intersection table during the generation of the voxel mesh. The summarized results are stored as generic attributes, one for each result parameter, with their corresponding city object in the 3DCityDB. Now all city objects in the simulation domain contain attributes for physical quantities (overpressure, overpressure impulse) and a set of probability values for damage classes like glass breakage, eardrum damage, destruction of masonry, or concrete walls and lethality. The semantically enriched city model now allows for thematic queries regarding the simulated blast event.

5.4 Visualization and result analysis

For visualization and result analysis a cloud-based 3D Webclient developed at the Chair of Geoinformatics, TUM is used. The browser based application uses the Cesium Earth Viewer (see https://cesiumjs.org/) to 3D visualize the city model. The city model is reduced both geometrically and semantically to the requirements of the specific application context (KOLBE et al. 2013; YAO et al. 2014; YAO & KOLBE 2014). Geometries are represented in the KML/COLLADA format, which is exported by the 3DCityDB Importer/Exporter KML/COLLADA exporter plugin (KOLBE et al. 2013). Thematic information, including the generic attributes created for the blast simulation results are exported with the Spreadsheet Generator Plugin and published via GoogleSpreadsheet allowing to apply spreadsheet calculations. The relation between both models is kept with a logical key (GMLID) (NADERI et al. 2012).

For demonstration purpose an artificial test scenario has been developed. We assume, that a



☑ surface_type: WallSurface
☑ MAX_glass: >0.7

	Fieldname	Value
☐	GMLID	DEBY_LOD2_490696...
☐	AVG_max_overpressure_impulse	198.468276197581
☐	AVG_overpressure_impulse	12.7948130908323
☐	AVG_glass	0.399231457991345
☐	AVG_safety_glass	0.373875496816395
☐	AVG_masonry_30cm	0.344842858824553
☐	AVG_rc_wall_30cm	0.012554469705641
☐	AVG_person_safe	0.00144987301410237
☐	AVG_lethality_1_percent	0
☐	AVG_lethality_50_percent	0
☐	AVG_eardrum_damage	0.377514194398846
☐	MAX_peak_overpressure	74340
☐	MAX_max_overpressure_impulse	1019
☐	MAX_overpressure_impulse	300.5
☑	MAX_glass	>0.7
☐	MAX_safety_glass	1
☐	MAX_masonry_30cm	1
☐	MAX_rc_wall_30cm	0.8686
☐	MAX_person_safe	0.3837

Object ID
DEBY_LOD2_42364665_5561ad7b-e8eb-4792-8f86-2609e59bdc11
DEBY_LOD2_42364669_df6637f9-22d0-48b4-b5be-e04de785fd9b
DEBY_LOD2_4906965_2a0e6e25-6959-4eaa-9775-211507bdc085
DEBY_LOD2_4906965_2fbeceeb-a71a-443b-826a-58e9569a2621
DEBY_LOD2_4906965_37f352ee-4e1c-4634-aa97-e8ebd370819a
DEBY_LOD2_4906965_4c53a234-9f14-4cf2-8ffd-c539c74b8f0d
DEBY_LOD2_4906965_8347e1b7-fbbe-408b-b581-cf39aa2857c3

Fig. 4: Cloud-based 3D Webclient example of a fictive blast scenario on TUM campus: Wall surfaces with a glass breakage probability greater 70% are highlighted in the 3D view.

bomb from world war two has been discovered during ground working in the inner court of TUM. For the test scenario only the CityGML building layer has been used, all buildings have been translated to a plane on elevation zero meters.

To display the analytic capabilities of the Webclient we will try to identify walls, in which windows are likely to break if the bomb cannot be defused and needs to be detonated on site.

First, we need to setup the query for the required information. As depicted in Fig. 4, we type the filter criteria in the attribute view to the right (see enlarged entries). After issuing the query all matching surfaces are highlighted (yellow) in the 3D view. For further analysis aggregate operations on the selected objects can be performed directly in the client. For example, by summing up the area of all currently selected surfaces we can determine the total affected wall surface. By multiplication with factors for window area on a wall and window price per area we can provide a rough cost estimation for broken windows.

6 Discussion and evaluation

Besides its benefits mentioned in the previous sections, there are some limitations and problems with the proposed approach which are discussed in this section.

One issue concerns the quality of the derived voxel model. Since all geometries are reassembled by finite volume elements for the field based representation the orientation of the voxel grid relative to the input geometry greatly influence the model quality. As depicted in Fig. 5, if the grid is aligned with the input features a smooth voxel representation of the building wall surfaces is created while the transformation with no alignment causes a ripple effect on the surface approximation. Possible solutions are the alignment of the voxel grid with its input geometries. As this is not always possible in the context of 3D city models containing real world objects which are not always rectangular an increased grid resolution will be required to avoid ripple effects.

Another problem influencing the model quality are invalid city model geometries like non planar surfaces. They cause the intersection algorithms to fail consequently leading to holes in the voxel model. As CFD simulations require objects to be represented by a closed shell of voxels, this is a big issue. As the quality and validation of 3D city models is a matter of ongoing research, this is not further discussed here (ALAM et al. 2013).

Regarding the different modelling paradigms between field based simulations and object based 3D city models with boundary representation geometries there are some issues, that require further research. Due to different scales between single objects in city models and field based simulations (e.g. size voxel << size WallSurface) simulation results cannot be mapped in full detail to the city model. When aggregating the simulation results attached to several voxels to a single city object information is lost, results may even be falsified statistically. This issue could be addressed by the generation of raster textures which are supported by the CityGML appearance model. A combination of textures representing in depth details delivered by a simulation and aggregated attributes that can be subject to queries would foster analytic capabilities including a high level of detail for simulation results in 3d city models.

Another issue when trying to represent and store simulation results in CityGML is the absence of dynamic attributes. CFD simulations usually provided time series of results, which are not

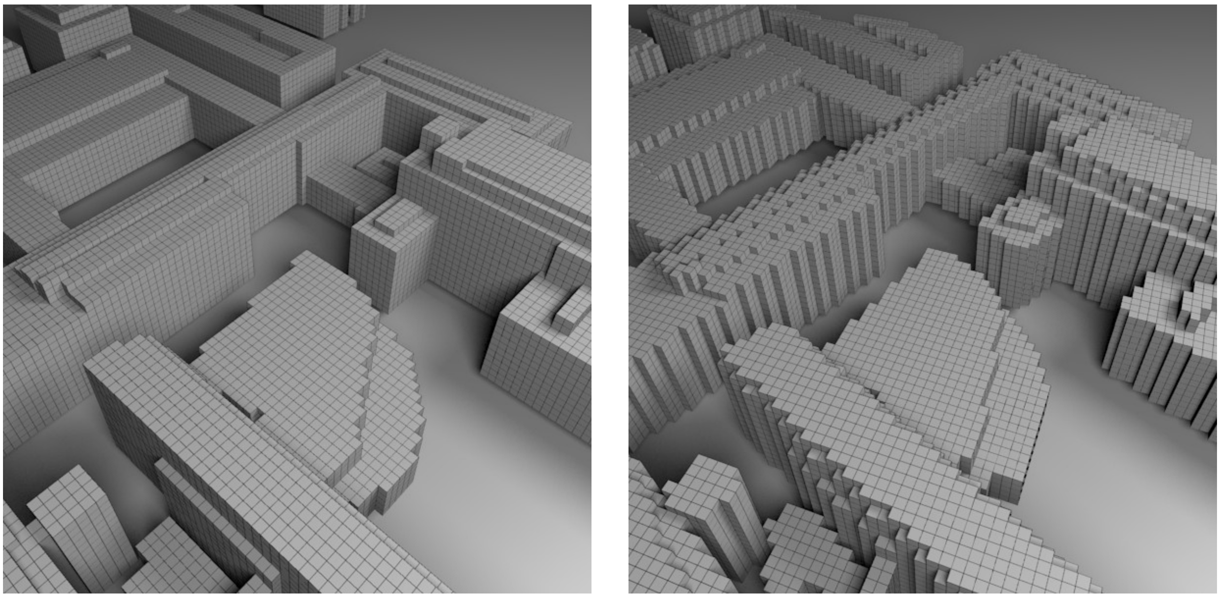


Fig. 5: Comparison of the model quality between CityGML buildings aligned with the voxel orientation before model creation (left image) and without alignment (right image).

included in the data model at this point. However, this topic is currently being researched by CHATURVEDI & KOLBE (2015) and is expected to be implemented for the next version of CityGML.

7 Conclusion

The central research question of this study was to relate sematic 3D city models and field based simulation models and their different modeling paradigms to allow information exchange between both of them. Based on the 3DCityDB an approach for deriving a field based geometry representation from a CityGML city model using spatial database operations provided by PostGIS was introduced. By mapping the results of 3D intersection tests between voxel and city model geometries to a relational database table information exchange between both systems with common database join operations could be realized. Accordingly, the rich semantic information of the city model is available to field based simulation tools. The other way round the back referencing of simulation results to their corresponding city objects is now feasible. Using the extension capabilities of CityGML a semantic enrichment of the city model with aggregated simulation results is performed based on the proposed approach. This permits visualization and assessment of simulation results within the city model and its framework. The generated information can be accessed in an interactive, cloud based environment allowing non-expert to perform analytical tasks. As simulation results are persistently stored with their city model objects they remain available for future analysis, possibly including new intelligence form various sources like simulations, and therefore pose a consistent added value for the data.

References

ALAM, N., WAGNER, D., WEWETZER, M., FALKENHAUSEN, J. V., COORS, V. & PRIES, M., 2013: Towards automatic validation and healing of CityGML models for geometric and semantic consistency. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, **II-2/W1**, 1-6.

BAYRISCHES LANDESAMT FÜR DIGITALISIERUNG, BREITBAND UND VERMESSUNG, 2016: LOD2 Arbeitsstand.

BILJECKI, F., STOTER, J., LEDOUX, H., ZLATANOVA, S. & ÇÖLTEKIN, A., 2015: Applications of 3D City Models: State of the Art Review. IJGI **4** (4), 2842-2889.

CHATURVEDI, K., KOLBE, T.H., 2015: Dynamizers - Modeling and Implementing Dynamic Properties for Semantic 3D City Models. Eurographics Workshop on Urban Data Modelling and Visualisation, Filip Biljecki & Vincent Tourre (Hrsg.).

CHEN, A.S., EVANS, B., DJORDJEVIC, S. & SAVIC, D. A., 2012: Multi-layered coarse grid modelling in 2D urban flood simulations. Journal of Hydrology **470**, 1-11.

FADL, M.S. & KARADELIS, J., 2013: CFD Simulation for Wind Comfort and Safety in Urban Area: A Case Study of Coventry University Central Campus. IJAEC **2** (2), 131-143.

GUTTMAN, A. & STONEBRAKER, M., 1983: R-trees. Electronics Research Laboratory, College of Engineering, University of California, Berkeley.

HANNA, S.R., BROWN, M.J., CAMELLI, F. E., CHAN, S. T., COIRIER, W. J., KIM, S., HANSEN, O. R., HUBER, A. H. & REYNOLDS, R. M., 2006: Detailed Simulations of Atmospheric Flow and Dispersion in Downtown Manhattan: An Application of Five Computational Fluid Dynamics Models. Bull. Amer. Meteor. Soc. **87** (12), 1713-1726.

HELLERSTEIN, J.M., NAUGHTON, J.F. & PFEFFER, A., 1995: Generalized Search Trees for Database Systems. Proceedings of the 21. International Conference on Very Large Data Bases.

HERRING, J., 2001: OGC Abstract Specifications Topic 1 – Feature Geometry (ISO 19107 Spatial Schema).

ISO 19109, 2005: Geographic information — Rules for application schema.

KLOMFASS, A., HERZOG, O., 2010: A Reliable, Fast and Easy-to-use Simulation Tool for Blast Waves in Urban Scenarios. Fraunhofer Institute for High-Speed Dynamics, Ernst-Mach-Institute. Freiburg, Germany.

KLOMFASS, A., THOMA, K., 2012: CFD and Holistic Methods For Explosive Safety and Risk Analysis. Process and Plant Safety **9**, 9-29.

KLOMFASS, A., ZWEIGLE, T. & FISCHER, K., 2013: A New AMR/ALE-Solver for the efficient Simulation of Detonations and Blast Waves. Intern. Symposium of Interaction of Effects of Munitions with Structures.

KOLBE, T.H., 2009: Representing and Exchanging 3D City Models with CityGML. 3D Geo-Information Sciences. Lecture notes in geoinformation and cartography, Lee, J. & Zlatanova, S. (Hrsg.), 15-31.

KOLBE, T.H., KOENIG, G., NAGEL, C. & STADLER, A., 2009: 3D-Geo-Database for CityGML. Proceedings of the 3rd International Workshop on 3D Geo-Information, Lecture notes in Geoinformation and Cartography, Lee, J. & Zlatanova, S. (Hrsg.).

KOLBE, T.H., YAO, Z., HERRERUELA, J., NAGEL, C., KUNDE, F., WILLKOMM, P. & HUDRA, G., 2013: Addendum to the 3D City Database Documentation Version 2.0.1.

NADERI, B., HERRERUELA, J. & KOLBE, T.H., 2012: Spreadsheet Generator Version 1 .0.0 Plugin for Importer/Exporter Version 1.4.0.

NAGEL, C. & KADEN, R., 2011: Importer/Exporter - Plugin API - Slides for internal discussion. Institute for Geodesy and Geoinformation Science, Technical University Berlin.

OPEN GEOSPATIAL CONSORTIUM, 2010: OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 2: SQL option.

OPEN GEOSPATIAL CONSORTIUM, 2011: OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture.

OPEN GEOSPATIAL CONSORTIUM, 2012: OGC City Geography Markup Language (CityGML) Encoding Standard. Version 2.0.0.

POSTGIS PROJECT STEERING COMMITTEE, 2015: PostGIS 2.2.0dev Manual.

RUTNER, M., MENSINGER, M. & TROMETER, S., 2008: Die Auswirkung von duktilen Anschlüssen auf die Explosionsresistenz lasttragender Stützen. Stahlbau **77** (6), 436-446.

STADLER, A., NAGEL, C., KÖNIG, G. & KOLBE, T.H., 2009: Making Interoperability Persistent: A 3D Geo Database Based on CityGML. 3D Geo-Information Sciences. Lecture notes in geoinformation and cartography, Lee, J. & Zlatanova, S. (Hrsg.), 175-192.

STOTER, J., KLUIJVER, H. de & KURAKULA, V., 2008: 3D noise mapping in urban areas. International Journal of Geographical Information Science **22** (8), 907-924.

THE POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2016: PostgreSQL 9.5.1 Documentation.

TROMETER, S., MENSINGER, M., 2014: Simulation von Detonationsszenarien im urbanen Umfeld. Geoinformationssysteme 2014, Tagungsband zur 1. Münchner GI-Runde, Kolbe, T. H., Bill, R., Donaubauer, A. (Hrsg.), 150-164.

VARDUHN, V., MUNDANI, R.-P. & RANK, E., 2015: Multi-resolution Models: Recent Progress in Coupling 3D Geometry to Environmental Numerical Simulation. 3D Geoinformation Science. Lecture notes in geoinformation and cartography, Breunig, M., Al-Doori, M., Butwilowski, E., Kuper, P. V., Benner, J., Haefele, K.H. (Hrsg.), 55-69.

WILLENBORG, B., 2015: Simulation of Explosions in Urban Space and Result Analysis Based on CityGML City Models and a Cloud Based 3D Web Client, Masterarbeit. Technische Universität München, München.

YAO, Z., KOLBE, T.H., 2014: Cloud-basierter Webclient für kollaboratives Arbeiten auf semantischen 3D-Stadtmodellen. Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation (DGPF) e.V, Band **23**, Beitrag 234.

YAO, Z., SINDRAM, M., KADEN, R. & KOLBE, T.H., 2014: Cloudbasierter 3D-Webclient zur kollaborativen Planung energetischer Maßnahmen am Beispiel von Berlin und London. Geoinformationssysteme 2014, Tagungsband zur 1. Münchner GI-Runde, Kolbe, T.H., Bill, R., Donaubauer, A. (Hrsg.), 40-52.