Automatically enhancing CityGML LOD2 models with a corresponding indoor geometry

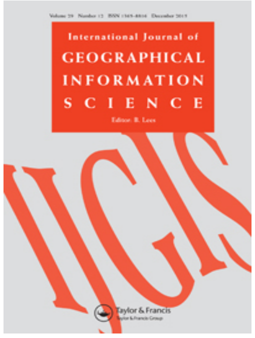
Boeters, R; Arroyo Ohori, GAK; Biljecki, F; Zlatanova, S

DOI
10.1080/13658816.2015.1072201

Publication date
2015

Document Version
Final published version

Published in
International Journal of Geographical Information Science



International Journal of Geographical Information Science

ISSN: 1365-8816 (Print) 1362-3087 (Online) Journal homepage: http://www.tandfonline.com/loi/tgis20

Automatically enhancing CityGML LOD2 models with a corresponding indoor geometry

Roeland Boeters, Ken Arroyo Ohori, Filip Biljecki & Sisi Zlatanova

International Journal of Geographical Information Science, 2015
Vol. 29, No. 12, 2248–2268, http://dx.doi.org/10.1080/13658816.2015.1072201

Automatically enhancing CityGML LOD2 models with a corresponding indoor geometry

Roeland Boeters, Ken Arroyo Ohori*, Filip Biljecki and Sisi Zlatanova

3D Geoinformation, Delft University of Technology, Delft, The Netherlands

(*Received 15 January 2015; accepted 8 July 2015*)

The international standard CityGML defines five levels of detail (LODs) for 3D city models, but only the highest of these (LOD4) supports modelling the indoor geometry of a building, which must be acquired in correspondingly high detail and therefore at a high cost. Whereas simple 3D city models of the exterior of buildings (e.g. CityGML LOD2) can be generated largely automatically, and are thus now widely available and have a great variety of applications, similarly simple models containing their indoor geometries are rare.

In this paper we present two contributions: (i) the definition of a level of detail LOD2+, which extends the CityGML LOD2 specification with indoor building geometries of comparable complexity to their exterior geometries in LOD2; and more importantly (ii) a method for *automatically* generating such indoor geometries based on existing CityGML LOD2 exterior geometries. We validate our method by generating LOD2+ models for a subset of the Rotterdam 3D data set and visually comparing these models to their real counterparts in building blueprints and imagery from Google Street View and Bing Maps. Furthermore, we use the LOD2+ models to compute the net internal area of each dwelling and validate our results by comparing these values to the ones registered in official government data sets.

Keywords: CityGML; 3D city model; indoor; LOD2+; net internal area

1. Introduction

A 3D city model is a computer representation of a mostly urban area, consisting of a collection of buildings, tunnels, bridges and other real-world objects (Döllner *et al*. 2006, Kolbe 2009). One of the main formats used for 3D city models in the GIS domain is CityGML, codified as an international standard defined by the Open Geospatial Consortium (2012). This standard is aimed at the creation of general purpose models that can be used in many different application areas (Gröger and Plümer 2012). The wide applicability of CityGML is enabled by the possibility of representing the same region at multiple levels of detail (LODs). This recognizes the fact that an increase in the detail of a model enables more applications, but comes with correspondingly higher acquisition, modelling and computational costs (Luebke *et al*. 2003, Coltekin and Reichenbacher 2011, Biljecki *et al*. 2014c).

Five LODs are defined in CityGML (LOD0–LOD4). Respectively, these are:

- **LOD0:** 2.5D building footprints and/or roof edge polygons.
- **LOD1:** Extruded footprints (prismatic models).

*Corresponding author. Email: g.a.k.arroyoohori@tudelft.nl

- **LOD2:** Simple models with differentiated roof structures and semantically enriched boundary surfaces.
- **LOD3:** Detailed architectural models with openings such as windows and doors.
- **LOD4:** LOD3 models with similarly detailed indoor geometries of buildings.

There is a progressive increase in the geometric detail and spatio-semantic coherence in the different LODs in CityGML (Stadler and Kolbe 2007). However, this applies only to the exterior of buildings, as their indoor information is only available in the highest LOD (LOD4). This does not correspond to actual application requirements, since there are many applications that *do* require indoor building geometries but *do not* require the (very high) level of indoor detail defined in CityGML LOD4. In fact, as the related work described in Section 2 shows, there is significant interest in modelling indoor building geometries at a level of detail lower than CityGML LOD4.

Among the LODs in CityGML, LOD2 models are notable because of two factors: (i) they are the ones with the highest level of detail that is commonly available in practice; and (ii) despite their relative simplicity they have a very wide range of applications. The availability of LOD2 models is largely explained because they are the highest LOD that is not difficult to generate automatically on a large scale. For instance, LOD2 models can be automatically generated from a combination of: airborne LiDAR point clouds (Rottensteiner 2003), 2D building footprints (Aringer and Roschlaub 2014), terrestrial LiDAR (Akmalia *et al*. 2014), aerial images (Hammoudi and Dornaika 2011) and freely available mapping information such as OpenStreetMap (Goetz and Zipf 2012). Some example applications of LOD2 models include: estimation of the solar potential of roofs (Santos *et al*. 2014), noise pollution analysis (Stoter *et al*. 2008), shadow modelling (Alam *et al*. 2013) and urban planning (Buhur *et al*. 2009).

However, when LOD2 models are supplemented *with indoor building geometries that have a level of detail comparable to their exterior geometries*, consisting of storeys within a building (Vandysheva *et al*. 2012), which we refer to in this paper as an *LOD2+ model* and define more extensively in Section 3, their applicability increases significantly. While some of the indoor information, such as the number of storeys, may be available in the form of attributes in the present models, having the storeys modelled geometrically enhances their usability. For example, such models could be used in urban heat simulations (Kastendeuch and Najjar 2009), the estimation of inhabitants per building (INSPIRE Thematic Working Group Buildings 2013), the perception of space and green areas from apartments (Yasumoto *et al*. 2011, Fisher-Gewirtzman 2012), heating demand estimation based on the volume to be heated (Kaden and Kolbe 2014), mass valuation of real estate (Tomić *et al*. 2012) and firefighting simulations (Chen *et al*. 2014).This bypasses the current need for high-detail LOD4 models, which are expensive to create and are usually not available. For instance, in emergency response it is crucial to determine on which floor a responder is located, which can be achieved with basic spatial operations on a LOD2+ model – which can be generated in a fairly easy manner – unlike LOD4 models for which extensive additional information is needed.

Moreover, by proposing a method to create LOD2+ models from LOD2 models and widely available ancillary data automatically, which is presented in Section 4, we hope to make such models widely available as well. The method that we have developed, originally described in Boeters (2013), does not require any additional geometric information apart from that already included in an LOD2 model. However, this method can optionally use ancillary data to improve the accuracy of the generated models, if such data are available. The method is robust and has been implemented successfully using Nef polyhedra from the Computational Geometry Algorithms Library (CGAL).[1] The implementation is open source and is available at https://github.com/tudelft3d/lod2plus.

In order to demonstrate the usefulness of the newly introduced LOD2+, we present an application in Section 6. Using the generated LOD2+ models, we compute the *net internal area* (NEN 2007) for each residence in a subset of the Rotterdam 3D LOD2 data set, a property enshrined in Dutch law that specifies what surface area can be used by its owner. This area has important implications for taxation and sale, and it is currently computed mostly manually. Our tests have shown that the data currently available in the registers contain large errors. We conclude with a discussion and some recommendations for future work in Section 7.

2. Related work

Work related to this paper exists in two forms: research efforts into the extension of the CityGML and other 3D standards with lower-than-LOD4 detail, and these work on the automatic generation of the indoor geometry. They are reviewed independently in the following sections.

2.1. Indoor models with lower-than-LOD4 detail

The representation of the indoor geometries of buildings in CityGML is not particularly well defined. Theoretically, CityGML LOD4 may contain anything from simple indoor geometries to detailed representations of furniture. However, LOD4 intrinsically requires a detailed exterior (and available models all conform to this practice), which means that there is still a lack of CityGML indoor models with lower-than-LOD4 detail. Researchers in the 3D GIS community recognize this problem and propose various levels of indoor detail. Their efforts are described below.

Hagedorn *et al*. (2009) describe a modelling scheme of four levels of detail destined for indoor routing, where LOD1 models individual storeys in 2D, LOD2 models individual rooms with doors and windows in 2D, LOD3 does the same in 3D and LOD4 contains architectural indoor models similar to CityGML LOD4.

Kemec *et al*. (2012) present their ideas on an LOD system for indoor building geometries in relation to disaster risk communication. They define three new LODs (1.5 with storeys, 2.5 with indoor buildings parts and 3.5 with apartments) that complement the standard LODs 1, 2 and 3, respectively.

Kang and Lee (2014) propose four LOD representations for indoor spatial data, ranging from floor blueprints (e.g. for indoor navigation) to architecturally detailed representations suited for indoor facility management.

Benner *et al*. (2013) and Löwner *et al*. (2013) propose the decomposition of CityGML LODs by focusing on four exterior and four indoor LODs, each semantically decomposed, resulting in numerous combinations.

Billen *et al*. (2012) propose three new internal levels of detail which are similar to the aforementioned examples, but add emphasis on the connections (openings) between the exterior and indoor features in their most detailed LOD.

2.2. Automatic modelling of the indoor geometry

Various methods are available to automatically model the indoor geometry of a building. The most common technique is to construct them from architectural drawings. Yin *et al*.

(2009) review various methods of doing so, which work on the basis of rasterized floor plans, CAD documents or structural construction drawings. Similar work based on detailed CAD floor plans has been done by Nagel (2014) and Bleifuss *et al*. (2009). However, extracting quality vector geometries from raster images is difficult, and as different symbols are used for features like windows and doors, these methods do not work for all drawings, requiring extensive interactive input from users in order to produce working 3D models.

Goetz (2013) describes a method of using volunteered geographic information to extend the building mapping features of OpenStreetMap in order to describe the indoor structure of buildings. The author also developed software where indoor building geometries can be semi-automatically generated from building evacuation plans or building blueprints.

Another possibility is using laser data and/or photographs acquired from inside the building (Okorn *et al*. 2009, Budroni and Böhm 2009, Xiong and Huber 2010, Khoshelham and Elberink 2012, Becker *et al*. 2013, Mura *et al*. 2013, Khoshelham and Diaz-Vilariño 2014, Rosser *et al*. 2015), which can be done by segmenting horizontal structures for the floor and ceiling and using the remaining points as potential information on the walls, or by applying shape grammars, among other possibilities. Johnston and Zakhor (2008) describe a similar method, but using laser scans from the outside of a building. As a limited number of laser pulses can enter the building via windows, planes can be fit through these points and used to reconstruct part of its indoor geometry.

Finally, for applications where realistic looks are more important than actual realism (e.g. gaming), Martin (2005) presented a method to apply procedural modelling to automatically produce floor plans.

While many of these solutions are capable of generating building indoor models of good quality, apart from the generation from floor plans, they are very difficult to apply to LOD2-like models on a large scale as they require either substantial manual work or entering the buildings.

3. A CityGML LOD2+ indoor model

3.1. Limitations of CityGML

Currently CityGML does not support storing a simple exterior geometry of a building with a correspondingly simple indoor geometry. The indoor information can be stored only in a LOD4 model, which is described as an extension of the LOD3 model with an architecturally detailed indoor geometry. Strictly speaking, CityGML does not provide any of the LODs with minimum requirements, and hence a simple exterior *could* be stored in an LOD4 model. However, this would run contrary to generally accepted practices.

While CityGML 2.0 does not support storeys, the standard (Open Geospatial Consortium 2012, p. 76) suggests an alternative method by using `CityObjectGroup` as a generic way to represent storeys by aggregating more detailed indoor features such as rooms. While we do not handle rooms, this method could be applied to aggregate all the surfaces of a storey. In our approach we do not follow this recommendation, in order to be consistent with the next version of the standard, which will likely include storeys.

3.2. Defining an LOD2+ model

LOD2 is not defined in the CityGML standard in a very precise manner. The standard describes LOD2 as a model with ‘differentiated roof structures and thematically

Table 1. Definition of the LOD2+ model by extending comparable concepts from the LOD2.

Exterior in LOD2	Interior in LOD2+
Buildings bodies are prisms	Storeys within building bodies are prisms
Simple roof shapes	Attic storey shapes corresponding to roof shapes
Thematically classified boundary surfaces	Thematically classified boundary surfaces
No openings in the exterior geometry	No openings in the indoor geometry

differentiated boundary surfaces'. Apart from this, the specifications are flexible. For instance, walls in LOD2 may represent projections from roof edges or the actual footprint of buildings, which may have a drastic impact on the intended application (Biljecki *et al.* 2014b). Neither is the representation of indoor geometries in CityGML well defined (partially discussed in Section 2).

As the goal of this paper is to create a joint exterior and indoor level of detail definition, we propagate the flexibility of the LOD2 definition to LOD2+, aiming for a corresponding indoor geometry to the LOD2 exterior geometry. Hence, here we define LOD2+ as an extension of LOD2 with an analogous indoor geometry, which lies fully within the boundaries of the LOD2 exterior, and whose spatio-semantic complexity is comparable to the spatio-semantic complexity of LOD2. Within these constraints, we purposefully define LOD2+ such that it can be modelled as automatically as LOD2.

The LOD2+ indoor geometry, containing a storey as its main feature, is defined in Table 1 by translating the spatio-semantic characteristics of an LOD2 model. Figure 1 shows a visual representation of LOD2+ .

3.3. Extending CityGML with an LOD2+ model

There are three ways to implement the described LOD2+ in CityGML 2.0: (1) store it as an LOD4 where the features are aggregated as multiple instances of `CityObjectGroup`; (2) develop an application domain extension (ADE); and (3) extend the CityGML 2.0 schema.

The first solution, while technically possible, is not desirable as it would exclude the joint storage of an LOD2+ and LOD4. The second solution extends CityGML by employing the ADE mechanism, which is being increasingly used by the 3D city

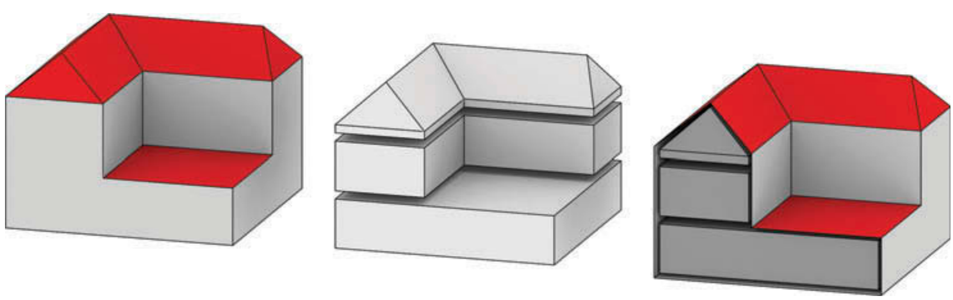


Figure 1. The LOD2+ model (shown on the right) is defined as the extension of the LOD2 model (left) with a corresponding indoor level of detail (middle). For LOD2+, model volumes are fitted inside the exterior shell, representing storeys. The storeys, being simple, can be automatically determined from the exterior characteristics of the LOD2 model using the methodology described in Section 4.

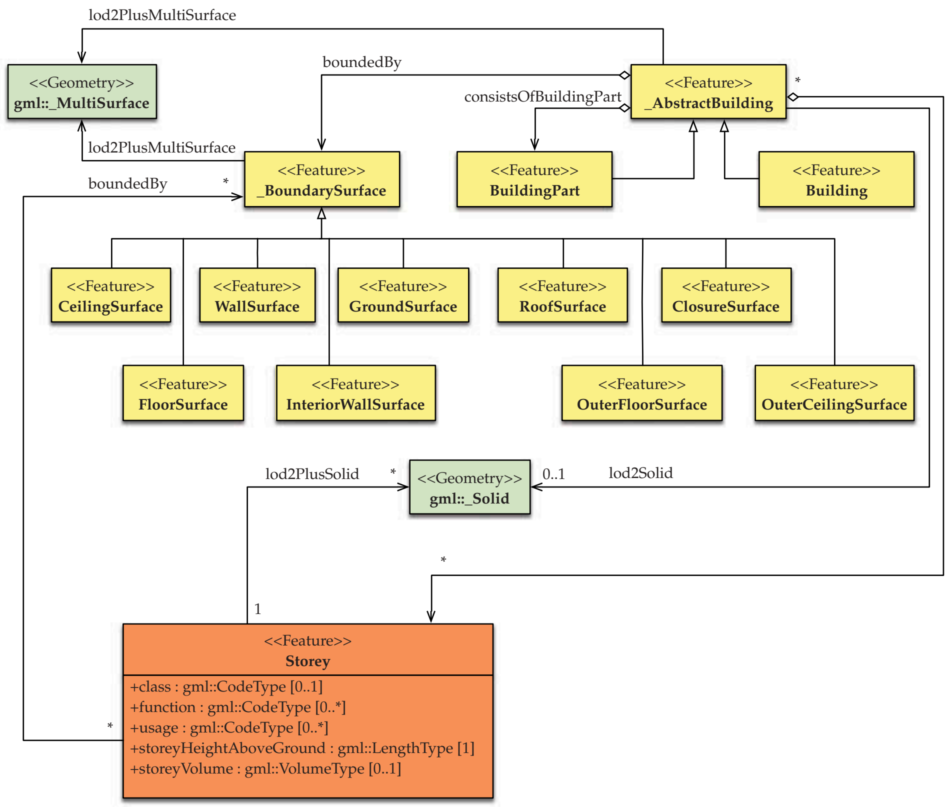


Figure 2. Simplified version of the CityGML Building module (yellow and green) with the LOD2+ extension (orange). Note that the surfaces can be reused in multiple storeys to increase consistency and reduce the storage footprint (Biljecki *et al.* 2015).

modelling community (Schulte and Coors 2009, Çağdaş 2013, Van Den Brink *et al.* 2013, Kim *et al.* 2014).

However, we have decided to proceed with the third solution of extending the CityGML 2.0 schema because we support the inclusion of storeys in the next version of the standard (Löwner *et al.* 2014), and believe that the results of this research might be important for the developers of the standard. A simplified UML diagram of the Building module with the LOD2+ extension is shown in Figure 2.

For the best results, each storey should be modelled as a valid shell according to the ISO 19107 standard (ISO 2003), such that calculations (e.g. volume computations) can be done. The LOD2+ building solid will then have holes (inner shells) which represent the storeys. Each face of the storey solid can be classified with existing feature types: `FloorSurface`, `InteriorWallSurface` and `CeilingSurface`. We believe that as indoor building geometries are inseparable from the exterior ones, the solution should be part of the CityGML standard.

4. Automatic generation of LOD2+ models from LOD2 models

In this section we explain our method of generating a LOD2+ model from a CityGML LOD2 model by following a series of geometric operations on volumetric (space-filling) objects. The approach consists of the following steps: repairing the CityGML LOD2 geometries, creating (solid) Nef polyhedra from them, carving out the hollow spaces that represent storeys and classifying the faces from the resulting Nef polyhedra. The validation is performed with City Doctor (Wagner *et al*. 2013), while the geometric computations use the Computational Geometry Algorithms Library (CGAL)[2]– mainly the Nef polyhedra (Bieri and Nef 1988) package. After the LOD2+ volumes have been obtained, individual faces are extracted from the Nef polyhedra and the correct semantic information for a (b-rep) LOD2+ model is obtained by classifying these faces. These steps are described in detail in the following sections.

4.1. Repairing the building geometries in the input LOD2 data

The geometries that are stored in the CityGML format are supposed to conform to the ISO 19107 definitions (ISO 2003) for polygons (as `GM_Ring` and `GM_Polygon`) and polyhedra (as `GM_Shell` and `GM_Solid`), which should ensure that these form valid 2D and 3D objects (Ledoux 2013). Mainly, a polygon should be closed ('[A GM_Ring] consists of a number of references to GM_OrientableCurves connected in a cycle'), and non-self-intersecting ('each ring is simple'). Similarly, a polyhedron should be closed ('[A GM_Shell] consists of a number of references to GM_OrientableSurfaces connected in a topological cycle') and non-self-intersecting ('GM_Shells are simple'). The standard also makes it possible to specify other practical requirements, such as a polygon being planar ('The default [surface in which to embed a polygon] is that [the rings of a polygon] are coplanar').

However, the validity criteria are difficult to enforce. Because of this, invalid geometries in CityGML models are prevalent in practice, such as buildings with missing faces (which therefore do not enclose any space). As the creation of Nef polyhedra (or any other volumetric representation) from CityGML requires the input to consist solely of valid volumes with planar faces, an automated polyhedra repair tool needs to be used.

For the purposes of this research, City Doctor (Wagner *et al*. 2013) was first used, which attempts to ensure that:

- A linear ring consists of at least four ordered points.
- All points of the linear ring are different except for the first and the last.
- Edges can only intersect at their start or end point.
- All faces of the solid need to be planar.
- A solid consists of at least four polygons.
- The normal vector of each face points out of the solid.
- All polygons bounding a solid should be connected.
- Each point is surrounded by one cycle of alternating edges and faces.

The results depend on the data set, but this procedure does manage to repair many problematic buildings. However, problems are still present in many cases, such as when multiple adjacent non-co-planar faces are missing in a shell, or when there are dangling faces along an edge, among others. Moreover, a tolerance needs to be set to define which faces are co-planar (set to 0.01 m), resulting in faces that are still not precisely co-planar.

Therefore, in order to increase the number of valid geometries, we also repair the following problems:

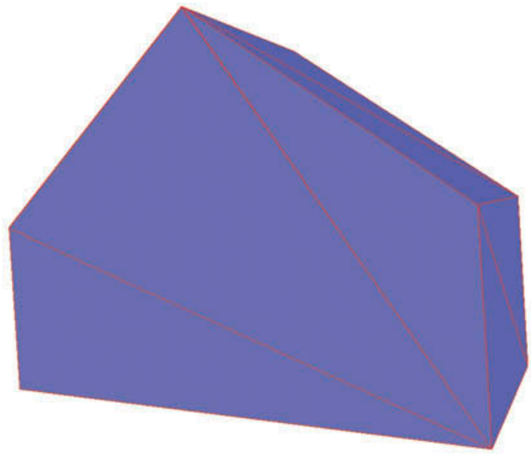


Figure 3. Each face of each polyhedron is triangulated to ensure that they are perfectly planar.

- Removing duplicate vertices by snapping those that are within a given tolerance threshold.
- Splitting non-planar faces into triangles (which are by definition perfectly planar) using a constrained triangulation (Figure 3).
- Adding missing faces to create closed polyhedra.

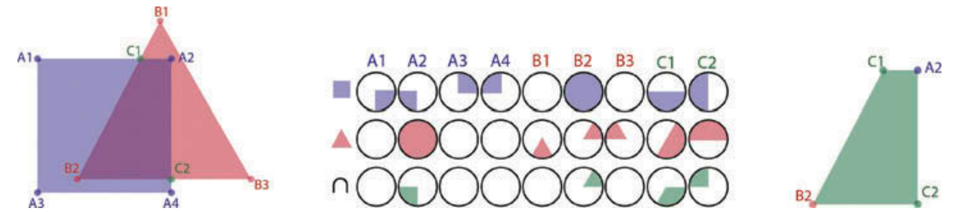
4.2. Conversion from CityGML to Nef polyhedra

The next step in our method is to transform the CityGML geometries into Nef polyhedra. CityGML uses a boundary representation (b-rep) scheme for its geometry in which all the polygonal faces of a volume are represented independently. Such an approach works well for exchange file formats such as CityGML, but applying geometric operations efficiently and robustly generally requires a representation with explicit knowledge about the topological relationships among faces and whether the interior or exterior lies on a given side of a face.

Nef polyhedra (Bieri and Nef 1988) fulfil these requirements in an elegant way. It is an n-dimensional data model that supports the representation of all n-dimensional polytopes that can be defined by combining point set intersection and complement operations on a finite number of open half-spaces (Arroyo Ohori *et al*. 2015b). The objects created with these can be then represented on a computer using the concept of a local pyramid, which can store the neighbourhood information around every vertex by projecting this neighbourhood onto the surface of an infinitesimally small hypersphere centred on the vertex.

In practice, this means that volumetric objects can be represented using two sets of simple 2D structures, which respectively contain a topological representation of the vertices/edges/faces of a volume and the edges/faces/volumes around a vertex. Furthermore, as shown in Figure 4, Boolean set operations (i.e. the union, intersection and complement) and Minkowski sums can be performed and stored at the local pyramid level.

This fact allows these geometric operations to be performed robustly, as they have good support for complex geometries, such as non-manifolds and disjoint volumes, and it



(a) The vertices from the input polygons and those at the intersection points of the edges of the polygons are found.

(b) The local pyramids for all candidate vertices are computed for each polygon, and their intersection is computed as the result.

(c) The end result is obtained from the computed local pyramids for the intersection.

Figure 4. Computing a polygon intersection operation on Nef polyhedra.

is easy to perform operations on them while ensuring that their output consists of a valid set of volumes.

In our implementation, if an input geometry forms a valid polyhedron or is possible to repair successfully, it is loaded into the half-edge `Polyhedron_3` data structure in CGAL (Kettner 1999), by incrementally adding adjacent faces to the polyhedral structure. From this data structure, it is possible to use functionality available in CGAL to create a `Nef_polyhedron_3` (Granados *et al.* 2003, Hachenberger 2006), which internally calculates a plane for each face (equivalent to a curve on the local pyramid of a vertex incident to it), and joins these curves in order to create the required internal structures.

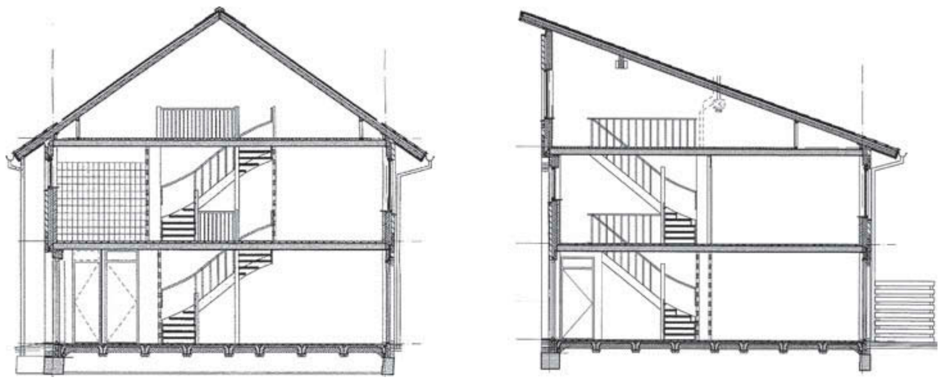
4.3. Geometric operations to obtain a volumetric representation of LOD2+

The first rule in the process of LOD2+ generation is to split a building into storeys with no space between them. For this, we primarily use the number of storeys of a building – something that can be obtained from registration data or from the model itself (Nouvel *et al.* 2014). However, when other heuristics are available, these can be incorporated into the process in order to achieve better results.

In our case, the only other heuristic used for this part of the process is a set of characteristic heights at which storeys can be reasonably expected to start/end. In the Netherlands, the eaves of a tilted roof (Figure 5) and the heights of house extensions (Figure 6) are typically good indicators of characteristic heights.

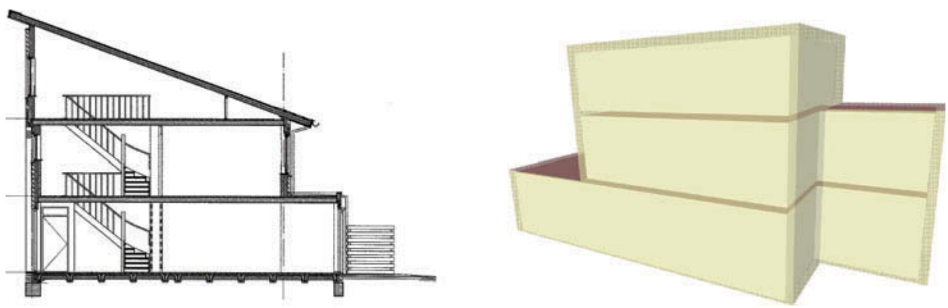
Based on the assumption that the storeys are of similar height, preliminary height values for each storey are obtained. These are then iteratively snapped to match the characteristic heights obtained previously if they are within a certain threshold distance and do not create overly tall or short storeys, after which the non-snapped storey heights are again distributed evenly. Using these heights, as shown in Figure 7, each storey of a building can be obtained by performing a set intersection operation between the (volumetric) building and a box-shaped polyhedron that is known to cover the building horizontally (e.g. an extrusion of its footprint), having bottom and top faces that are parallel to the ground plane.

After this, each (volumetric) storey is reshaped by subtracting from it the spaces occupied by the (thick) walls, floor and ceiling. Approximate thickness values for each of



(a) (b)

Figure 5. Two different buildings where the highest storey starts at the eaves of the roof. Taken from BAG Rotterdam.



(a) (b)

Figure 6. Two different buildings where the height of an extension indicates the height of a storey. Blueprint (a) taken from BAG Rotterdam.

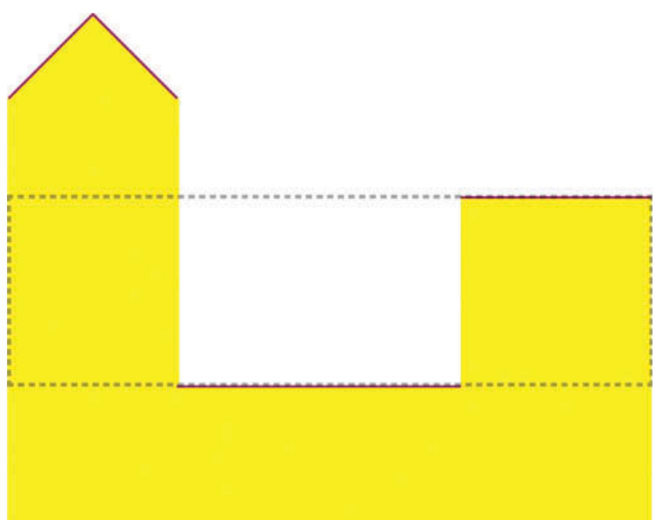


Figure 7. A set intersection operation used to obtain each storey of a building. Note that the output of this operation can produce multiple disjoint polyhedra.

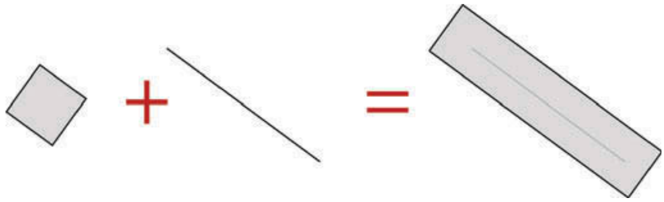


Figure 8. The Minkowski sum of a rotated square kernel with a line resulting in a correctly oriented line with the thickness of the kernel.

these can be used based on available data, such as the age of the building, construction materials, type of building (e.g. high-rise) and building height. This subtraction is done by first computing the Minkowski sum of each face of the building and a cubical kernel of the corresponding thickness that has been rotated to be oriented along the face (as shown in Figure 8), which is then subtracted from the volumetric storey. These operations can also be performed robustly on CGAL Nef polyhedra, since a Minkowski sum of a polyhedron and a kernel can be computed by doing a convex decomposition of the polyhedron, computing the Minkowski sum of each convex part and the kernel, and computing their set union (Hachenberger 2006, De Berg *et al*. 2008).

4.4. Surface classification

After the volumes for each storey have been obtained, every face of each storey volume is classified into its corresponding CityGML class: `CeilingSurface`, `InteriorWallSurface` or `FloorSurface`. Similarly, each face of the repaired input volume, representing the exterior of the building, is classified as `RoofSurface`, `WallSurface` or `FloorSurface`. This is done using a simple method that computes this based on the normal vector of each face (as shown in Figure 9), which nevertheless yields a correct classification in almost every case.

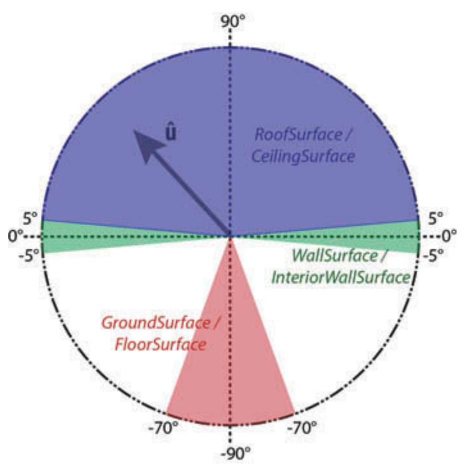


Figure 9. Classification of the surface performed on the basis of the pitch angle of the normal vector of each face. The angles in the figure show which angle results in which surface type.

5. Creation of LOD2+ models from Rotterdam 3D

We used the methodology described in the previous section in order to generate LOD2+ models from a subset of the CityGML LOD2 data set Rotterdam 3D.[3] These were validated by visually inspecting a sample of the resulting models, comparing them to building blueprints, street-level imagery from Google Street View and oblique aerial views from Bing Maps.

Rotterdam 3D is the 3D model of the city of Rotterdam, created on the basis of building footprints from the *Basisregistratie Adressen en Gebouwen* (BAG – key register for addresses and buildings)[4] data set and point cloud data with a point density of at least 15 points per m^2 in the harbour area of Rotterdam, and at least 30 points per m^2 elsewhere. In this case we used a subset of Rotterdam 3D consisting of the area of Hoogvliet-Zuid, which is shown in Figure 10 . This is the southern part of a borough in the southwest of Rotterdam, and was selected as it has varied building types, such as terraced houses, flats, high-rises and detached houses. It also has buildings with a significant difference in their construction date due to recent urban renewal in the borough. Each building has an identifier that links the buildings to their counterparts in the BAG data set, and their geometries contain roof, wall and ground surface members. The buildings are not valid solids as they contain various problems, such as no walls being modelled at places where neighbouring buildings touch (e.g. terraced houses).

Using the methodology explained in Section 4.1, the models were first repaired with valid volumes of 71.7% obtained. Since no additional validation was done by us at this step, some invalid buildings were still processed. Depending on the building, this resulted in either incorrect or correct results due to the automatic repair performed in CGAL. The generated volumes were then loaded into Nef polyhedra and characteristic heights were obtained. In cases where the BAG data set indicated underground levels in the building, the volumes were extended downwards as the Rotterdam 3D data set does not model underground levels.

In order to determine appropriate thickness values, the buildings were first classified into similar subsets by their type, construction year and number of storeys, and a sample of building blueprints were analysed for each set. The aim was to inspect at least 3% of the buildings in each set. Unfortunately, this was not possible for all of them, as for some sets there were no building blueprints available or some blueprints were of very poor quality. Based on these, representative thickness values were obtained for external walls (t_{ext}) and shared walls (t_{shared}). These are shown in Table 2.

Figure 10. Hoogvliet-Zuid, a subset of the Rotterdam 3D data set, shown with textures.

Table 2. Input thicknesses determined from building blueprints for different building categorizations.

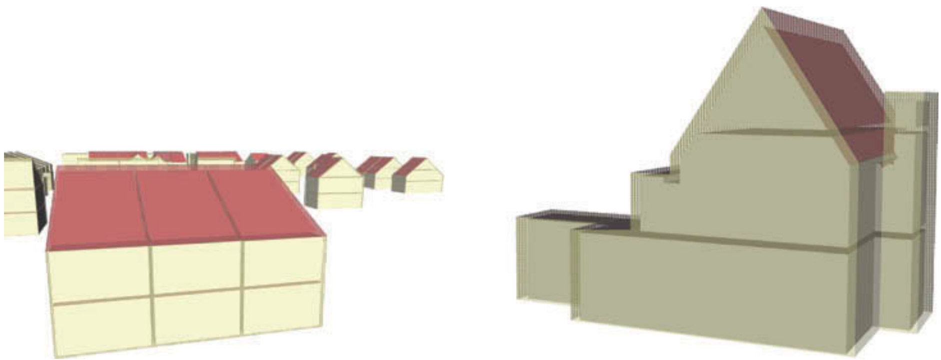
Type	Year y	Storeys x	t_{ext} [cm]	t_{shared} [cm]
Non-stacked	$y < 1970$	$x \leq 2$	27	11
		$x \geq 3$	27	12
	$1970 \leq y \leq 1985$	$x = 2$	27	10
		$x = 3$	28	12
		$x = 4$	27	9
	$y > 1985$	$x = 2$	28	13
		$x = 3$	30	12
		$x = 4$	25	12
Stacked	$y < 1970$	$x \leq 5$	29	12
		$5 < x \leq 10$	38	11
		$x > 10$	25	9
	$1970 \leq y \leq 1985$	$x \leq 5$	28	11
		$5 < x \leq 10$	26	11
		$x > 10$	29	12
	$y > 1985$	$x \leq 5$	30	12
		$5 < x \leq 10$	38	13
		$x > 10$	35	15
Other types	$y < 1970$	$x = 1$	14	14
		$x \geq 2$	31	11
	$1970 \leq y \leq 1985$	$x = 1$	14	14
		$x \geq 2$	30	10
	$y > 1985$	$x = 1$	14	14
		$x \geq 2$	36	13

Unfortunately it was not possible to determine values for the thickness of the ceiling/floor and roof using this method, due to a lack of sufficient blueprints with side-views. However, realistic values were assigned for these parameters based on consulting building experts from Rotterdam. For all buildings, the roof thickness was therefore set to 30 cm and the ceiling/floor thickness to 20 cm. Since all storeys are initially the same height, the ceiling/floor thickness is shared between two storeys resulting in a floor thickness of 10 cm and a ceiling thickness of 10 cm. However, the thickness for the classified `GroundSurface` is set to 1 cm to ensure that the ground level floor does not intersect the bottom plane of the exterior shell, but is still practically at the same height as the terrain.

The LOD2+ models for all buildings in the data set were therefore generated using these values for the heights and thickness values, with some of the results shown in Figures 11–13.

6. Case study: computing and validating the net internal area

To showcase the utility of our LOD2+ definition and in order to further validate our method, we used the generated models used to estimate the net internal area (*gebruiksoppervlakte*) as defined in the Dutch national standard NEN 2580:2007 (NEN 2007). This area has important implications for taxation and sale, and it is currently computed mostly manually. It represents the surface area of a residence that can be used, disregarding load-bearing structures, locations where the net height is lower than 1.5 m, large voids between storeys, and elevator/pipe/cable shafts.

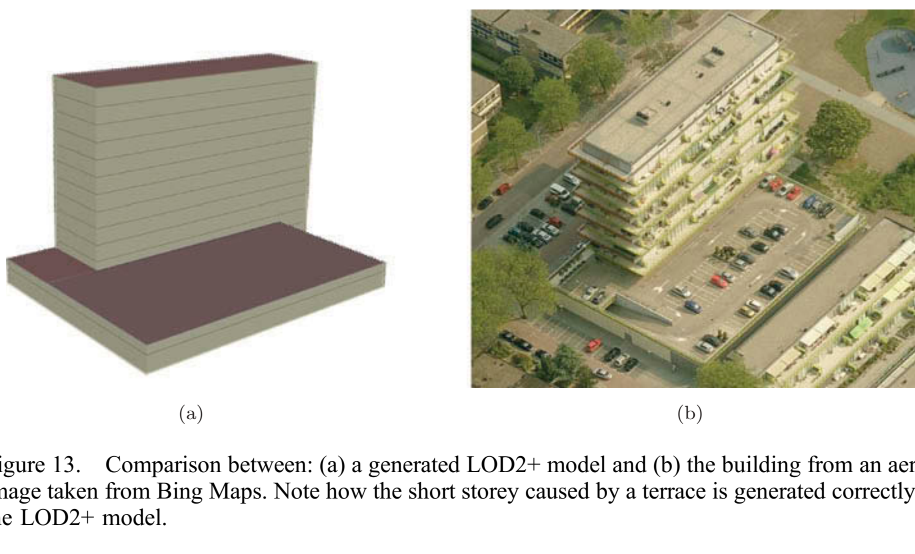


(a) (b)

Figure 11. LOD2+ models for various gabled houses. Because the exterior shell is transparent, the storey solids are visible.

(a) (b)

Figure 12. Comparison between: (a) a generated LOD2+ model and (b) the building from an aerial image taken from Bing Maps.



(a) (b)

Figure 13. Comparison between: (a) a generated LOD2+ model and (b) the building from an aerial image taken from Bing Maps. Note how the short storey caused by a terrace is generated correctly in the LOD2+ model.

Unfortunately, some load-bearing structures (e.g. internal structural walls and columns), large voids and shafts cannot be directly obtained using this method. However, as these values are not expected to greatly affect the net internal area, our value would be expected to be only a slight overestimation compared with the registered values, which are included in both the BAG and *Waarde Onroerende Zaken* (WOZ – value of immovable property)[5] official data sets. By comparing a large number of the registered values in BAG and WOZ to those computed using our LOD2+ models, it is possible to further validate our methodology for generating these models.

Based on our generated LOD2+ models, the net internal area was computed for all residences by calculating the total surface area and subtracting areas where the height of the ceiling is less than 1.5 m. This was also done by subtracting from them a 1.5 m-high box-shaped polyhedron at the appropriate height and computing the area of the base on the resulting polyhedron.

On average, the entire process including the determination of the net internal area takes 16 seconds per building on a computer system with a quad-core CPU of 3.20 GHz and 8 GB of RAM. Note, however, that the developed software does not utilize all CPU cores. In order to process the entirety of the Rotterdam 3D data set, which has approximately 125,000 buildings, it would take approximately 23 days to produce a LOD2+ model of the entire city using a single computer.

Figures 14 and 15 show the automatically computed net internal area compared to that registered in the BAG data set. Surprisingly, it can be seen that although not all parameters from the definition of net internal area are taken into account because of a lack of information, for the majority of the buildings the net internal area is *smaller than the values registered in BAG*. Furthermore, only 38% of the buildings have values that are within the $1.15\sqrt{A}$ tolerance that is allowed in the NEN 2580 standard.

Upon further investigation, it was discovered that for many buildings the net internal area calculated from the LOD2+ model would be very close to the values from BAG if the surface area where the net height is less than 1.5 m was not subtracted. In this case 77% of the buildings satisfy the allowed tolerance. Since some municipalities began registration of net internal area before NEN 2580 was introduced, we believe that some of these areas were registered using a different net surface area definition that did not exclude the areas where the net height is less than 1.5 m. For many buildings the net internal area should probably need to be recalculated in order to comply with NEN 2580.

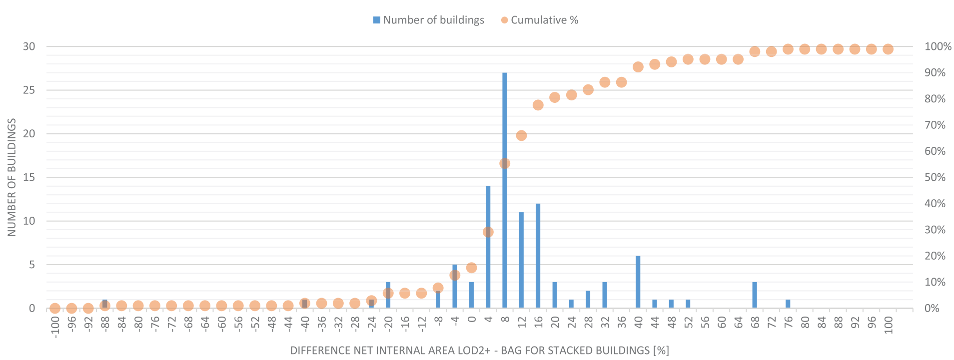


Figure 14. Histogram of the differences between the net internal area calculated for stacked buildings from the LOD2+ model and BAG. When the difference is larger than zero, the net internal area of the LOD2+ buildings is larger than that of BAG.

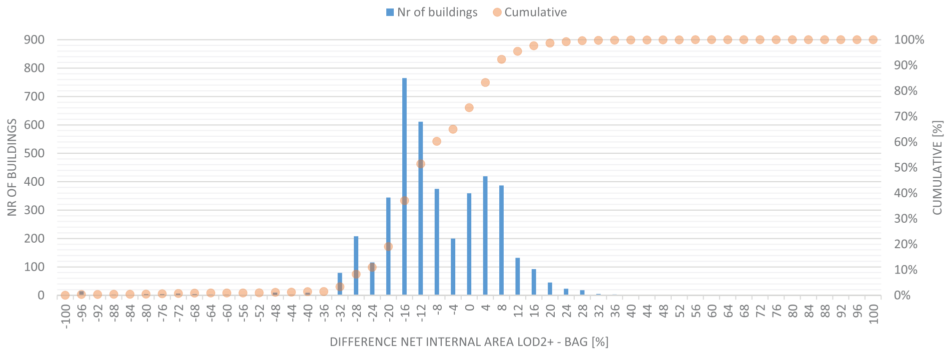


Figure 15. Histogram of the differences between the net internal area calculated for non-stacked buildings from the LOD2+ model and BAG. When the difference is larger than zero, the net internal area of the LOD2+ buildings is larger than that of BAG.

Tests comparing our obtained value for the net internal area with that registered in WOZ show similar results, with WOZ data (now being discontinued) being a somewhat better fit to our computed values. More detailed information on all tests is available in Boeters (2013).

In order to determine the reasons for the largest differences between computed and registered net internal area, a random sample from the 5% of the building models causing the largest differences were examined manually and compared to their blueprints and street imagery from Google Street View. In decreasing order of importance, the results of this analysis indicate that the difference is mostly due to: (i) the surface area where the net height is below 1.5 m not being subtracted from the net internal area in BAG; (ii) large step wells being present in the building; (iii) the roof not being correctly modelled in the 3D model; and (iv) the building still being under construction at the time of laser data acquisition. More detailed information on this validation is also available in Boeters (2013).

7. Discussion and future work

The definition of a CityGML LOD2 that also contains indoor building information with a level of detail comparable to its exterior information opens the door for the many applications that do require indoor building information but do not require the (very high) level of indoor detail defined in CityGML LOD4. As these models, referred to in this paper as LOD2+, can be generated in a similarly automated manner to LOD2 models, their creation does not come with significantly increased acquisition costs compared with LOD2 models. Moreover, their increased applicability is in line with the stated objective of CityGML, which is to create general purpose models that can be used in multiple application areas.

We have presented a method to automatically create LOD2+ models from LOD2 which is robust, using Nef polyhedra in order to support geometric operations on volumetric objects. Since the vast majority of 3D city models do not currently contain indoor buildings geometries (Morton *et al.* 2012), this method could also contribute to an increase in the use of models where this information is available. Our method of generating these models was implemented using CGAL, and the implementation has been released as an open source and is freely available at https://github.com/tudelft3d/lod2plus.

Our definition of LOD2+ has been realized in the CityGML procedural modelling engine developed by Biljecki *et al*. (2014a), as another automatic method for deriving models in this LOD.

This method has been tested in a case study with a subset of the Rotterdam 3D data set, using the generated LOD2+ models in order to compute the net internal area of each residence, which was cross-validated with the values registered in the official BAG and WOZ data sets. Based on these areas, it was discovered that the registered values for many buildings do not seem to conform to the NEN 2580 standard.

An important challenge in increasing the applicability of such a method is the fact that buildings in many CityGML models do not form valid closed objects. A future possibility to ensure that this is the case is to repair the model using shrink-wrapping (Zhao *et al*. 2013), which uses carving operations on a constrained tetrahedralization of the input. This should always be able to generate a valid volume from *arbitrary* input, but in extreme cases the output can have a significantly changed shape to the input.

In the future, we plan to similarly extend CityGML with LOD1+ and LOD3+ models, with a level of indoor detail corresponding to LOD1 and LOD3, respectively, and to investigate their possible applications. For generating the LOD3+ model we plan to investigate the use of procedural modelling for the generation of indoor geometries (Gröger and Plümer 2010, Becker *et al*. 2013, Peter *et al*. 2013, Ilčk and Wimmer 2013). We also plan to investigate whether LODx+ fits within the framework of continuous LODs (Arroyo Ohori *et al*. 2015a).

Acknowledgements

We would like to thank the anonymous reviewers who greatly improved the paper with their very helpful comments.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This research was financially supported by the Dutch Technology Foundation STW, which is part of the Netherlands Organisation for Scientific Research (NWO), and which is partly funded by the Ministry of Economic Affairs (project code 11300).

Notes

1. http://www.cgal.org
2. http://www.cgal.org
3. http://www.rotterdam.nl/rotterdam_3d
4. http://www.kadaster.nl/bag
5. http://www.kadasterdata.nl/woz-loket

ORCID

Roeland Boeters http://orcid.org/0000-0002-7736-3593
Ken Arroyo Ohori http://orcid.org/0000-0002-9863-0152
Filip Biljecki http://orcid.org/0000-0002-6229-7749
Sisi Zlatanova http://orcid.org/0000-0002-8766-0487

References

Akmalia, R., *et al.*, 2014. TLS for generating multi-LOD of 3D building model. *IOP Conference Series: Earth and Environmental Science. Proceedings of the 8th International Symposium of the Digital Earth (ISDE8)*, 18, 1–8.

Alam, N., Coors, V., and Zlatanova, S., 2013. Detecting shadow for direct radiation using CityGML models for photovoltaic potentiality analysis. *In*: C. Ellul, *et al.*, eds. *Urban and regional data management*. Boca Raton, FL: CRC Press, 191–196.

Aringer, K. and Roschlaub, R., 2014. Bavarian 3D building model and update concept based on LiDAR, image matching and cadastre information. *In*: U. Isikdag, ed. *Innovations in 3D Geo-information sciences*. Cham: Springer International Publishing, 143–157.

Arroyo Ohori, K., *et al.*, 2015a. Modeling a 3D city model and its levels of detail as a true 4D model. *ISPRS International Journal of Geo-Information*, 4, 1055–1075. doi:10.3390/ijgi4031055

Arroyo Ohori, K., Ledoux, H., and Stoter, J., 2015b. An evaluation and classification of nD topological data structures for the representation of objects in a higher-dimensional GIS. *International Journal of Geographical Information Science*, 29 (5), 825–849. doi:10.1080/13658816.2014.999683

Becker, S., *et al.*, 2013. Combined grammar for the modeling of building interiors. *In*: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences. Proceedings of the ISPRS Acquisition and Modelling of Indoor and Enclosed Environments workshop*, December. Cape Town: ISPRS, 1–6.

Benner, J., *et al.*, 2013. Enhanced LOD concepts for virtual 3D city models. *In*: U. Isikdag, ed. *ISPRS annals. Proceedings of the ISPRS 8th 3D GeoInfo conference & WG II/2 workshop*, November. Istanbul: ISPRS, 51–61.

Bieri, H. and Nef, W., 1988. Elementary set operations with d-dimensional polyhedra. *In*: H. Noltemeier, ed. *Computational geometry and its applications., Vol. 333 of lecture notes in computer science*. Berlin: Springer, 97–112.

Biljecki, F., Ledoux, H., and Stoter, J., 2014a. Error propagation in the computation of volumes in 3D city models with the Monte Carlo method. *In*: S. Li and S. Dragicevic, eds. *ISPRS annals. proceedings of the ISPRS/IGU joint international conference on geospatial theory, processing, modelling and applications*, October. Toronto: ISPRS, 31–39.

Biljecki, F., Ledoux, H., and Stoter, J., November. 2014b. Height references of CityGML LOD1 buildings and their influence on applications. *In*: M. Breunig, *et al.*, eds. *Proceedings of the 9th 3DGeoInfo conference 2014*. Dubai: UAE.

Biljecki, F., Ledoux, H., and Stoter, J., 2015. Improving the consistency of multi-LOD CityGML datasets by removing redundancy. *In*: M. Breunig, *et al.*, eds. *Lecture notes in geoinformation and cartography. 3D geoinformation science*. Dubai: UAE: Springer International Publishing, 1–17.

Biljecki, F., *et al.*, 2014c. Formalisation of the level of detail in 3D city modelling. *Computers, Environment and Urban Systems*, 48, 1–15. doi:10.1016/j.compenvurbsys.2014.05.004

Billen, R., *et al.*, 2012. Developing an ontology of space: application to 3D city modeling. *In*: T. Leduc, G. Moreau, and R. Billen, eds. *Usage, usability, and utility of 3D City Models – European COST Action TU0801*. October. Nantes, France: EDP Sciences.

Bleifuss, R., *et al.*, 2009. Entwicklung einer CityGML-Erweiterung für das Facility Management am Beispiel Landeshauptstadt München. *In*: *Symposium und Fachmesse für Angewandte Geoinformatik (AGIT*). Salzburg: Universität Salzburg, 122–131.

Boeters, R., 2013. Automatic enhancement of CityGML L*oD2 models with interiors and its usability for net internal area determination*. Master's thesis. Delft University of Technology.

Budroni, A. and Böhm, J., 2009. Toward automatic reconstruction of interiors from laser data. *3D Virtual Reconstruction and Visualization of Complex Architectures*., Vol. XXXVIII-5/W1 of *ISPRS Archives* ISPRS.

Buhur, S., *et al.*, 2009. 3D city modelling for planning activities, case study: Haydarpasa train station, Haydarpasa port and surrounding backside zones, Istanbul. *In*: C. Heipke, *et al.*, eds. *ISPRS archives. Proceedings of the WG I/2, I/4, IV/2, IV/3, VII/2 ISPRS hannover workshop 2009 on high-resolution earth imaging for geospatial information*, June. Hannover: ISPRS, 6.

Çağdaş, V., 2013. An application domain extension to CityGML for immovable property taxation: a Turkish case study. *International Journal of Applied Earth Observation and Geoinformation*, 21, 545–555. doi:10.1016/j.jag.2012.07.013

Chen, L.-C., *et al*., 2014. The application of geometric network models and building information models in geospatial environments for fire-fighting simulations. *Computers, Environment and Urban Systems*, 45, 1–12. doi:10.1016/j.compenvurbsys.2014.01.003
Coltekin, A. and Reichenbacher, T., 2011. High quality geographic services and bandwidth limitations. *Future Internet*, 3 (4), 379–396. doi:10.3390/fi3040379
De Berg, M., *et al*., 2008. *Computational geometry: algorithms and applications*. 3rd ed. Berlin: Springer-Verlag.
Döllner, J., Baumann, K., and Buchholz, H., 2006. Virtual 3D city models as foundation of complex urban information spaces. *In*: M. Schrenk, ed. *11th international conference on urban planning and spatial development in the information society (REAL CORP)*. Vienna: CORP – Competence Center of Urban and Regional Planning, 107–112.
Fisher-Gewirtzman, D., 2012. 3D models as a platform for urban analysis and studies on human perception of space. *In*: T. Leduc, G. Moreau, and R. Billen, eds. *Usage, usability, and utility of 3D city models – european COST action TU0801*. Nantes: EDP Sciences.
Goetz, M., 2013. Towards generating highly detailed 3D CityGML models from OpenStreetMap. *International Journal of Geographical Information Science*, 27 (5), 845–865. doi:10.1080/13658816.2012.721552
Goetz, M. and Zipf, A., 2012. Towards defining a framework for the automatic derivation of 3D CityGML models from volunteered geographic information. *International Journal of 3-D Information Modeling*, 1 (2), 1–16. doi:10.4018/IJ3DIM
Granados, M., *et al*., 2003. Boolean operations on 3D selective Nef complexes: data structure, algorithms and implementation. *In*: *Proceedings of the 11th Annual European Symposium on Algorithms*, September. Budapest: Springer-Verlag, 174–186.
Gröger, G. and Plümer, L., 2010. Derivation of 3D indoor models by grammars for route planning. *Photogrammetrie - Fernerkundung - Geoinformation*, 2010 (3), 191–206. doi:10.1127/1432-8364/2010/0049
Gröger, G. and Plümer, L., 2012. CityGML – Interoperable semantic 3D city models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 71, 12–33. doi:10.1016/j.isprsjprs.2012.04.004
Hachenberger, P., 2006. Boolean operations on 3D selective Nef Complexes data structure, algorithms, optimized implementation, experiments and applications. Thesis (PhD). Saarland University.
Hagedorn, B., *et al*., 2009. Towards an indoor Level-of-Detail model for route visualization. *In*: *10th International Conference on Mobile Data Management: systems, Services and Middleware*, May. Taipei: IEEE, 692–697.
Hammoudi, K. and Dornaika, F., 2011. A featureless approach to 3D polyhedral building modeling from aerial images. *Sensors*, 11 (1), 228–259. doi:10.3390/s110100228
Ilčk, M. and Wimmer, M., 2013. Challenges and ideas in procedural modeling of interiors. In: UDMV '13: proceedings of the Eurographics Workshop on Urban Data Modelling and Visualisation, May. Girona: Eurographics Association.
INSPIRE Thematic Working Group Buildings, 2013. D2.8.III.2 INSPIRE Data Specification on Buildings – Technical Guidelines. [online] [Accessed 9 July 2015].
ISO, 2003. 19107: geographic Information – Spatial Schema. Technical report.
Johnston, M. and Zakhor, A., 2008. Estimating building floor-plans from exterior using laser scanners. *In*: *Proceedings of Three-Dimensional Image Capture and Applications*. San Francisco, CA: SPIE.
Kaden, R. and Kolbe, T.H., 2014. Simulation-Based total energy demand estimation of buildings using semantic 3D city models. *International Journal of 3-D Information Modeling*, 3 (2), 35–53. doi:10.4018/IJ3DIM
Kang, H.-Y. and Lee, J., 2014. A study on the LOD(Level of Detail) model for applications based on indoor space data. *Journal of the Korean Society of Surveying, Geodesy, Photogrammetry and Cartography*, 32 (2), 143–151. doi:10.7848/ksgpc.2014.32.2.143
Kastendeuch, P. and Najjar, G., 2009. Simulation and validation of radiative transfers in urbanised areas. *Solar Energy*, 83, 333–341. doi:10.1016/j.solener.2008.08.006
Kemec, S., Zlatanova, S., and Duzgun, S., 2012. A new LoD definition hierarchy for 3D city models used for natural disaster risk communication tool. *In*: T. Bandrova and C. König, eds. *Proceedings of the 4th international conference on cartography and GIS*. Bulgaria: Albena, 95–104.

Kettner, L., 1999. Using generic programming for designing a data structure for polyhedral surfaces. *Computational Geometry — Theory and Applications*, 13 (1), 65–90. doi:10.1016/S0925-7721(99)00007-3

Khoshelham, K. and Diaz-Vilariño, L., 2014. 3D modelling of interior spaces: learning the language of indoor architecture. *In*: *ISPRS Archives. Proceedings of the ISPRS Technical Commission* V *Symposium*, June. Riva del Garda: Copernicus Publications, 321–326.

Khoshelham, K. and Elberink, S.O., 2012. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12 (12), 1437–1454. doi:10.3390/s120201437

Kim, Y., Kang, H., and Lee, J., 2014. Developing CityGML indoor ADE to manage indoor facilities. *In*: *Innovations in 3D Geo-information sciences*. Isikdag: Springer International Publishing, 243–265.

Kolbe, T.H., 2009. Representing and exchanging 3D city models with CityGML. *In*: S. Zlatanova and J. Lee, eds. *3D Geo-Information sciences*. Berlin: Springer, 15–31.

Ledoux, H., 2013. On the validation of solids represented with the international standards for geographic information. *Computer-Aided Civil and Infrastructure Engineering*, 28 (9), 693–706. doi:10.1111/mice.2013.28.issue-9

Löwner, M.O., *et al*., 2013. New concepts for structuring 3D City Models – an extended level of detail concept for CityGML buildings. *In*: *Proceedings of the 13th International Conference Computational Science and Its Applications – ICCSA 2013*, June. Vietnam: Springer-Verlag, 466–480.

Löwner, M.O., Benner, J., and Gröger, G., 2014. Aktuelle trends in der Entwicklung von CityGML 3.0. *In*: E. Seyfert, *et al*., eds. *Geoinformationen öffnen das Tor zur Welt, 34. Wissenschaftlich-Technische Jahrestagung der DGPF*, March. Hamburg: DGPF.

Luebke, D., *et al*., 2003. *Level of detail for 3D graphics*. San Francisco, CA: Morgan Kaufmann Pub.

Martin, J., 2005. Algorithmic beauty of buildings methods for procedural building gen*eration*. Master's thesis, Trinity University.

Morton, P.J., *et al*., 2012. Virtual city models: avoidance of obsolescence. *In*: H. Achten, *et al*., eds. *Digital physicality - proceedings of the 30th eCAADe conference*, September. Prague: Czech Technical University in Prague., 213–224.

Mura, C., *et al*., 2013. Robust reconstruction of interior building structures with multiple rooms under clutter and occlusions. *In*: *Proceedings of the International Conference on Computer-Aided Design and Computer Graphics (CAD/Graphics*), November. Guangzhou: IEEE, 52–59.

Nagel, C., 2014. Spatio-Semantic modelling of indoor environments for indoor navigation. Thesis (PhD). TU Berlin.

NEN, 2007. NEN 2580:2007 nl: areas and volumes of buildings - Terms, definitions and methods of determination. Technical report.

Nouvel, R., *et al*., 2014. Urban energy analysis based on 3D city model for national scale applications. *In*: *Proceedings of the Fifth German-Austrian IBPSA Conference (BauSIM 2014*), September. Aachen: IBPSA, 83–90.

Okorn, B., *et al*., 2009. Toward automated modeling of floor plans. *In*: *Proceedings of the Symposium on 3D Data Processing, Visualization and Transmission* [online], Paris. Available from: http://ri.cmu.edu/pub_files/2010/5/2009%203DPVT%20plan%20view%20modeling%20v13%20(resubmitted).pdf [Accessed 27 July 2015].

Open Geospatial Consortium, 2012. OGC City Geography Markup Language (CityGML) Encoding Standard 2.0.0. [online] [Accessed 9 July 2015].

Peter, M., Becker, S., and Fritsch, D., 2013. Grammar supported indoor mapping. *In*: *Proceedings of the 26th International Cartographic Conference*. Dresden: ICA, 1–18.

Rosser, J., Morley, J., and Smith, G., 2015. Modelling of building interiors with mobile phone sensor data. *ISPRS International Journal of Geo-Information*, 4 (2), 989–1012. doi:10.3390/ijgi4020989

Rottensteiner, F., 2003. Automatic generation of high-quality building models from lidar data. *IEEE Computer Graphics and Applications*, 23 (6), 42–50. doi:10.1109/MCG.2003.1242381

Santos, T., *et al*., 2014. Applications of solar mapping in the urban environment. *Applied Geography*, 51, 48–57. doi:10.1016/j.apgeog.2014.03.008

Schulte, C. and Coors, V., 2009. Development of a CityGML ADE for dynamic 3D flood information. *In*: *Proceedings of the Applied Geoinformatics for Society and Environment (AGSE)* 2009 *Conference*, August. Stuttgart: AGSE, 1–12.

Stadler, A. and Kolbe, T.H., 2007. Spatio-semantic coherence in the integration of 3D city models. *In*: A. Stein, ed. *Proceedings of the WG II/7 5th international symposium spatial data quality 2007*, June. Enschede: ISPRS, 8.

Stoter, J., De Kluijver, H., and Kurakula, V., 2008. 3D noise mapping in urban areas. *International Journal of Geographical Information Science*, 22 (8), 907–924. doi:10.1080/13658810701739039

Tomić, H., Roić, M., and Mastelić Ivić, S., 2012. Use of 3D cadastral data for real estate mass valuation in the urban areas. *In*: *Proceedings of the FIG 3rd International Workshop on 3D Cadastres*, October. Shenzhen: FIG, 73–86.

Van Den Brink, L., Stoter, J., and Zlatanova, S., 2013. UML-based approach to developing a CityGML application domain extension. *Transactions in GIS*, 17 (6), 920–942. doi:10.1111/tgis.2013.17.issue-6

Vandysheva, N., *et al*., 2012. The 3D Cadastre Prototype and Pilot in the Russian Federation. *In*: *FIG Working Week 2012*, May.

Wagner, D., *et al*., 2013. Geometric-semantical consistency validation of CityGML models. *In*: J. Pouliot, *et al.,* eds. *Progress and new trends in 3D geoinformation sciences*. Berlin: Springer, 171–192.

Xiong, X. and Huber, D., 2010. Using context to create semantic 3D models of indoor environments. *In*: *Proceedings of the British Machine Vision Conference (BMVC)*. Aberystwyth: BMVA.

Yasumoto, S., *et al*., 2011. The use of a virtual city model for assessing equity in access to views. *Computers, Environment and Urban Systems*, 35 (6), 464–473. doi:10.1016/j.compenvurbsys.2011.07.002

Yin, X., Wonka, P., and Razdan, A., 2009. Generating 3D building models from architectural drawings: a survey. *IEEE Computer Graphics and Applications*, 29, 20–30. doi:10.1109/MCG.2009.9

Zhao, J., Ledoux, H., and Stoter, J., 2013. Automatic repair of CityGML LoD2 buildings using shrink-wrapping. *ISPRS 8th 3DGeoInfo Conference & WG II/2 Workshop*, Vol. II-2/W1 ISPRS Archives. Istanbul: ISPRS, 309–317.