

TRIANGULAR MESH APPROACH FOR AUTOMATIC REPAIR OF MISSING SURFACES OF LOD2 BUILDING MODELS

Hanis Rashidan¹, Alias Abdul Rahman¹, Ivin Amri Musliman¹, Gurcan Buyuksalih²

¹3D GIS Research Lab, Universiti Teknologi Malaysia, Johor Bahru, Malaysia
mhanis27@gmail.com, alias@utm.my, ivinamri@utm.my

²Istanbul, Turkey
gurcan.buyuksalih@gmail.com

Commission 4, WG 7

KEY WORDS: 3D Building Models, Automatic Repair Technique, Geometric Errors.

ABSTRACT

3D city models are increasingly being used to represent the complexity of today's urban areas, as they aid in understanding how different aspects of a city can function. For instance, several municipalities and governmental organisations have constructed their 3D city models for various purposes. These 3D models, which are normally complex and contain semantics information, have typically been used for visualisation and visual analysis purposes. However, most of the available 3D models open datasets contain many geometric and topological errors, e.g., missing surfaces (holes), self-intersecting surfaces, duplicate vertices, etc. These errors prevent the datasets from being used for advanced applications such as 3D spatial analysis which requires valid datasets and topology to calculate its volume, detect surface orientation, area calculation, etc. Therefore, certain repairs must be done before taking these models into actual applications, and hole-filling (of missing surfaces) is an important one among them. Several studies on the topic of automatic repair of the 3D model have been conducted by various researchers, with different approaches have been developed. Thus, this paper describes a triangular mesh approach for automatically repair invalid (missing surfaces) 3D building model (LOD2). The developed approach demonstrates an ability to repair missing surfaces (with holes) in a 3D building model by reconstructing geometries of the holes of the affected model. The repaired model is validated and produced a closed-two manifold model.

1. INTRODUCTION AND MOTIVATION

3D city models are increasingly being used to represent the complexity of today's urban areas, as they aid in understanding how different aspects of a city can function (Vitalis et al., 2019). For instance, several municipalities and governmental organisations have constructed 3D city models for various purposes, some of them have been reported in studies such as Singapore (NRF, 2018) (Gobeawan et al., 2018), Brisbane city in Australia (Alatalo et al., 2017), Vienna city in Austria (Agugiaro, 2016), Lyon city in France (Nouvel et al., 2015), Rotterdam city in the Netherlands (Boeters et al., 2015), New York City in the United States (Chaturvedi et al., 2015), Istanbul city in Turkey (Buyuksalih et al., 2013), Berlin city in Germany (Carrión et al, 2010), and Stuttgart city in Germany (Ewald & Coors, 2005). Semantically enriched 3D city models have the potential to be powerful hubs of integrated information for computer-based urban spatial analysis. The 3D city models, which contain 3D data of urban objects like buildings, roads, and waterways, may serve for various applications. Some notable applications are smart cities as reported by Ruohomäki et al., (2018) and Chaturvedi & Kolbe, (2016), noise propagation as studied by Lu, Becker, & Löwner, (2017), emergency response, solar power as reported by Biljecki et al., (2017), shadows cast as studied by Herbert & Chen, (2015), routing analysis as studied by Hildebrandt & Timm, (2014), change detection as studied by Sharkawi & Rahman, (2014), 3D cadastre as studied by Van Oosterom, (2013), facility management as studied by Zlatanova & Beetz, (2012), visibility analysis as studied by Yang, Putra, & Li, (2007), visualisation

for navigation as studied by Musliman, Rahman, & Coors, (2006), communication of urban information as studied by Döllner, et al., (2006), and urban planning.

To allow for the development of advanced applications, a 3D city model should describe the geometry and attributes of all the individual elements that are typically present in a city. Therefore, 3D city models require a proper data standard for realising efficient 3D data analysis, management, interaction, visualisation, update, and storage. One of the main practices for storing and exchanging 3D city models is through CityGML - as an international standard established by OGC (Groger et al., 2012). CityGML is an open data model and XML-based data exchange format describing the most relevant urban and landscape objects along with their spatial and non-spatial attributes, relations, and their complex hierarchical structures in five levels of detail (LoD).

There is a huge concern for realising 3D city models to be fully implemented in the industrial area which is the issue in 3D data quality. As highlighted by Stoter et al. (2020) and Biljecki et al. (2019), (2016), most of the available 3D city models open datasets contain many geometric and topological errors, e.g. duplicate vertices, missing surfaces, self-intersecting volumes, etc. For 3D models to be used as an input in most city simulation and modelling software, the models should be geometrically and topologically valid. Often, these errors are not visible at the scale on which the datasets are visualised, and some users are unaware of the issue due to the models appear to be valid in the software viewer.

Therefore, the validation process means the need to check whether a given 3D primitive respects the standardised definitions ISO 19107 - an international standard of geometry data model. Biljecki et al. (2016) highlighted those common types of geometric errors are normally to be found in 3D models datasets (consist of 3.6M buildings), and most of the datasets are invalid and perfectly valid datasets are very rare to be found. Hence, validation of the 3D models geometry and the detection of the errors can be achieved through several tools such as Val3Dity (Biljecki et al., 2019), and CityDoctor (Wagner et al., 2013). These validators can validate CityGML datasets regarding their semantic and geometrical correctness or regarding their geometric-topological consistency. Figure 1 shows the types of geometries (primitives) used in 3D city models (CityGML).

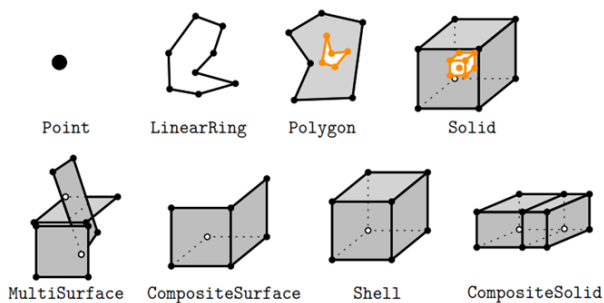


Figure 1. The 3D geometric primitives are used in CityGML (Biljecki et al., 2016).

There are many different errors in 3D city models, of which some are common across multiple datasets. The common errors found in 3D data validation are geometries not properly snapped, invalid semantics, non-planarity, and invalid polygon orientation. For instance, a common cause for invalid solids geometries are gaps, the intersection between surfaces, degenerated edge and surface primitives, and non-manifold situations which result in a non-watertight solid. Thus, Stoter et al. 2020 stated that there is a need for implementing an automatic repair technique to handle these problems, and a 3D model should be validated and repaired before it can be further used (Ledoux, 2013, Biljecki et al. 2016).

3D repair has become an important topic in recent years because of the increasing availability of 3D city models worldwide. Therefore, this paper describes a triangular mesh approach for automatically repair invalid 3D models but focussing on the repairing LOD2 solid building model. The remainder part of the paper is structured as follows; Section 2 the related works, Section 3 the triangular mesh approach, and Section 4 Conclusion.

2. THE RELATED WORKS

The following sub-sections describe the major recent works in the 3D city modelling quality aspects, particularly on repairing and reconstructing an invalid 3D building model.

2.1 Local repair method

Among the earliest study in repairing 3D city models were investigated by Alam et al. (2014). He described a local method that checks and iterates through all polygons and solids for specific errors. The details explanation of the method is as follows – if any defects are encountered, then local repair

techniques are applied. For example, on the polygon level, the method is able to repair the problem of self-intersection and consecutive vertices. Furthermore, the method can repair the shell and solid with the multiple connected components, simple self-intersections (overlapping polygons), and face orientations. However, the method was unable to repair automatically the overused edges (i.e. edge bounding more than 2 faces) as shown in Figure 2.



(a) Self-intersections (b) Non-manifold vertex
Figure 2. Some limitations of the detect and local repair method (Alam et al., 2014).

2.2 Voxelization of Polygonal Mesh

A voxel-based repair method takes a different approach to the polygonal model (e.g. as experimented by Alam et al., 2014), however, the voxel method can be considered as equivalent to the volumetric methods as described by Attene et al., (2013). In general, the voxel approach works as follows – the input of the 3D model converted into 3D binary grids by morphological operators and disregarding of inner shells to produce a manifold result. The following figure illustrates the voxelization of the 3D model.

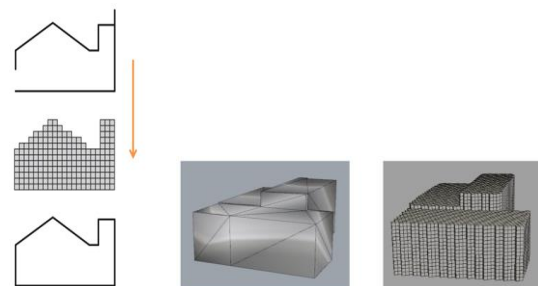


Figure 2. The voxelization repair concept for 3D building model (Mulder, 2015).

2.3 Shrink-wrapping Method

Figure 3 shows another repair method called Shrink-wrapping as reported Zhao et al. (2013). The method is based on graph which applies a constrained tetrahedralization on all faces of the model and its convex hull. The method produces a volumetric representation that made up of polyhedron (i.e. a 3D shape with flat polygonal faces, straight edges and sharp corners). The method also utilizes a heuristic carving process for deleting unnecessary polyhedron from the model.

The Zhao et al., (2013) method able to effectively repair the gaps, self-intersections, and holes in the LoD2 CityGML datasets. The resulting repair technique produces some unwanted shapes as illustrate in Figure 13. In addition, the Shrink-wrapping method is sensitive to floating point arithmetic. The rounding of point precision may result in triangles being decomposed into edges or vertices, makes the

building model invalid. Finally, the occurrence of overlaps in the input model may produce small errors. The Shrink-wrapping method for automatic repair of invalid 3D models has some advantages - the surface semantics remains intact and maintains the original geometry of the input 3D models.

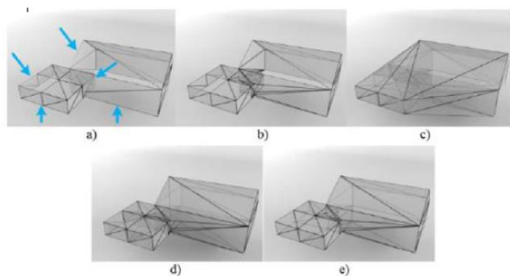


Figure 3. Illustration of the Shrink-wrapping method (Zhao et al. 2013).

2.4 Summary of the related works

In addition to these existing repair methods and algorithms, most of the contemporary 3D model repair research and development has been focusing on the repairing of generic 3D models (LoD2), common geometric errors (gaps, consecutive points, self-intersection, non-manifold edges, semantic attributes preservation, tilted surface, etc.), and volume-based models. Although, these approaches are important, as they provide solution to automatically fix most of the geometric errors found in 3D city models. However, the capability of the existing repair methods for 3D buildings models are still limited to be recognise as a general solution. Therefore, there is a need to conduct, at an advanced level, a systematic study to improve the repair algorithms of 3D models by developing a repair framework for more complex structure such as LoD3 (with windows and doors) and also indoor components. However, this paper initially describes the experiment on the repairing technique - triangular approach for missing surfaces.

3. THE TRIANGULAR MESH APPROACH

Triangular meshes are used widely to represent an object in 3D modelling. A triangle mesh is a type of polygon mesh comprises a set of triangles (typically in three dimensions) that are connected by their common edges. It is a collection of cells: vertices, edges, and triangles. Each edge is the line segment between 2 vertices and is relatively open. The 3D building model (see Figure 4) normally stored as a polyhedron (a three-dimensional shape with flat polygonal faces). In order to carry out a repairing process, the target model must be converted into 3D triangular mesh.

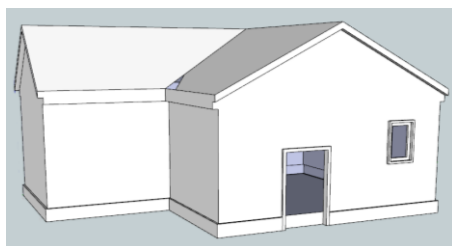


Figure 4. The building model (single storey house) used for the case study.

3.1 Validate 3D Model

Geometry of the 3D model need to be validated to check whether it respects the standardised definitions (ISO 19107 - International standard of geometry data model). For instance, existence of holes and non-watertight boundary in the model. The *val3dity* tool (Ledoux, 2019) was used to perform the validation on the data model. The tool is an open-source software to validate 3D primitives according to the international standard and outputs a report that identify the errors as shown in Figure 5 below.

```

+++++++ SUMMARY ++++++
Input file type:
OBJ
+++++
Total # of Features:    1
# valid:                0 (0.0%)
# invalid:              1 (100.0%)
Types:
GenericObject
+++++
Total # of primitives:  1
# valid:                0 (0.0%)
# invalid:              1 (100.0%)
Types:
Solid
+++++
Errors present:
302 -- SHELL_NOT_CLOSED
1 primitive(s)
+++++++
    
```

Figure 5. Report of geometry validation shows an error existed in the model.

Based on the validation report obtained, it is shows that the model has geometric error which is due to the existing holes in the model. Figure 6 shows several locations of the errors due to missing surfaces which create the holes in the model.

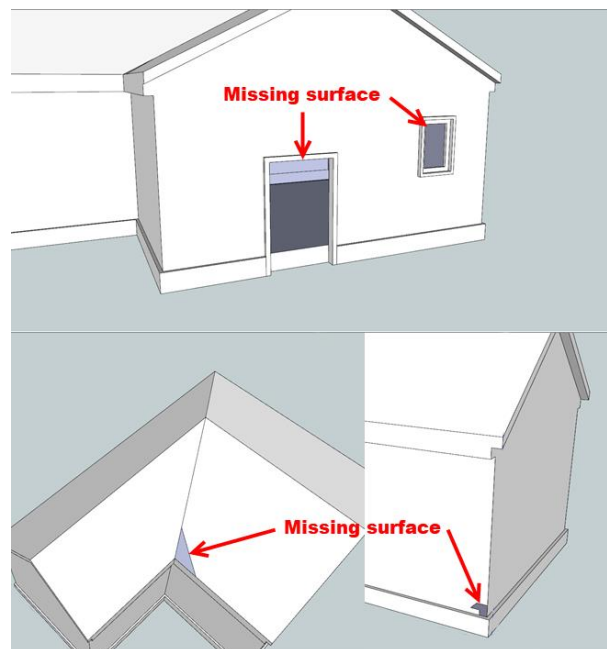


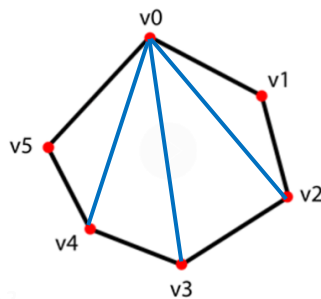
Figure 6. Missing surfaces available in the model.

3.2 Fixing the Geometry

In the triangle-based approaches, the holes are patched by dealing with the triangles directly. Holes with regular boundary over a relatively planar region can be patched via planar

triangulation. Figure 7 shows the algorithm developed based on triangulation method for repairing 3D model due to the holes error (missing surfaces).

Algorithm: Triangulation of polygon



- Step 1:** Read surfaces of 3D model
- Step 2:** Access vertices (v) of the surface
- Step 3:** Calculate number of triangles Δ_{total} can be formed:
 $\Delta_{total} = \text{total_vertices} - 2$
- Step 4:** Connecting the vertices:
 $\Delta_n = v[0], v[n+1], v[n+2]$
 where, n is the number of triangles formed.
- Step 5:** Loop Step 4 until $n == \Delta_{total}$
- Step 6:** Goto Step 2 for another surface

Figure 7. Triangulation technique to repair holes in 3D model.

In the repairing process, geometrically invalid data is identified, and the algorithm is used to cover the hole with newly created triangles. Then new model can be regenerated so that is it in closed-two manifold condition. Euler's formula for triangulation is used to calculate and detect any non-manifold model. The formula is derived as:

$$T - E + V = 2, \text{ for a simple closed surface}$$

Where, T triangles, E edges and V vertices.

Next, we need to detect the holes in the triangular mesh model. Since a vertex-based data structure is used in this work, all boundary vertices can be identified by checking the numbers of their triangles and edges. For example, if the two numbers of a vertex are not equal, the vertex is a boundary vertex. In the triangle-based approaches, the holes can be patched by dealing with the triangles directly. Holes with regular boundary over a relatively planar region can be patched via planar triangulation:

$$\text{Triangle(s) (patch)} = v[0], v[n+1] \text{ and } v[n+2]$$

Where, v is the vertices, and n is the current number of triangles formed.

After the model is repaired, it will then re-validate to confirm the error is removed and watertight model is obtained. Figure 8 shows the outcome of the implemented repair method, after re-validate the model, there is no error exists, and the watertight solid model is obtained.

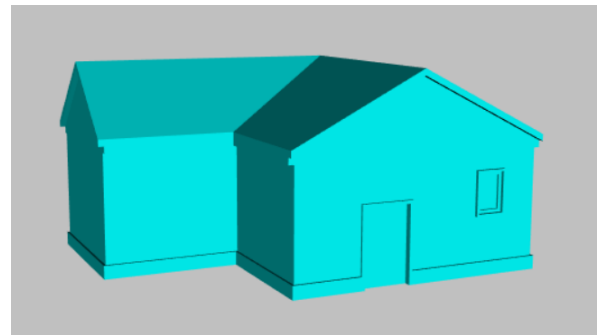


Figure 8. The repaired 3D model (solid).

4. CONCLUSION

In this paper, we proposed a triangular mesh repair algorithm for filling holes (missing surfaces) in 3D building model. The outcome shows that, the repaired model demonstrate new surfaces are added to cover the holes. After re-validate the model, report from the *val3dity* tool shows that no error detected, and watertight model was obtained. The work from this paper is our initial outcome for the development of automatic repair method for 3D city model. As discussed in the Section 2, most of the existing repair methods and research has been focusing on repairing generic 3D solid models, and common geometric errors (gaps, consecutive points, self-intersection, non-manifold edges, semantic attributes preservation, tilted surface, etc.). Although, these approaches are important, as they provide solution to fix some geometric errors found in 3D city models. However, the capability of the existing repair methods for 3D building model are still limited (indoor and complex structures) and require more studies to produce a complete repair solution which is our future work.

REFERENCES

- Aguiaro, G. (2016). First Steps Towards an Integrated CityGML-based 3D Model of Vienna. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 3(4), (pp.139-146).
- Alam, N., Wagner, D., Wewetzer, M., von Falkenhausen, J., Coors, V., & Pries, M. (2014). Towards automatic validation and healing of CityGML models for geometric and semantic consistency. In *Innovations in 3D Geo-Information Sciences* (pp. 77-91). Springer, Cham.
- Alatalo, T., Pouke, M., Koskela, T., Hurskainen, T., Florea, C., & Ojala, T. (2017). Two real-world case studies on 3D web applications for participatory urban planning. Paper presented at the Proceedings of the 22nd International Conference on 3D Web Technology (pp.1-9).
- Attene, M., Campen, M., & Kobbelt, L. (2013). Polygon mesh repairing: An application perspective. *ACM Computing Surveys (CSUR)*, 45(2), 1-33.
- Biljecki, F., & Tauscher, H. (2019). Quality of BIM-GIS conversion. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 4.
- Biljecki, F., Ledoux, H., Du, X., Stoter, J., Soon, K. H., & Khoo, V. H. S. (2016). The most common geometric and semantic errors in CityGML datasets. *ISPRS Annals of*

- Photogrammetry, Remote Sensing & Spatial Information Sciences, 4.
- Boeters, R., Arroyo Otori, K., Biljecki, F., & Zlatanova, S. (2015). Automatically enhancing CityGML LOD2 models with a corresponding indoor geometry. *International Journal of Geographical Information Science*, 29(12), 2248-2268.
- Buyukaslil, I., Isikdag, U., & Zlatanova, S. (2013). Exploring the processes of generating LOD (0-2) CityGML models in greater municipality of Istanbul. Paper presented at the 8th 3DGeoInfo Conference & WG II/2 Workshop, Istanbul, Turkey, 27–29 November 2013, ISPRS Archives Volume II-2/W1.
- Carrión, D., Lorenz, A., & Kolbe, T. H. (2010). Estimation of the energetic rehabilitation state of buildings for the city of Berlin using a 3D city model represented in CityGML. Paper presented at the ISPRS Conference: International Conference on 3D Geoinformation. XXXVIII-4.
- Chaturvedi, K., & Kolbe, T. H. (2016). Integrating dynamic data and sensors with semantic 3D city models in the context of smart cities. *ISPRS Ann Photogramm Remote Sens Spatial Inf Sci*, 4, 31-38.
- Chaturvedi, K., Yao, Z., & Kolbe, T. H. (2015). Web-based Exploration of and interaction with large and deeply structured semantic 3D city models using HTML5 and WebGL. Paper presented at the Bridging Scales-Skalenübergreifende Nah-und Fernerkundungsmethoden, 35. Wissenschaftlich-Technische Jahrestagung der DGPF.
- Döllner, J., Kolbe, T. H., Liecke, F., Sgouros, T., & Teichmann, K. (2006). The virtual 3d city model of berlin-managing, integrating, and communicating complex urban information. Paper presented at the Proceedings of the 25th international symposium on urban data management UDMS 2006 in Aalborg, Denmark, 15-17 May 2006.
- Ewald, K., & Coors, V. (2005). Appraisal of standards for 3D City Models. Paper presented at the Ninth International Conference on Information Visualisation (IV'05).
- Gröger, G., Kolbe, T. H., Nagel, C., & Häfele, K. H. (2012). OGC city geography markup language (CityGML) encoding standard.
- Gobeawan, L., Lin, E., Tandon, A., Yee, A., Khoo, V., Teo, S., . . . Wise, D. (2018). Modeling Trees for Virtual Singapore: From Data Acquisition to CityGML Models. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, Vol. 42 Issue 4/W10, p55-62.
- Herbert, G., & Chen, X. (2015). A comparison of usefulness of 2D and 3D representations of urban planning. *Cartography and Geographic Information Science*, Vol 42(1), p22-32.
- Hildebrandt, D., & Timm, R. (2014). An assisting, constrained 3D navigation technique for multiscale virtual 3D city models. *GeoInformatica*, 18(3), 537-567.
- Ledoux, Hugo (2019). val3dity: validation of 3D GIS primitives according to the international standards. *Open Geospatial Data, Software and Standards*, 3(1), 2018, pp.1
- Ledoux, Hugo (2013). On the validation of solids represented with the international standards for geographic information.
- Computer-Aided Civil and Infrastructure Engineering, 28(9):693-706.
- Lu, L., Becker, T., & Löwner, M.-O. (2017). 3D complete traffic noise analysis based on CityGML Advances in 3D Geoinformation (pp. 265-283): Springer.
- Mulder, D. T. (2015). Automatic repair of geometrically invalid 3D city building models using a voxel-based repair method. MSc thesis, 111 pages. Delf University of Technology, Netherlands.
- Musliman, I. A., Rahman, A. A., & Coors, V. (2006). 3D navigation for 3D-GIS—initial requirements Innovations in 3D geo information systems (pp. 259-268): Springer.
- NRF (2018). Virtual Singapore. National Research Foundation, Singapore. Accessed June 2020 from <https://www.nrf.gov.sg/programmes/virtual-singapore>
- Nooruddin, F. S., & Turk, G. (2003). Simplification and repair of polygonal models using volumetric techniques. *IEEE Transactions on Visualization and Computer Graphics*, 9(2), 191-205.
- Nouvel, R., Brassel, K.-H., Bruse, M., Duminil, E., Coors, V., Eicker, U., & Robinson, D. (2015). SimStadt, a new workflow-driven urban energy simulation platform for CityGML city models. Paper presented at the Proceedings of International Conference CISBAT 2015 Future Buildings and Districts Sustainability from Nano to Urban Scale.
- Ruohomäki, T., Airaksinen, E., Huuska, P., Kesäniemi, O., Martikka, M., & Suomisto, J. (2018). Smart City Platform Enabling Digital Twin. Paper presented at the 2018 International Conference on Intelligent Systems (IS).
- Sharkawi, K. H., & Rahman, A. A. (2014). Towards an Efficient City Inventory Management System for Urban Authorities in Developing Countries—The Case of 3D Change Detection. *The Electronic Journal of Information Systems in Developing Countries*, 60(1), 1-13.
- Stoter, J. E. (2020). State of the Art in 3D City Modelling, Six Challenges Facing 3D Data as a Platform. *GIM International*. Accessed May 2020.
- van Oosterom, P. (2013). Research and development in 3D cadastres. *Computers, Environment and Urban Systems*, 40, 1-6.
- Vitalis, S., Labetski, A., Arroyo Otori, K., Ledoux, H., & Stoter, J. (2019). A Data Structure to Incorporate Versioning In 3D City Models. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 4.
- Wagner, D., Wewetzer, M., Bogdahn, J., Alam, N., Pries, M., & Coors, V. (2013). Geometric-semantical consistency validation of CityGML models Progress and new trends in 3D geoinformation sciences (pp. 171-192): Springer.
- Yang, P. P.-J., Putra, S. Y., & Li, W. (2007). Viewsphere: a GIS-based 3D visibility analysis for urban design evaluation. *Environment and Planning B: Planning and Design*, 34(6), 971-992.
- Zhao, J., Ledoux, H., Stoter, J., & Feng, T. (2018). HSW: Heuristic Shrink-wrapping for automatically repairing solid-

based CityGML LOD2 building models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 146, 289-304.

Zhao, Z., Ledoux, H., & Stoter, J. E. (2013). Automatic repair of CityGML LOD2 buildings using shrink-wrapping. In 8th 3DGeoInfo Conference & WG II/2 Workshop, Istanbul, Turkey, 27–29 November 2013, *ISPRS Archives Volume II-2/W1*. ISPRS.

Zlatanova, S., & Beetz, J. (2012). 3D spatial information infrastructure: The case of Port Rotterdam. Usage, Usability, and Utility of 3D City Models—European COST Action TU0801, 03010.