MSc thesis in Geomatics

Automatic enhancement of CityGML LoD2 models with interiors and its usability for net internal area determination

Roeland Boeters

June 2013

TU Delft Delft University of Technology

Gemeente Rotterdam

Department of GIS Technology, OTB Research Institute for the Built Environment

On cover:
LoD2+ model for part of Hoogvliet-Zuid (transparent building exteriors with visible inner shells for each storey). The base map used is TOP10NL.

AUTOMATIC ENHANCEMENT OF CITYGML LOD2 MODELS WITH INTERIORS AND ITS USABILITY FOR NET INTERNAL AREA DETERMINATION

A thesis submitted to the Delft University of Technology in partial fulfilment of the requirements for the degree of

Master of Science in Geomatics

by

ROELAND BOETERS

June 2013

Delft University of Technology	
Graduation professor	Dr. dipl.-ing. Sisi Zlatanova
Supervisor	Ken Arroyo Ohori, M.Sc.
Co-reader	Ir. Paul de Ruiter
External examiner	Dr. ir. Bastiaan van Loenen

Municipality of Rotterdam	
Supervisor	Christian Veldhuis

Roeland Boeters: *Automatic enhancement of CityGML LoD2 models with interiors and its usability for net internal area determination,* A thesis submitted to the Delft University of Technology in partial fulfilment of the requirements for the degree of Master of Science in Geomatics, © June 2013

"He who has gone, so we but cherish his memory, abides with us,
more potent, nay, more present than the living man."

— Antoine de Saint-Exupéry

In memory of my father Bert Boeters.

1952 – 2003

ABSTRACT

Where Level of Detail (LoD) allow City Geography Markup Language (CityGML) models to be shared over different application domains, only one LoD defines building interiors. As there are potentially numerous applications with a need for building interiors (but not with all the details as defined in LoD4), in order to attain a larger applicability the LoD system should be extended with building interiors in accordance with the existing exterior definitions. Since CityGML LoD2 is a common LoD, in this research it is investigated how to automatically generate interiors for LoD2 (resulting in a model called LoD2+) in which the details are comparable to those of the exteriors.

A key application in the Netherlands for LoD2+ is the determination of net internal area of buildings (the surface area which can be used by building owners). This property is useful for many applications which require the size of buildings, but is especially important for real estate taxations and sales. The registration of these values by Dutch municipalities (via Basisregistratie Adressen en Gebouwen (BAG) - Key Register for Addresses and Buildings) is at the moment still a manual process (determined from building blueprints), and is therefore error-prone.

This research dealt with the design of a finite set of shape rules to generate valid LoD2+ city models from a CityGML LoD2 model with a small amount of semantic information. Here LoD2+ is defined as the existing building solids complemented by inner shells for each storey in a premises where space is allocated for walls, roofs, ceilings and floors. Using Boolean set operations (e.g. union, intersection and difference) the building exteriors can be split in the right amount of storeys according to BAG and erosion can be applied to offset walls, roofs, ceilings and floors to obtain storey solids according to LoD2+.

From LoD2+ buildings in which interior surfaces are classified, the net internal area of the majority of premises can be estimated. By calculating these values from LoD2+ city models a comparison can be done, resulting in the validation of both the net internal area in BAG and the LoD2+ city model. Significant errors are found in the BAG registration, since net internal area was mostly not recalculated after the net internal area standards were changed/introduced.

For this research prototype software has been developed in C++ for the conversion of LoD2 to LoD2+ buildings and the calculation of net internal area. This software package is open source and freely available at `http://lod2plus.googlecode.com`.

ACKNOWLEDGEMENTS

What you are about to read, is the final work on my Geomatics studies: the master thesis. I would have never successfully managed to finish this piece of work, without the help of the people mentioned on this page. Many thanks to everyone who helped me, be it content-wise, emotionally, in any way they could have.

I would like to thank all people at Delft University of Technology involved in my project, in particular I would like to thank my supervisor Ken Arroyo Ohori for the weekly meetings which motivated me to keep me going, and his very useful advice. Furthermore I thank my graduation professor Sisi Zlatanova for her guidance and feedback. Hugo Ledoux' help on getting me started on my graduation project at the municipality of Rotterdam, is much appreciated as well. Additionally I would like to thank Sjors Donkers for his advice and suggestions during my project. John Zhao provided great assistance with programming and debugging tasks, thank you.

I would also like to express my utmost gratitude to the people involved at the municipality of Rotterdam for giving me the opportunity for this research but also for their feedback and suggestions. In particular Christian Veldhuis for his daily supervising tasks and useful feedback. The helpful discussions with Joris Goos on helping me to define my research objective at the beginning of the project were much appreciated as well.

Most of all I would like to thank my girlfriend Michelle Voogt for her support, love and encouragement during the challenging times of my studies.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ADE Application Domain Extensions
BAG Basisregistratie Adressen en Gebouwen
BGR Basis Gebouwen Registratie
BIM Building Information Modelling
BRA Basis Registratie Adressen
CAD Computer Aided Design
CAM Computer Aided Manufacturing
CNC Computer Numerical Control
CSD Construction Structure Drawing
CSV Comma-separated Values
CGA Computer Graphics Architecture
CGAL Computational Geometry Algorithms Library
CityGML City Geography Markup Language
CSG Constructive Solid Geometry
GML Geography Markup Language
IFC Industry Foundation Classes
LoD Level of Detail
OGC Open Geospatial Consortium
OSM OpenStreetMap
SNC Selective Nef Complex
UHI Urban Heat Island
UML Unified Modelling Language
VGI Volunteered Geographic Information
WOZ Waarde Onroerende Zaken
XML Extensible Markup Language

1 INTRODUCTION

Three-dimensional city models are virtual representations of mostly urban areas. Such a model may be a collection of buildings, tunnels, bridges, water bodies, city furniture, transportation objects, vegetation, etcetera. A standard for 3D city models which is becoming more popular nowadays is CityGML (Stoter et al., 2011). Gröger et al. (2006) describe it as: "a common semantic information model for the representation of 3D urban objects that can be shared over *different applications*".

For the latter part of this statement, the notion of Level of Detail (LoD) is of much importance. Since a model is always an abstraction of the real world, it must be defined what is exactly represented by the model. LoD formalizes this definition of the amount of detail present in a model. This is of significance to be able to not only compare data of various sources but also for the users to see if the data fits their needs. Another purpose of LoD is found in computer visualization where it is used to visualize objects further away with less detail and objects closer to the user with more detail (Rossignac and Borrel, 1993). As 3D city models are mainly focussed on urban areas the definitions of LoD are mainly based on the details of buildings, and specifies for example whether roofs are modelled, whether doors are present, whether buildings have interiors and so on.

In CityGML five LoDs are defined (see Figure 1), which mainly distinguish the amount of detail of building exteriors. Where LoD1 is characterized by simple extruded buildings, LoD3 buildings are true architectural models. LoD2 buildings mainly differ from LoD1 buildings by the fact that also simplified roof structures are modelled. CityGML LoD2 models can nowadays be created relatively easily using airborne laser altimetry (Vosselman and Dijkman, 2001). An example of such an LoD2 model from the municipality of Rotterdam (Rotterdam 3D) is shown in Figure 2. The building footprints of the 3D model correspond to available building footprints from a key register. Only LoD4 contains building interiors (with a lot of detail), which is how LoD4 differs from LoD3. Therefore building interiors always have to be in the same LoD.

Given that CityGML models should be usable throughout different application domains, these different LoDs are required. Such an LoD system in CityGML is however only accomplished for the building exteriors and not for the interiors. This also needs to be done for interiors because there are numerous applications that do require building interiors but do not all require the same level of detail.

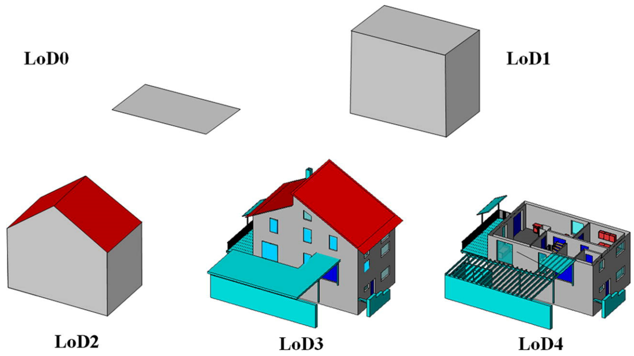


Figure 1: The difference between the five LoDs is shown in this figure. (Gröger and Plümer, 2012).

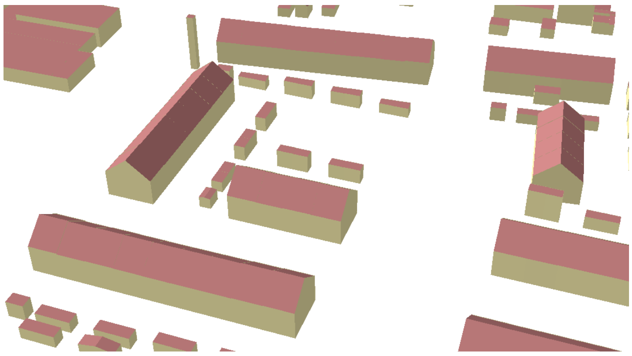


Figure 2: CityGML LoD2 model of the municipality of Rotterdam (Rotterdam3D)

CityGML LoD2 models are the most common models, not only because LoD2 is relatively easy to generate, but also because of its wide applicability. For example, since simple roofs are modelled in LoD2, it can be used to classify these roofs according to their solar potential. In this way citizens can be optimally advised whether it is useful to mount solar panels on their roofs, and on which part of their roof they should be mounted (Jochem et al., 2009). Also in noise mapping LoD2 models are highly relevant. Noise propagation is a typical three-dimensional problem as people living on the upper floors of a high-rise building may be affected by highway noise, but people on lower floors may be not (Stoter et al., 2008). The presence of roof structures in the model makes a noise propagation simulation more accurate, where otherwise the shapes of buildings are incomplete. Other examples in which LoD2 city models can be used, are: water drainage simulations in urban areas, wind simulations and urban development (Blaauboer et al., 2012).

The applicability of LoD2 when also interiors (with level of detail comparable with the LoD2 exteriors) are modelled (which is from now on referred to as LoD2+), would significantly increase. LoD2+ city models could for example greatly assist in urban heat simulations. In big cities the temperature difference between urban and rural area increases significantly, called: the Urban Heat Island (UHI) effect. Scientists have shown that as a consequence of the UHI effect more (elderly) people are dying (Watkins et al., 2007). To mitigate the UHI effect, more accurate heat simulations are helpful. For this not only building exteriors are required, but also interior properties (net/gross building volume is an important parameter for heating/cooling inertia of buildings) (Kastendeuch and Najjar, 2009). Examples of other applications that do require modelled interiors, but not to the accuracy as defined by CityGML LoD4 are: indoor architectural planning, public safety simulations, human space use/need and energy efficiency studies. It is very likely that new applications are devised when it is possible to enhance LoD2 with interiors in an automatic manner.

A key application in the Netherlands for LoD2+ is the net internal area of buildings (or better: residential units). This property relates addresses to a surface area which actually can be used by the owner(s). In the Netherlands the net internal area for all addresses is registered, which is a task of the municipality (by law through the *BAG - Key Register for Addresses and Buildings*). The property is useful for many applications that require the size of buildings, but is especially important for real estate taxations and sales. The value is defined by the Dutch standard *NEN 2580:2007* which describes which parts of a residential unit can be considered as usable surface area.

Usually this quantity is acquired by analysing building blueprints. The municipality of Rotterdam has been registering this quantity for decades now, but this has always been a manual process. As this registration is a manual process and due to changes in registration methods over the years, errors may have slipped into the system. Furthermore buildings may have been modified over the years, which intentionally or unintentionally did not get registered. Additionally the Dutch standard has not always been the leading standard for the net internal area, as registration of the net internal area started long before the introduction of *NEN 2580*.

Checking the quality of the registered net internal area is a very expensive task as it involves the validation of many objects. It is therefore desired to test the quality of the existing values of net internal area with another, more automated method (able to handle a large amount of buildings). To be able to calculate the net internal area with sufficient accuracy, floor plans, evacuation maps or Industry Foundation Classes (IFC)/Building Information Modelling (BIM) models are expected to help significantly. Where IFC/BIM models are at this time, (by far) not available for all buildings, floor plans or evacuation maps would not yield an efficient method at all as it requires a lot of manual labour.

The question thus rises, whether interiors can automatically be generated from the exteriors of the buildings from an LoD2 model (such that the level of detail of the exteriors is in accordance with the interiors) and whether it can be used to estimate net internal area.

1.1 RESEARCH OBJECTIVE

Based on the problem introduced in the previous section, the research question of this master thesis is:

Is it possible to automatically enhance LoD2 to LoD2+ city models (with an according level of detail), and use it for net internal area determination of buildings?

To be able to answer this question the following sub questions have been defined:

1. What is LoD2+?
    a) How to extend the CityGML schema to feature LoD2 building interiors (LoD2+)?
2. How to automatically enhance a LoD2 to LoD2+?
    a) What information is needed to be able to automatically generate LoD2+?
    b) What geometric operations are needed to automatically enhance LoD2 to LoD2+?
    c) Are the generated LoD2+ city models valid?
        i. Are the modelled interiors accurately representing reality?
        ii. Does the enhancement algorithm produce valid geometry?
3. How to determine net internal area from the *enhanced* LoD2 city model?
    a) Is there still data missing from the LoD2+ model to be able to determine net internal area? If so, is it possible to add this data?
    b) How to calculate the net internal area from the enhanced LoD2+ model and for which building types can this be done?
    c) What is the quality of the determined net internal area and how can it be used for validation of the BAG data?
4. Is LoD2+ the best option to serve as a basis for applications such as determining net internal area?

1.2 THESIS OUTLINE

This thesis starts by outlining previous work on (semi-)automatic interior modelling In Chapter 2. The data sets as well as the methodological concepts that are used in this research are elucidated in Chapter 3. The definition of LoD+ as well as the research methodology is found in Chapter 4. The design and implementation of LoD2+ generation algorithms is discussed in Chapter 5. This chapter also contains a brief discussion on the model results and validity. The produced LoD2+ model is used to estimate the net internal area of buildings, which is discussed in Chapter 6. Conclusions on this research as well as recommendations for future work are given in Chapter 7.

2 RELATED WORK

From Chapter 1 it has become apparent that the LoDs in CityGML need to be extended to incorporate different levels of detail for interiors as well. The little research that has been done on extending the CityGML with interiors for all LoDs is discussed in Section 2.1. Furthermore it is desired that the modelling of these building interiors can be automated in a way. Section 2.2 elaborates on current techniques for automatically producing interior models from other data sources. When little or no data is available procedural modelling is used to quickly generate large scenes (see Section 2.3). Observations relating the literature review in this section to this research are found in Section 2.4.

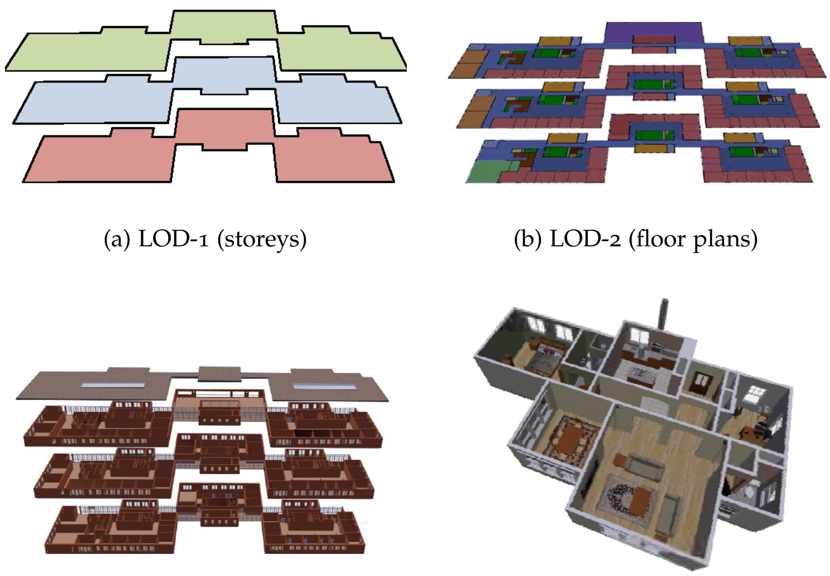
2.1 INTERIOR LODS FOR CITYGML

Hagedorn et al. (2009) recognize the problem of not having building interiors with different levels of detail and present a LoD system for indoor models. Their focus is on indoor navigation, and therefore much attention is paid to connectivity between rooms and floors. They describe four LoDs for indoor modelling:

- "**LOD-1**: Showing a building's access points, showing a building's outer shell, and showing a building's building parts;
- **LOD-2**: Showing various spaces, showing indoor routes, and providing 2D floor plans;
- **LOD-3**: Showing space height, showing the shape of doors and windows, and providing higher realism;
- **LOD-4**: Showing all interior and exterior details of buildings, providing highest degree of realism." (Hagedorn et al., 2009)

The differences between these LoDs for interiors are shown in Figure 3. As can be seen there is only three-dimensional information present in LOD-3 and LOD-4.

Recently a change request has been submitted (Behrens, 2012) for CityGML to extend building interiors according to Hagedorn et al. (2009). Although the LoDs they presented are certainly useful for indoor navigation purposes, they do not show much resemblance with the existing CityGML LoDs. The interiors of LOD-1 and LOD-2 do not contain 3D information, but LoD2 does for example show the location of doors (whilst CityGML LoD2 exteriors do not contain this information).



(a) LOD-1 (storeys) (b) LOD-2 (floor plans)

(c) LOD-3 (spaces) (d) LOD-4 (architectural interior)

Figure 3: Interior LoD system for indoor navigation according to Hagedorn et al. (2009)

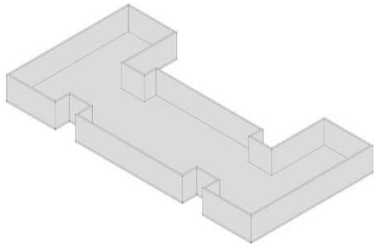
Also Kemec et al. (2012) have recently presented their first ideas on a LoD system for building interiors in relation with disaster risk communication. They define three new LoDs:

- **LoD1,5** - LoD1 with indoor modelled storeys
- **LoD2,5** - LoD2 with indoor modelled compartments
- **LoD3,5** - LoD3 with indoor modelled apartments

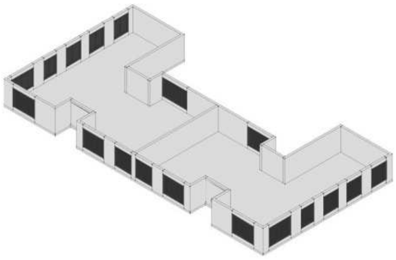
For LoD4 there is no LoD4,5 as it already contains modelled interiors with the highest detail. For LoD1.5 only the storeys are modelled. For LoD2,5 also compartments are modelled, although the exact definition is unclear. For LoD3,5 apartments are modelled, meaning different addresses can be distinguished in 3D. Visualizations of their proposed interior LoD system can be seen in Figure 4.

2.2 AUTOMATIC INTERIOR MODELLING

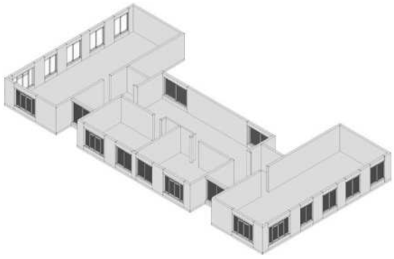
More research has been done on automatic building interior generation. The most common technique is to construct 3D interiors from architectural drawings (see Figure 5). Yin et al. (2009) describe various methods to do so. Different kinds of input can serve as a basis for constructing 3D models, such as: rasterized floor plans, Computer Aided Design (CAD) documents or Construction Structure Drawing (CSD). The biggest problem is the lack of generality, for example different symbols may be used in 2D drawings for walls or doors. Also vector-



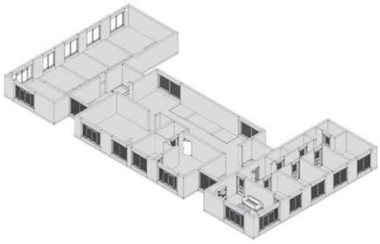
(a) LoD1,5 (storeys)



(b) LoD2,5 (compartments)



(c) LoD3,5 (apartments)



(d) LoD4,5 (identical to CityGML LoD4)

Figure 4: Interior LoD system proposed by Kemec et al. (2012) for visualization purposes of 3D urban disaster situations.

ization of raster images is a difficult task. They state that reconstruction from 2D drawings therefore still requires elaborate help from users. Goetz and Zipf (2011) go one step further, as they describe a

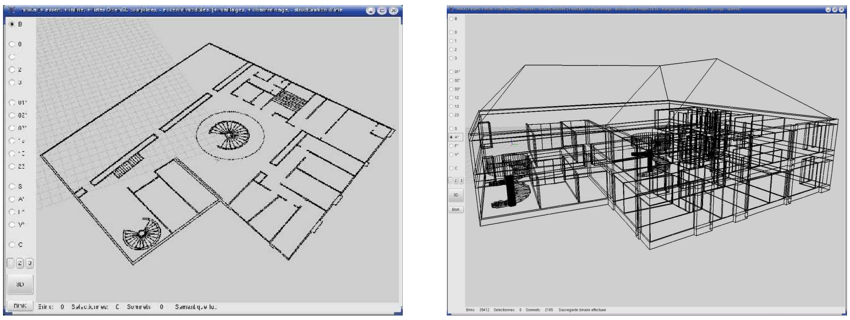


Figure 5: 2D architectural drawings may serve as input for 3D building interior construction (Horna et al., 2007).

method of using Volunteered Geographic Information (VGI) to extend the building mapping features of OpenStreetMap (OSM) with the capability of describing the inner structure of buildings. To this extent also software has been developed where building interiors can be (semi-)automatically generated from for example evacuation plans or building blueprints.

Another method is described by Okorn et al. (2010) and Budroni and Böhm (2009) as they describe a method for the reconstruction of building interiors from laser data acquired from within the building. The latter have developed a sweep algorithm that first segments horizontal structures as either floor or ceiling after which the remaining points are used in a segmentation of vertical structures to determine the potential wall points. Finally floor plans of rooms are estimated by intersecting the directions of the walls. Johnston and Zakhor (2008) on the other hand developed an algorithm to estimate building floor plans from the exterior using laser scanners for civilian and military applications. Since laser pulses can enter the building via windows, planes can be fit through these points thereby reconstructing floor plans of buildings.

2.3 PROCEDURAL MODELLING

A method that does not require this excessive data acquisition is procedural modelling. Procedural modelling is a method to create 3D models and textures from sets of rules (in a shape grammar) that may make use of randomness to simulate a realistic looking scene. Procedural modelling has first been done for modelling plants by Deussen et al. (1996) using L-system grammars, after which it has also been implemented for modelling building architecture instantly (Wonka et al., 2003). Figure 6 shows an example of a procedurally modelled city. Watson et al. (2008) state that the use of procedural modelling nowadays is mostly found in the game and film industry because it is the most cost-effective solution. Where the budgets for productions are decreasing, they must improve the tools from which procedural modelling is a result. They describe urban content as one of the main drivers as cities are huge and using existing, non-procedural tools it will take hundreds of man-years to complete such large cities.

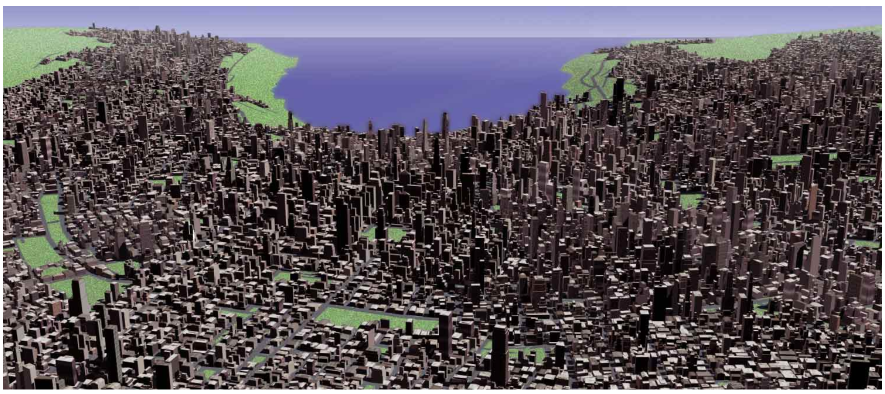


Figure 6: Example of a procedurally modelled city with approximately 26,000 buildings (Parish and Müller, 2001)

Martin (2005) presented a method to apply procedural modelling to automatically produce building floor plans. He starts off from the footprint of a building and applies graph theory to create a general structure of rooms. Having the connectivity between rooms, the rooms are placed inside the footprint (using Monte Carlo simulation to expand or shrink the rooms to the proper size). In this method no three-dimensional data has served as an input and furthermore this algorithm has not been extended to multiple storey buildings. Also this algorithm focusses more on placing different kind of rooms in a two-dimensional floor plan, while keeping the connectivity structure (including doors) between the rooms.

2.4 observations

Several observations can be made from the literature review done in this chapter, that are of relevance for this research:

- Although there is definitely a need for an interior LoD system, proposed systems are weakly defined. Furthermore their aim is not for a generic applicability, but the proposed LoDs are directed towards the use in one specific application domain. Where the connection for the proposed LoDs by Hagedorn et al. (2009) with CityGML is not that clear, more resemblance is found in the LoDs by Kemec et al. (2012) since they build upon CityGML.
- Automatic modelling techniques for interiors require a large amount of 2D or 3D interior information (such as laser data from the inside of the building or floor plans). When using floor plans still a significant amount of manual labour is required, even when digital information is used (Yin et al., 2009).
- Since procedural modelling is mainly used in computer visualizations or in the game/film industry, the focus is on creating a realistic looking scene. In this research there is no need for realistic looks, but there is a need for scenes that do match reality and therefore can be used for many applications. Shape grammars that are used in procedural modelling are therefore expected to be useful in the automatic generation of LoD2+.

The need for modelled building interiors is evident. This research is unique in the sense that an automated way of modelling building interiors will be researched, resulting in a method with which models can be generated that are useful for a broad range of applications. Since current techniques require a substantial amount of information on building interiors (albeit 2D floor plans or raw 3D data), large interior city models cannot be created and therefore their applicability remains small. This research will pioneer options to create 3D building interiors starting from 3D building exteriors and a finite amount of semantics.

3 BACKGROUND INFORMATION

One part of this research is to extend the LoDs in CityGML to feature building interiors. To this end more background information on CityGML is found in Section 3.1, on topics like geometry and topology, Level-of-Detail and the extensibility of CityGML. BAG is the key register of addresses and buildings in the Netherlands. Since this key register contains geometry of buildings it is commonly used as a basis for 3D city models, such as is also the case for the 3D city model of the municipality of Rotterdam. Information on BAG as well as information on the net internal area of buildings is found in Section 3.2. CityGML and BAG are the ingredients of the 3D city model of Rotterdam (Rotterdam 3D). The creation of this model as well as the model format is discussed in Section 3.3.

Automatic modelling of building interiors requires fundamental knowledge on architectural properties of buildings. When little information is at hand on each individual building generalizations have to be done on for example wall and floor thickness in order to still produce realistic building interiors. More on architectural properties of buildings is found in Section 3.4. As discussed in Section 2.3, when little is known about buildings, procedural modelling techniques can be used to still be able to produce realistic scenes. For this shape grammars are of much importance. Shape grammars describe the finite set of rules to produce these scenes. The principles of shape grammars are found in Section 3.5. In procedural modelling Boolean set operations as intersection or union are commonly used. Therefore more attention to these type of operations is paid in Section 3.6.

Since this research aims at the automatic generation of large interior building models that have a large applicability, it is of the essence to produce valid models. To this end, in Section 3.7, it is discussed what exactly valid 3D geometry is.

3.1 CITYGML

CityGML is a format for the representation and exchange of virtual 3D city models based on Geography Markup Language (GML) as described by Portele (2007). The standard is issued by the Open Geospatial Consortium (OGC).[1] The aim of CityGML is to create a basic common model such that the data can be used by different users and applications. Four different aspects are represented by CityGML: semantics, geometry, topology and appearance. Not only buildings can

1 http://www.opengeospatial.org/standards/citygml

be modelled using CityGML; different modules allow modelling of different object types (see Figure 7). For the purpose of this research only

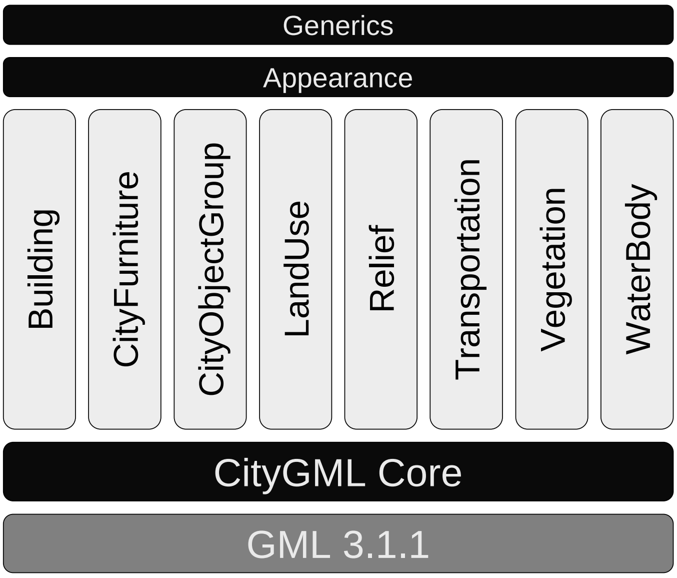


Figure 7: Modularization in CityGML (Kolbe, 2009)

semantics, geometry and topology are discussed in relation to buildings (as described by Kolbe (2009)). Furthermore the defined LoDs are discussed in more detail in this section.

3.1.1 *Geometry and topology*

In CityGML objects are implicitly represented by their boundaries (B-rep). The geometry in CityGML can be modelled as *MultiSurface*, when the topological relationships between surfaces are not known (surfaces may overlap, intersect, etc.), or as *CompositeSurface*, where surfaces must be topologically connected. As a consequence solids are typically modelled as *CompositeSurface*, to ensure valid geometry. Furthermore when connectivity between solids is known they can be topologically connected as *CompositeSolid*, otherwise *MultiSolid* must be used.

Topology is achieved by using the *XLink* mechanism from Extensible Markup Language (XML). Using *XLink*, surfaces can be reused at other places in the data model by calling the *gml:id* from the feature defined before. An example is shown in Listing 1.

Listing 1: Example of XLink mechanism

```
<bldg:BuildingPart>
  ...
  <bldg:lod2Solid>
    ...
    <gml:surfaceMember>
      <gml:Polygon gml:id="wallSurface4711">
```

```
        <gml:exterior>
          <gml:LinearRing>
            <gml:pos srsDimension="3">32.0 31.0 2.5</gml:pos>
            ...
          </gml:LinearRing>
        </gml:exterior>
      </gml:Polygon>
    </gml:surfaceMember>
  </bldg:lod2Solid>
  ...
</bldg:BuildingPart>
...
<bldg:BuildingPart>
  ...
  <bldg:lod2Solid>
    ...
    <gml:surfaceMember xlink:href="#wallSurface4711"/>
    ...
  </bldg:lod2Solid>
...
</bldg:BuildingPart>
```

3.1.2 *Level-of-Detail*

In CityGML five LoDs are defined (see Figure 8), to facilitate different application requirements. LoD0 is a 2.5D terrain model. LoD1 buildings are modelled as simple extruded blocks. LoD2 contains the same extruded blocks with some roof structures defined (as well as simple balconies and stairs). For LoD2 in specific the CityGML standard specifies a positional and height accuracy of at least 2 m. True architectural models can be defined in LoD3, with which also detailed wall structures, roofs, windows and doors can be modelled. As already mentioned, LoD4 is the only Level of Detail (LoD) in which interiors of buildings are modelled, with structures as rooms, stairs and furniture. An example of such an LoD4 building is shown in Figure 9. The five LoDs can be summarized as:

- **LoD0** - 2.5D digital terrain model
- **LoD1** - Simple extruded blocks
- **LoD2** - Simple extruded blocks + modelled roofs
- **LoD3** - Architectural models (exterior only)
- **LoD4** - Architectural models (also interiors)

3.1.3 *Semantics*

Semantics give a meaning to the geometry in the model. By using semantics, it can be known that one polygon is part of the roof of a building while the other polygon represents a wall. in CityGML this is done by using feature types. For LoD2 buildings the following feature types can be distinguished (for the complete Unified Modelling Language (UML) diagram of the building module please refer to Appendix A):

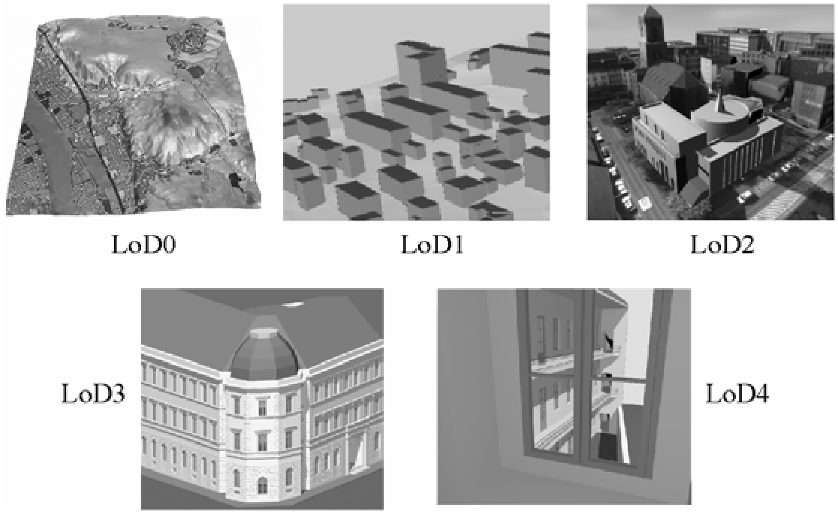


Figure 8: Examples for the five LoDs in CityGML (Kolbe, 2009)

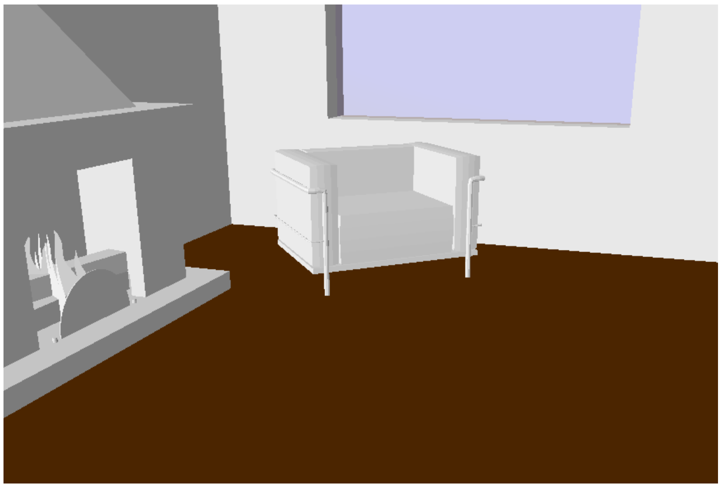


Figure 9: Example of an LoD4 building

- Building
- BuildingPart
- BuildingInstallation (e.g. stairs, balconies, etc.)
- RoofSurface
- WallSurface
- GroundSurface
- ClosureSurface (virtual surface to model unclosed volumes as solids be able to calculate volumes)

On the other hand for LoD4, in which also interiors can be modelled, the following feature types can be distinguished additionally:

- IntBuildingInstallation
- Room
- BuildingFurniture
- Opening
  - Window
  - Door
- CeilingSurface
- InteriorWallSurface
- FloorSurface
- OuterCeilingSurface
- OuterFloorSurface

3.1.4 *CityGML extensibility*

CityGML models can be extended in two different ways. Firstly by using generic city objects and generic attributes. Generic objects may only be used if appropriate classes or attributes are not defined within the CityGML schema. An identifier for each generic city object is mandatory, and additionally a class, function and usage can be defined (see Figure 10). Generic attributes must also have an identifier and additionally the data type must be specified (e.g. string, integer, double, date, etc.).

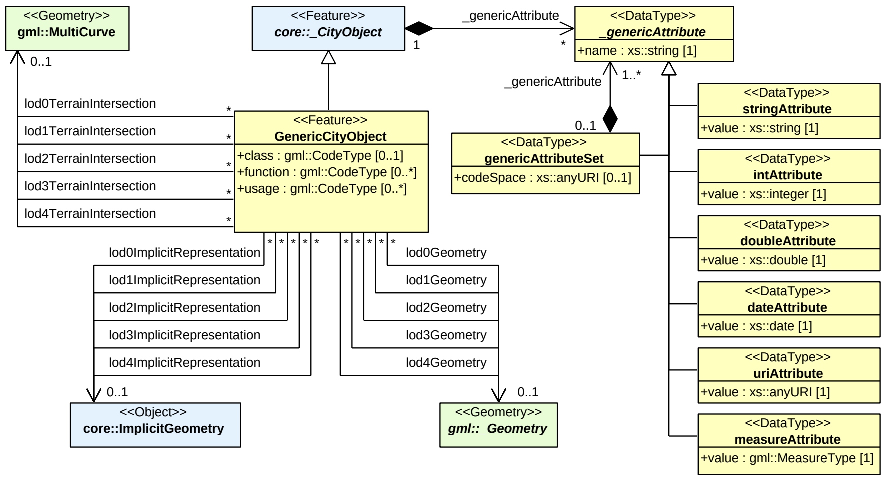


Figure 10: UML diagram of generic objects and attributes

The second possibility for extending CityGML is using Application Domain Extensions (ADE). The latter option requires that it is defined in the XML schema definition. The following CityGML ADEs are currently available or under development (CityGML, 2012):

- CityGML Noise ADE
- CityGML Tunnel ADE
- CityGML Bridge ADE
- CityGML GeoBIM (IFC) ADE
- CityGML CAFM ADE
- CityGML Hydro ADE
- CityGML UtilityNetwork ADE
- CityGML Immovable Property Taxation ADE

As can be seen these ADEs are all directed towards the use in one particular application domain.

3.2 BAG (BASISREGISTRATIE ADRESSEN EN GEBOUWEN)

In July 2009 the key register for addresses and buildings (Basisregistratie Adressen en Gebouwen) has been introduced in the Netherlands (by law) as part of the system of key registers (Ministerie van

VROM, 2009). The BAG registration is a combination of the former Basis Gebouwen Registratie (BGR) - key register for buildings - and the Basis Registratie Adressen (BRA) - key register for addresses (Fuld and Rietdijk, 2004). Municipalities are obliged to implement and maintain this key register, and governmental bodies are obliged to use the data. For these purposes a national database has been set-up in which municipalities copy their data and users can get their data from.

Figure 11 shows a general schema of BAG depicting the connectivity between the different features of the former registers. The BGR contains information on premises, mooring locations and pitch locations. In fact BGR does not contain information on buildings, but only on premises. Premises can be described as the smallest architectural objects, which are durable connected to the ground and can be entered and locked. On the other hand residential units are the smallest units (within one or more premises) with residential, recreational or commercial purpose (Rietdijk, 2009). The BRA register contains information on residential units with number indications or addresses (Rietdijk, 2009).

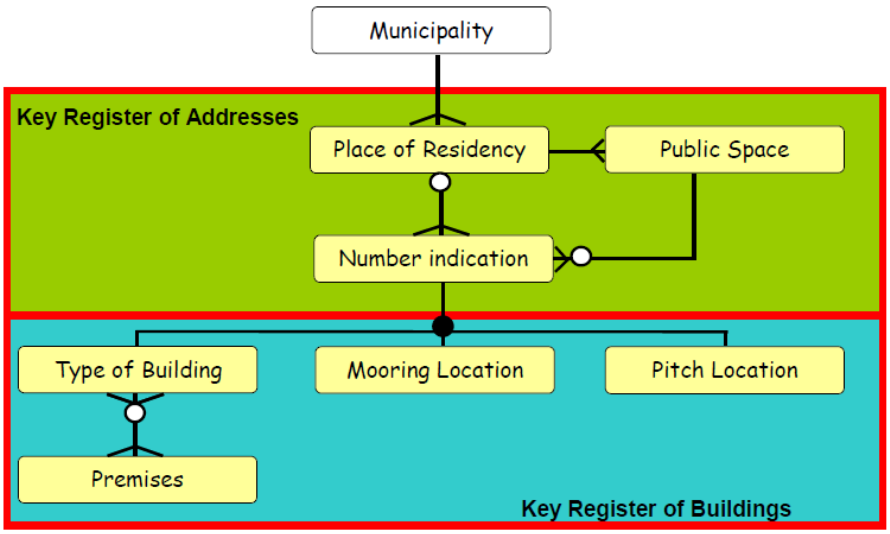


Figure 11: This figure shows a schema of all the object classes present in BAG. Taken from Fuld (2007).

In terms of geometry there is also a difference between the two former registers. Where BGR contains polygons to indicate the locations, BRA only contains 2D point features (see Figure 12). The geometry of the premises is the largest extent of the building as seen from above. This geometry may thus be larger than the exterior walls in case of overhanging roofs, balconies or building canopies.

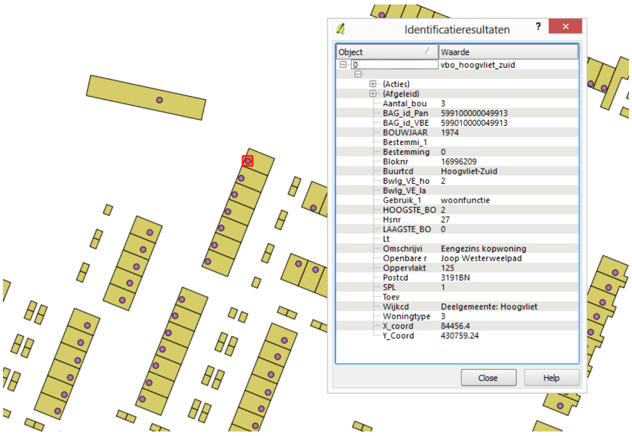


Figure 12: This figure shows sample data from BAG. The BAG building footprints (premises) are shown in yellow. The residential units are points (no geometry) of which also the attributes are shown.

3.2.1 *BAG attributes*

Premises and residential units are of interest to this research, because of the relation between premises and the 3D city model and because of the relation between premises and the net internal area of residential units. Therefore the other features such as public spaces, number indications, mooring locations and pitch locations will be left out of scope.

Residential units are point features, and premises are polygonal features. For this research, apart from the geometric information also the semantic information is of importance. While there is core information which municipalities are obliged to provide, municipalities can store additional information. In Table 1 the attributes that are linked to the premises are shown, whilst in Table 2 the attributes belonging to residential units are shown. The tables indicate which attributes are part of the core of BAG and which attributes are additionally acquired by the municipality of Rotterdam.

There are no quality checks performed on the BAG data. A key register is however supposed to be authentic and reliable. It can be argued that the BAG data is reliable because it is supported by documents. However BAG data is still corrected when feedback is received from users (e.g. Kadaster, tax authorities or brokers). At the municipality of Rotterdam there were 868 occasions of feedback in the year 2012. The largest part of feedback was about the net internal area ($\approx 30\,\%$).

Attribute	Core	Non-core
Premises identification number	✓	
Identification premises detected	✓	
Year of construction	✓	
Status	✓	
Validity start date	✓	
Validity end date	✓	
Data in research	✓	
Mutation document date	✓	
Mutation document number	✓	
Building type		✓
Number of floors		✓
Lowest storey		✓
Highest storey		✓
District heating		✓

Table 1: Attribute information corresponding to premises in BAG.

Attribute	Core	Non-core
Residential unit identification number	✓	
Identification residential unit detected	✓	
Main address residential unit	✓	
Secondary addresses residential unit	✓	
Purpose of use	✓	
Net internal area	✓	
Status	✓	
Corresponding premises	✓	
Validity start date	✓	
Validity end date	✓	
Data in research	✓	
Mutation document date	✓	
Mutation document number	✓	
Residential unit type		✓
Building type (CBS)		✓
Destination		✓
Lowest floor residential unit		✓
Highest floor residential unit		✓
Entrance floor		✓
Target group		✓
Subsidy		✓

Table 2: Attribute information corresponding to residential units in BAG.

Important attributes for this research are the number of floors as well as the building types and/or destination. The building type is only defined for residences whereas the destination types is also defined for school and restaurants for example.

3.2.2 *Net internal area*

One of the attributes in BAG that is connected to residential units, is the net internal area (Dutch: 'gebruiksoppervlakte'). The net internal area is supposed to represent the surface area that can be 'used' by the user of the building. Since this definition is rather vague, its value is defined by the Dutch standard NEN 2580:2007. This standard has been introduced in 1997. Its definition is given by Table 3. Where this standard has already been introduced in 1997, BAG has only been introduced in 2009. However many municipalities have been registering net internal area for many more years, even before NEN 2580 was introduced, and therefore discrepancies are expected between area calculations done before and after this introduction.

Area	+/-	Condition
Gross floor area (total floor area up to the exterior shell of the exterior walls)	+	
Load bearing structure	-	
Voids in-between storeys	-	if $\geqslant 4\,\mathrm{m}^2$
Cavities	+	if $\leqslant 0.5\,\mathrm{m}^2$
Locations where net height $< 1.5\,\mathrm{m}$	-	
Elevator / pipe /cable shafts	-	if $\geqslant 0.5\,\mathrm{m}^2$

Table 3: Definition of net internal area according to NEN 2580:2007. The +/- signs indicate whether the area should be subtracted or added. Some rules have additional conditions.

Figure 13 and Figure 14 show the NEN 2580:2007 standard more clearly. There is however one exception for BAG. Recently (January 2013) the Waarderingskamer (institution concerned with taxation of real estate objects) decided to deviate from NEN 2580 at one point (Waarderingskamer, 2013). The inner load bearing structure is not subtracted any more, such that is not required to be able to judge whether inner walls are load bearing or not. Although this decision simplifies the process a bit, again discrepancies in the net internal area are expected between buildings for which it has been calculated before or after January 2013.

Little is known about the quality of the net internal area. According to Rietdijk (2009) the maximum allowable deviation is 1.15 times the square root of the true net internal area, with the exception that when the object has a net internal area of $1\,\mathrm{m}^2$ the maximum allow-

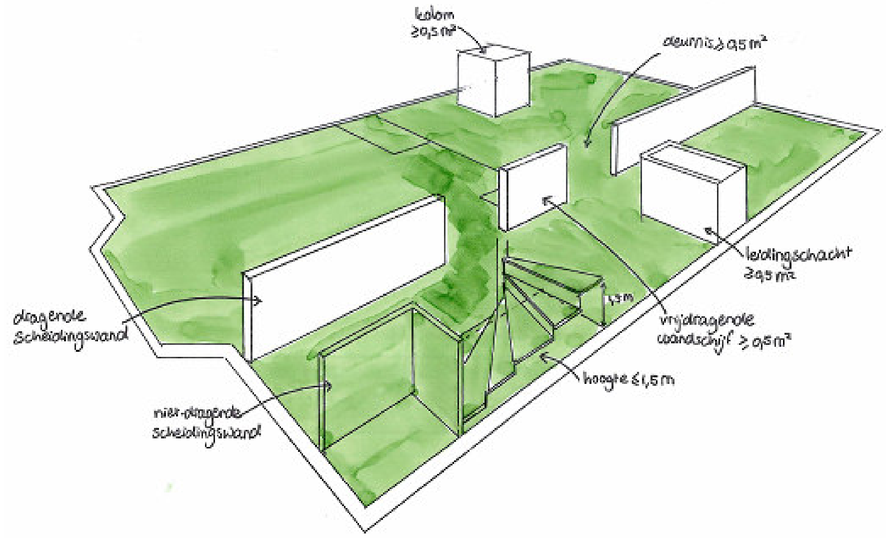


Figure 13: Sketch showing which surface areas are included (green) or excluded (white) in the net internal area according to NEN 2580:2007 (Ministerie van VROM, 2010).

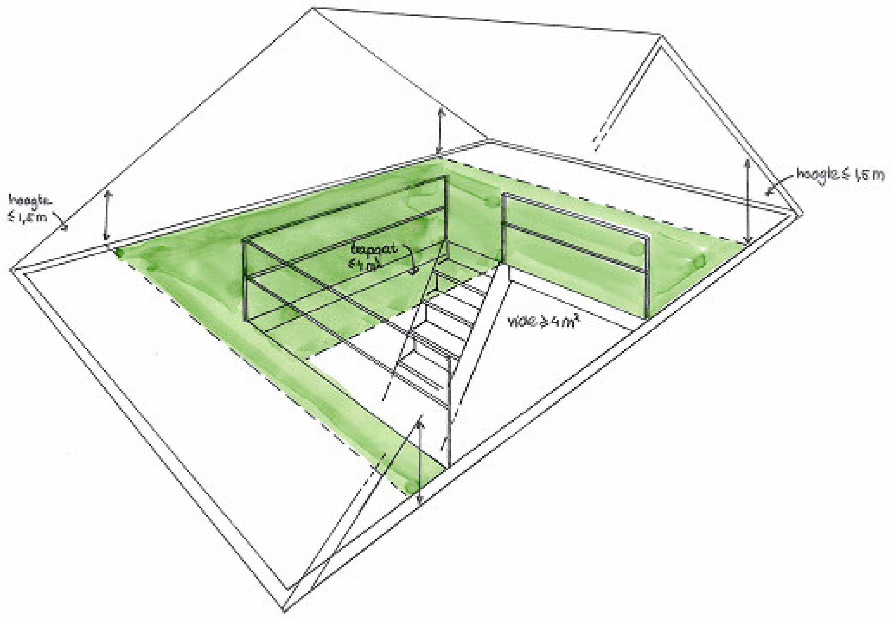


Figure 14: Sketch showing which surface areas are included (green) or excluded (white) in the net internal area according to NEN 2580:2007 on the attic floor (Ministerie van VROM, 2010).

able deviation is 100 %. For a residential unit of $120\,m^2$ the maximum allowed difference is then $\sim 12.6\,m^2$.

3.2.3 *BAG-WOZ linking*

Another key register that uses a lot of information on buildings and addresses is Waarde Onroerende Zaken (WOZ), which is used for the taxation of real estate properties. Since BAG and WOZ are partially registering the same information it has been decided that WOZ has to use information from buildings and addresses from BAG (Waarderingskamer, 2010). Since both key registers contain information on net internal area from different sources, discrepancies are expected.

3.3 ROTTERDAM 3D

Rotterdam 3D is the three-dimensional model of the city of Rotterdam. The basis for this product is point cloud data, with a point density of at least 15 points per m^2 in the harbour area of Rotterdam and a point density of at least 30 points per m^2 elsewhere. There are three variants of the product being: the unfiltered point cloud, filtered point cloud (points of the terrain) and extracted point cloud (objects and vegetation).

3.3.1 *Creation of the model*

The 3D model of Rotterdam is constructed from the extracted point cloud data described before and the BAG. The bottom plane of the 3D buildings correspond exactly to the BAG footprint. The height (z) of the bottom face is set at 0, but the relative height to the terrain is kept in the attribute data. A number of roof types (and combinations of them) are modelled (shown in Figure 15). No overhanging roofs are modelled. Furthermore the buildings are complemented with texture information. A sample dataset of Rotterdam 3D with textures is shown in Figure 16.

Since the 3D city model is based on geometry of BAG one might think that there is already a correlation between the net internal area defined in BAG and the 3D model, however this is not the case. Only the premises contain geometry, which is acquired by land surveyors. On the other hand the net internal area is linked to the residential units (point data) and is mostly determined from building blueprints. As such there is no correlation between net internal area and Rotterdam 3D.

3.3.2 *Model format*

The model is stored in CityGML. Each building has an identifier that links to the buildings in BAG, and contains roof, wall and ground surface members. Furthermore textures are attached to each building object. Exteriors are defined by a list of coordinates, so no topological

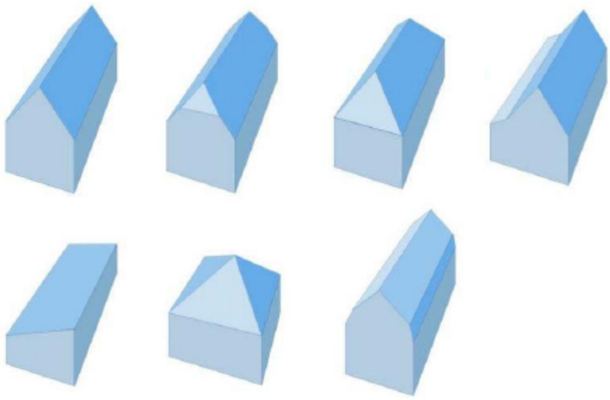


Figure 15: Roof types modelled in Rotterdam 3D (Gemeente Rotterdam, 2009).

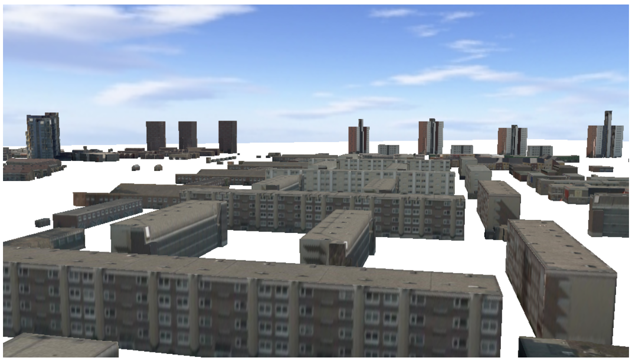


Figure 16: Sample dataset of Rotterdam 3D (Hoogvliet-Zuid) with textures.

relationship between polygons are present (see Listing 2). Furthermore the buildings are not solids as they contain holes. For example there are no walls modelled at places where neighbouring buildings touch. Therefore healing of the data is needed, which is further discussed in Section 5.1.

Listing 2: XML snippet from Rotterdam 3D

```
....
<bldg:boundedBy>
 <bldg:RoofSurface gml:id="fme−gen−affodf06−e851−4a88−9f2d−a077faa7eedd">
  <bldg:lod2MultiSurface>
   <gml:MultiSurface srsName="EPSG:28992" srsDimension="3">
    <gml:surfaceMember>
     <gml:Polygon gml:id="2ce1e1fd−6e96−463e−ae6f−7478bf5aa02b">
      <gml:exterior>
       <gml:LinearRing gml:id="48429c0c−0fb0−4d3f−803f−1cf8272c376a">
        <gml:posList>84720.57524 429696.1895 4.586 ...</gml:posList>
       </gml:LinearRing>
      </gml:exterior>
     </gml:Polygon>
    </gml:surfaceMember>
    <gml:surfaceMember>
     <gml:Polygon gml:id="7eac0b82−c7ab−471e−80c2−28252a442d7c">
      <gml:exterior>
       <gml:LinearRing gml:id="280dc09d−10e3−4e38−9120−ec47689ebb93">
```

```
      <gml:posList>84710.52906 429699.7537 8.28 ...</gml:posList>
    </gml:LinearRing>
....
```

3.3.3 *Test site Hoogvliet-Zuid*

Hoogvliet-Zuid is the southern part of a borough in the south-western part of Rotterdam (see Figure 17). Approximately 35,000 people are living in the whole of Hoogvliet. The borough which has a large variety of building types, e.g.: terraced houses, flats, some high-rise and a few detached houses. Also it has buildings with a significant difference in built years because of recent renovations of the borough (see Figure 18).

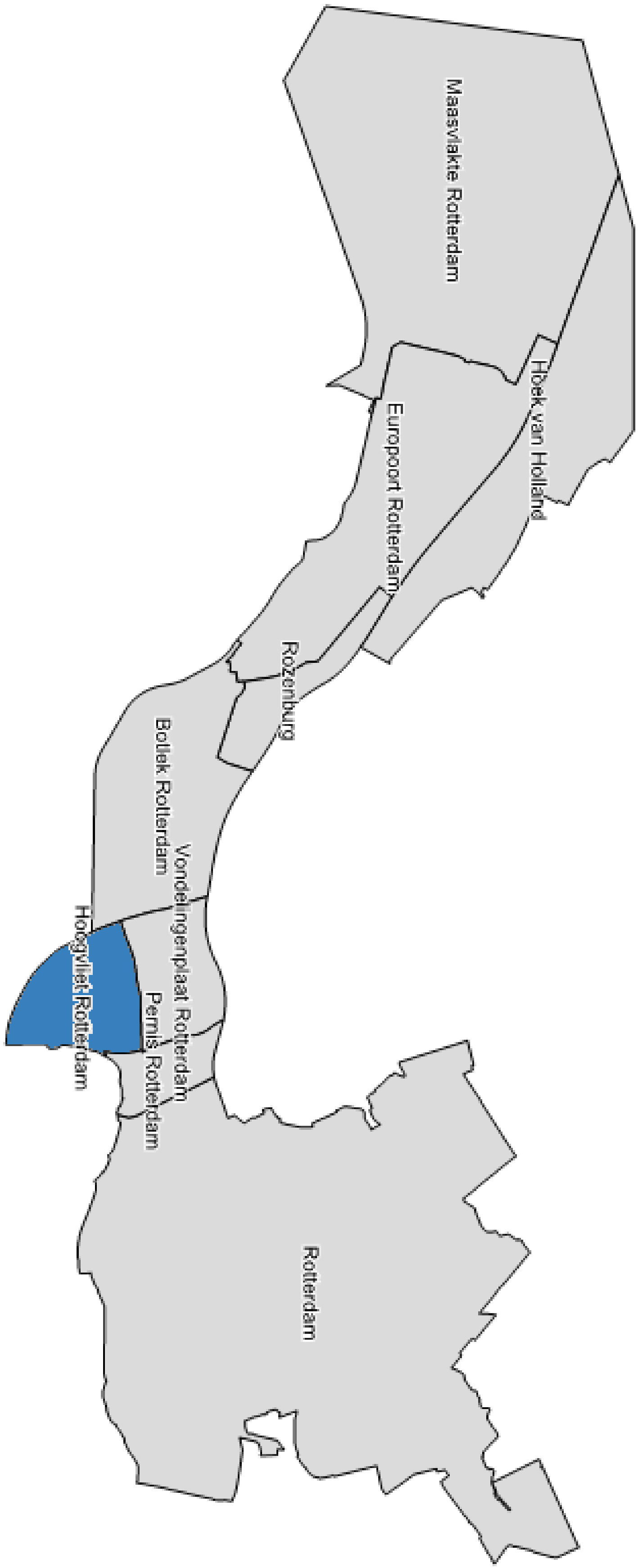


Figure 17: Location of Hoogvliet (blue) with respect to the city of Rotterdam.

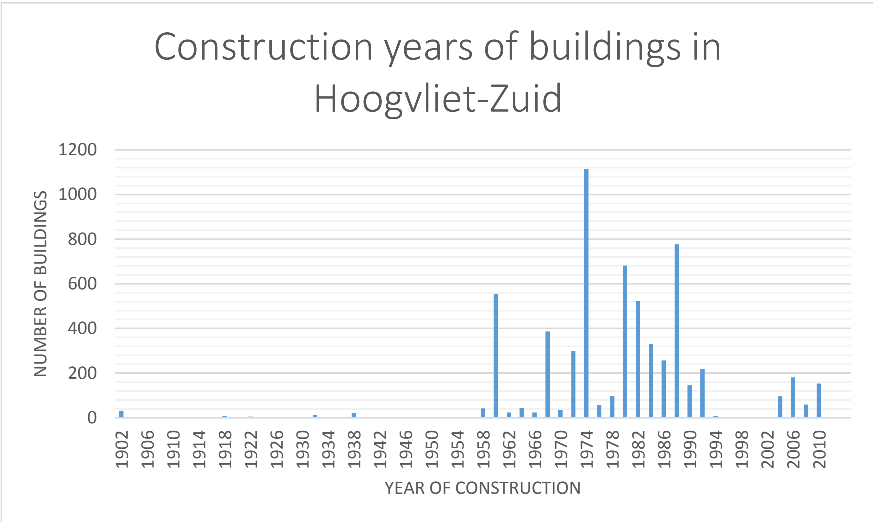


Figure 18: Histogram of construction years of buildings in Hoogvliet-Zuid

3.4 ARCHITECTURAL PROPERTIES

To be able to model building interiors, a basic understanding of architectural properties is needed. Very little research is done on linking non-geometric data to geometric properties of buildings. For this research different architectural properties are of relevance, such as: load bearing structures, non-load bearing walls, floors and ceilings.

According to Lichtenberg (2005) 25 % of the gross volume of a building is taken by the building's structure. This 25 % consists of material and voids for cables and pipes. The most of this space is assumed to be taken by the load bearing structure. There are a lot of different load bearing structure types, e.g.:

- Load bearing cavity walls
- Truss structure
- Tubular constructions
  - Framed tube
  - Trussed tube
  - Bundled tube
- Load bearing prefab walls

This list is not complete, as this research does not aim to completely cover all architectural properties. From experience we can say that most of the houses and apartments in the Netherlands do have load bearing cavity walls, which is especially true for Rotterdam as most of the city is rebuilt after World War II.

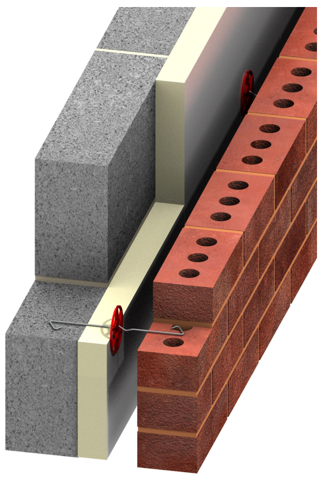


Figure 19: Illustration of a cavity wall

The sizes (thicknesses) of cavity walls vary. According to ISSO (2011) it is mainly dependent on the year of construction of buildings. They describe a varying wall thickness of 240 mm to 370 mm. The thickness of the cavity (in which the insulation is placed) depend on the year of construction as described in Table 4.

Year of construction	Cavity thickness
1930 to 1970	30–60 mm
1970 to 1985	± 70 mm
after 1985	± 100 mm
New buildings can also	
have 120 mm cavity walls	

Table 4: Cavity thickness in relation to year of construction

The wall thickness does however not only depend on construction years, but also on the materials used (e.g. concrete, wood, steel), the size of the building and the type of building. Thijssen and Meijer (1988) for example show an old Dutch regulation which describes a minimum wall thickness for different storeys.

The same holds true for floors/ceilings. No research is done on determining characteristic values for floor thicknesses for different building typologies. Various non-scientific sources describe an average floor thickness of ±200 mm, but this depends again on used materials, years of construction, size of building and type of building.

3.5 SHAPE GRAMMARS

A procedural modelling algorithm is usually described by a shape grammar. This shape grammar holds a description of the set of rules that is used to produce a scene. A shape grammar is defined as follows (Stiny, 1980):

- A finite set of shapes $\mathbb{S} = \{S_1 ... S_n\}$
- A finite set of attributes $\mathbb{A} = \{A_1 ... A_n\}$
- A finite set of shape operations $\mathbb{O} = \{O_1 ... O_n\}$
- A finite set of production rules
  $\mathbb{R} = \{R_1 ... R_n\} : predecessor \rightarrow (conditions) \rightarrow successor$

The production rule is executed (i.e. the predecessor is transformed in the successor shape) if the conditions are met. This process can be an iterative one, as the successor can become again a predecessor. The grammar consists of at least three rules: a start-rule, a transformation rule and an end-rule. For each shape that has a symbol, it is checked whether there exists a production rule where the predecessor shape is the same. When there is such a production rule, it is executed and the symbol is removed from the predecessor shape (although a new symbol may be attached to the successor shape). An example showing how the principles of these shape rules is shown in Figure 20. Hence, there are two steps involved in computations using a shape grammar:

1. The recognition of a particular shape
2. Replacement of the shape

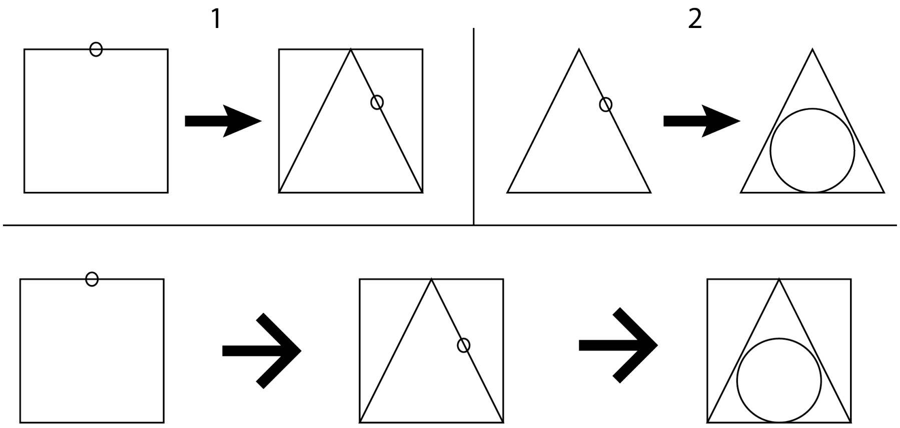


Figure 20: This figure illustrates a simple example of the execution of two production rules. For each shape with a symbol (o) it is checked whether a production rule exists. If it exists it is executed.

Stiny (1980) also described two types of shape operations: Boolean set operations and transformation operations. Boolean set operations are operations such as union, intersection and difference. Transformation operations include translation, rotation, reflection, scale or finite compositions between them.

One of the popular shape grammar implementations is the Computer Graphics Architecture (CGA) shape grammar (used in CityEngine), with which cities can procedurally be modelled, is described by Watson et al. (2008).

3.6 BOOLEAN OPERATIONS ON 3D GEOMETRY

As discussed in Section 3.5 Boolean operations (such as set union, intersection, difference and complement) are important operations in procedural modelling. Figure 21 shows an example of how different Boolean operations can be combined to construct 'complex' geometry.

A way of combining simple primitives (such as cubes, spheres, cylinders and pyramids) to obtain complex geometry is known as Constructive Solid Geometry (CSG). Requicha and Voelcker (1985) believe that Boolean operations in CSG are mainly important because of the desire to be able to "define solid objects through 'additions' and 'subtractions' of parametrized solid primitives such as cuboids and cylinders" and "model and simulate manufacturing processes such as milling an drilling and integrated-circuit fabrication processes".

One can imagine that producing building interiors from exteriors can be achieved by simulating manufacturing processes as well and therefore this section focusses on creating a basic understanding of

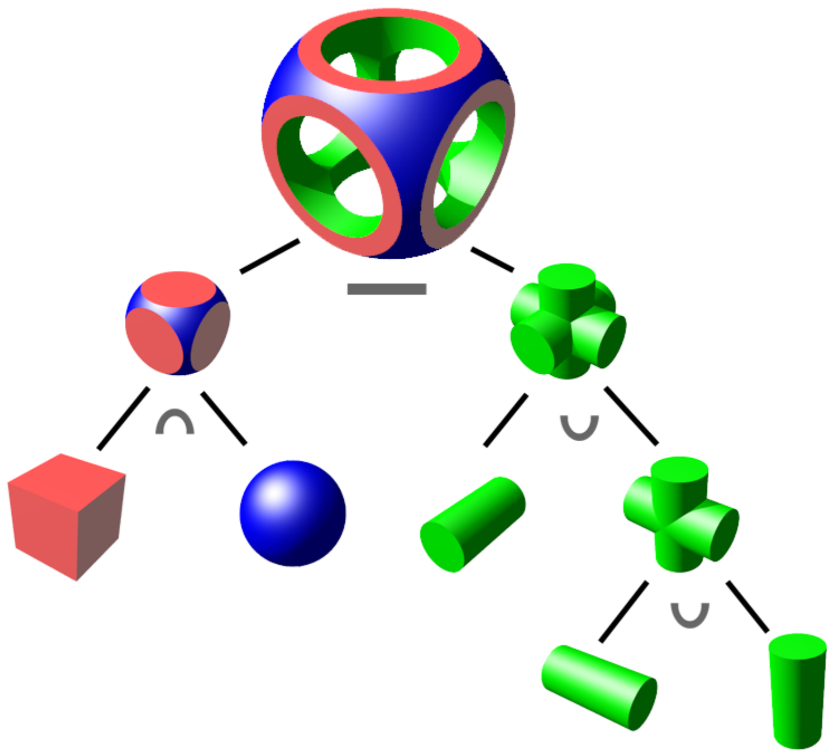


Figure 21: By combining Boolean operations as intersection (∩), union (∪) and difference (−) complex solids can be constructed such as the one in this figure.

Boolean operations. However, in general buildings are too complex to be generated using CSG and therefore Boolean operations on general polyhedra is discussed here.

Since Boolean operations only work on bounded volumes, valid input is required. The building exteriors need to form at least solids which are watertight. Additionally other conditions need to be checked (see Section 3.7). If these conditions are not met, which is generally the case for buildings in Rotterdam 3D, prior healing is mandatory (see Section 5.1.1).

In principal Boolean operations on polyhedra do not guarantee solids. For example the intersection of two cubes touching each other at one of the faces results in a plane, i.e. a degenerate volume. Using regularized operations a solid can still be guaranteed (Tilove and Requicha, 1980). The regularization of a polyhedron is the inside of the polyhedron covered by a tight skin. Therefore the boundary is not included which results in the removal of dangling faces as well. Furthermore the intersection of two solids may result in multiple disjoint solids. A polyhedron type, which is very suitable for Boolean set operations and supports regularization operations as well as the handling of disjoint volumes, is the Nef polyhedron (Nef, 1978).

3.6.1 *Nef polyhedra*

"Nef polyhedra are the smallest family of solids containing the half-spaces and being closed under Boolean operations". (Granados et al., 2003) Nef objects can represent any object in $\mathbb{R}^d$, but existing implementations are limited to three dimensions. A Nef polyhedron is generated by combining set intersection and complement operations on a finite number of open half-spaces. Nef polyhedra are defined by local pyramids which characterise the local space around 0D (vertex), 1D (edge), 2D (facet) and 3D (volumes) faces. An example of a Nef polyhedron with its corresponding local pyramids is given in Figure 22.

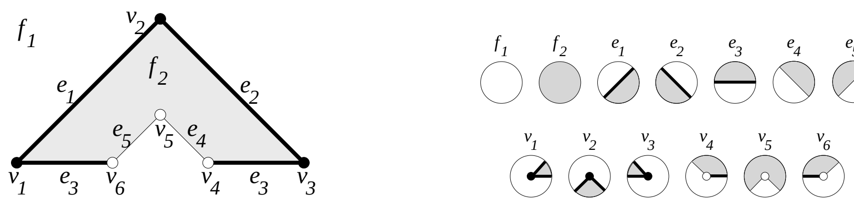


Figure 22: An example of a Nef polyhedron with its corresponding local pyramids (Granados et al., 2003)

3D Nef polyhedra are implemented in Computational Geometry Algorithms Library (CGAL) by Granados et al. (2003). They use two data structures for this, being: sphere maps and Selective Nef Complex (SNC) representation.

3.6.1.1 *Sphere maps*

In case of three-dimensional polyhedra the local pyramids from Figure 22 correspond to spheres instead of circles. Sphere maps represent the local neighbourhood of each vertex of the Nef polyhedron. It is obtained by intersecting a small sphere with each vertex (see Figure 24). An edge intersecting the sphere results in a vertex on the sphere (*svertex*). A face intersecting the sphere results in an edge on the sphere (*sedge*). Finally a volume intersecting the sphere results in a face on the sphere (*sface*). An example of a sphere map is given in Figure 23.

3.6.1.2 *Selective Nef Complex*

Next to the sphere maps also a so-called SNC is constructed such that the representation becomes more easily accessible. The SNC is constructed from the sphere maps of a polyhedron. In this data structure they store:

- **edges** - two oppositely oriented edges (identified by a *svertex* in the sphere map as shown in Figure 24)

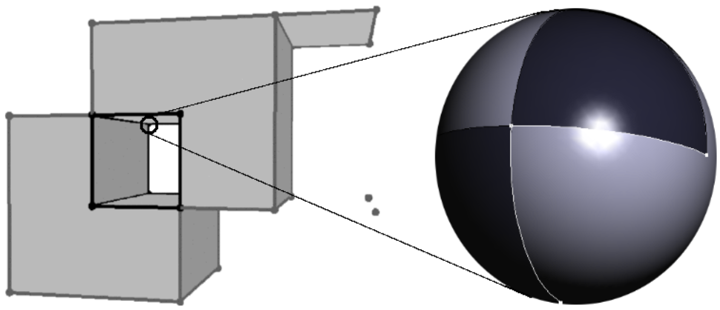


Figure 23: Example of a sphere map (Hachenberger and Kettner)

- **edge uses** - oriented edges corresponding to an incident oriented facet (identified from a *sedge* in the sphere map as shown in Figure 24)
- **facets** - boundary cycles of oriented edge-uses
- **shells** - connected set of facets, edges and vertices
- **volumes** - a set of shells defining the volume (one outer shell, and zero or more inner shells for holes)

For each of the items a set-selection mark is stored, which indicates whether it belongs to the solid or not.

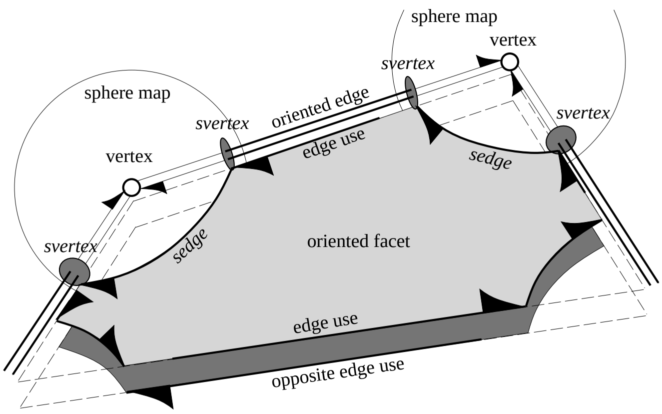


Figure 24: Construction of sphere maps is done by intersecting a small sphere with a vertex. The different features that are stored in the Selective Nef Complex data structure relate to the sphere maps as shown in this figure. (Granados et al., 2003)

3.6.1.3 *Boolean operations on Nef polyhedra*

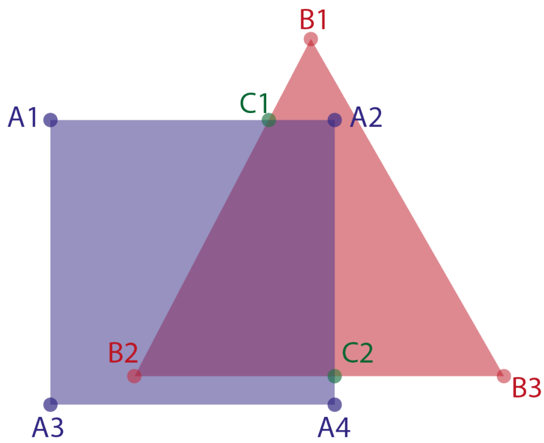
Using the two data structures described before, Boolean operations can now be implemented as follows:

1. First all the candidate vertices are found, which are all the vertices from both original polyhedra plus the points where edges intersect with edges or edges intersect with faces.

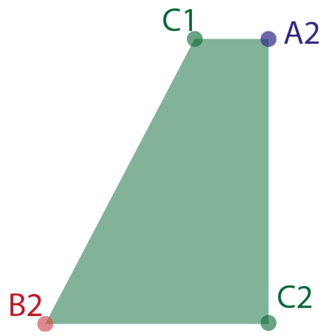
2. For each candidate vertex the local sphere map must be constructed (if not yet available). Now the two sphere maps of a candidate vertex (one for each polyhedron) are combined using Boolean operations.

3. For each of the constructed sphere maps it is determined whether the vertex is part of the result and the resulting Nef polyhedron is constructed from there.

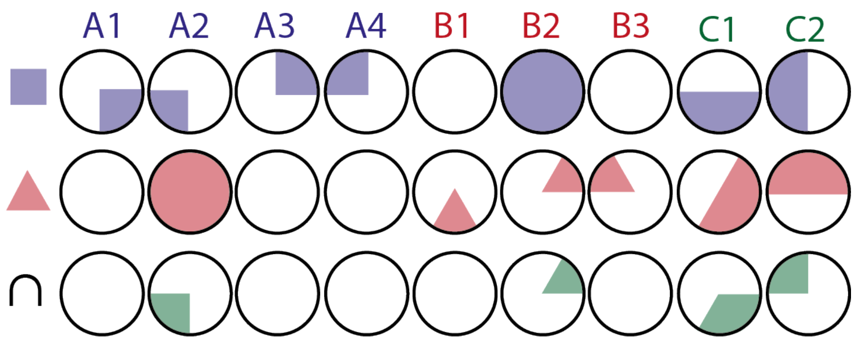
A simple example in Figure 25 illustrates this procedure for the *set intersection* operation on two 2D polygons. Analogously the Boolean set operations union, difference and symmetric difference can be performed.



(a) Step 1: First all candidate vertices are found which are the vertices of both original polygons, and also the vertices at intersection points (in 2D the vertices where edges intersect each other).



(b) Step 3: From the combined sphere maps it can be seen which vertices are part of the result and from there the final polygon can be constructed.



(c) Step 2: For all candidate vertices the sphere maps are constructed (if not yet available) and using Boolean operations they are combined. The combined sphere maps for the intersection operation are shown on the bottom row.

Figure 25: Illustration of the procedure (steps 1 to 3) to perform Boolean operations on Nef polyhedra.

3.6.2 *Minkowski sum on 3D polyhedra*

The Minkowski sum is defined as the vector sum of all points belonging to two point sets. Figure 26 shows an example illustrating the vector sum of all points belonging to two 2D triangles.

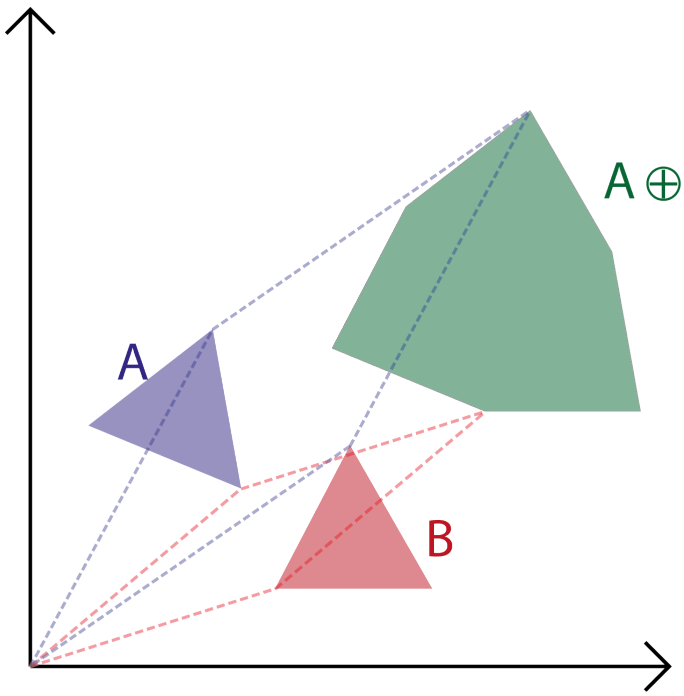


Figure 26: The Minkowski sum of polyhedra A and B is defined as the vector sum of all points belonging to A and B.

This is specifically useful for applications as robot motion planning or tight passages simulation (see Figure 27). Minkowski sum is also known as dilation, which is usually referred to in image processing where a structuring element is moved over a 2D raster image (Gorte, 2006). An example of the Minkowski sum of two three-dimensional polyhedra is shown in Figure 28.

The Minkowski sum requires the convex-decomposition of non-convex polyhedra and consequently performing a union operation on the pair-wise Minkowski sums of the convex polyhedra (Hachenberger, 2007).

3.7 GEOMETRIC VALIDITY

When automatically enhancing the CityGML models with interiors it is desired to have a valid geometric output. This way ensuring that the model can be used by future applications. It is already stated that the preferred output of the produced algorithms is solid geometry. In order to satisfy these two objectives, it must be known what a valid solid is, which is described in this section.

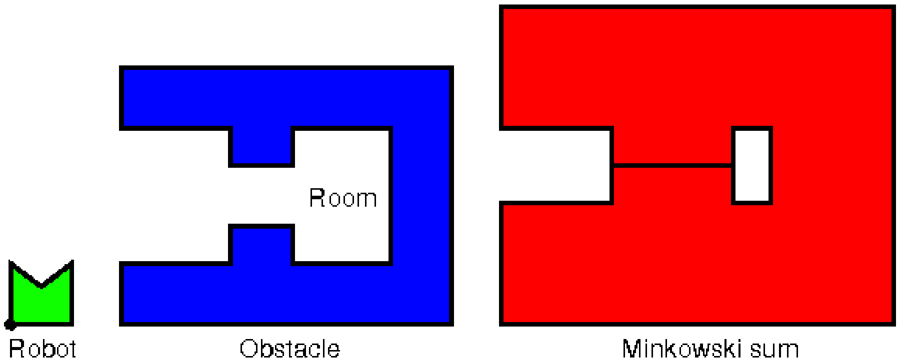


Figure 27: Minkowski sum is also used in the case of tight passage planning. The boundary of the Minkowski sum of a robot and a polygon describes all the legal positions of the robot (Hachenberger, 2013).

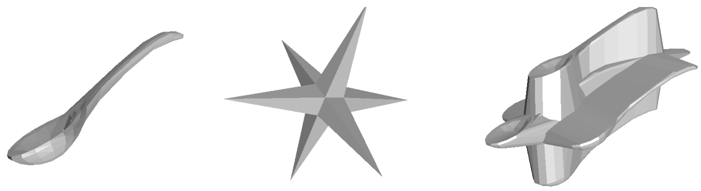


Figure 28: Minkowski sum of a spoon and a star (Hachenberger, 2007)

Janssen et al. (2012) have done research on valid 3D geometries, in order to develop software to validate solids (based on ISO19107). The following validity requirements are extracted from their work:

- At the level of vertices:
  - **Duplicate points** - The same vertex may not be used more than once in the same linear ring.
  - **Ring not closed** - The last vertex of a linear ring must be the same as the first.
- At the level of edges:
  - **Inner ring wrong orientation** - Inner rings must have an opposite orientation in relation to outer rings.
  - **Non planar surface** - The vertices of a surface must lie in one plane.
  - **Inner ring intersects outer** - An inner ring (hole) must be disconnected from the outer ring.
  - **Inner ring outside outer** - An inner ring (hole) must be inside the outer ring.
  - **Interior of ring is not connected** - The interior of a linear ring must be connected.
- At the level of faces:
  - **Not valid 2-manifold** - Each edge must be connected to two neighbouring faces.

  - **Surface not closed** - The solid must be watertight and thus not contain holes.
  - **Dangling faces** - Faces that do not bound the interior of the solid may not be present.
  - **Face orientation incorrect edge usage** - The correct oriented edge must be used in relation to the orientation of the face.
  - **Free faces** - All faces must be connected to the solid.
  - **Surface self intersect** - Faces should not intersect each other.
  - **Vertices not used** - Unconnected vertices may not be present.
  - **Surface normals bad orientation** - The normals of all faces of the solid should point outwards.

- At the level of solids:
  - **Shells faces adjacent** - The faces of shells (exterior/interior) of the solid may not be adjacent .
  - **Shell interior intersects** - Interior shells may not intersect.
  - **Inner shell outside outer** - Inner shells may not be outside the outer shell.
  - **Interior of shell not connected** - The interior of a shell must be connected.

4 LOD+ DEFINITION AND RESEARCH METHODOLOGY

One goal of this research is to investigate whether it is possible to automatically generate LoD2+ city models (see Section 1.1). Currently, in only two previous studies the desire has been expressed for an interior LoD system and a initial proposal for such a system has been given (see Section 2.1). The proposed interior LoDs did either not fit the amount of detail of the exterior LoDs in CityGML or the interior LoDs were weakly defined. The objective of this research is to have a LoD system with both exteriors and interiors which have a comparable amount of details. To this end a LoD+ system will be designed. The design process of this is discussed in Section 4.1.

Having LoDs featuring exteriors and interiors a methodology for generating LoD2+ model is needed. Current methods for the automatic generation are limited because they require a substantial amount of information from the interiors (see Section 2.2). From Section 2.3 and Section 3.5 it is known that using a finite set of shape rules and a limited amount of input parameters virtual cities can be generated. This general idea will be used for the generation of LoD2+ interiors as well. The required input parameters and data, derived from the LoD2+ definition (as given in Section 4.1), are discussed in Section 4.2. Furthermore certain requirements are set on the generated LoD2+ city model having implications for the modelling methodology as well.

In Section 3.6 it is discussed that Boolean set operations are important in CSG because of the desire to simulate manufacturing processes. How this links to the automatic generation of LoD2+, along with which geometric (Boolean set) operations are needed for this, is discussed in Section 4.3. Using this defined methodology algorithms will be developed and implemented to automatically generate LoD2+ city models (see Chapter 5).

Once such a LoD2+ city model is generated, it is desired to use it for net internal area (see Section 3.2.2 for its definition) calculations. The Dutch NEN standard describes which parts of buildings are to be excluded or included in the usable surface area. The influence of each of these specific exceptions on the net internal area for specific buildings is to be determined. The method for this is described in Section 4.4.

It is expected that it is very hard to determine the accuracy of the modelled interiors. Even more so because this is dependent on the accuracy of the available CityGML LoD2 model. Since a wrongly modelled interior, results in an incorrect net internal area it is assumed that using the calculated net internal area from the LoD2+ city model, an indication of the quality of the modelled interiors can be given. Furthermore it is desired to use the calculated net internal area for the validation of the net internal area in BAG. The validation is thus twofold, and its method is described in Section 4.5.

The complete research methodology work flow diagram, as described before, is shown in Figure 29. The implementation of the methodological concepts are described for the LoD2+ generation in Chapter 5 and for the net internal area determination in Chapter 6.

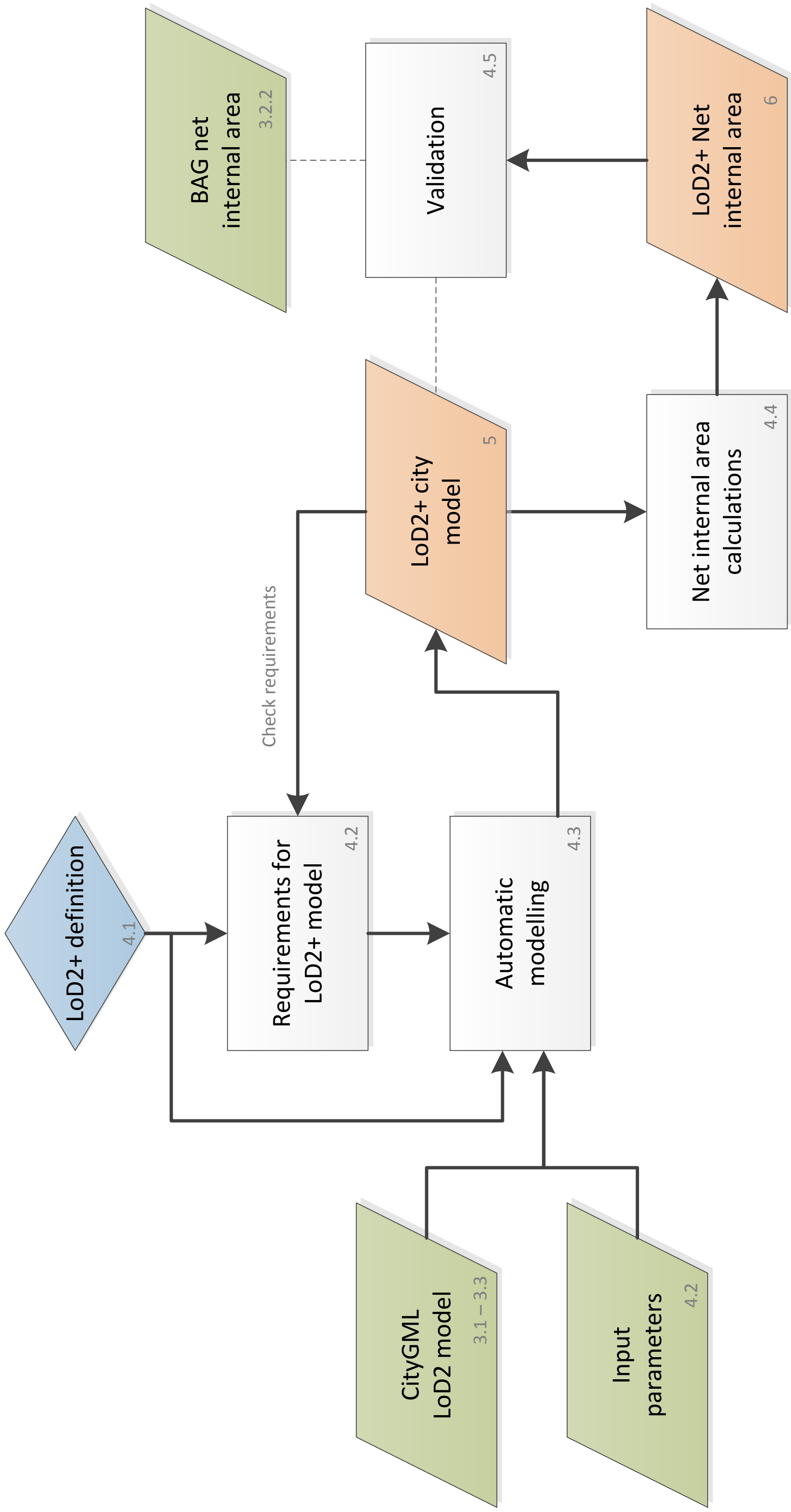


Figure 29: Research methodology work flow diagram with corresponding section numbers.

4.1 LOD2+ DEFINITION

LoD2+ cannot be modelled without the notion of the appropriate amount of detail. To this end, in this section the current CityGML LoDs are extended to feature building interiors as well.

One goal of this research is to define interior LoDs which are comparable to the exterior LoDs. Therefore it can be compared what features are already present for the exterior LoDs and what interior features would correspond to this. In Table 5 the features that are already present in the exterior LoDs and the possible corresponding features for the interior LoDs are given. As LoD1 buildings are simple extruded blocks, LoD1+ could be simple storey blocks by splitting the building at the right heights. These simple storeys would then correspond to LoD1 identical to the description of Kemec et al. (2012). Given the current definition of LoD2 and LoD3, it makes sense to model no doors and windows in LoD2+ as they are not present in LoD2 exteriors either, but to do model it in LoD3+ where they are present for the exteriors. Furthermore since LoD1 does not yield thematic boundary surfaces (e.g. *WallSurface*, *RoofSurface* or *GroundSurface*) LoD1+ would be storey solids without semantics. Since LoD4 already features building interiors it does not need to be designed.

To distinguish LoD2+ from LoD1+ the surfaces should be offset from the exterior boundary surfaces (e.g. wall, ceiling and floor thickness should be modelled). Figure 30 shows a visual representation of LoD2+. Furthermore it is proposed to model the interior LoDs with the same hierarchy as in the Dutch BAG system. That is: LoD2 would correspond to premises and LoD3 would correspond to residential units. The difference is explained in Section 3.2. To distinguish LoD3+ from LoD2+ also entrances and windows of residential models can be modelled, as they are also modelled for LoD3 exteriors. Also stairwells located in the premises but not outside the residential units can be modelled in LoD3+ to make the exterior LoD as much as comparable to the interior LoD. Where LoD4+ does in fact geometrically not differ from LoD4 it does incorporate storeys, premises and residential units.

The proposed LoD+ system, in which the level of detail of the interiors are as much in accordance as possible with the level of detail of the exteriors, can be summarized as follows:

- **LoD1+** - 'LoD1' + 'modelled storeys'
- **LoD2+** - 'LoD2' + 'LoD1+' + premises + interior features with thickness offset
- **LoD3+** - 'LoD3' + 'LoD2+' + residential units
- **LoD4+** - The same as LoD4, architectural model (interior) + 'LoD3+ features'

LoD	Exterior features	Corresponding interior
1	Simple blocks	Simple storeys
	No thematic boundary surfaces	No thematic boundary surfaces
2	Roof structures	Propagated for ceiling structures
	Thematic boundary surfaces	Thematic boundary surfaces
	Outer building installations	-
3	Doors	Entrances of residential units
	Windows	Windows of residential units
	Outer building installations	Inner building installations

Table 5: Since the interior LoDs should be comparable with the exterior LoDs in this table the exterior features with the possible corresponding interior LoD features are given.

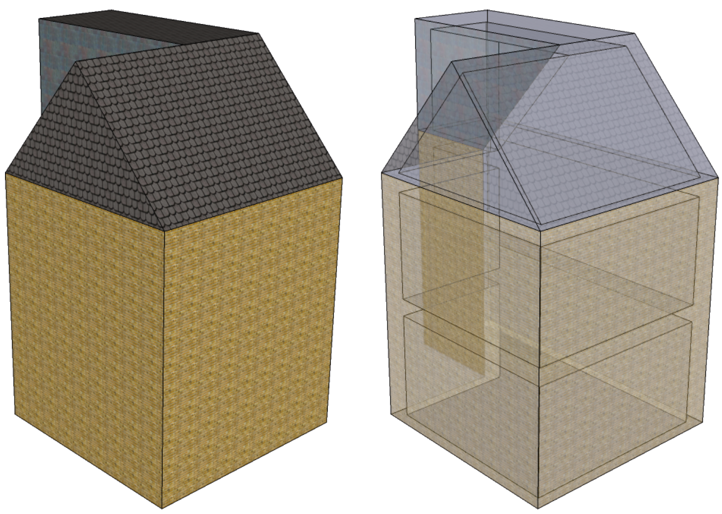


Figure 30: This figure shows the difference between LoD2 (left) and LoD2+ models (right). For LoD+ models volumes are fit inside the exterior shell which represent storeys.

Since LoD4 already contains building interiors, LoD4+ is in fact the same as LoD4. LoD+ is not proposed as an extension to the original CityGML LoDs, so this yields no problems in terms of incoherency.

To increase the applicability each storey should be modelled as a closed shell, such that volume calculations 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, i.e.: *FloorSurface*, *InteriorWallSurface* and *CeilingSurface*. Although LoD+ will extend the current LoDs, it should not be modelled as ADE. As building interiors are inseparable from the exteriors, it should be part of the CityGML standard. To this end the CityGML building module UML diagram (Gröger et al., 2012) is extended to feature LoD1+, LoD2+ and LoD3+. A simplified UML diagram of the building module with the LoD+ extension is shown in Figure 31.

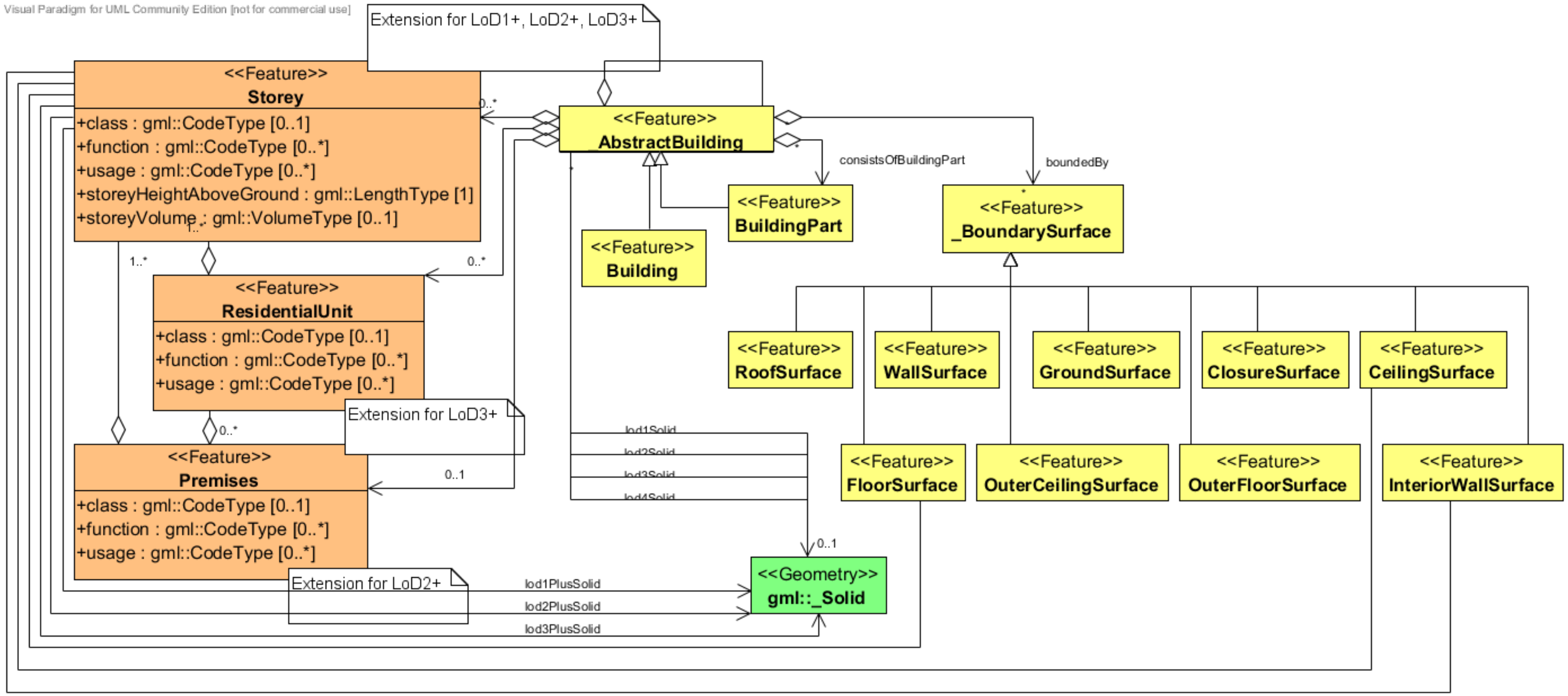


Figure 31: This figure illustrates a simplified version of the CityGML building module (yellow and green) with the LoD+ extension (orange). LoD1+ only contains storey solids, whereas LoD2+ and LoD3+ feature premises and residential units respectively.

4.2 INPUT DATA AND REQUIREMENTS

From Section 4.1 it is known what LoD2+ city models should look like. Inner shells represent each storey within a premises. Furthermore offsets are applied from the exterior shells to incorporate wall, roof, ceiling and floor thickness.

To be able to generate such a city model certain data is required. First of all we need a CityGML LoD2 city model because it is desired to enhance such a model with interiors. However since LoD2+ buildings relate to premises, clearly the buildings in the city model need to relate to premises as well. In the city model from the municipality of Rotterdam (Rotterdam 3D; see Section 3.3) this is the case which makes it suitable for the generation of LoD2+ interiors.

Additionally certain input data is required. First because for each storey a shell should be modelled, we need to know the number of storeys. From Table 1 it is known that the lowest and highest storey of a premises is part of BAG at the municipality of Rotterdam. The lowest storey is usually 0, but may be negative in case of basements.

Next to that, we know from Section 4.1, that an offset is desired to incorporate wall, roof, ceiling and floor thickness. Naturally this offset is different from building to building. As described in Section 3.4 these thickness values are at least depending on:

- Building type
- Year of construction
- Number of storeys

The methodology for the determination of the thickness values is discussed in Section 4.2.1. Also these parameters are part of BAG at the municipality of Rotterdam. To summarize the following data is needed as input to the automatic generation of LoD2+ city models:

- CityGML LoD2 model based on BAG premises polygons
- BAG data including building type, year of construction and number of storeys

Further requirements for the LoD2+ city models can be set. First of all, as discussed in Section 1.1, the storey shells are required to be valid (for the validation methodology see Section 4.5). Performance-wise there are no requirements for the municipality of Rotterdam, as it is expected that the automatic generation of LoD2+ city models is much faster than manually checking building blue prints.

4.2.1 *Thickness input*

As discussed in Section 4.1 the inner shells should be offset from the exterior shell by a wall (also shared walls between neighbouring buildings), ceiling, floor and roof thickness. It appears not to be easy to determine generic values from literature (see Section 3.4). To still be able to determine these offset values, building blueprints are available (as these documents are registered in BAG to support certain values in the registration; see Appendix B for a few example blueprints).

The assumption is that when small sets of buildings are created which are based on the three aforementioned parameters, the thickness values for these buildings can be better approximated than by using average values for all of the buildings. Thus by determining small sets of comparable buildings and consequently determining the thickness values for these buildings the parametrized solid erosion can be done more accurately.

It is assumed that the building type has the most influence on the thickness values as the material and the wall type depend very much on the building type. For example high rise buildings may have curtain walls, while most normal houses have cavity walls. The year of construction is assumed to have the next largest influence (mainly because of the evolution of cavity walls over the years). After that the number of storeys is assumed to have the least influence. When categorizing according to these parameters, the set of buildings should first be split according to building type. After that each building type is split according to the years of construction. Finally each of these smaller sets is again split by the number of storeys, resulting in the smallest sets. For each of these small sets the average thickness values are determined by randomly inspecting building blueprints.

Each splitting step is limited to three levels to reduce the amount of labour ($3 \times 3 \times 3 = 27$ sets for which thickness values need to be determined).

4.3 LOD2+ GENERATION METHODOLOGY

Having defined how LoD2+ should be modelled and knowing which input data is required we want to know which geometric operations are required to do the automatic generation of LoD2+ buildings.

In Section 3.6 it is discussed that Boolean set operations are important in CSG to be able to simulate manufacturing processes and furthermore it is hypothesized that the simulation of manufacturing processes may generate LoD2+ building interiors. The methodology to generate LoD2+ interiors is thus to choose the right manufacturing processes which can generate these interiors, and consequently determining the needed Boolean set operations to simulate these manufacturing operations. This methodology is chosen because of its intuitive approach but even more so because machining processes work on solids such that the output is guaranteed to be solid geometry.

As described in Section 4.1 LoD1+ building interiors are simple volumes where a building is split in the right number of storeys and no thickness offset is modelled for walls, roofs, ceilings and floors. In fact this is the first step to generate LoD2+ building interiors. A logical first operation is thus to split a building into multiple solids which represent each storey.

This first operation can be simulated using a simple cutting operation. Such a cutting operation in Computer Aided Manufacturing (CAM) or Computer Numerical Control (CNC) is commonly done using a circular saw blade as shown in Figure 32. In fact such a sawing operation removes material thereby separating multiple parts.

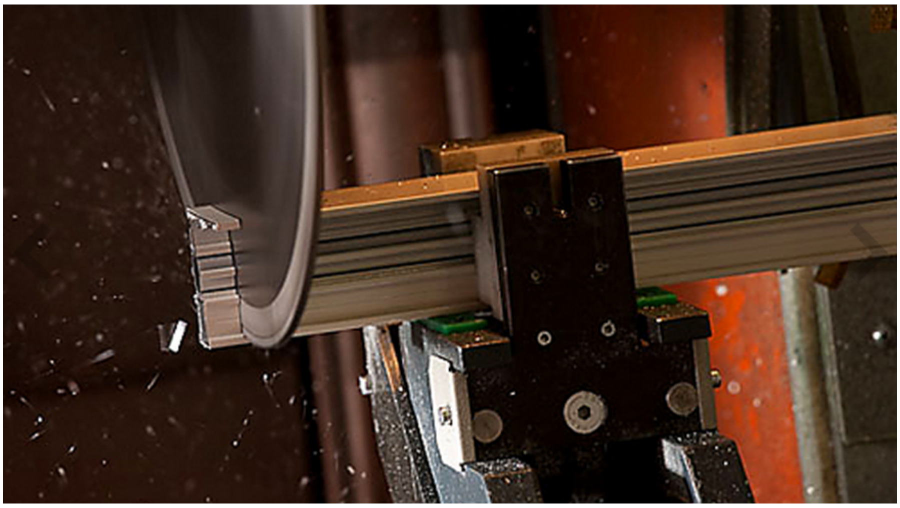


Figure 32: In CAM/CNC cutting operations can be done using circular saw blades. Here an aluminium profile is cut to the correct length.

Such an cutting operation is in Boolean set operations equal to a difference operation since material is removed from the original piece of material. In fact for the generation of LoD2+ interiors it is not necessary to remove material and thus this cutting operation can be reduced to an intersection operation of the building with a box shaped solid having the desired minimum and maximum height.

This operation produces solids for each storey, but does not result in the correct offset yet. For this a second machining process is required. The solid thus needs to shrink which is also known as erosion as described by Heijmans and Ronse (1990). However for each face the offset may be different. E.g. the offset thickness for one wall may be 30 cm and for the other wall 12 cm. In fact we thus need a parametrized erosion of the solid where a different offset can be applied for each surface.

In CNC this erosion can be done by using milling operations (Wang, 1988). This is the moving of a robot along the surface of the solid thereby removing material usually done using a milling cutter. An example of this is shown in Figure 33. In milling the removed material (pocket) is exactly described by the Minkowski sum of the milling cutter and the surface over which it is moved (Yao and Gupta, 2004).

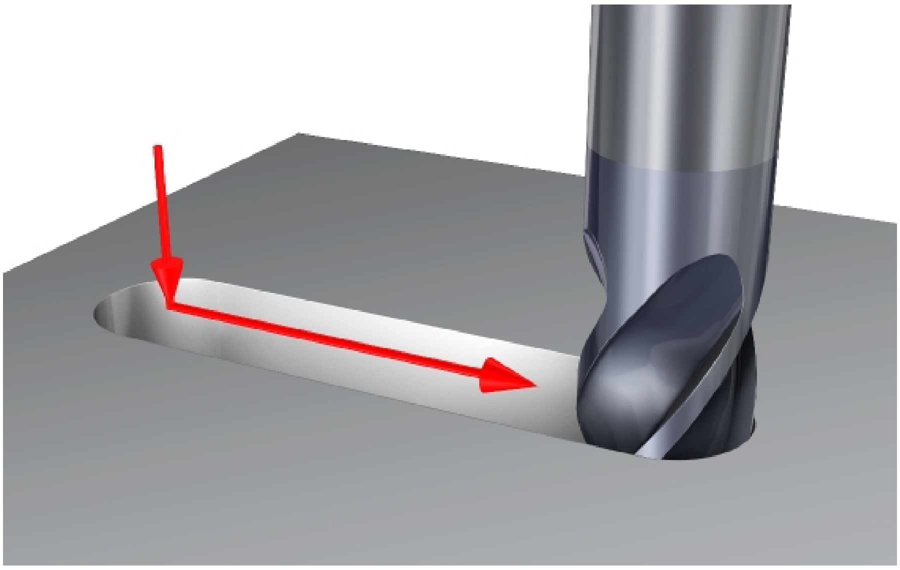


Figure 33: This figure illustrates the milling process where a milling cutter removes material (pocket) from the piece of material by rotation and movement.

Therefore the methodology to generate LoD2+ interiors is to split buildings by intersecting solid volumes with the original buildings and to subsequently remove material by simulating a milling process (using a Minkowski sum). This is an intuitive approach as it can be physically performed as well. Furthermore Boolean set operations are used because then it is relatively easy to guarantee the output to be solid geometry as well.

When these storey solids are created, then the surface still has to be classified (as described in Section 4.1). Since ceilings, floors and walls have different orientations it is expected that the normal vector of each facet can be used for this classification.

4.4 NET INTERNAL AREA APPROACH

From a LoD2+ model with classified interior facets (e.g. *InteriorWallSurface*, *FloorSurface*, *CeilingSurface*), the surface area of the floors can already be calculated. This is the gross floor area from which the surface area of the exterior walls is subtracted. Clearly, this *FloorSurface* area is not identical to the net internal area as defined by NEN 2580. Therefore other parts of the building have to be subtracted as described in Section 3.2.2.

For each of these different parts of a building there are three options:

1. It may be possible to geometrically determine the influence (from the LoD2+ model) of this part of a building and subtract this area from the *FloorSurface* area.

2. Using semantic information of the building the influence might be determined, and subsequently be subtracted.

3. The influence may not be determined at all.

When the influence of a certain parameter can be geometrically determined, it is assumed that the accuracy of the surface area that has to be subtracted is sufficiently high. For option two when semantic information should be used to determine the influence, the accuracy is expected to be less. Intuitively the surface area for these parameters change from building to building. To be able to take these parameters into account, for the same building categories as described in Section 5.2.4 the minimum and maximum surface area should be determined. Then by subtracting those minimum and maximum values from the *FloorSurface* area a range is determined in which the sum of net internal area should be located.

When more than one residential unit is located within a premises, certain areas in that premises do not belong to any of the residential units. To compare the sum of the net internal area of residential units within that premises to the calculated area from a LoD2+ model at least the following areas need to be subtracted which may all be depending on the size of the premises, number of storeys and/or number of residential units:

- Corridors
- Stairwells
- Elevators
- Maintenance rooms
- Sheds

FloorSurface area	1100 m^2	
Parameter	Min. area [m^2]	Max. area [m^2]
Corridors	40	80
Stairwells	30	50
Elevators	20	35
Maintenance rooms	5	15
Sheds	80	140
Subtract	175	320
Net internal area	**780**	**925**

Table 6: When minimum and maximum surface areas can be determined for each parameter for a certain building a range of values can be calculated between which the BAG net internal area must fall.

An example calculation for a premises with multiple residential units is shown in Table 6. For option three when the influence cannot be determined at all, the resulting net internal area will not be very accurate but may still be useful in the validation of the net internal area registered in BAG.

In the end the calculation of the net internal area for any premises will thus result in a range of net internal area in which the true net internal area is expected to be located.

4.5 VALIDATION METHODOLOGY

One of the goals of this research is to produce valid LoD2+ city models. As discussed in Section 1.1 this validity is twofold:

- The LoD2+ city model needs to consists of valid geometry
- The modelled interiors need to be accurate

For the geometric validity, software is available (see Section 3.7). However to achieve geometric validity, in the process of generating LoD2+ buildings the following is taken into consideration:

- The input data is assumed to contain valid geometry (through a healing process described in Section 5.1.1).
- Regularized operations (see Section 3.6) will be used to discard dangling faces and non-manifold geometry.
- All degenerate cases are handled by the Nef polyhedra implementation.
- Intersection operations may result in multiple disjoint volumes, but these may be handled separately.

Besides the geometric validity, it thus also needs to be known whether the building interiors are actually accurately representing reality. The accuracy of the modelled interiors is depending on the accuracy of the modelled exteriors in the LoD2 model. As it is difficult to draw conclusions on the basis of the modelled LoD2+ buildings, the validation is largely done on the basis of the calculated net internal area in Chapter 6 because this can be done automatically for a large number of buildings. It is assumed that if the building interiors are accurately modelled, the net internal area will be accurate as well. On the other hand, if there are big flaws in the modelled interiors it is assumed that there are large differences between the net internal area registered in BAG and the net internal area calculated from this model. Therefore by inspecting the largest differences between the net internal area in BAG and the area calculated from the LoD2+ model the majority of issues with the model is expected to be found. The following possible issues in the LoD2+ model may be found:

- When storeys are modelled at incorrect heights, the heights of these storeys may be too small such that the net height is not 1.5 m and thus the complete storey does not add to the total net internal area of the premises.
- When at slanted roofs the storeys are not modelled at the correct height, the surface area below 1.5 m net height will significantly change.
- When the incorrect number of storeys are modelled the net internal area will significantly differ from the net internal area in BAG.
- When the surface classification is not properly done, there may be too much or too little *FloorSurface* area resulting in an incorrect net internal area.

Since there are many parameters involved in the calculation of net internal area from its definition, the calculated net internal area can be assumed to follow a normal distribution according to the central limit theorem (Petrov, 1995) when the minimum and maximum influence for each of these parameters can be determined as described in Section 4.4. The available BAG net internal area can then be compared against the confidence region it falls in (see Figure 34). If we now assume that for the example building in Table 6 780 m^2 and 925 m^2 correspond to the -3σ and 3σ values respectively we can calculate in which confidence region the BAG area falls and validate according to this.

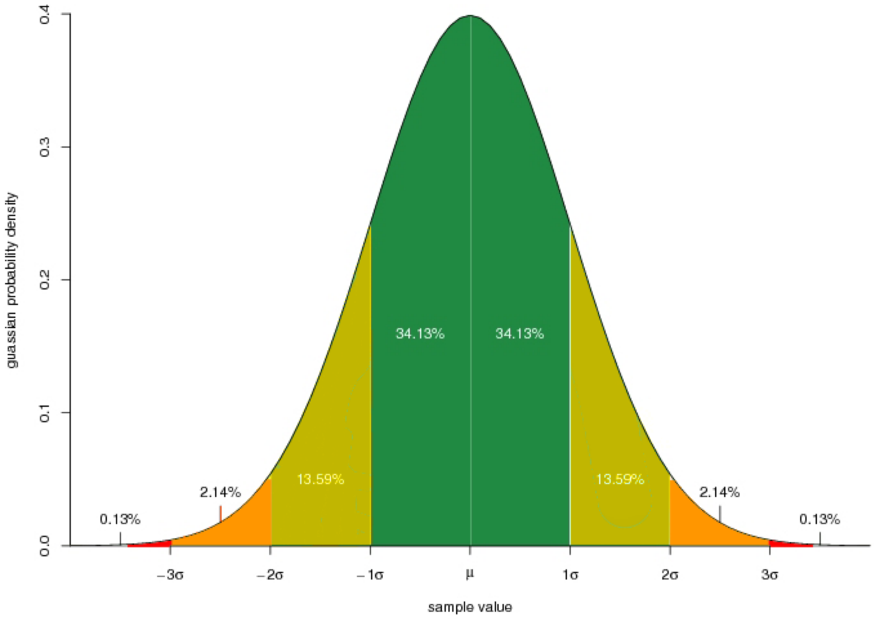


Figure 34: If a normal distribution can be determined for the net internal area of a stacked building, the net internal from BAG can be validated against the confidence region in which it falls.

5 GENERATION OF LOD2+

LoD2+ is an extended version of CityGML LoD2 in which building interiors are modelled with less detail than in LoD4. The definition of it is discussed in Section 4.1 and can be summarized as inner shells for each storey where thickness offsets for walls, ceilings and floors are taken into account. Furthermore the boundary surfaces of the storey solids are thematically classified.

In Section 4.3 the methodology for the generation of LoD2+ was discussed. It is argued that the simulation of machining processes may intuitively generate LoD2+ interiors. Furthermore Boolean set operations make it relatively easy to create solid volumes which are valid.

Boolean set operations require valid input geometry however. Therefore pre-processing of the data is needed as well as converting the CityGML buildings to an appropriate data format. A short description on that is given in Section 5.1.

In Section 4.3 it is also discussed that the storeys without offset can be generated by intersection operations of solid bounding volumes with a minimum and maximum height and the original building. When this is done then there needs to be a parametrized solid erosion of the storey. For this the combination of set difference and Minkowski sum is used. To know how much the storey solid should shrink for different sets of buildings based on building type, year of construction and number of storeys thickness values are determined from blueprints. The implementation of the geometric algrithms and the determination of the thickness values are described in Section 5.2.

The results of applying the designed shape rules to the test site Hoogvliet-Zuid (see Section 3.3.3 for more information on the test site) are discussed in Section 5.3. In this section the geometric validity and the performance is discussed as well as a visual inspection on the modelled interiors for different types of buildings.

5.1 DATA STRUCTURE AND PRE-PROCESSING

Before the city model can be enhanced, pre-processing of the data is required. First of all we must ensure the data contains valid geometry. Second the CityGML data needs to be converted to Nef polyhedra to be able to perform Boolean operations on the building models.

5.1.1 *Healing the geometry*

Validation on the Rotterdam 3D dataset show a larger number of problems. The largest problem is that no walls are modelled between neighbouring buildings causing 95 % of the buildings to not be solids because of missing faces in the shells (determined using 3D validator software of Janssen et al. (2012)). The main goal for healing geometry is therefore to ensure that each shell is closed such that solids are formed. Other problems which are present and are attempted to be healed are non-planar surfaces and wrong orientation of facets.

The first attempt to heal the geometry is done using the City Doctor Healing tool by Wagner et al. (2013). The following checks are implemented and a attempt is done to repair these errors:

- 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 are only allowed to intersect at their start or end point
- Faces of the solid need to be planar
- A solid consists of at least four polygons
- The normal vector of faces is points out of the solid
- All polygons belonging to a solid should be connected
- Each point is surrounded by one cycle of alternating edges and faces

Their software heals about 60 % of the dataset. Problems are still present when multiple adjacent non-coplanar faces are missing in a shell. Another issue that is not yet solved are dangling faces along an edge, which are not removed. Furthermore a tolerance is used for faces which are not co-planar (tolerance is set to 0.01 m), resulting in faces which are still not exactly co-planar. A second attempt to make sure each building is valid solid is done while converting the CityGML data to Nef polyhedra.

Covering all invalidity issues is not the scope of this research, but is necessary because of the need of valid input geometry. It is therefore not validated what part of the model is in fact valid geometry, however the input is healed as much as possible and discarded at run-time if the geometry is still not valid.

5.1.2 *CityGML to Nef polyhedra*

Since in Rotterdam 3D for each face the xyz-coordinates are stored in a position list, no topological relationships exist. To be able to construct Nef polyhedra from the CityGML file, these topological relationships have to be constructed. That is, a face-vertex relationship needs to be known. Each face of the solid then consists of a number of vertex indices and furthermore no duplicate vertices are present.

To eliminate duplicate vertices a tolerance is used to snap vertices together.

Having this topological structure, a polyhedron can be constructed by incrementally adding facets to a polyhedral surface as implemented by Kettner (1999). From this data structure a conversion algorithm is available to automatically construct Nef polyhedra, that is if the polyhedra are valid and closed. If the polyhedra are not, another attempt is done to heal: non-coplanar facets (by triangulating the polyhedral surface; see Figure 35 for an example), surfaces with boundaries (unclosed polyhedra), self-intersecting boundaries and disjoint facets (CGAL, 2013).

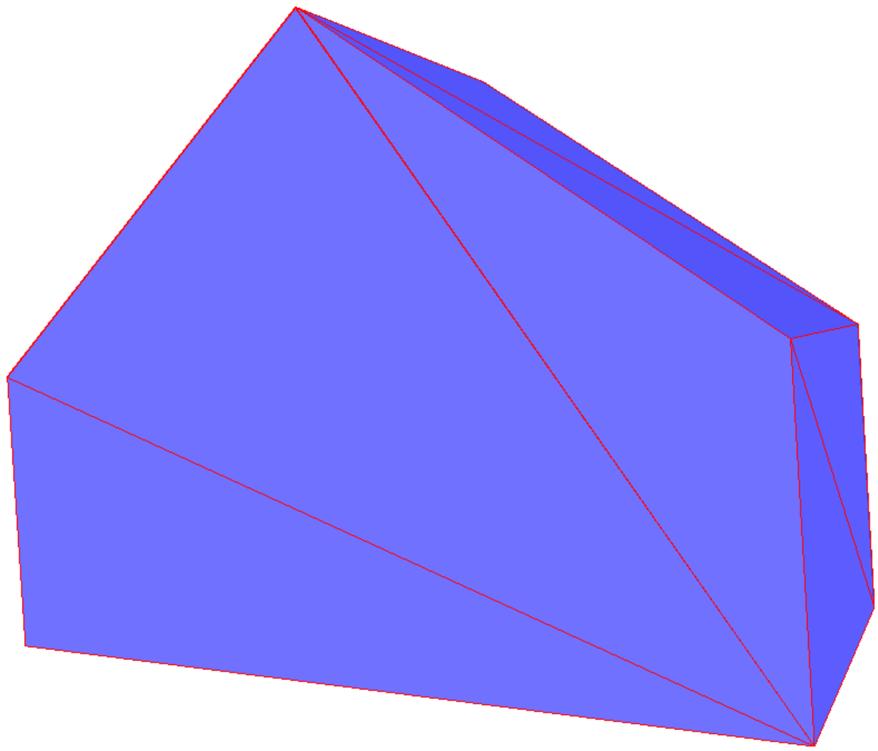


Figure 35: Triangulation of the exterior shell is done to ensure each face is co-planar.

Another step that is done in this process is extruding the building downwards if a basement is present. A basement is indicated in the BAG data by a negative storey number (see Section 3.2). Since the buildings in Rotterdam 3D are set to zero height (see Section 3.3), the basements can be modelled by shifting each of vertices having a zero height downwards. The distance over which the vertices are translated downwards is equal to the amount of negative storeys divided by the total amount of storeys times the original height of the building. This thus results in storeys below the ground that have similar heights as the storeys above the ground.

Now that we are able to construct Nef polyhedra from CityGML data, the LoD2+ generation rules can be designed and implemented which is described in Section 5.2.

5.2 GENERATION RULES AND DATA INPUT

For the generation of LoD2+ interiors multiple generation rules are needed. The methodology for this is discussed in Section 4.3. The implementation is discussed in this section in more detail. Since buildings from the city model can be seen as individual objects (individual solids which are not overlapping due to BAG geometry properties) they can also be enhanced individually. Therefore the generation rules discussed in this section apply to each individual building.

In Section 4.3 it is said that the storeys without offset can be generated by intersection operations of solid bounding volumes with a minimum and maximum height and the original building. For this it thus also needs to be known what the minimum and maximum height of each storey should be. The splitting techniques and the determination of the storey heights are discussed in Section 5.2.1.

When this is done then there needs to be a parametrized solid erosion of the storey as discussed in Section 4.3 as well. For this the combination of set difference and Minkowski sum is used. Each facet of the storey solid thus needs to be buffered and consequently subtracted from the original storey solid. How this is implemented is discussed in Section 5.2.2.

It is also discussed that surface classifcation of the storey solid needs to be done. It was hypothesized that the normal vector of a facet can be used for this as floors, ceilings and walls have different directions in space. This surface classification is not only needed to conform to the LoD2+ specifications as dicussed in Section 4.1, but also to be able to offset the right thickness as different offsets may be applied to for example exterior walls and roofs. The implementation details of the surface classification algorithm are discussed in Section 5.2.3.

To know how much the storey solid should shrink for different sets of buildings based on building type, year of construction and number of storeys thickness values are determined from blueprints. The implementation of the geometric algrithms and the determination of the thickness values are described in Section 5.2.

Although the generation rules are not imported into a real shape grammar implementation this is a possibility such that also new rules can be added to make more detailed representations such as LoD3+. A visual summary of the shape rules discussed in this section is given in Section 5.2.5.

5.2.1 *Splitting buildings into storeys*

The first shape rule in the process of LoD2+ generation is to essentially split a building into storeys (that would thus be equal to LoD1+ if no roofs would be modelled). Since all buildings from Rotterdam 3D are set at zero height and can be assumed level, this basically yields splitting the building solid (obtained through the healing process) in the vertical direction. As each input building is a solid, using Boolean operations the building can be intersected with multiple box-shaped solids (representing the storeys) at different heights to obtain a solid for each storey of the building with the right shape.

As discussed in Section 3.2 the number of storeys (minimum and maximum storey level) is stored in BAG at the municipality of Rotterdam. The first step is therefore to split the building in the right amount of storeys. To ensure that horizontally the complete building is covered by the *intersection solid*, all points belonging to the building are projected on the xy-plane. The convex hull of the projected points covers the complete building. In fact, since buildings are processed one-by-one, a bounding box or the complete xy-plane would yield the same result. Now the *intersection solid* can be created by defining a minimum and maximum z-value for the convex hull. The complete building can now subsequently be intersected with the constructed intersection solid for each storey.

This intersection may generate disjoint solids. For example consider the premises in Figure 36, which is one connected building. In this case when intersection is done with the grey box, two solids are produced. Each produced solid must be handled separately in the following steps, such that in the end each individual solid can be stored as such.

Determination of storey heights

To be able to split buildings, it must be known at what heights storeys start and end. A first assumption may be that each storey has equal height. Clearly this is a simplistic assumption, for example for high rise buildings the first floor is mostly higher than the remaining floors. Also below slanted roofs the storey may be higher to compensate for lost space. Unfortunately no building is the same and therefore a storey cannot be guaranteed to be modelled at the right height, but geometrically still some information can be extracted from the 3D model. For example at the eaves of the roof, usually a new storey starts. Two examples of this are shown in Figure 37. Also when a building is extended the extension has usually the same height as the original storey (see Figure 38). Let us call these heights the characteristic heights of a building.

These characteristic heights can be obtained in the following way. The eaves of the roof is determined from the lowest z-coordinate

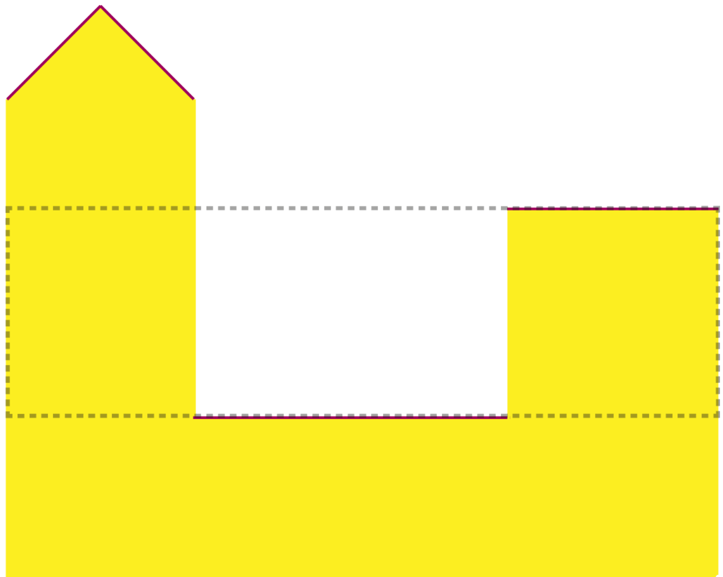
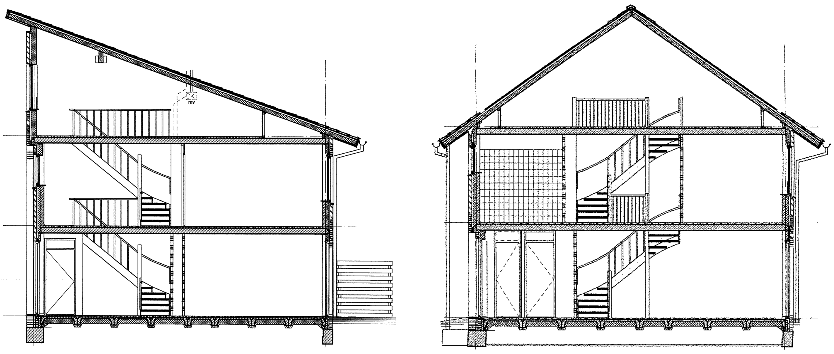


Figure 36: Intersection operations may produce multiple disjoint solids. These volumes need to be handled separately so that in the output each individual solid can be identified.

of the roof polygons of which the normal vector is under an angle smaller than 80° or larger than 100°, thereby excluding flat roof parts. The building extension heights are extracted by comparing the height of the flat roof with the maximum building height. When there is a significant difference, this height is marked as characteristic height as well.

Since the initial assumption is that each storey has the same height, the roof thickness is subtracted from the total height of the building. Subsequently the total height can be divided by the amount of storeys, to obtain the heights at which the building must be split. Each initial height can be compared to the characteristic heights. A snapping tolerance is implemented, such that each splitting height is snapped to the characteristic height whenever the distance is smaller than a defined value (0.5 m is used here). Afterwards the lower and higher floors are redistributed (starting from the last snapped floor such that previously snapped floors are not changed). This process is done from lower to higher storeys.

Due to the fact that it is not known what the quality is of the registered number of storeys in BAG, an additional check may be implemented. For normal houses, based on experience, it is very unlikely that the storey height is smaller than ~2.3 m including floor and ceiling thickness. Furthermore it is unlikely that the storey height for this building type is larger than ~4.0 m.



(a) The highest storey starts approximately at the eaves of the roof.

(b) The highest storey starts approximately at the eaves of the roof (if not taking into account roof overhang).

Figure 37: Two buildings where the highest storey starts at the eaves of the roof. Figures taken from BAG Rotterdam.

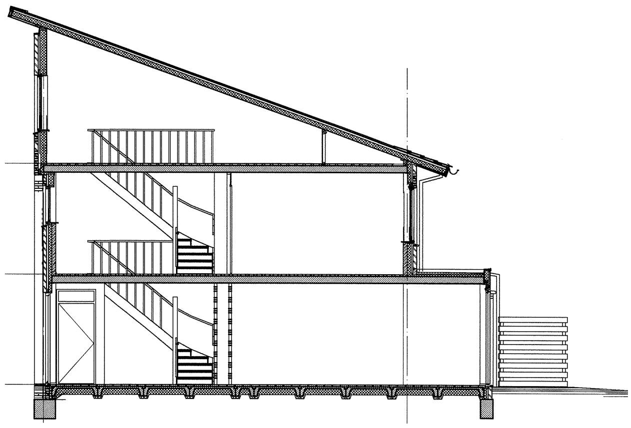


Figure 38: The height of the extension of a house may be an indicator for the height at which a new storey starts. Figure taken from BAG Rotterdam.

5.2.2 *Parametrized solid erosion*

The intersection procedure described in the previous section produces in fact LoD1+ storey solids (that is, if no roofs were modelled for the buildings) because geometrically no interior features are modelled. For LoD2+ also interior walls, floor surfaces and ceiling surfaces should be modelled. To do this an offset must be applied to the shell of each solid.

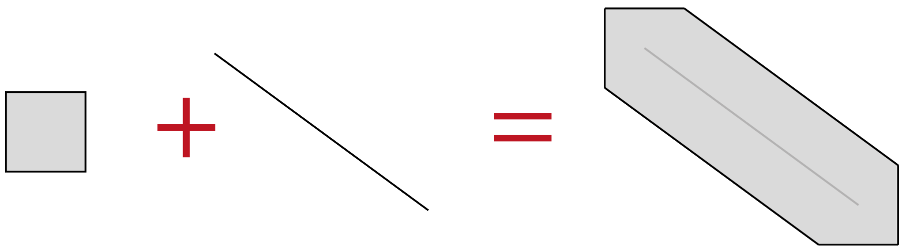
To determine the correct thickness offset, the faces of the solid need to be classified (see Section 5.2.3) and furthermore for each classified surface a thickness must be set (see Section 5.2.4).

As discussed in Section 4.3, in this research the Minkowski sum (Hachenberger, 2007) is used (see Section 3.6) to produce an offset to the exterior shell. Two steps are required for this:

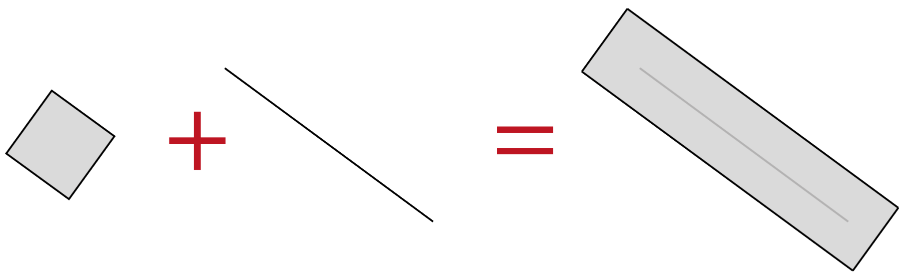
1. Buffering each face of a solid using the Minkowski sum
2. Subtracting each buffered face from the original solid (set difference Boolean operation)

This robot can have different shapes. If an exact buffer is required in all directions, the robot should be a sphere where the radius equals the desired offset. Unfortunately the Minkowski sum is an expensive operation and runs in $O(n^3m^3)$ where n and m are the sum of vertices, halfedges and shalfedges of polyhedron 1 and polyhedron 2 respectively (Hachenberger, 2007). A quick performance test shows that Minkowski sum of a triangular face with an approximated sphere (with 80 triangular facets and 42 vertices) takes about 2-3 times longer than Minkowski sum with a cube whereas the accuracy in the perpendicular offset is then still limited. Therefore a cube is chosen as robot, which is expected to be good enough as the walls of most buildings are perpendicular to each other.

The Minkowski sum is the vector sum of the point sets of both polyhedra. Therefore when using a cube for applying the offset, a rotation should be applied. This is illustrated for a 2-dimensional case in Figure 39. The offset to the line is not the same for both cases. A rotation thus needs to be applied, such that the square is aligned with the edge. Furthermore the robot must be scaled, such that the radius of an inscribed sphere equals the desired offset.



(a) Minkowski sum of a square with a line results in the wrong offset of the line.



(b) Minkowski sum of a rotated square with a line results in the correct offset of the line.

Figure 39: Difference between Minkowski sum with and without a rotation applied to the robot.

In three dimensions, i.e. the Minkowski sum of a cube with a face, two rotations about two axes are required to align the two polyhedra.

It is sufficient to align the two normal vectors of the face and the robot. The normal vectors of a cube with length 2, centered around the origin, are aligned with all three axes. The robot rotation can be done by applying two rotation matrices which transform for example the x-axis onto the normal vector of the face. The rotation of the cube in three dimensions can be achieved by applying two rotations:

1. A rotation θ about the x-axis, corresponding to a pitch motion
2. A rotation ϕ about the z-axis, corresponding to a yaw motion

These two rotations are shown in Figure 40. Note that the pitch rotation is defined positive in the negative z-direction.

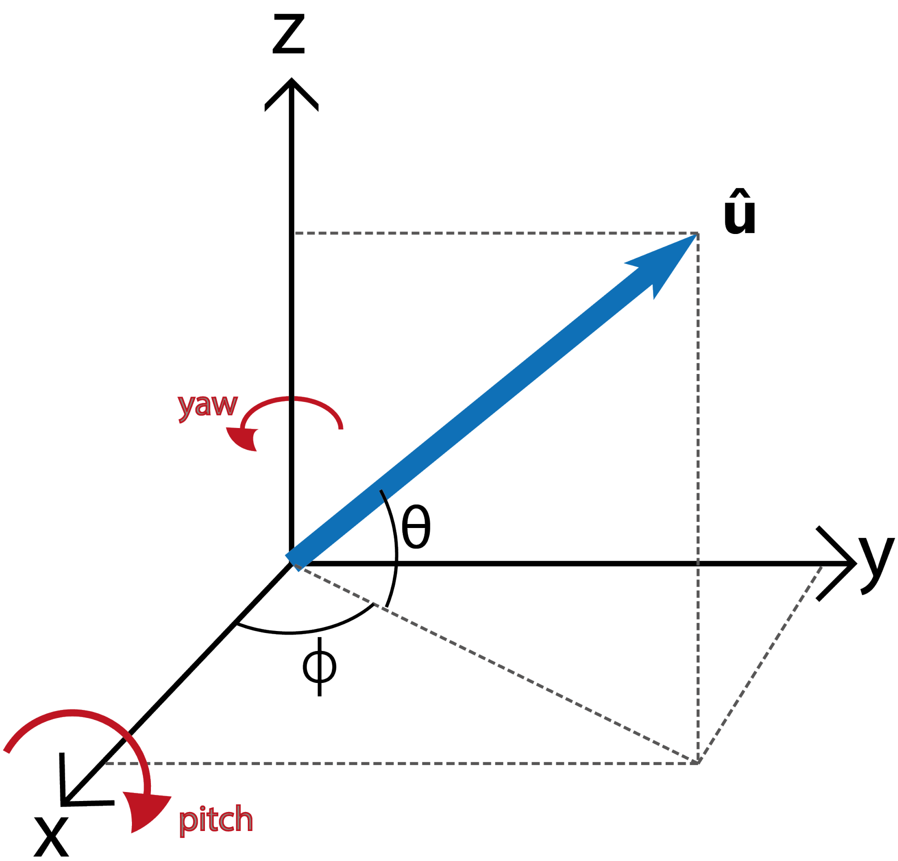


Figure 40: To rotate the x-axis onto the normal vector û it needs to be rotated by a yaw angle ϕ and a pitch angle of $-\theta$. Note that the pitch motion is defined positve in the negative z-direction.

The Euler rotation matrices corresponding to the rotations θ and ϕ are given by Equation 1 and Equation 2 respectively.

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \tag{1}$$

$$R_z = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Given the x, y and z components of a unit normal vector $\hat{\mathbf{u}}$, i.e.:

$$\hat{\mathbf{u}} = \begin{bmatrix} x \\ y \\ z \end{bmatrix},$$

the pitch angle θ and yaw angle θ can be calculated using Equation 3 and Equation 4 respectively.

$$\theta = \arcsin\left(\frac{z}{R}\right) = \arcsin\left(\frac{z}{\sqrt{x^2+y^2+z^2}}\right) \quad (3)$$

$$\phi = \arctan\frac{x}{y} \quad (4)$$

In case the angle between two faces is larger than 180° also an in-plane offset is required as otherwise not enough material is removed from the solid. When not doing so the yellow area in Figure 41 is not taken into account when buffering the two red faces and thus incorrect solids are formed. Although the Minkowski sum with a three-dimensional cube already results in a in-plane offset, this offset is generally larger than the perpendicular offset because the cube cannot be aligned with all three edges of the triangular face. By using different methods for buffering faces, this may be overcome. Figure 42 shows an example of a buffered triangular face.

The complete parametrized solid erosion operation is summarized by the following set of operations on each facet of the solid:

1. Transform Minkowski robot (a cube)
   a) Determine desired offset on basis of surface type
   b) Scale robot to match the desired offset
   c) Determine angles θ and ϕ from the normal vector of the face
   d) Compute rotation matrices about the x-axis and z-axis.
   e) Apply rotation matrices to robot
2. Compute Minkowski sum of transformed robot and triangular face
3. Subtract the Minkowski sum from the solid (storey)

This process is repeated for each face of the solid, after which geometrically the final storey solid is formed. Semantically still some work needs to be done.

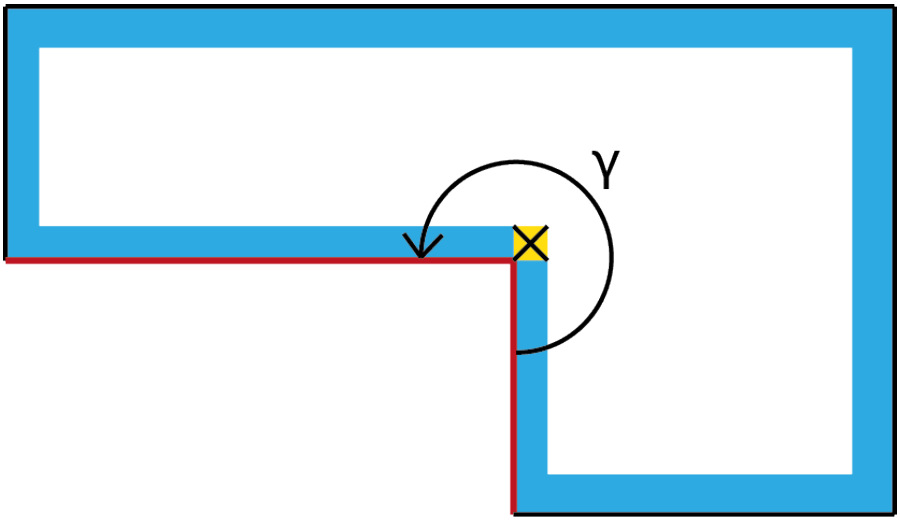


Figure 41: If the angle γ between faces (red lines) is larger than 180°, an in-plane offset is required to exclude the correct parts from the solid. In this case when only buffering the faces in perpendicular direction (blue area) the yellow square would be excluded, which is incorrect.

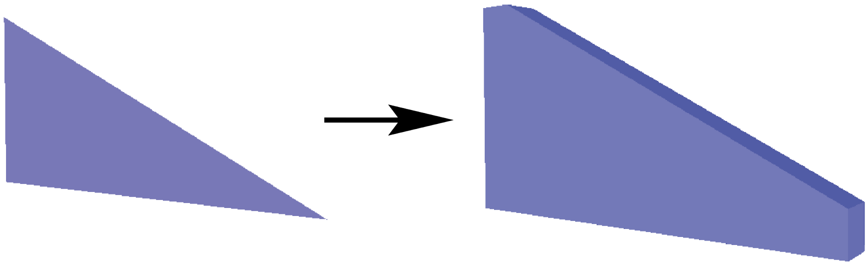


Figure 42: A triangular face buffered using Minkowski sum. The perpendicular offset can be set exactly, whereas the in-plane offset is generally larger.

5.2.3 *Surface classification*

The different features that are modelled in LoD2+ are *WallSurface*, *RoofSurface*, *GroundSurface*, *InteriorWallSurface*, *CeilingSurface* and *FloorSurface*. In the process of LoD2+ generation surfaces have to be classified. First, before the erosion process, the surfaces have to be classified to be able to apply the right offset. After this the surfaces have to be classified again[1] to be able to store the right feature types in the CityGML output.

In the first step, to determine the right offset, the surfaces can largely be classified on the basis of the direction of the normal vector of a face. For this the pitch angle is sufficient, and the horizontal direc-

1 The Nef implementation in CGAL does currently not allow additional information to be stored for each face.

tion is thus irrelevant. When the normal vector points approximately in horizontal direction, the surface is classified as *WallSurface*. When the normal vector points downwards, the surface is classified as *FloorSurface*. All other surfaces are classified as *RoofSurface*. An illustration of the three cases, along with the angles that are used to classify the faces is shown in Figure 43.

In this first step a few additional classification criteria are needed, which are:

- Slanted surfaces are always classified as *RoofSurface*. Since the building is split in vertical direction, the only faces that can be slanted are those that are part of the roof.
- Horizontal faces with a normal vector upwards can be either *RoofSurface* or *CeilingSurface*. Only the horizontal faces that belong to the most upper storey are classified as *RoofSurface*. The remaining faces are classified as *CeilingSurface*.
- Only the horizontal faces with the normal vector pointing downwards belonging to storey level 0, are classified as *GroundSurface*. The remaining horizontal faces with the normal vector pointing downwards are classified as *FloorSurface*.
- Walls that are shared between neighbouring buildings are often less thick and thus need a different offset opposed to normal walls.

To determine which walls are shared walls between neighbouring buildings, some more processing is required. Determination of these shared walls from the 3D model is an expensive task, as it requires intersection operations on a large amount of three-dimensional objects. Instead it is more efficient to process each building in the 3D model separately and determine the shared walls using other data.

The buildings in the 3D model are modelled on the building polygons from BAG. Shared walls can thus be found in BAG data, but some operations are required. First of all the BAG buildings are polygons with no topological relations. However using a topology generator, polygons can be converted to poly-lines while generating the topological relationship of neighbouring polygons (e.g. by using Esri ArcMap). When doing so poly-lines are created with one neighbour id (in case of outer walls) or two neighbour id's (in case of walls touching two buildings). Selecting only the poly-lines with two neighbour id's, results in all the poly-lines that are in fact shared walls (see Figure 44). To make it more easy to process the data, poly-lines can be split into single line segments. The xy-coordinates of the line segments can now be extracted along with the adjacent polygon id's (BAG id's). The ArcMap model showing this process is shown in Figure 45.

Now for each classified wall during the LoD2+ generation, an additional check can be performed to determine whether it is a shared

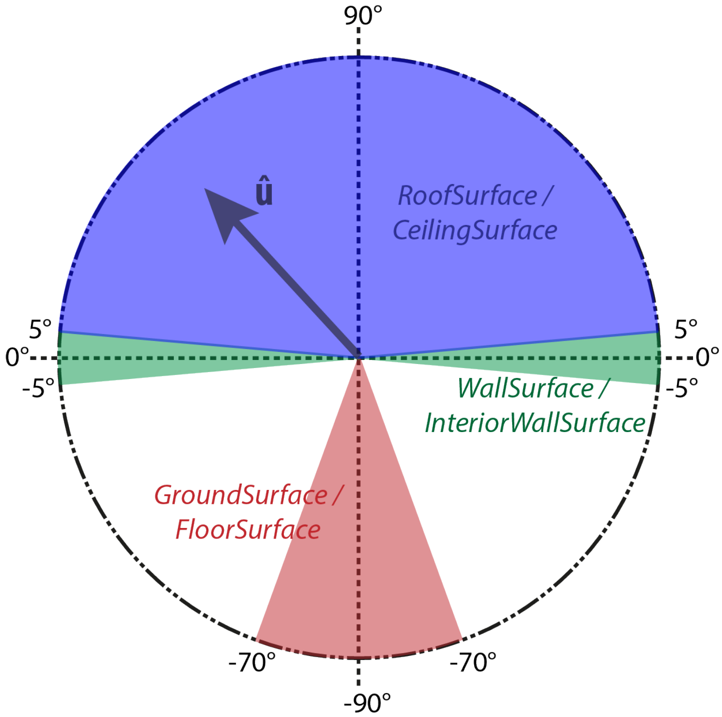


Figure 43: Classification of the surface is done on the basis of the pitch angle of the normal vector of each face. The angles in the figure show which angle results in which surface type.

wall. Each point of the wall is projected on the xy-plane and consequently the distance between each of points and the lines belonging to shared walls of the BAG premises is computed. When all three points are within a certain distance from the line, it is classified as shared wall. The snap-rounding tolerance is set to 0.01 m to account for differences in accuracy. Now the desired offset can be applied for shared walls.

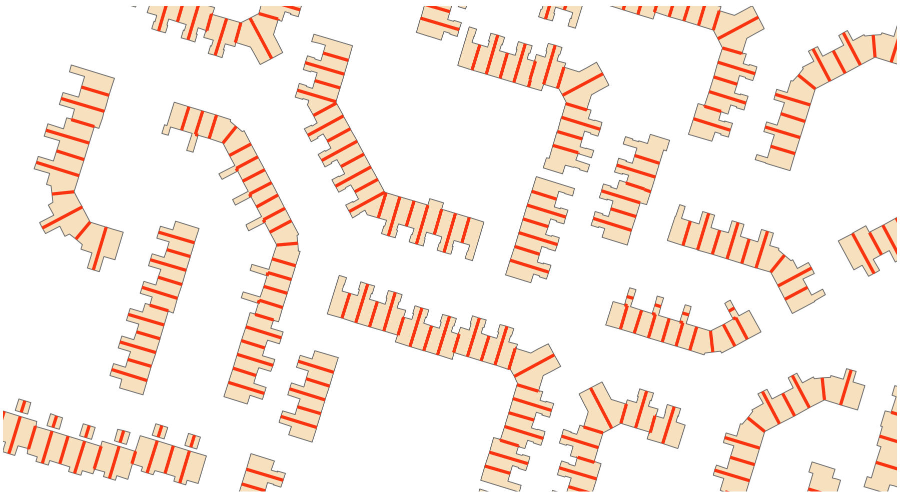


Figure 44: Map showing the shared walls (thick red lines) in part of the test site. For each of the lines the xy-coordinates are extracted with additionally the neighbouring BAG id's.

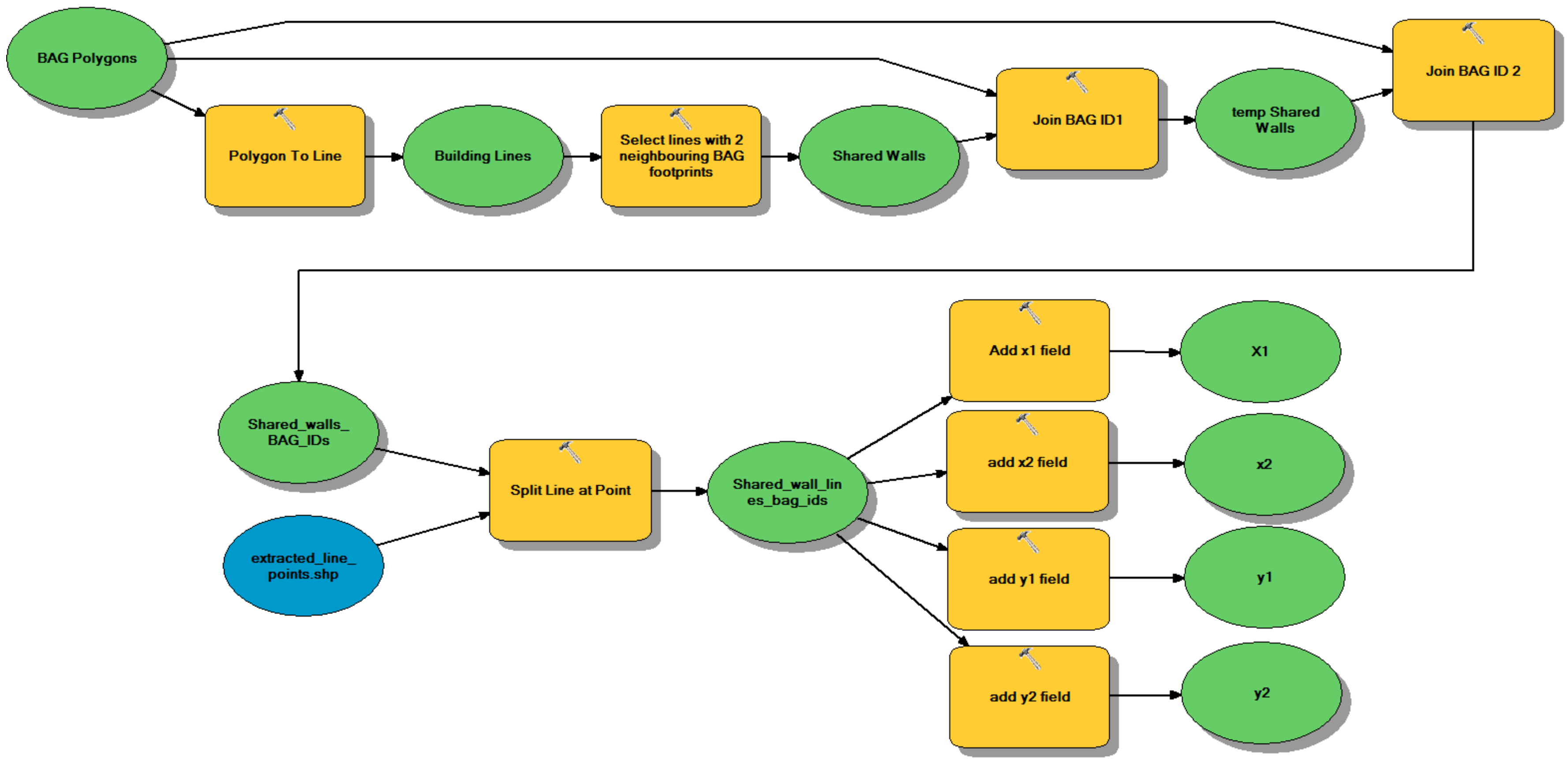


Figure 45: This figure illustrates the designed model for ArcMap to determine the shared walls from the 2d BAG data. The output are the xy-coordinates of the two points of a line with the two neighbouring BAG id's.

The second classification of the surface, to be able to write the right feature types in the CityGML output, is less complex. Each surface can be classified with the interior feature types *InteriorWallSurface*, *CeilingSurface* and *FloorSurface* using the pitch angle shown in Figure 43.

5.2.4 *Thickness input*

In Section 4.2.1 it is discussed that the thickness offsets of the different surface types cannot be determined from literature alone. Therefore for different building categories the thickness values have to be determined from building blueprints. To do so, the complete building set is split into smaller sets based on building type, year of construction and number of storeys.

The first category is the building type, see Section 4.2.1. In Figure 46 a histogram is shown for each of the building types present in Hoogvliet-Zuid. The majority of buildings are normal houses (non-stacked), and therefore this building type is treated as one of the categories. The second type that is important is the stacked building (with more than one residential unit per premises; e.g. flats). The third building type category should then cover the remaining building types, however garages still make up quite a substantial part of the remaining building types. These can however still be covered later, because they normally only have one storey.

The second category (years of construction) is split according to the relation between cavity sizes of cavity walls given by ISSO (2011) in Section 3.4. For each of the nine sets that are now generated the histogram of the amount of storeys are plot such that these sets can approximately be split evenly again.

The sets can now further be split according to the amount of storeys (approximately evenly). Buildings with building type *Other* and one storey are always handled separately to cover the garages as well. The result of this splitting process is shown in Figure 47 along with the number of buildings in each category for Hoogvliet-Zuid.

According to these categories the average thickness values are determined. The aim was to inspect at least 3 % of the buildings in each set. Unfortunately this was not possible for all of the sets, as for some sets there were no building blueprints available or the blueprints were of very low quality such that it was impossible to determine the values (see Appendix B for example building blueprints). Furthermore only one blueprint also contained a side view, such that it was not possible to determine floor/ceiling thickness.

The result of this process is shown in Table 7. It was expected that thickness values increase for higher buildings, but there does not seem such a relationship. Although it seems to be the case that the wall thickness is indeed larger for newer buildings, the building type and the amount of storeys seem to have a lesser influence. Furthermore the thickness values are not deviating very much from each other, with the exclusion of the building type *other* with *one storey* because they are usually garages with single brick walls. It can also be seen that for most sets the standard deviation of the wall thickness and shared wall thickness is mostly smaller than the overall standard deviation. From this it can be concluded that there is a coherence between thickness values for each of the buildings in each of the sets, but this may also be induced by the fact that buildings in each individual set may be designed by particular architects at that time.

Due to the fact that a few cm larger or smaller thickness may have a substantial impact on the net internal area later on, the determined thickness values are used as input to the parametrized solid erosion algorithm. An example calculation shows that for a rectangular building of 8×8 m with three storeys, a difference of 5 cm results in a difference of $\sim 5\,\mathrm{m}^2$ $(4 \times 8 \times 3 \times 0.05$ m).

Building Type	Construction year y	Nr. of Storeys x	t_{wall} [cm]	σ_{wall} [cm]	$t_{sharedWall}$ [cm]	$\sigma_{sharedWall}$ [cm]
Non-stacked	$y < 1970$	$x \leqslant 2$	27	1.1	11	0.0
		$x \geqslant 3$	27	0.0	12	2.4
	$1970 \geqslant y \leqslant 1985$	$x = 2$	27	0.0	10	0.6
		$x = 3$	28	2.2	12	2.6
		$x = 4$	27	0.0	9	0.9
	$y > 1985$	$x = 2$	28	1.3	13	1.7
		$x = 3$	30	4.6	12	1.7
		$x = 4$	25	0.0	12	0.0
Stacked	$y < 1970$	$x \leqslant 5$	29	1.2	12	2.1
		$5 > x \leqslant 10$	38	3.0	11	0.0
		$x > 10$	25	2.1	9	1.4
	$1970 \geqslant y \leqslant 1985$	$x \leqslant 5$	28	2.2	11	0.4
		$5 > x \leqslant 10$	26	*	11	*
		$x > 10$	29	*	12	*
	$y > 1985$	$x \leqslant 5$	30	*	12	0.6
		$5 > x \leqslant 10$	38	*	13	*
		$x > 10$	35	10.7	15	*
Other types	$y < 1970$	$x = 1$	14 * *	*	14**	
		$x \geqslant 2$	31	7.1	11	0.0
	$1970 \geqslant y \leqslant 1985$	$x = 1$	14	3.5	14 * **	
		$x \geqslant 2$	30	3.5	10	*
	$y > 1985$	$x = 1$	14 * *		14 * *	
		$x \geqslant 2$	36	6.6	13	2.5
		Average [cm]	27.6		11.7	1.1
		St. dev. [cm]	6.6		1.5	

Table 7: This table shows the determined input thicknesses from building blueprints for different building categorizations. * Only one building blueprints availble ** No readable blueprints available, therefore the same thickness is used as Other types -> 1970-1985 -> 1 storey. *** No information found in blueprints, therefore the same as outer wall thickness.

Unfortunately it has proven not possible to determine values for the thickness of the ceiling/floor and roof using this method. Based on expert knowledge assumptions[2] are made for these parameters. For all buildings the roof thickness is set to 30 cm and the ceiling/floor thickness is set 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 as well as a ceiling thickness of 10 cm. On the other hand the thickness offset for the classified *GroundSurface* is set at 1 cm to ensure the floor on ground level is not intersecting the bottom plane of the exterior shell, but is still approximately at the same height as the terrain which is intuitively almost always the case.

2 Common values are taken from online building construction communities such as http://bouwinfo.be/forum/forum.php.

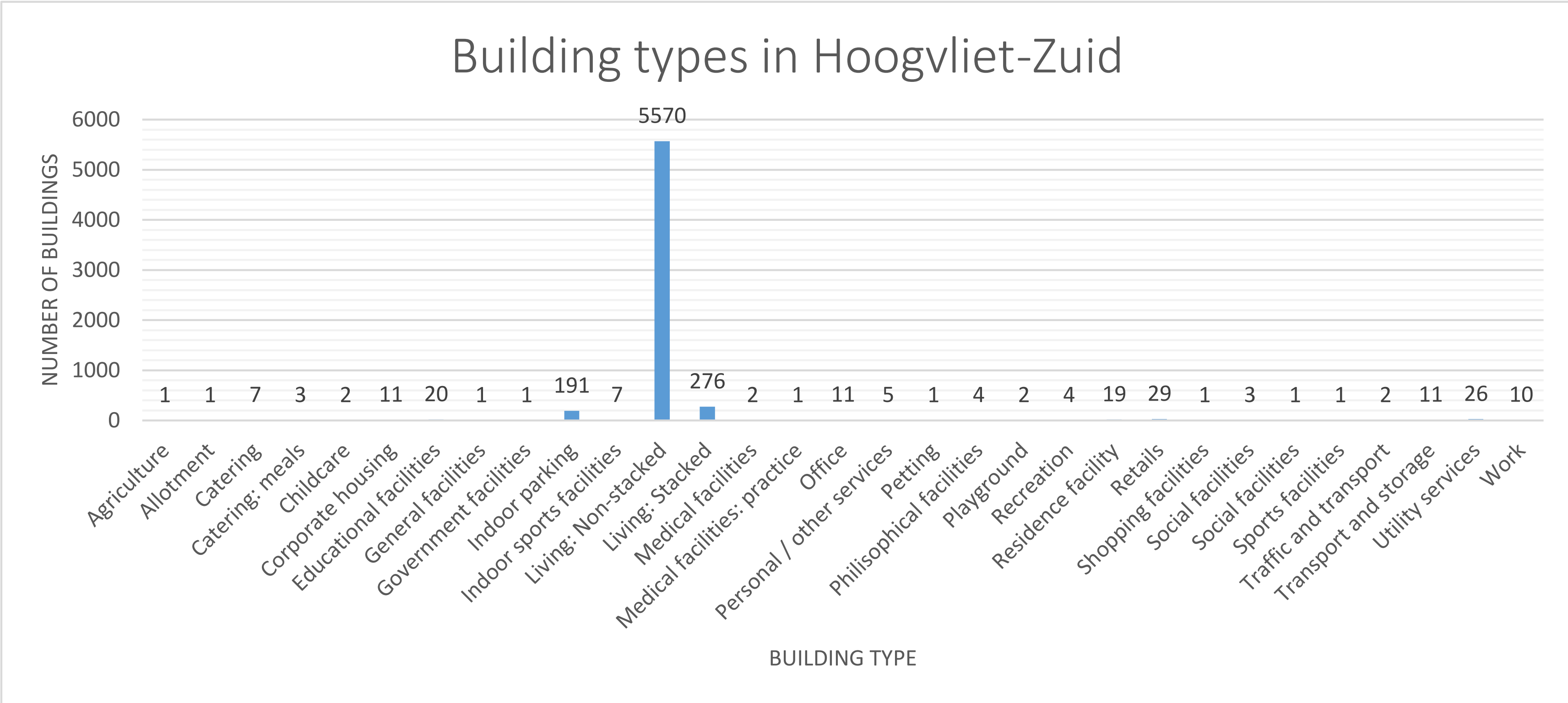


Figure 46: This histogram shows the different building types present in Hoogvliet-Zuid. The majority of the buildings are normal houses (Living: Non-stacked), followed by flats/appartments (Living: Stacked).

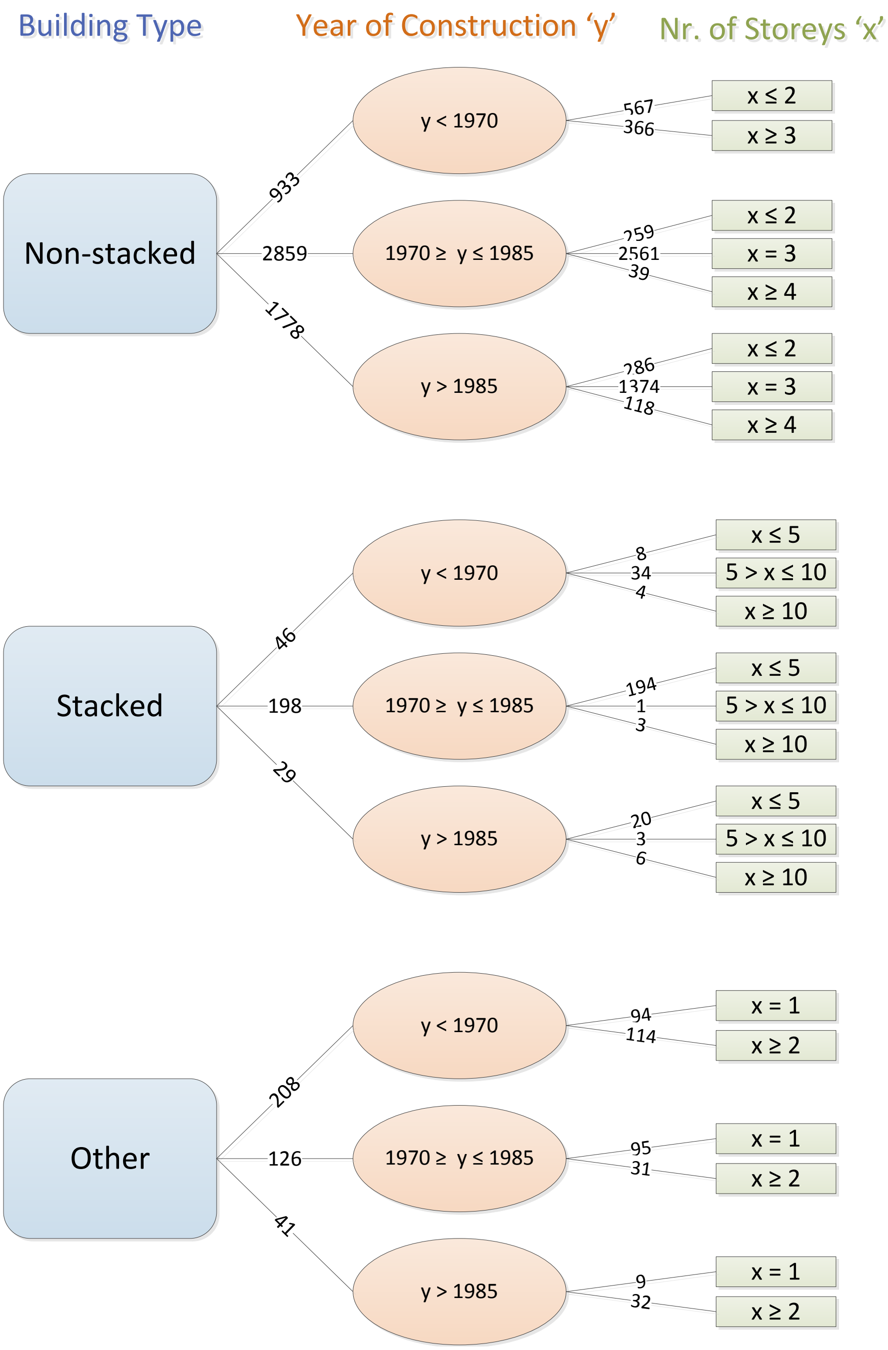


Figure 47: Classification of the buildings is done in three steps. First by building type, second by year of construction and finally by the number of storeys. Each of the categories is shown in this figure along with the number of buildings in Hoogvliet-Zuid belonging to this category.

5.2.5 *Shape grammar summary*

The generation rules described in this section may be formalized in a shape grammar. In fact by designing only two parametric shape rules the geometry of LoD2+ is formed. The shape rules are visualized in Figure 48.

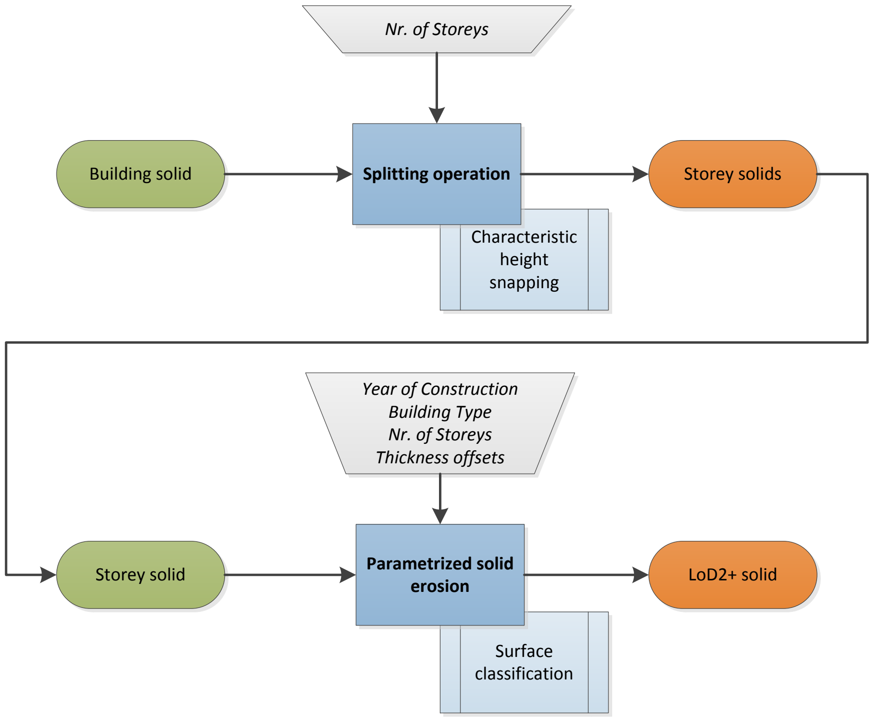


Figure 48: This diagram shows the parametric shape rules that are designed: the building splitting operation and the parametrized solid erosion. The parameters which alter the shape rules as well as the sub-processes are shown in the diagram as well.

5.3 MODEL RESULTS AND VALIDITY

Given the LoD2+ generation rules described in Section 5.2 software has been developed which creates building interiors for each building in a CityGML LoD2 model. After the generation of the interior solids, they are output to a CityGML file according to the specifications which can be found in Section 4.1. Since the algorithms described in this chapter produce solids, they need to be inverted to yield inner shells in the LoD2 exterior shells. For more implementation details on of the software, please refer to Appendix C. In this section the output is discussed. In Figure 49 the LoD2+ model for part of Hoogvliet-Zuid is shown.

The first shape rule is to split storeys, which is a relatively simple operation. The determination of the start and end heights of the

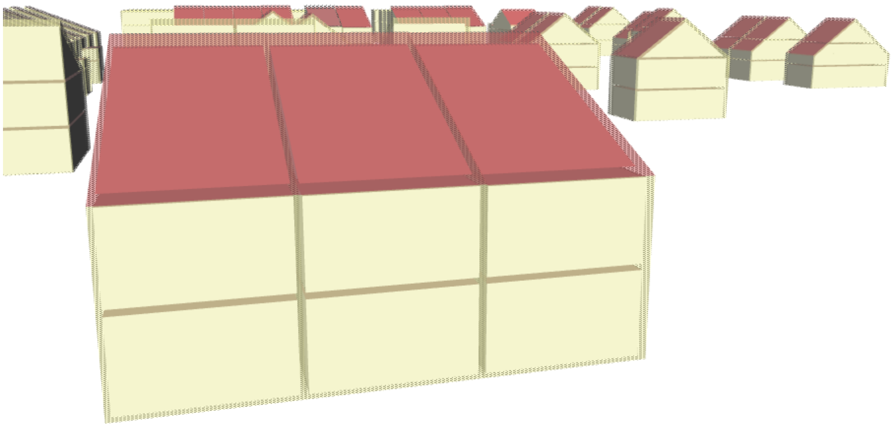


Figure 49: LoD2+ result for a small part of Hoogvliet-Zuid. The exterior shell is transparent such that the storey solids are visible.

storey using the snapping operation works out usually well (for example see Figure 50), however is depending on the tolerance used. An example is shown in Figure 51. Two buildings in this row of houses are not snapped to the eaves of the roof, because the distance between the initial height and the characteristic height is somewhat larger than the tolerance while this is not the case for the other buildings. A larger tolerance may solve this problem, but may introduce others. For example the attic storey height may become too small if splitting the building at the eaves of the roof if the distance between the initial height and the characteristic height is too large.

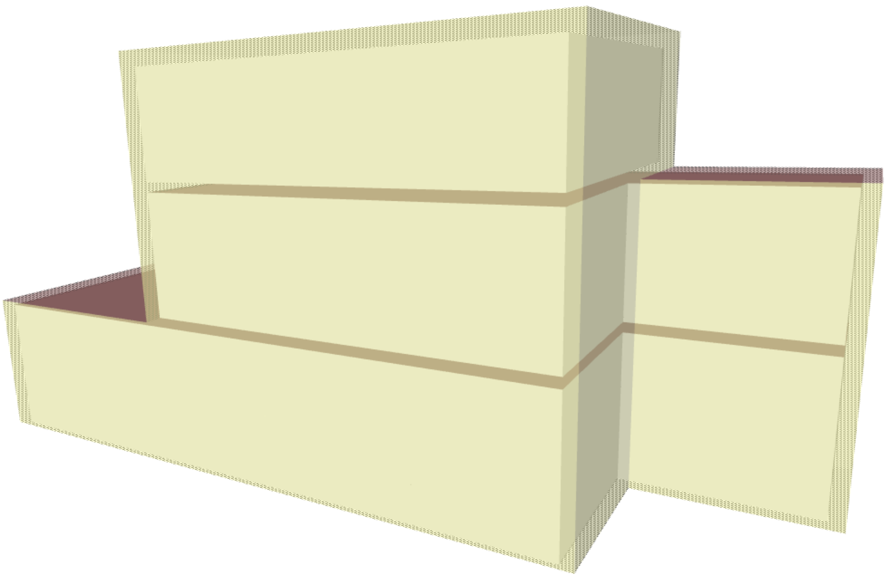


Figure 50: Using the storey height snapping algorithm the storeys are aligned with lower roof elements which would otherwise not be the case.

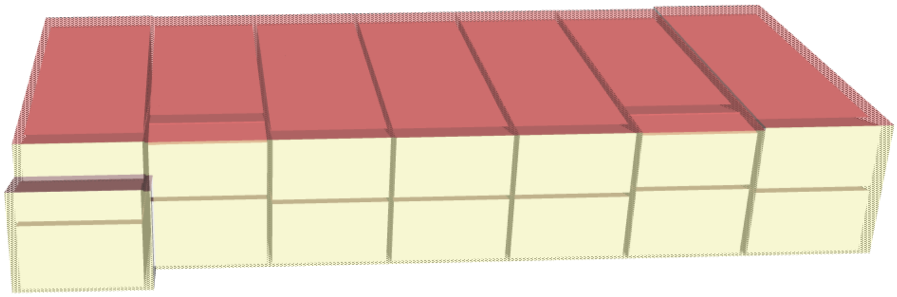


Figure 51: Since a tolerance is used for snapping storey heights to characteristic building heights, odd results may appear where neighbouring buildings got split differently.

The parametrized solid erosion usually works well as for most buildings the walls are perpendicular to each other and therefore no in-plane offset is required. Small problems may arise when the angle between faces is larger than 180° or where neighbouring faces have a different thickness offset. An example of this is shown in Figure 52. In this case the Minkowski robots for both the roof surface and the wall surface leave an indentation in the storey solid because the angle between facets is larger than 180°.

5.3.1 *Validity*

Since only Boolean operations are used in the process of LoD2+ generation and the polyhedra are based on Nef, it can be argued that the LoD2+ storey solids should be valid geometry:

- Each volume is handled individually (in case intersection results in multiple disjoint solids) and therefore each solid is output individually to the CityGML LoD2+ output file.
- Since regularized operations are used, dangling faces are removed and non-manifold geometries are eliminated.
- Degeneracies are explicitly handled in the Nef polyhedron implementation by Granados et al. (2003).
- Non-planar surfaces cannot be present as the output is a triangulation.
- Self-intersections are not possible since only Boolean operations are used to split buildings and remove material.

Using validator software by Janssen et al. (2012) the output is validated. When checking the CityGML output with the software frequently still the error *degenerate face* arises (see Section 3.7 for more information on this error). This error is not expected as Nef polyhedra are handling all degeneracies (Hachenberger, 2007). Investigation

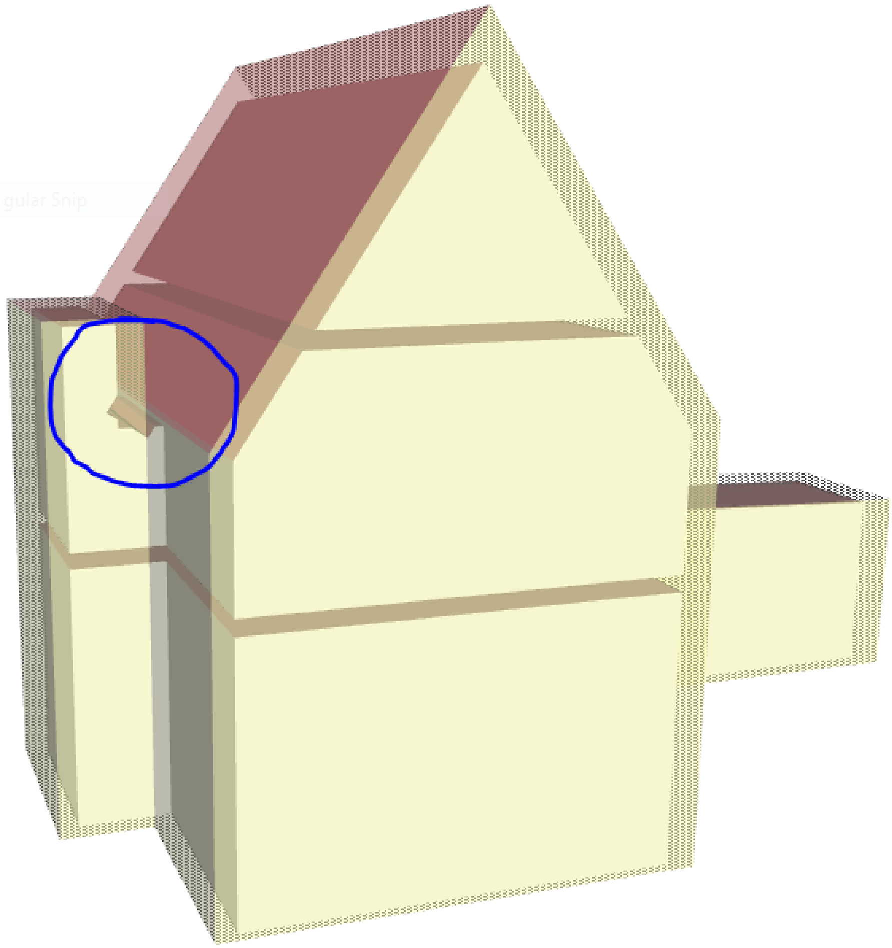


Figure 52: At angles larger than 180° extra faces are formed. The Minkowski robot for both faces can be identified in the produced solid due to the in-plane offset.

shows that the validation software uses a tolerance to determine if a face is degenerate. Indeed the output contains some very small triangles (see Figure 53). This is inconvenient (for the storage space needed as well as the viewing performance), but is not necessary invalid. Simplification of the surface using quadric edge collapse decimation (Garland and Heckbert, 1997) may eliminate this issue but is not further investigated in this research.

5.3.2 *Performance*

The input dataset consists of 6,225 premises of which in the end LoD2+ could be generated for 4,461 premises. Invalid input (e.g. missing facets, non-planar facets, dangling facets) is the reason that 28.3 % cannot be converted. Since no extensive validation is done within the developed software, some invalid buildings are still processed resulting in either still correct output (due to the repair done within

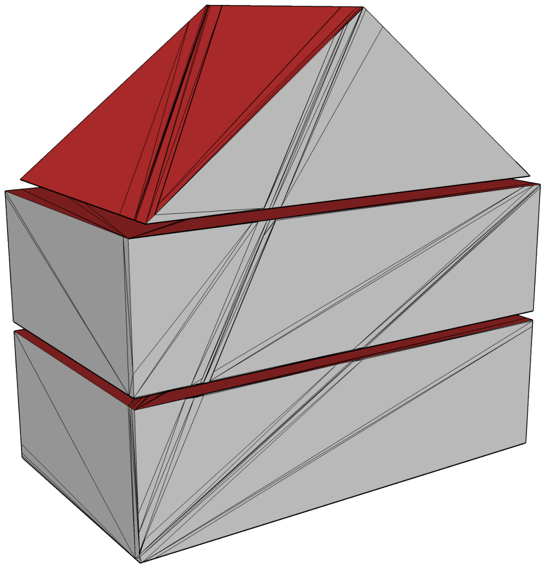


Figure 53: Along the process of generating LoD2+ buildings very small triangles are formed which validator software may classify as degenerate.

CGAL), incorrect output or very occasionally crashing of the software. The latter possibility is handled by resuming the conversion from the next building on restart of the software.

The time it takes to produce LoD2+ from a LoD2 building depends very much on the complexity of the building. It may take from 2 s for very simple buildings such as garages to 400 s for tall buildings with a large number of storeys and complex exterior shapes. On average the whole process (including determination of the net internal area described in Chapter 6) takes 16 s per building (see Figure 54) on a computer system with a quad-core CPU of 3.20 GHz and 8 GB of RAM, however the developed software does not utilize all CPU cores and thus performance improvements can be achieved. For the city of Rotterdam which has approximately 125.000 premises this would mean that it takes approximately 23 days to produce a LoD2+ model of the whole city.

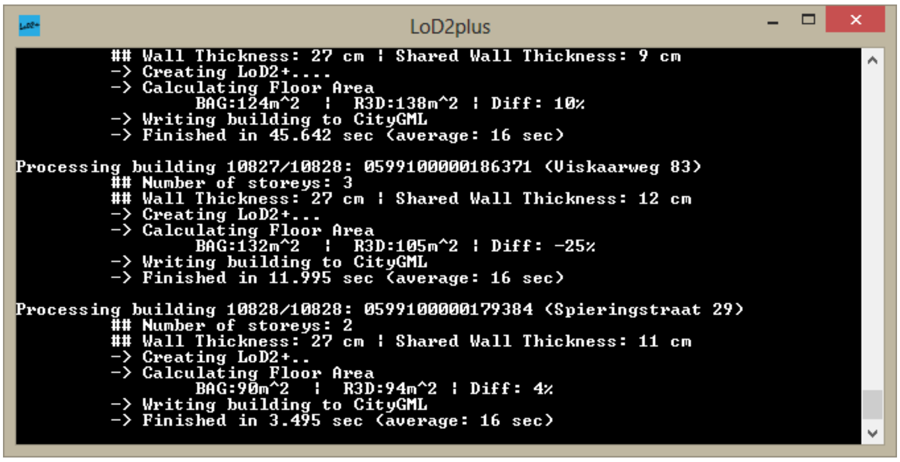


Figure 54: Screenshot of the software showing the average time it takes to generate a LoD2+ building after having processed the complete dataset of Hoogvliet-Zuid.

6 NET INTERNAL AREA CALCULATION

From Chapter 5 it is now known how to produce an LoD2+ model. In this model the *InteriorWallSurface*, *FloorSurface* and *CeilingSurface* features are modelled. From the floor polygons it is thus already easily calculated what the exact *FloorSurface* area is. This is however not conforming the definition of the net internal area (NEN 2580; see Section 3.2.2).

To calculate the net internal area according to its definition still some parameters (parts of the building which do not contribute to the net internal area) have to be taken into account. In Section 4.4 it was discussed that for all parameters there are three options:

- It is possible to determine the influence of that part of the building geometrically and subtract this area.
- It is possible to determine the influence of that part of the building using semantic information and subtract an area range.
- It is not possible to determine the influence of that part of the building at all.

In the first section of this chapter (Section 6.1), it is discussed what surface areas should still be subtracted to get the true net internal area and also which areas can still be calculated and how. Those parameter(s) that can still be taken into account are discussed in more detail.

In Section 6.2 it is discussed why and how, even though not all parameters from the net internal area definition can be estimated, the calculated area from a LoD2+ model is still useful for the validation of BAG. WOZ is another key register for the value for real estate properties. It is known by municipalities that, although both BAG and WOZ register the net internal area of residential units, there are differences between the net internal area values. In Section 6.3 it is discussed what the differences are between the two registers.

In Section 4.5 it was discussed that the validation of the modelled interiors will be done on the basis of the net internal area because it can be done automatically for a large number of buildings. By calculating the net internal area from the LoD2+ city model, both the accuracy of the modelled interiors and the quality of the net internal in BAG can be evaluated by manually inspecting samples from the largest differences between BAG and the net internal area calculated from the LoD2+ model. An extensive evaluation on the causes of the largest differences between BAG and Rotterdam 3D is found in Section 6.4.

6.1 NET INTERNAL AREA PARAMETERS AND ALGORITHMS

From a LoD2+ building the surface area can be calculated which excludes the load bearing walls (exterior). It is assumed that all exterior walls are load bearing. Since the LoD2+ model storey shells are triangulated this surface area relates to the sum of the area of all triangles which are classified as *FloorSurface*. From Section 3.2.2 it is known that certain areas still need to be subtracted. These are:

- Voids in-between storeys (if $\geqslant 4\,\mathrm{m}^2$)
- ~~Inner load bearing walls~~ (see Section 3.2.2)
- Locations where net height $< 1.5\,\mathrm{m}$
- Elevator/pipe/cable shafts (if $\geqslant 0.5\,\mathrm{m}^2$)

On the other hand cavities which are smaller than $0.5\,\mathrm{m}^2$ should be added to the gross floor area if it is not included yet. Since the gross floor area from LoD2+ takes into account the full exterior geometry, it is already included. For each of the other parameters described here it is discussed whether they can still be subtracted and what the consequence would be of not doing so. If the area can still be taken into account it is described how.

6.1.1 *Voids in-between storeys*

Voids in-between storeys are holes in the floor which can facilitate the movement of people (step wells) or are present because of other design reasons (see Figure 55). According to NEN 2580 these holes only

Figure 55: A storey which is only partly spanned by a floor.

have to be subtracted from the gross floor area if they are larger than $4\,\mathrm{m}^2$. By inspecting building blue prints it is found that most normal houses do not have step wells larger than $4\,\mathrm{m}^2$. If it can be determined what kind of buildings do possess large step wells or loft storeys, for these categories the size of the step wells can be subtracted. In the scope of this research this is however found too difficult to determine from the available building blueprints.

6.1.2 *Locations where net height < 1.5 m*

In NEN 2580 it is defined that locations where the net height (the height between floor and ceiling) is less than 1.5 m are not considered as usable surface area. This mostly occurs below slanted roofs on an attic floor for example, but a slanted roof may span more than one storey.

Since geometrically the interior volumes are available, using Boolean operations the area below 1.5 m height can be easily calculated. By extruding each *FloorSurface* triangle of the LoD2+ model by 1.5 m solids are created which may or may not be (partly) outside the storey solid. For storeys of sufficient height with vertical walls these extruded solids are completely located within the storey solid. When slanted roofs are present or the storey is not of sufficient height, the extruded solids will be (partly) outside.

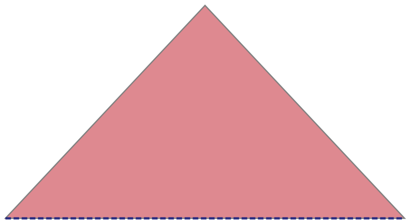
By subtracting (Boolean set difference) each extruded triangle from the original LoD2+ storey the part that is outside the storey solid remains. Then by projecting the resulting solid(s) on the xy-plane the *FloorSurface* area below 1.50 m is covered twice. Division by two results in the area that thus needs to be subtracted from the LoD2+ *FloorSurface* area to take that part into account. This procedure is illustrated by Figure 56.

6.1.3 *Elevator/pipe/cable shafts*

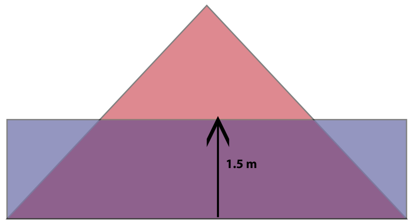
Elevator, pipe or cable shafts are mostly relevant for tall buildings. These are vertical connections within a building which are used for cables or water pipes for example. Since in normal house these pipes and cables usually are placed within wall cavities, for non-stacked buildings this can be assumed to have a negligible effect on the net internal area (Briels, 2004). For larger buildings with multiple residential units this may not be the case (see Section 4.4).

Also elevators are usually not present in normal houses. High buildings with multiple residential units may have an elevator (which is also registered in BAG; see Section 3.2). Although it is known which buildings have elevators, it is not known what size they have and how many elevators are present in a specific building. In Section 4.4 it was discussed how these parameters can still be taken into account if more information is available.

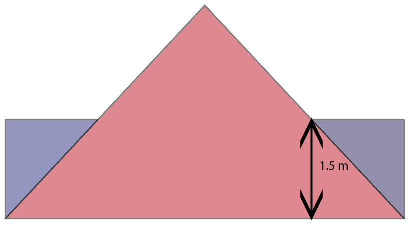
Unfortunately, during this research it has proven not to be possible to determine these values for the building categories as only



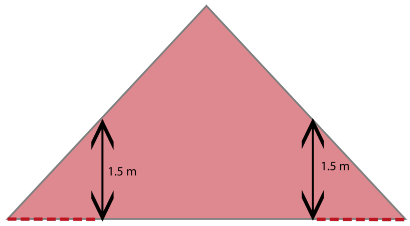
(a) Side view on a storey solid where the dotted line represents the *FloorSurface*



(b) Extrusion of a FloorSurface facet by 1.5 m



(c) Subtracting the storey solid from the extruded *FloorSurface* facet



(d) Projection on the xy-plane results in the FloorSurface where net height is smaller than 1.5 m

Figure 56: Procedure for determining the area where the net height is less than 1.5 m

blueprints of the residential units are available and not for the whole premises. Therefore no information could be extracted containing the influence for each of the parameters on the net internal area. When more information can be obtained, this method can be applied. However the result may have less meaning than in the case of one residential unit per premises (non-stacked buildings), since the registered net internal area corresponds to the residential unit which relate one-on-one to the premises in the latter case. If we do plot the differences between the surface area of stacked buildings from the LoD2+ buildings and BAG we find that for approximately 85 % of the buildings the area from the LoD2+ buildings is larger (see Figure 57).

For this reason the comparison done in Section 6.2 and Section 6.3 are done on non-stacked buildings only.

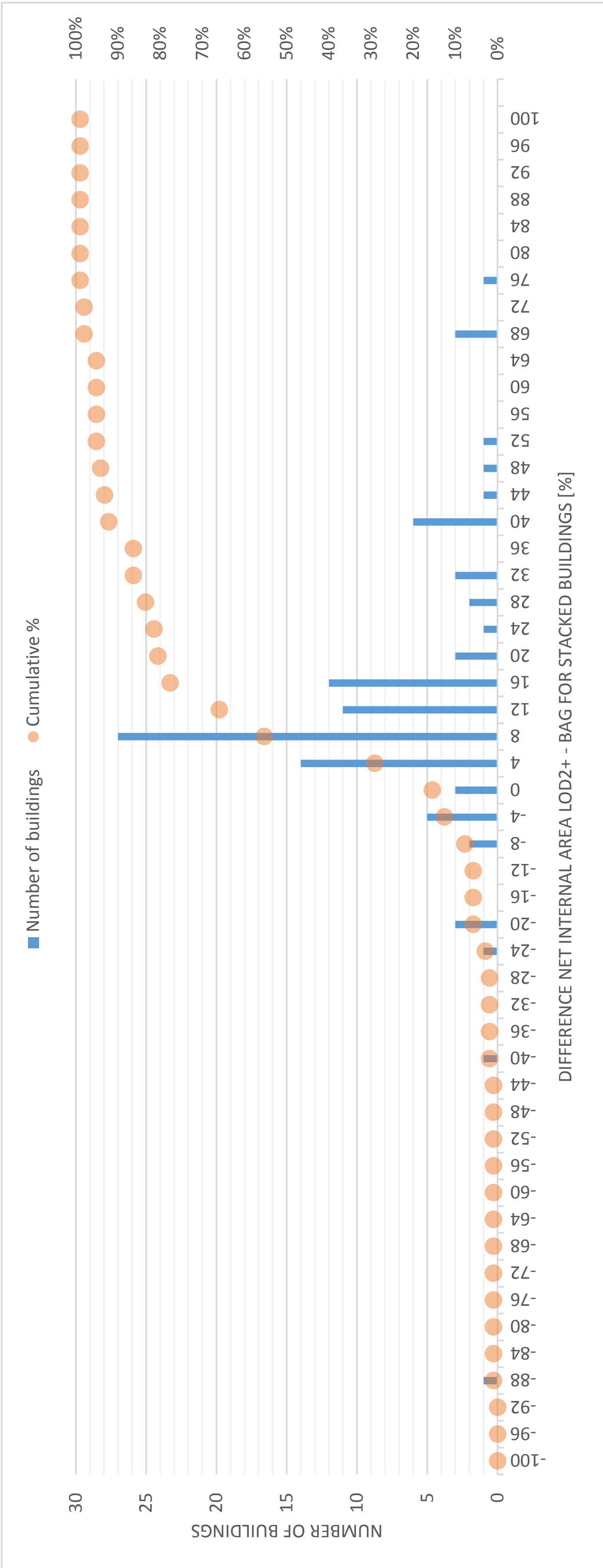


Figure 57: Histogram of the differences between the net internal area calculated for stacked buildings from the LoD2+ model and BAG. When the difference is larger than zero, the net internal area of the LoD2+ buildings is larger than those of BAG.

6.2 COMPARISON WITH BAG

As discussed in Section 6.1 the net internal area can only be calculated to a certain extent. In the LoD2+ model the load bearing structure is modelled. Geometrically also the surface area where the net height is smaller than 1.5 m can be determined. Other parameters like voids in-between storeys and elevator/cable/pipe shafts are more difficult to determine and are therefore out of scope of this research.

It is also discussed that the net internal area for premises with more than one residential unit is too difficult to determine as required information is lacking. Therefore the comparison of the calculated net internal area from a LoD2+ model with the available net internal area in BAG is done one *non-stacked* buildings only for the test-site Hoogvliet-Zuid.

The difference between the net internal area from LoD2+ buildings and BAG is shown in Figure 58. From this figure it can be seen that although not all parameters from the definition of net internal area (NEN 2580) are taken into account because of lack of information, for the majority of the buildings the net internal area is smaller than the values available in BAG (73.4 %). Furthermore we observe that most frequently the net internal area from the LoD2+ model is 16 % smaller than registered in BAG. On the other hand from the absolute differences in percentages in Figure 59 we can see that for approximately 95 % of the buildings the calculated net internal area differs less than 30 % from BAG.

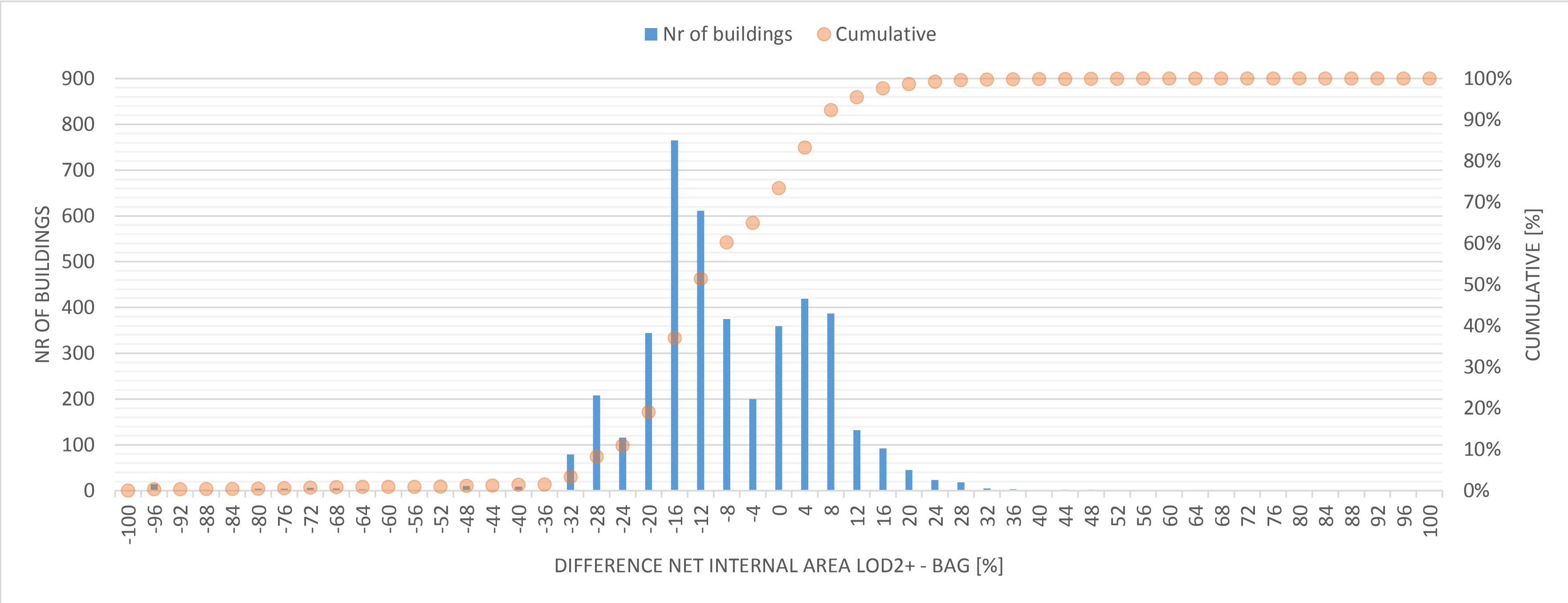


Figure 58: Histogram of the differences between the net internal area calculated for non-stacked buildings from the LoD2+ model and BAG. When the difference is larger than zero, the net internal area of the LoD2+ buildings is larger than those of BAG.

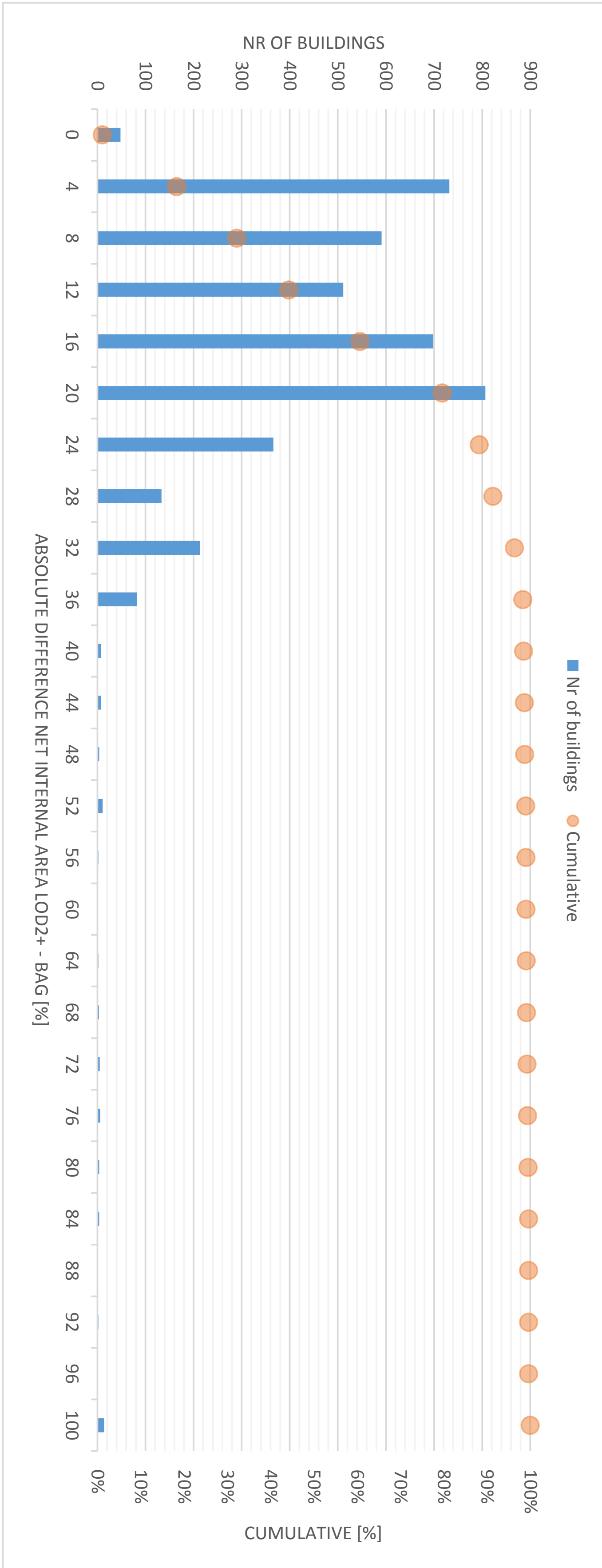


Figure 59: Histogram of the absolute differences in percentages between the net internal area calculated for non-stacked buildings from the LoD2+ buildings and BAG.

From Section 3.2.2 it is known that the maximum allowed deviation is $1.15\sqrt{A}$ where A is the true net internal area. When assuming the calculated net internal area from the LoD2+ buildings is 100 % correct, we see that only 38 % of the buildings satisfy this allowance. This means either that the modelled interiors are not accurate or there are significant errors within the BAG registration.

The latter seems to be the case. When investigating individual buildings it is found that for many buildings the net internal area calculated from the LoD2+ model would compare to the values from BAG if the surface area where the net height is less than 1.5 m was not subtracted. This is shown in Figure 60. This figure shows that then the differences between the calculated areas and the BAG areas are much smaller. In this case 77 % of the buildings satisfy the allowed deviation of $1.15\sqrt{A}$.

Since municipalities started the registration of net internal area before NEN 2580 was introduced, it is reasonable to argument that before that time a different definition was used which did not exclude surface areas where the net height is less than 1.5 m. The following classification is used to determine for how many buildings this is at least the case:

- Buildings with area below 1.5 m at least 10 m^2
- Calculated *FloorSurface* area at least 2 times closer to BAG area than the calculated net internal area

For example a building which has a net internal area registered in BAG of 126 m^2. According to the LoD2+ model the area is actually 128 m^2 and 16 m^2 should be subtracted because the net height is less than 1.5 m. The difference between the net internal area is then 14 m^2 whereas the difference with the *FloorSurface* area is only 2 m^2. Since the latter value is more than 2 times as small, it is assumed that for BAG indeed this parameter of the net internal area definition was not taken into account. Applying this classification for all the buildings, yields the result in Figure 61. It can be seen that according to this figure for many buildings the net internal area should probably be recalculated.

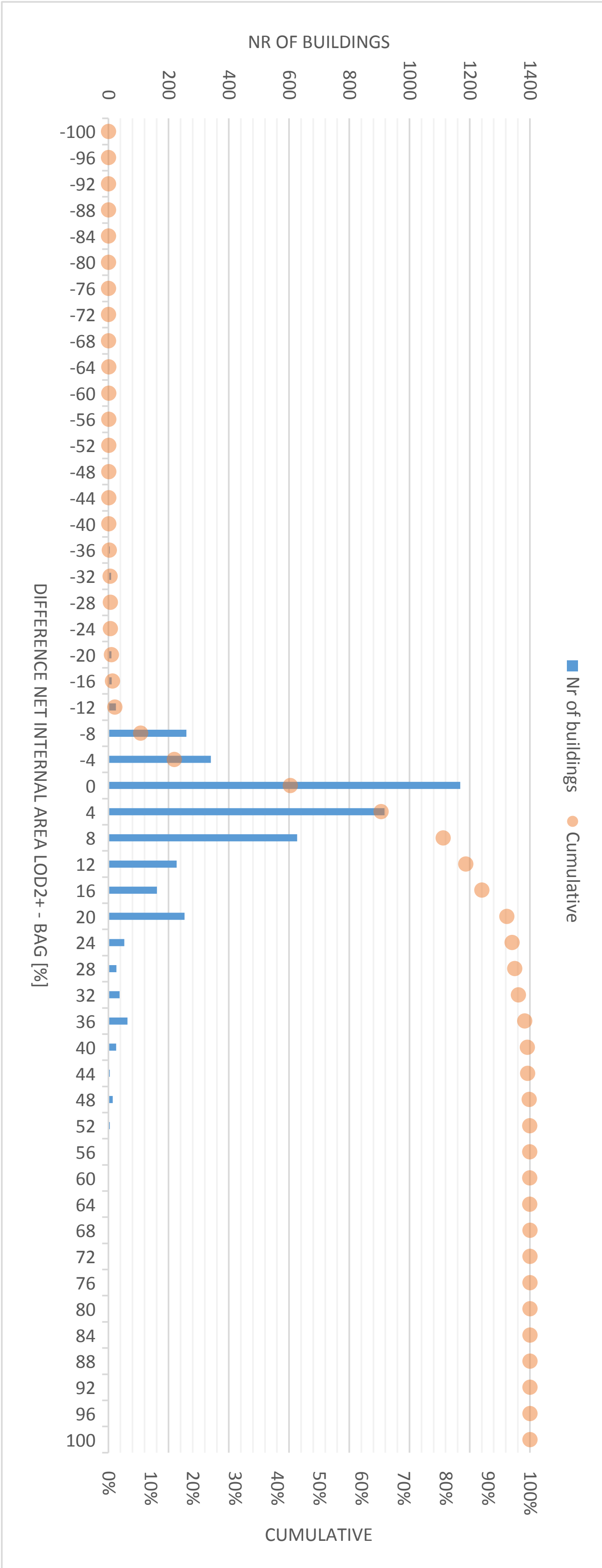


Figure 60: In this histogram the differences are shown between the LoD2+ *FloorSurface* area and the BAG net internal area.

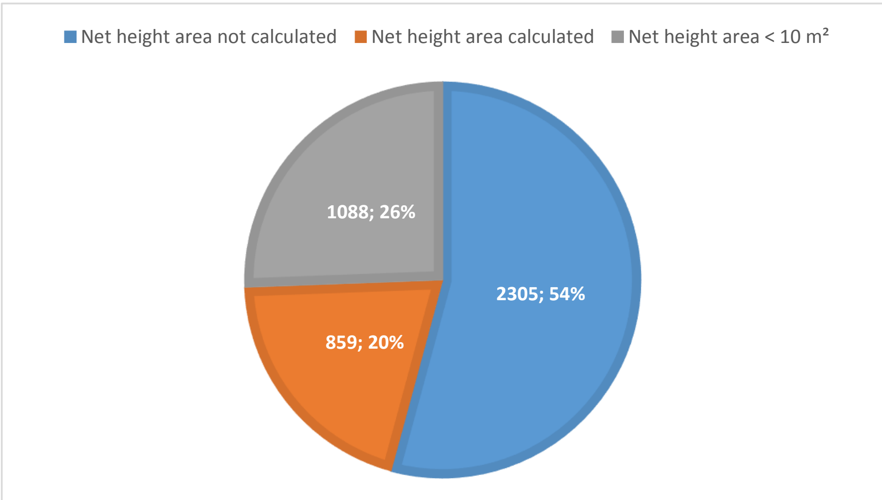


Figure 61: Diagram showing for which part of the buildings the surface area below 1.5 m is probably not taken into account for BAG, for which part it is probably taken into account and the buildings for which this area is smaller than 10 m^2 (also flat roofs).

6.3 COMPARISON WITH WOZ

WOZ is another key register in which the net internal area of residential units is registered (see Section 3.2). Although recently it is decided that for this registration information from BAG should be used, at least for the net internal area of residential units large differences between BAG and WOZ exist.

We can perform the same comparison as for BAG with a WOZ dataset for Hoogvliet-Zuid. The result of the differences between the net internal area calculated from LoD2+ buildings and WOZ is shown in Figure 62. Compared to BAG the differences are smaller.

To improve both registers for each residential unit it can be determined for which of the registers (BAG or WOZ) the difference to the net internal area calculated from the LoD2+ buildings is the smallest. This is shown in Figure 63. From this figure it can be seen that probably the WOZ register is more accurate. This is interesting because in Section 3.2.3 it is discussed that the BAG registration should become the leading register.

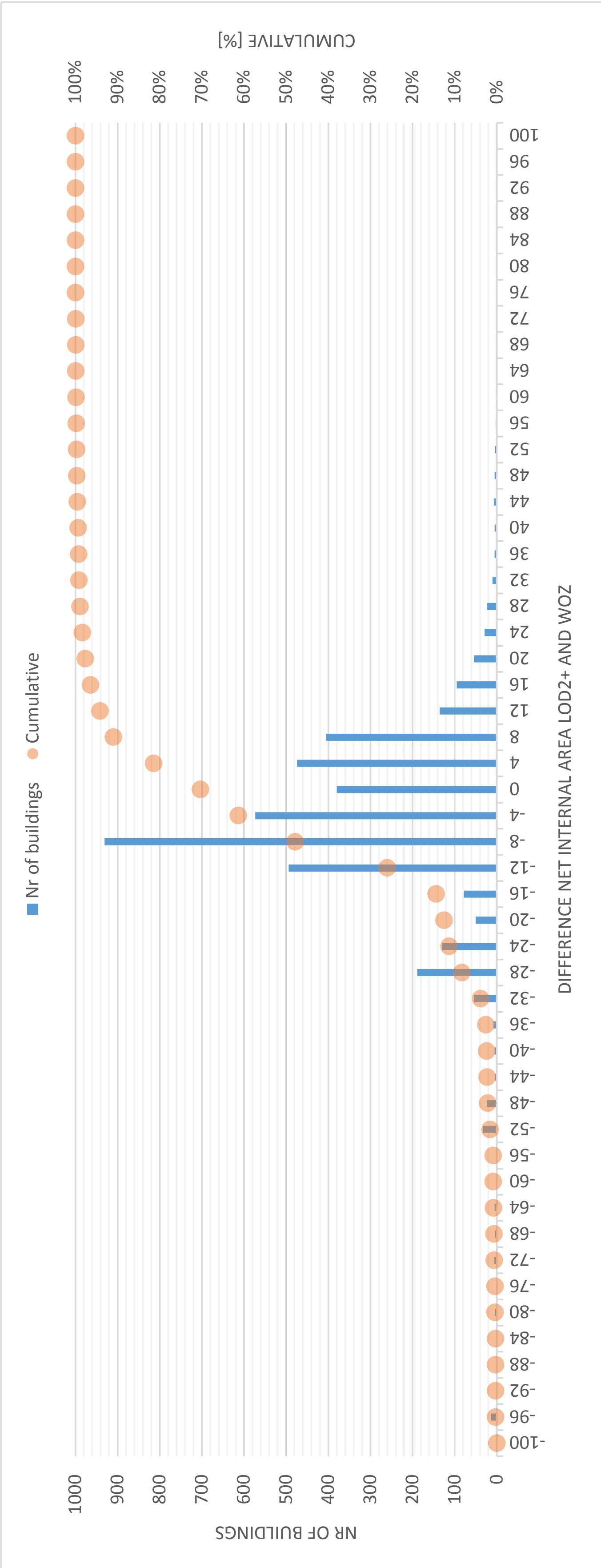


Figure 62: Histogram of the differences between the net internal area calculated for non-stacked buildings from the LoD2+ model and WOZ. When the difference is larger than zero, the net internal area of the LoD2+ buildings is larger than those of WOZ.

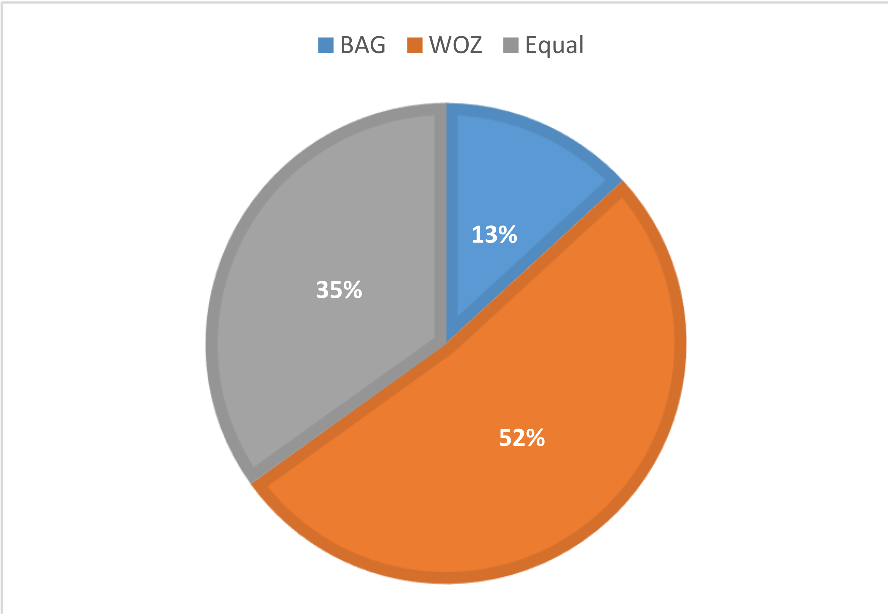


Figure 63: This diagram shows for which register the differences to the net internal area calculated from LoD2+ buildings are smallest.

6.4 VALIDATION OF MODELLED INTERIORS AND BAG

We now know what the differences are between the net internal area calculated from LoD2+ buildings and the values registered in BAG (and WOZ). It is uncertain however what the quality is of the calculated net internal area.

It is impossible to calculate the accuracy of the calculated net internal because there is no dataset available which is 100 % correct. Therefore it is expected that manually inspecting the 5 % largest differences (both negative and positive; see Figure 58) using building blueprints and street imagery results in the determination of the problems. For both the 5 % largest negative differences and the 5 % largest positive differences the problems are determined by randomly inspecting 20 buildings in both sets.

For the largest negative differences the followings problems are found in decreasing order of occurrence:

- The surface area where the net height is below 1.5 m is not subtracted from the net internal area in BAG (66 %).
- Building was still under construction at the moment of laser data acquisition resulting in a erroneous 3D model of the exterior (building height too low) which is propagated to the interiors (19 %).
- The registered net internal area in BAG does not match the building blueprints (10 %).
- The building is incorrectly healed during the healing process, see Figure 64 (5 %).

For the largest positive differences (the differences are smaller in this case) the following problems are found in decreasing order of occurrence:

- Large step wells are present in the building, see Figure 65 (30 %).
- The roof is not correctly modelled in the 3D model, see Figure 66 (25 %).
- The registered net internal area in BAG does not match the building blueprints (15 %).
- Balcony/roof terrace is incorrectly modelled as building exterior, see Figure 67 (15 %).
- Attic floor below slanted roof is modelled too low resulting in a larger surface area (10 %).
- Incorrect number of storeys are modelled (incorrectly registered in BAG) (5 %).

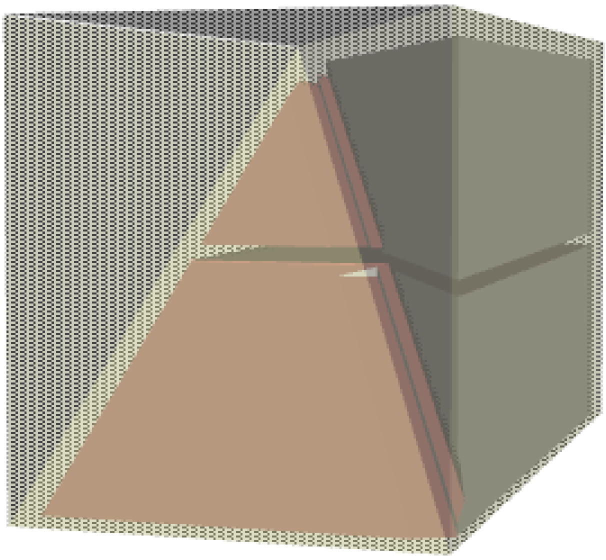


Figure 64: Missing facets for this building were incorrectly healed during the healing process. This results in an incorrect net internal area.

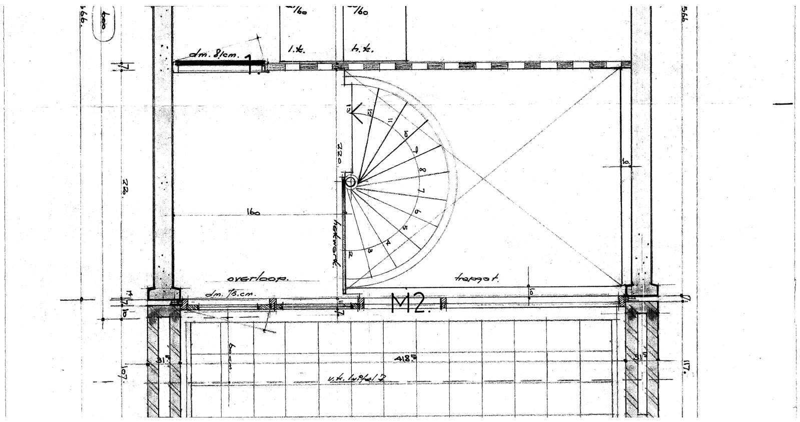
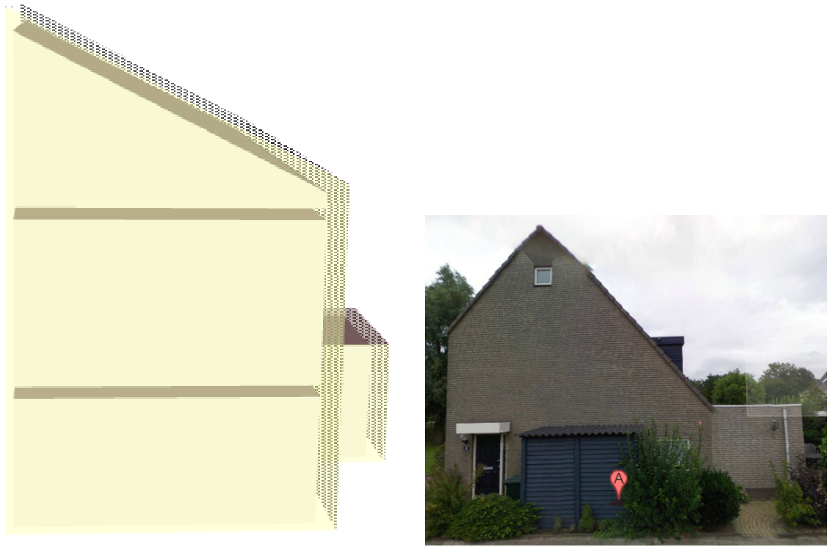


Figure 65: Some buildings do possess large step wells (as can be seen in this building blueprint), and therefore the net internal area calculated from the LoD2+ building may be too large because this is not taken into account.

From this investigation it can be seen that most problems are caused by either problems with the BAG registration or problems with the 3D model, but not with the modelled interiors. The only problem that is determined to be a modelling issue, is the problem where floors are modelled too high or too low resulting in an incorrect determination of the surface area below 1.5 m. This means that at least for the application of net internal area determination there is no need for more detail in the building interiors, but the quality of the 3D model or the BAG data should be improved first.



(a) Incorrect 3D model

(b) The same building on Google Streetview ©

Figure 66: For some buildings the exterior shell is incorrectly modelled due to the fact that the roof structure is modelled incorrectly.

(a) Incorrect 3D model

(b) The same building on Google Streetview ©

Figure 67: For some buildings the exterior shell is incorrectly modelled due to the fact that the balcony is modelled as building instead.

7 CONCLUSION AND RECOMMENDATIONS

This research dealt on the one hand with the extension of CityGML LoDs to feature building interiors. On the other hand it was questioned whether for CityGML LoD2 buildings the interiors according to the defined extension could automatically be modelled from the building exteriors and a limited amount of semantics using a finite set of shape rules. Conclusions on this are given in Section 7.1.

The resulting model that was created during this research is used in the determination of the net internal area of buildings. Several conclusions can be drawn on this and are discussed in Section 7.2.

Since no research has been done before on the automatic modelling of building interiors from building exteriors, this type of research is still in its infancy. To this end in Section 7.3 recommendations are given for future research on this topic.

7.1 DISCUSSION ON LOD2+ GENERATION

This thesis has shown that it is definitively possible to automatically generate building interiors. In this research the generation of LoD2 buildings is done mostly using Boolean set operations. Where CityGML LoD2 models are currently the models with the highest amount of details that can still be (semi-)automatically generated, it is found that LoD2+ is the most detailed interior model that can still be generated in an automated fashion.

The first objective of this research was to define the amount of detail and features in LoD2+. The current CityGML LoDs are inconsistent with respect to interiors and exteriors as only LoD4 has building interiors. When extending the LoDs with interiors either the existing LoDs should be replaced with new definitions which include building interiors or LoD4 needs to be changed such that there can be both exterior LoDs and interior LoDs. Since building interiors are important for many applications and are in fact inseparable from the exterior it is proposed to replace the current LoDs thereby significantly increasing its applicability.

The proposed LoD+ system yields simple storey volumes for LoD1+. For LoD2+ these storey volumes are offset from the exterior shell to incorporate the thickness of for example walls and ceilings and form in fact inner shells of the LoD2 building solid. In LoD2+ also the different premises can be distinguished. Furthermore for these models the faces are semantically enriched. LoD3+ are the same storey shells as in LoD2+ but now also with residential units (e.g. different apartments within a flat can be distinguished). Additionally doors and windows (as they are defined for LoD3 exteriors) can be modelled for the residential units in the interior as well. For CityGML LoD4 the interiors are already described. Geometrically they are not changed for LoD4+, but LoD4+ models should also incorporate storeys, residential units and premises.

Given the LoD+ extension that was designed in Section 4.1 it is investigated how to automatically generate LoD2+ buildings from the available building exteriors and what data and input parameters are needed. Since LoD2+ yield storey shells within a premises, a CityGML LoD2 model is needed where the exteriors represent the premises. Furthermore data is required in which the amount of storeys for a premises is present. Also years of construction and building type are needed because they influence the thickness values of for example walls and floors. This made the BAG registration suitable, because this data is present in the registration at the municipality of Rotterdam.

The generation methodology started from procedural modelling because this method allows the generation of virtual cities with a limited amount of rules and semantic information. It was found that Boolean set operations (e.g. set union and intersection) are very suitable for the generation of LoD2+ interiors. Boolean set operations can simulate machining processes such as milling and cutting, and therefore it makes it an intuitive approach to fabricate interiors from exteriors and furthermore solid geometry can be guaranteed. Using set intersection operations and the combination of Minkowski sum and set difference, cutting and milling operations can be simulated with which the LoD2+ interiors are modelled. Classification of surfaces yielded a simple approach using the normal vector of facets.

The geometric validity of the storeys can be guaranteed relatively easily using Boolean set operations (regularized operations and the handling of disjoint volumes) because of the Nef implementation in CGAL.

Although this research has shown that it is relatively easy to generate LoD2+ interiors from LoD2 buildings there are some disadvantages to the generation method discussed in this thesis. These disadvantages are listed below:

- For an accurate LoD2+ model it must be known at what height each storey of a building starts. From Section 6.4 it is known that sometimes the attic floor is modelled too low. The exact locations of storeys are unknown and therefore it is hard to say how realistic the results from the snapping algorithm to characteristic heights are (as described in Section 5.2.1). However, the evaluation in Section 6.4 show that the heights of storeys is not one of the major issues in the modelling of interiors.
- The Minkowski sum used in the parametrized solid erosion is an effective way of buffering each facet of the storey solid

in the perpendicular direction. Since the Minkowski robot cannot be rotated such that it is aligned with each of the edges of the triangular facet, the in-plane offset is inaccurate resulting in the excessive removal of material (see Figure 52). To overcome this problem another method for buffering each facet should be used. For example the Minkowski sum can be applied to each individual edge of the facet, such that the in-plane offset is also accurate. Moreover the offset of the neighbouring facets (with possible other thickness values) should be taken into account, such that also at wall-ceiling, wall-floor and floor-ceiling connections the interior shells are nicely formed.

- Another problem of using Minkowski sum is that the LoD2+ storey shells contain very small facets. This results in very large (file size) city models, and furthermore the performance when viewing the data is limited. Also using available 3D validator software the interior shells are considered not valid because of degenerate faces where there are in fact no degenerated faces, but a tolerance is used within these tools. When using other methods for the parametrized solid erosion this problem may be eliminated at once, or methods must be used to simplify the resulting interior shells.

- During this research it has proven to be very difficult to determine thickness values from literature or building blueprints and the defined building categories for *walls* and *shared walls*, and impossible for roofs and floors/ceilings. It needs to be investigated whether thickness values can be determined from different sources and also different types of (wall) structures can be taken into account.

- Valid input geometry is very important for the generation of LoD2+ buildings when using Boolean operations as union, intersection or difference. This research has shown that using the current available techniques invalid LoD2 buildings can only be healed to a certain extent. In this research for the test-site Hoogvliet-Zuid in the end 71 % of the input buildings could be enhanced with interiors because of invalid geometry such as missing facets.

- The parameters that are used from the BAG registration of Rotterdam such as the building types and the number of storeys are not part of the BAG core attributes. Its applicability for other municipalities may thus be limited when these properties are not available. Furthermore municipalities need to have a CityGML LoD2 building where the footprints correspond to the BAG geometry.

7.2 DISCUSSION ON NET INTERNAL AREA DETERMINATION

The key application for LoD2+ models in the Netherlands is the net internal area determination of buildings. This research has shown that validation of the registered net internal area in BAG is certainly possible using LoD2+ models mainly because of the 'low' BAG quality.

It can be well argued that LoD2+ is not the correct LoD for the application of net internal area of residential units as in LoD2+ premises are modelled and not residential units which are defined by LoD3+. Since no information is available on the layout of premises with multiple residential units, LoD2+ is the highest which can be automatically generated from the exteriors of premises. The consequence of this drawback is however limited, since approximately 90 % of the buildings in Hoogvliet-Zuid are *non-stacked* buildings.

The largest problem encountered during this research is that for a lot of parameters from the net internal area standard (NEN 2580) it could not be determined to what extent they influence the net internal area of individual buildings. From this definition only the influence of the surface below slanted roofs below 1.5 m could be determined. The remaining parts of a building that should still be subtracted from the *FloorSurface* area to obtain the net internal area could not be determined mainly because of low quality building blue prints.

Even though not all the parts from the complete net internal area definition can be automatically calculated from the LoD2+ model, it is found that the net internal area from most LoD2+ buildings is significantly lower than the net internal area in BAG. This was not expected as the 3D model is based on BAG geometry, and therefore 3D buildings are mostly larger than buildings really are. Furthermore some parameters of the net internal area definition are not subtracted, meaning the net internal area would even become smaller. Largely this is caused by missing information. When more information is available the net internal area can be estimated more accurately, but significantly more manual labour is required to link this information to individual building categories.

The reason that the calculated net internal from the LoD2+ model is significantly lower than BAG is the fact that NEN 2580 was only recently introduced, whereas the registration of net internal area started many years before. The calculation of net internal area of residential units done before the introduction of NEN 2580 was not redone and therefore these calculations do not fit the current standard. The main problem is that for these older registered values the surface area below 1.5 m is not subtracted. The calculation of the net internal area from LoD2+ buildings may well be used to check for which buildings this is the case. The usefulness of LoD2+ for this application is mainly defined by the low quality of the registered net internal area in BAG. Moreover it is found that when the calculated net internal area from the LoD2+ building is lower than the registered net internal area in BAG it is likely that the LoD2+ is more correct, the other way around BAG is mostly more accurate.

Since no data set is available with net internal area values which are known to be correct, the accuracy of the calculated net internal area from LoD2+ buildings is hard to determine. It should thus not be used as a method to calculate the net internal area, but rather as a method to validate the net internal area registered in BAG. From manual inspection of the largest differences between the net internal area calculated from the LoD2+ buildings and BAG it is determined that most of the problems are either in the BAG registration or the 3D city model. Therefore it is needed for the application of net internal area to improve the modelling techniques for the LoD2+ interiors, but rather to improve the 3D model and the BAG registration first.

7.3 RECOMMENDATIONS FOR FUTURE WORK

As discussed before this type of research to automatically produce building interiors from available building exteriors is rather new and therefore continued research is important. The following recommendations can be given for future research on this topic:

- **LoD+ applications** - The net internal area of premises is only one application which requires modelled interiors but not CityGML LoD4. To adopt the CityGML standard to feature interiors in all LoDs the significance of this must be shown by demonstrating its usefulness for other types of applications.
- **LoD3+ generation requirements** - LoD2+ is a good basis for LoD3+ as well. Using Boolean set operations the interior storey solids may be further split, to represent residential units in case of apartments. When it can be determined where these residential units are located in the premises, this is 'easily' done. The data that is needed to go from LoD2+ to LoD3+ is:
  - The boundaries of the residential unit (including possibly the entrance doors of residential units and windows)
  - The thickness offsets of the boundary surfaces of the residential unit

  When LoD3+ buildings can be automatically generated even more applications become available, e.g.:
  - The net internal area of all residential units can be calculated.
  - Door to door navigation (for pedestrians) is possible (for each address)
  - Public safety mapping can become more advanced

- Addresses may three-dimensionally be registered

- **Buffering techniques** - It should be researched how using different buffering techniques (as an alternative to Minkowski sum) the modelled interiors can be better formed at locations where the angle between faces is larger than 180° or where different surface types are connected which should be offset with different thickness values.

- **Thematic boundaries classification** - Nef polyhedra where also attribute information can be stored for each facet would yield significant improvements as the classification can largely be skipped, taking the original features from the LoD2 model. The Nef polyhedra implementation in CGAL is prepared for this, but the implementation is not yet done.

- **Boolean set operations for other features** - Using Boolean operations on Nef polyhedra not only storeys can be modelled, but may well be used for example for rooms or elevator shafts.

- **Geometric properties from textures** - Characteristic heights (to split buildings) which are determined from the geometry alone in this research, may possibly be detected from building textures as well if available (using pattern recognition for example).

- **Thickness influence** - In this research it is not investigated what the influence is of the differing thickness values for different building categories. It may be that this influence is negligible with respect to the quality of the net internal area in BAG. Further research is needed to determine this influence to see if an average value for all buildings can be used.

- **Merging of terraced houses** - In Figure 51 it is seen that improvements can be made by merging complete rows of houses. This should then not only be done for the interiors but also for the exteriors as described by Commandeur (2012).

CITYGML BUILDING MODULE

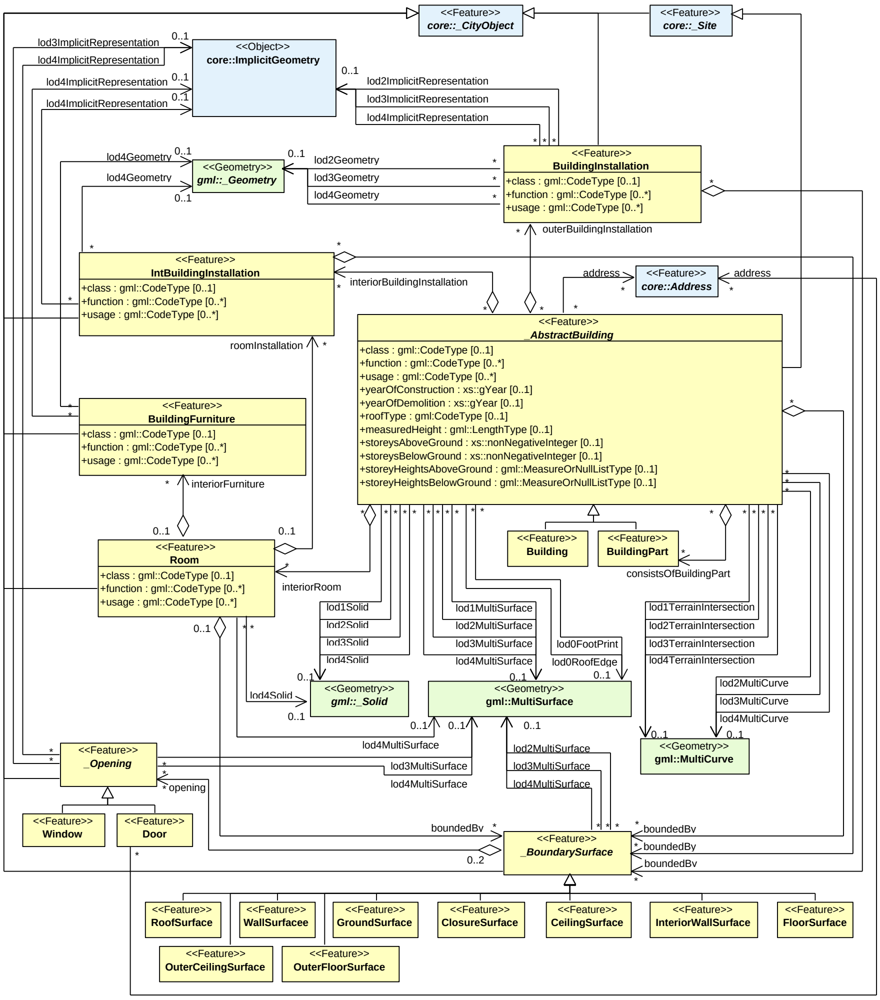


Figure 68: UML diagram of the building module from CityGML.

B

BUILDING BLUEPRINT EXAMPLES

In this research often use is made of building blueprints which are stored in BAG to support the registration. This is mainly done to determine thickness values for walls and shared walls. The assumption was that these values could be easily determined from the available building blueprints as they need to support the net internal area calculation in BAG. However frequent problems were encountered, where either no blueprints are available or blueprints are of bad quality (or contain no dimensions). To give an impression of what these building blueprints look like, three examples are given in Figure 69, Figure 70 and Figure 71.

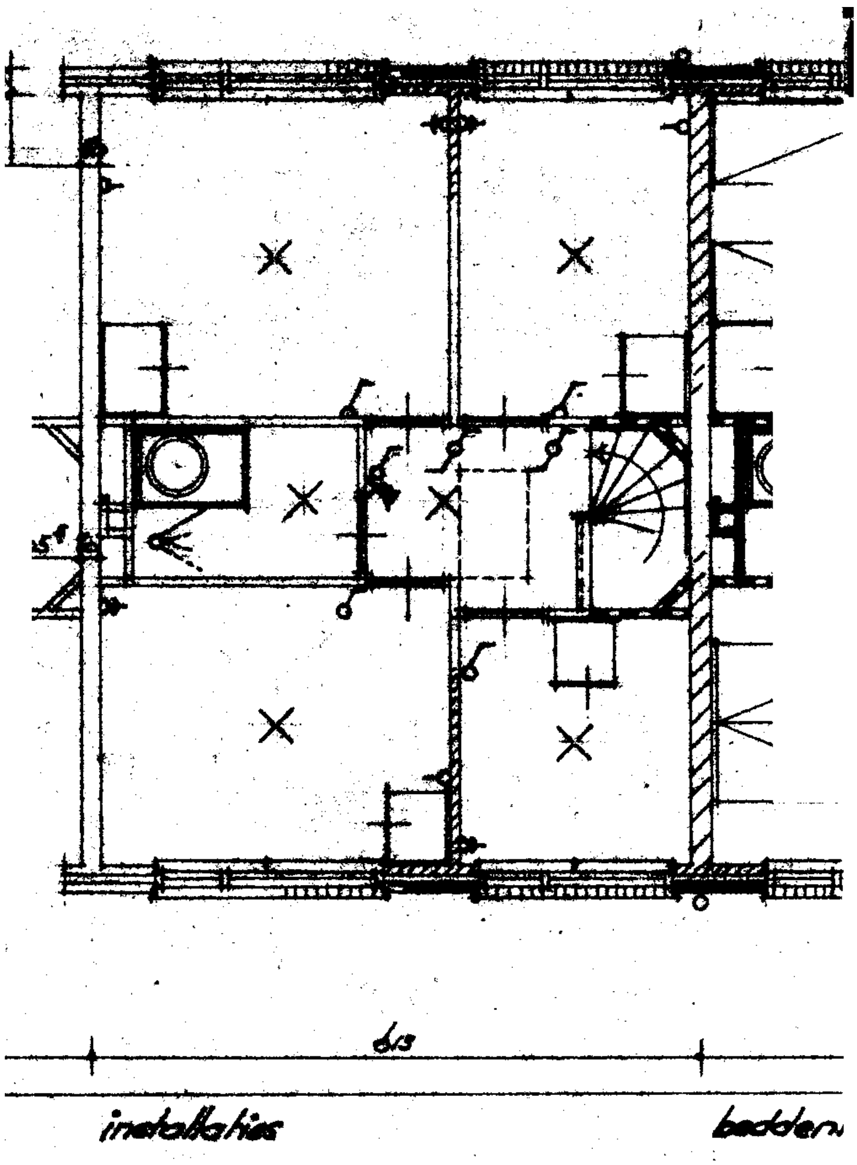


Figure 69: A lot of building blueprints did not contain any dimensions, which makes it impossible to include anything about the thickness of walls.

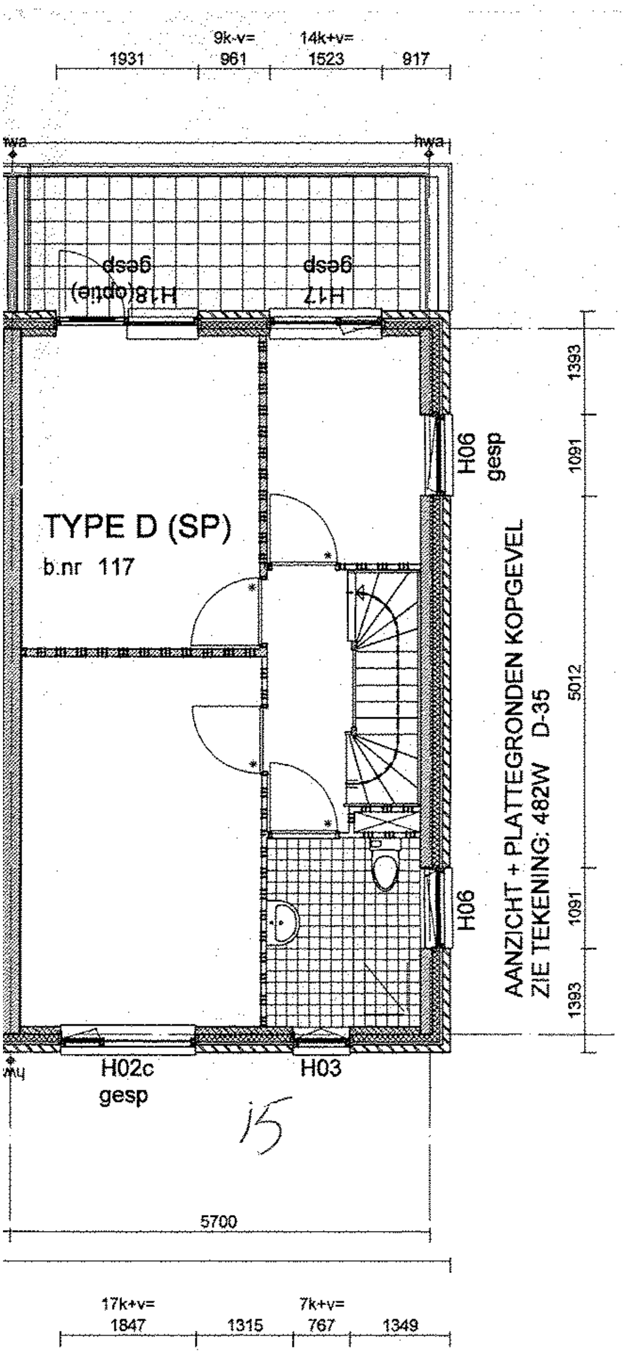


Figure 70: Some building blueprints are of better quality and do contain dimensions, but not for walls.

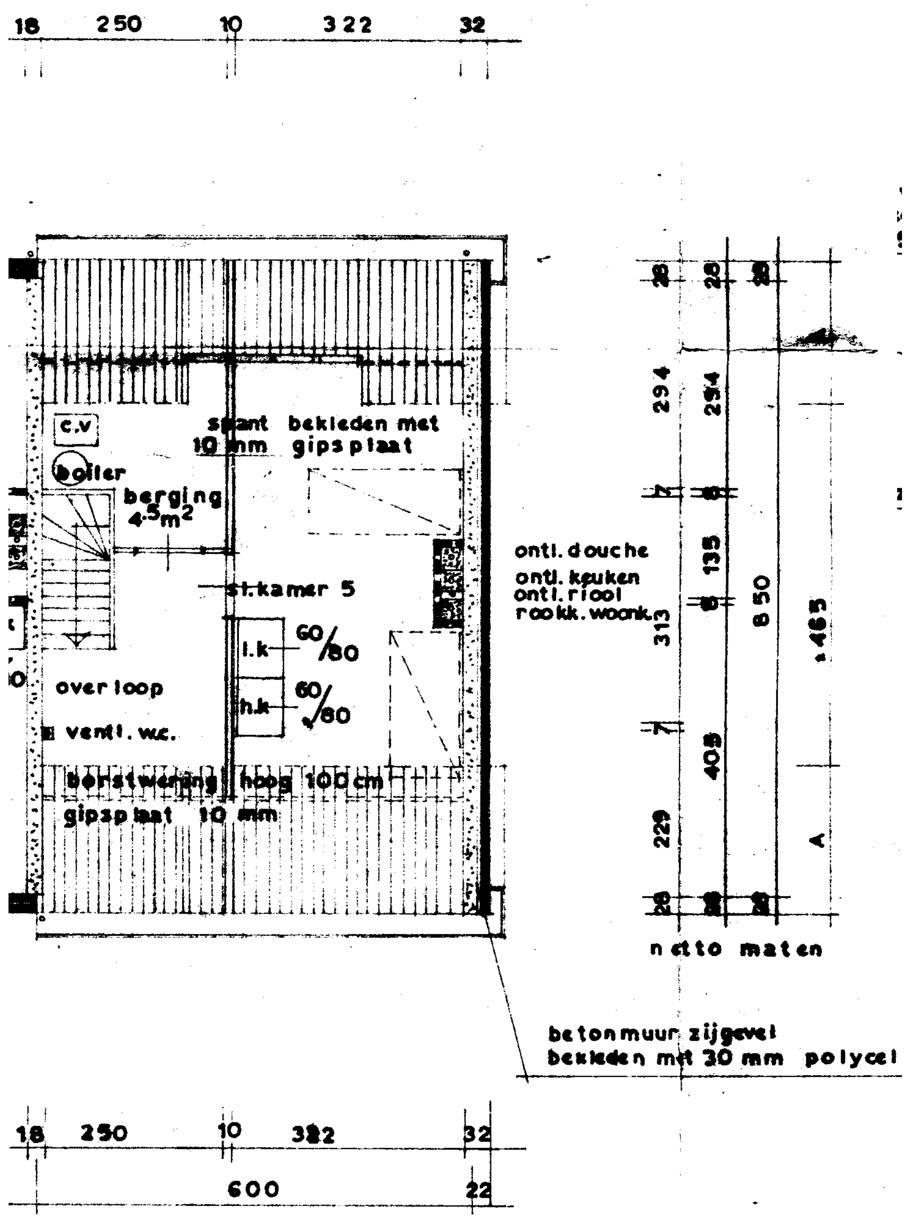


Figure 71: Only for a few buildings, blueprints were available which are actually useful for the purpose of this research.

C IMPLEMENTATION DETAILS

This chapter contains the implementation details on the software designed to generate LoD2+ buildings from a CityGML LoD2 model. First the input files are discussed in Section C.1. The details on how each building is converted is discussed in Section C.2. Lastly the output of the software is discussed in Section C.3.

C.1 INPUT FILES

For the conversion of the CityGML LoD2 model to LoD2+ and the calculation of the net internal area several input files are required. Each of them (along with a short description) are discussed below:

- **CityGML LoD2 file** - One of the input files is naturally the CityGML LoD2 model. It contains valid geometry and is thus healed (e.g. using City Doctor Healing tool by Wagner et al. (2013)). The features *RoofSurface*, *WallSurface* and *GroundSurface* are present in the model for each building. Additionally each building has a "gml:id" which corresponds to the database guid of the BAG premises from the municipality of Rotterdam. The coordinates of each polygon in the model are given by a gml:posList.
- **BAG attributes** - This file is a Comma-separated Values (CSV) file with at least the following attributes with the exact attribute names given on the first line of the file:
  - Street name - 'Openbare ruimte'
  - House number - 'Hsnr'
  - Destination - 'Bestemming'
  - BAG net internal area - 'Oppervlakte'
  - Number of storeys - 'Aantal_bouwlagen_Pand'
  - Lowest storey - 'LAAGSTE_BOUWLAAG'
  - Highest storey - 'HOOGSTE_BOUWLAAG'
  - BAG premises id - 'BAG_id_Pand'
  - Year of construction - 'BOUWJAAR'
  - Guid - 'GUID'

  This file should not only contain the BAG premises but also each individual residential unit, if also premises with multiple residential units should be handled.

- **Shared walls file** - This file is also a CSV file which contains the coordinates of the shared walls present in the dataset. Each lines contains two BAG id's and four ordinates. I.e.: BAG_ID1;BAG_ID2;X_1;Y_1;X_2;Y_2. The file is constructed using the ArcMap model described in Section 5.2.3.

C.2 PROCESSING

The algorithms to generate LoD2+ buildings and to calculate the net internal area for each building are discussed in Chapter 5 and Chapter 6. These algorithms are applied to all buildings individually, to limit the memory usage and the chance of lost data due to software crashes. The processes are executed in the following order:

1. Read LoD2 building from the input CityGML file
2. Create LoD2+ building
3. Calculate net internal area for this building
4. Write the LoD2+ to the CityGML LoD2+ model file for the complete model as well as to the individual CityGML LoD2+ file

C.3 OUTPUT

The output of the developed software is the following:

- **CityGML LoD2+ file** - The main output file is the LoD2+ model created using the algorithms as discussed in Chapter 5. The output file follows the definition of LoD2+ discussed in Section 4.1. It is therefore no valid CityGML, but can still be viewed using some CityGML viewers (e.g. FME data inspector).
- **Statistics file** - This CSV file contains important information on the net internal area. It contains the following parameters:
  - BAG id
  - Net internal area according to BAG
  - Net internal area according to the LoD2+ model
  - Surface area which has a net height of less than 1.5 m
  - Year of construction of the building
  - Building type
  - Number of storeys
- **Individual buildings** - For viewing and validation purposes also each individual building is written in a separate CityGML LoD2+ file. This makes it easier to compare the net internal area of BAG with the net internal area calculated from the LoD2+ buildings.

Cliff Behrens. CityGML change request: Storeys and LODs for BIM interiors, 2012. URL `http://www.opengeospatial.org/standards/cr`. (Cited on page 7.)

Jan Blaauboer, Joris Goos, Hugo Ledoux, Friso Penninga, Marcel Reuvers, Jantien Stoter, and George Vosselman. 3D Pilot Eindrapport werkgroep Technische specificaties voor de opbouw van 3D IMGeo-CityGML. (november), 2012. (Cited on page 2.)

Boris Briels. Onderzoek afmetingen leidingkokers en ruimtelijke consequenties. pages 1–16, 2004. (Cited on page 85.)

Angela Budroni and Jan Böhm. Toward automatic reconstruction of interiors from laser data. Citeseer, Institute for Photogrammetry, Terrestrial Positioning Systems and Computer Vision Group, 2009. (Cited on page 9.)

CGAL. OFF_to_nef_3, 2013. URL `http://www.cgal.org/Manual/latest/doc_html/cgal_manual/Nef_3_ref/Function_OFF_to_nef_3.html`. (Cited on page 57.)

CityGML. CityGML-ADEs, 2012. URL `http://www.citygmlwiki.org/index.php/CityGML-ADEs`. (Cited on page 17.)

Tom Commandeur. Master thesis: Footprint decomposition combined with point cloud segmentation for producing valid 3D models, 2012. (Cited on page 106.)

Oliver Deussen, Pat Hanrahan, Bernd Lintermann, Radomír Měch, Pharr Matt, and Prusinkiewicz Prusinkiewicz. Visual models of plants interacting with their environment. In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pages 397–410. ACM, 1996. (Cited on page 10.)

Marthe Fuld. Key Registers for Addresses and Buildings. September 2007. (Cited on page 18.)

Marthe Fuld and Marcel Rietdijk. Met zijn allen recht op het doel af. De strategie voor de implementatie van de basisregistraties voor adressen en gebouwen., 2004. (Cited on page 18.)

Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. Proceedings of the 24th annual conference on Computer graphics and interactive techniques, pages 209–216, 1997. (Cited on page 79.)

Gemeente Rotterdam. Programma van eisen voor de opbouw en levering van een 3D-model van de gemeente Rotterdam uit laseraltimetrie en/of stereofotografie. 2009. (Cited on page 24.)

Marcus Goetz and Alexander Zipf. Extending OpenStreetMap to Indoor Environments: Bringing Volunteered Geographic Information to the Next Level. Proceedings of the UDMS: Urban and Regional Data Management: 2011, pages 47–58, 2011. (Cited on page 9.)

Ben Gorte. Advanced remote sensing: image processing, 2006. (Cited on page 35.)

Miguel Granados, Peter Hachenberger, Susan Hert, Lutz Kettner, Kurt Mehlhorn, and Michael Seel. Boolean operations on 3D selective Nef complexes: Data structure, algorithms, and implementation. Algorithms-ESA 2003, pages 654–666, 2003. (Cited on pages 32, 33, and 78.)

Gerhard Gröger and Lutz Plümer. CityGML – Interoperable semantic 3D city models. ISPRS Journal of Photogrammetry and Remote Sensing, 71:12–33, July 2012. (Cited on page 2.)

Gerhard Gröger, Thomas H. Kolbe, and Angela Czerwinski. OpenGIS City Geography Markup Language (CityGML), Implementation Specification Version 0.3. 2006. (Cited on page 1.)

Gerhard Gröger, Thomas H. Kolbe, Claus Nagel, and Karl-Heinz Häfele. OGC City Geography Markup Language (CityGML) Encoding Standard Version 2.0. 2012. (Cited on page 44.)

Peter Hachenberger. Exact Minkowksi sums of polyhedra and exact and efficient decomposition of polyhedra in convex pieces. Algorithms–ESA 2007, 2007. (Cited on pages 35, 36, 61, 62, and 78.)

Peter Hachenberger. 3D Minkowski Sum of Polyhedra, 2013. URL `http://www.cgal.org/Manual/latest/doc_html/cgal_manual/Minkowski_sum_3/Chapter_main.html`. (Cited on page 36.)

Peter Hachenberger and Lutz Kettner. Boolean Operations on 3D Selective Nef Complexes: Optimized Implementation and Experiments. (Cited on page 33.)

Benjamin Hagedorn, Matthias Trapp, Tassilo Glander, and Jürgen Döllner. Towards an Indoor Level-of-Detail Model for Route Visualization. IEEE, 2009. (Cited on pages 7, 8, and 11.)

Henk Heijmans and Christian Ronse. The algebraic basis of mathematical morphology I. Dilations and erosions. Computer Vision, Graphics, and Image Processing, 50(3):245–295, June 1990. (Cited on page 48.)

Sebastien Horna, Guillaume Damiand, Daniel Meneveaux, and Yves Bertrand. Building 3D indoor scenes topology from 2D architectural plans. 2007. (Cited on page 9.)

ISSO. Bijlage 1A Tips voor het herkennen van bouwkundige constructies. In ISSO-PUBLICATIE 82.1 HANDLEIDING ENERGIEPRESTATIE ADVIES WONINGEN. Rotterdam, 2011. (Cited on pages 28 and 70.)

Paul Janssen, Jan Kooijman, and Hugo Ledoux. Eindrapportage Activiteit 2 : 3D Geometrie validator . Inhoudsopgave. 3D Pilot Fase II, 2012. (Cited on pages 35, 56, and 78.)

Andreas Jochem, Bernhard Höfle, Martin Rutzinger, and Norbert Pfeifer. Automatic roof plane detection and analysis in airborne lidar point clouds for solar potential assessment. Sensors, 2009. (Cited on page 2.)

Matthew Johnston and Avideh Zakhor. Estimating building floor-plans from exterior using laser scanners. Proceedings of Three-Dimensional Image Capture and Applications, 2008. (Cited on page 10.)

Pierre P. Kastendeuch and George Najjar. Simulation and validation of radiative transfers in urbanised areas. Solar Energy, 2009. (Cited on page 3.)

Serkan Kemec, Sisi Zlatanova, and Sebnem Duzgun. A new LoD definition hierarchy for 3D city models used for natural disaster risk communication tool. 2012. (Cited on pages 8, 9, 11, and 42.)

Lutz Kettner. Using Generic Programming for Designing a Data Structure for Polyhedral Surfaces. 1999. (Cited on page 57.)

Thomas H. Kolbe. Representing and exchanging 3D city models with CityGML. 3D geo-information sciences, 2009. (Cited on pages 14 and 16.)

Jos Lichtenberg. Slimbouwen. Uitgeverij Æneas BV, 2005. (Cited on page 28.)

Jess Martin. Algorithmic beauty of buildings methods for procedural building generation. page 4, 2005. (Cited on page 10.)

Ministerie van VROM. Wetgeving basisregistraties adressen en gebouwen. 2009. (Cited on page 17.)

Ministerie van VROM. Kwaliteit van de basisregistraties adressen en gebouwen. In Verdiepingsinformatie kwaliteit en de meting daarvan tijdens de toelatingsaudit, 2010. (Cited on page 22.)

Walter Nef. Beitrage zur Theorie der Polyeder, volume 1. Lang, 1978. (Cited on page 31.)

Brian Okorn, Xuehan Xiong, Burcu Akinci, and Daniel Huber. Toward automated modeling of floor plans. In Proceedings of the Symposium on 3D Data Processing, Visualization and Transmission, volume 2, 2010. (Cited on page 9.)

Yoav I.H. Parish and Pascal Müller. Procedural modeling of cities. Proceedings of the 28th annual conference on Computer graphics and interactive techniques, 2001. (Cited on page 10.)

Valentin V. Petrov. Limit theorems of probability theory. 1995. (Cited on page 52.)

Clemens Portele. OpenGIS® Geography Markup Language (GML) Encoding Standard, 2007. (Cited on page 13.)

Ari Requicha and Herbert B. Voelcker. Boolean operations in solid modeling: Boundary evaluation and merging algorithms. Proceedings of the IEEE, 1985. (Cited on page 30.)

Marcel Rietdijk. Catalogus basisregistraties adressen en gebouwen, 2009. (Cited on pages 18 and 21.)

Jarek R. Rossignac and Paul Borrel. Multi-resolution 3D approximations for rendering complex scenes. 1993. (Cited on page 1.)

George Stiny. Introduction to shape and shape grammars. Environment and planning B, 7(3):343–351, 1980. (Cited on pages 29 and 30.)

Jantien Stoter, Henk de Kluijver, and Vinaykumar Kurakula. 3D noise mapping in urban areas. International Journal of Geographical Information Science, 2008. (Cited on page 2.)

Jantien Stoter, Gerbrand Vestjens, George Vosselman, Joris Goos, Edward Verbree, Rick Klooster, and Marcel Reuvers. CityGML wordt standaard voor Nederlandse 3D geodata. GeoInside, 19(2):26–29, 2011. (Cited on page 1.)

C.C.F. Thijssen and C.J. Meijer. Bouwconstructieve analyse van naoorlogse meergezinshuizen in de non-profit huursector 1946-1965. 1988. (Cited on page 29.)

Robert Tilove and Aristides Requicha. Closure of Boolean operations on geometric entities. Computer-Aided Design, 12(5):219–220, September 1980. (Cited on page 31.)

George Vosselman and Sander Dijkman. 3D building model reconstruction from point clouds and ground plans. International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences, 34(3/W4):37–44, 2001. (Cited on page 1.)

Waarderingskamer. Handreiking voor het opstellen van een gemeentelijk stappelplan voor de koppeling BAG en WOZ. Den Haag, 2010. (Cited on page 23.)

Waarderingskamer. Meetinstructie Gebruiksoppervlakte woningen, 2013. (Cited on page 21.)

Detlev Wagner, Mark Wewetzer, Jürgen Bogdahn, Nazmul Alam, Margitta Pries, and Volker Coors. Geometric-semantical consistency validation of CityGML models. Progress and New Trends in 3D Geoinformation Sciences, 2013. (Cited on pages 56 and 113.)

Wen Ping Wang. Solid modeling for optimizing metal removal of three-dimensional NC end milling. Journal of Manufacturing Systems, 7(1):57–65, January 1988. (Cited on page 48.)

Richard Watkins, John Palmer, and Maria Kolokotroni. Increased temperature and intensification of the urban heat island: implications for human comfort and urban design. Built Environment, 2007. (Cited on page 3.)

Benjamin Watson, Pascal Müller, Peter Wonka, Chris Sexton, Oleg V. Veryovka, and Andy Fuller. Procedural urban modeling in practice. Computer Graphics and Applications, IEEE, 28(3):18–26, 2008. (Cited on pages 10 and 30.)

Peter Wonka, Michael Wimmer, François Sillion, and William Ribarsky. Instant architecture. Number 3. ACM, 2003. (Cited on page 10.)

Zhiyang Yao and Satyandra K. Gupta. Cutter path generation for 2.5D milling by combining multiple different cutter path patterns. International Journal of Production Research, 42(11):2141–2161, June 2004. (Cited on page 48.)

Xuetao Yin, Peter Wonka, and Anshuman Razdan. Generating 3d building models from architectural drawings: A survey. Computer Graphics and Applications, IEEE, 29(1):20–30, 2009. (Cited on pages 8 and 11.)

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "The Elements of Typographic Style".