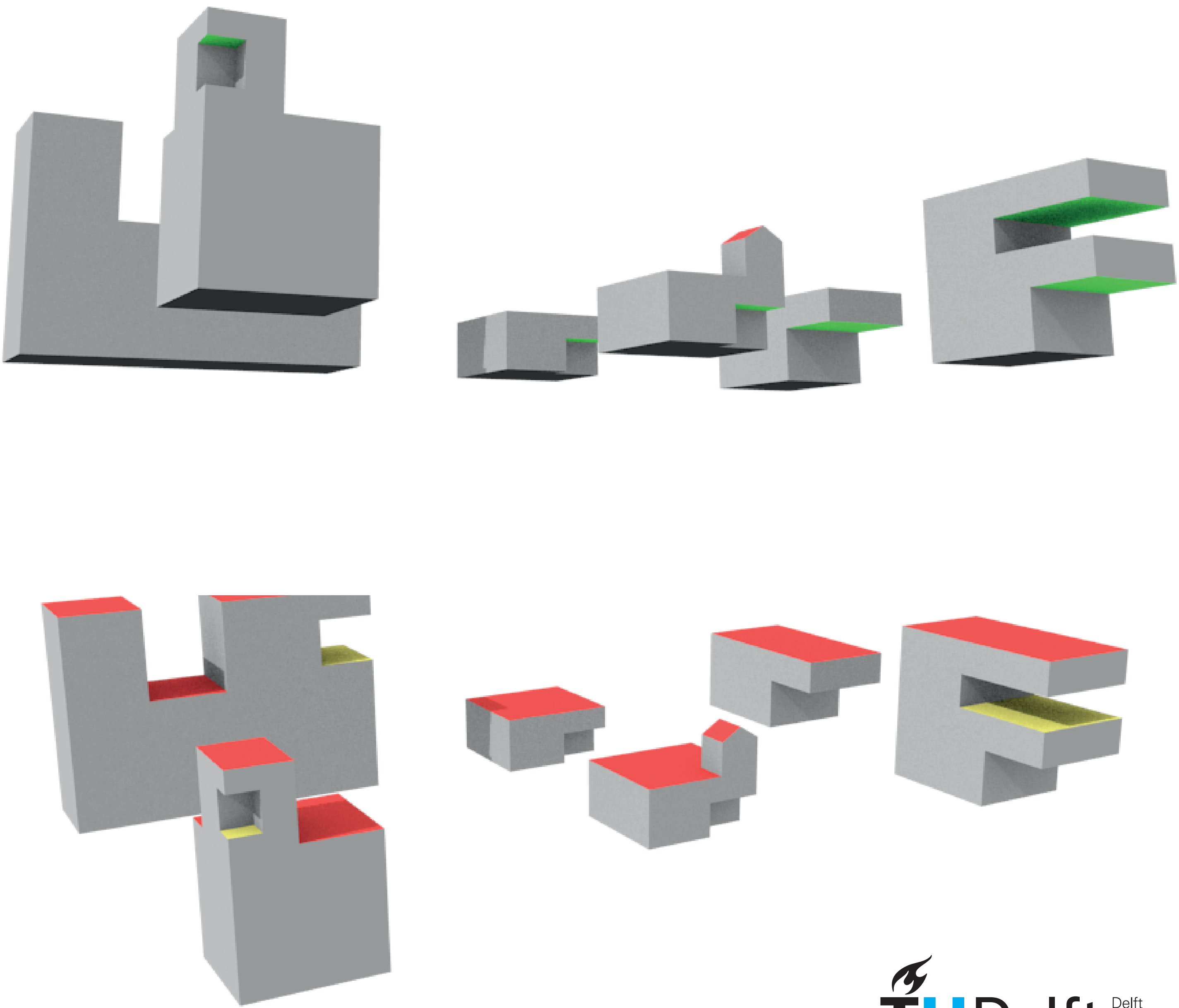
Automatic thematic and semantic classification of 3D city models

Merwin Rook

Delft University of Technology



TU Delft Delft University of Technology

Geomatics for the Built Environment

AUTOMATIC THEMATIC AND SEMANTIC CLASSIFICATION OF 3D CITY MODELS

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

Master of Science in Geomatics for the Built Environment

by

Merwin Rook

November 2016

The work presented in this thesis was carried out in the:

3D geoinformation group
Department of Urbanism
Faculty of Architecture & the Built Environment
Delft University of Technology

Supervisors: Filip Biljecki
Abdoulaye Diakité
Co-reader: Wilko Quak

ABSTRACT

Developments in remote sensing created new possibilities to capture the human environment on a large scale. This data is used in the generation of large scale Three dimensional (3D) city models, which has led to a higher demand for 3D city models in a wide range of fields.

The lack of semantic information in many 3D city models is a considerable limiting factor in their use, as a lot of applications rely on semantic information. This research is a first step in creating an automatic workflow, that semantically labels a plain 3D city model, with level of detail 1 or 2, represented by a triangulated polygon mesh, with semantic and thematic information as defined in the CityGML standard.

The first step in this labelling process is the reconstruction of the building entities and (parts of) the terrain: the thematic features. Next, two methods are proposed to semantically label the surfaces in the previously defined building entities.

The first implemented method is a best practice of methods that are tested, which aim at labelling the 3D city model with the classes WallSurface, RoofSurface, and GroundSurface. The second method is an explorative approach and a proposal that additionally recognises the classes OuterCeilingSurface and OuterFloorSurface. In this approach, an proposal is made that extends the current class definitions of a RoofSurface and a WallSurface in CityGML. This research shows that by extending these defintions, a CityGML file with Level of Detail 1 and 2 can be semantically labelled automatically.

The results show that a high semantic classification accuracy is possible. The accuracy depends on a number of factors, floating point precision errors is the biggest limitation factor in the thematic and the semantic labelling.

ACRONYMS

GIS Geographical Information System 3
3D Three dimensional iii
2D Two dimensional 3
SDI Spatial data infrastructure 13
LoD Level of Detail x
SVM Support Vector Machine 17
MVS Multi-View System 17
C-Map Combinatorial Map 17
LiDAR Light detection and ranging 11
CRF Conditional random field 20

ACKNOWLEDGEMENTS

I would like to thank my supervisors Filip Biljecki and Abdoulaye Diakité for their time, their effort and their willingness to help. Above all, I would like to thank Filip for his patience and Abdou for his endless optimism during the process of writing this thesis.

Thereby, I would like to thank my parents for their endless support during my working and study career in the past 10 years.

CONTENTS

1 INTRODUCTION

A Geographical Information System (GIS) operates with the largest scope of spatial objects: the spatial and the semantic, their relationships and the means to analyse these different components (Zlatanova, 2000). In a GIS, the information about these spatial objects is maintained and analysed (Stoter and Zlatanova, 2003). Geographical information however, is still largely presented in a Two dimensional (2D) field. This 2D geo-information is available in large amounts, in different types, at different scales while covering many application domains (Stoter and Van Oosterom, 2002).

In the last years, the need for 3D information is rapidly increasing, as 2D GIS has shown its limitations in some applications, such as: noise prediction, water management and flood modelling, air pollution modelling and geology. Other disciplines that can benefit from 3D geo-information are: 3D urban planning and real estate market analysis, environmental monitoring, telecommunications, public rescue operations or landscape planning (Stoter and Zlatanova, 2003; Gröger and Plümer, 2012; Vosselman et al., 2001).

Laser scanning and photogrammetry, created new possibilities to capture and model the human environment in three dimensions (Verma et al., 2006). This 3D spatial modelling is the key and the basis for 3D GIS (Yanbing et al., 2007). These new sensing technologies however, only capture the geometry.

The integration of this data in a GIS requires additional semantic information. Semantic information, in the scope of this research, is information about what a surface represents in the real world. For example, a surface with attached the information that it represents a wall, a terrain, or a roof surface. Next to semantic information of single surfaces, single buildings should be individually identifiable. Creating possibilities to select and query single buildings, which are used to evaluate the results in spatial analysis.

In other words, 3D city models without this information do not support most 3D GIS applications, because it is not possible to identify the surfaces of interest, e.g. roof surfaces to estimate the solar irradiation on them, or walls to calculate the total facade area. Therefore, their use for GIS purposes is hindered by the lack of thematic and semantic information (Brodeur, 2012).

Next to geometrical models created from sensor data, other sources of information are getting integrated into GIS too. For example, CAD models are currently integrated into GIS and vica versa (Mommers, 2015; de Laat and van Berlo, 2011), the large amount of available CAD data creates major possibilities for integration in a GIS to create urban scenes (Stoter and Zlatanova, 2003). Other sources of 3D city models are 3D modelling software, e.g. Esri's CityEngine. The integration of all this different data in a GIS however, needs more advanced integration of semantics. Therefore these semantics must first be added to this data in order to make them usable in a GIS (de Laat and van Berlo, 2011). In other words, the usability of 3D city models from all different sources can be highly increased if semantic information is added to the geometry. While models without semantic information may still be valuable for visualisation, their full potential in a GIS is limited by the lack of semantic information (Brodeur, 2012). Therefore, the concept of semantic enrichment, i.e. adding of semantic information to the geometry, is necessary to create models that meet the requirements of relevant applications (Henn et al., 2012).

The process of automatically adding semantic information to 3D city models is scarcely researched. Therefore, this research aims to develop a method to automatically enrich 3D city models with semantic and thematic information, as defined in the CityGML standard. CityGML is developed with in mind a broad spectrum of use cases. Thereby, it can be easily extended for a specific nationwide, or application specific demand. Making the use of the thematic and semantic classes as defined in CityGML a logical decision.

Although some research in creating semantically rich 3D models has been carried out (Verdie et al., 2015; Xiong et al., 2013), research in enriching existing 3D city models is almost non-existing and holds many scientific and software opportunities (Biljecki and Arroyo Ohori, 2015). This research aims at exploiting these opportunities and bridging the gap in the lack of research that currently exists in enriching 3D city models. As adding semantic information to these models is usually done manually, frequently on a small scale or for an individual building, and is therefore labour intensive and costly. Thereby, the thematic labelling is currently mostly neglected and left aside. Also, the LoD of the model must be detected, the LoD determines the presence of the semantic classes in the city model.

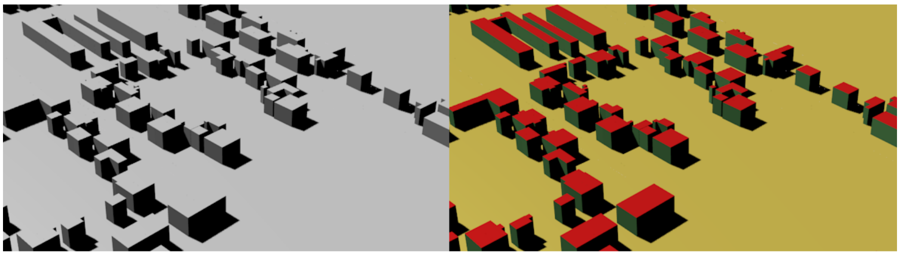


Figure 1.1: Left: an thematically/semantically unlabelled 3D city model. Right: the same model after the labelling process, where the semantic classes; roof, wall or terrain, are represented by a different colour. Source: Rotterdam municipality (2015)

1.1 RESEARCH QUESTIONS

This research aims at semantically enriching 3D city models in an automated way, in order to increase the usability of these models for GIS analysis. Here fore, thematic and semantic information is added to a dataset by only processing and analysing the geometry of the 3D city model. More precise, this research aims at creating a work flow that takes a polygon mesh, or a soup of polygons, as input, and generates a model whereby the different spatial features: walls, roofs, ground, terrain, and two more additional classes are differentiated and recognized. Figure 1.1 is an example of a unclassified and a classified city model of Rotterdam, what we seek to achieve automatically. Next, these spatial features are merged to create single buildings, following the specifications of the CityGML standard. This is translated in the following research question:

"How to automatically enrich a LoD 1 or 2 3D city model with thematic

and semantic information as defined in CityGML, by only utilising the models geometry?"

To bring the answer to this problem, it has been subdivided in several sub questions:

- How is the LoD of the 3D city model detectable?
- What semantic and thematic classes can be distinguished by only using geometric properties, dependent on the LoD?
- How can these geometric properties be used in the classification of the 3D city model?
- Can methods established in remote sensing, e.g. classification of point clouds be used?
- How accurate is the classification process?

1.2 RESEARCH SCOPE AND PURPOSE STATEMENT

The recognition of WallSurfaces, RoofSurfaces an GroundSurfaces in a soup of polygons has been successfully researched (Diakité et al., 2014), in order to create a semantically rich 3D city or building model. However, in order to create a semantically and thematically rich CityGML dataset with LoD 1 and 2, existing research still holds three limitations:

First, in existing research, not all the in CityGML defined semantic classes for a LoD 2 model are detected. The semantic classes that additionally need to be detected are OuterCeilingSurface and OuterRoofSurface (Figure 1.3). All the semantic classes, except 'relief' which represents the terrain, are visualised in Figure 1.3. This selection of semantic classes is based on the CityGML class taxonomy, as explained in the chapter related work and literature research, section 2.3.2. Thereby, these semantic classes are expected to be detectable in the 3D city models, by using the methods proposed later in this research. The classes BuildingInstallation and ClosureSurface will not be detected, because the class BuildingInstallation is not mandatory in the CityGML standard. The class ClosureSurface cannot be automatically detected, because it is not bound to any specifications and only functions to close a feature its space, so the volume of the feature can be computed. Therefore, the following semantic classes will be detected.

- Relief: Terrain surface, containing all ground surfaces that are not part of a building.
- RoofSurface
- WallSurface
- GroundSurface, which is the terrain surface that is part of a building.
- OuterCeilingSurface
- OuterRoofSurface

Second, in CityGML, single building entities, defined as AbstractBuilding in CityGML, must be stored individually. Facilitating the possibility to query and select these individual buildings. This demands the recognition and re composition of these individual buildings, aggregating the components of a building into single entities, specified by a semantic class. So, besides the labelling of the semantic classes, this research aims at retrieving the thematic building entities of the 3D city model. This means that this research aims at reconstructing, recreating and aggregating the single buildings. Thereby, polygons which are not part of a building will be thematically classified as terrain.

Third, current research does not focus on the detection of the LoD of the 3D city model. This limits the possibility to create the correct output for the algorithm, because the presence of the semantic classes in the dataset fully depends on the LoD of the model.

This research aims to find a solution for these three limitations. Therefore, the goal is to develop an automatic work flow that takes a 3D virtual city model as input, and turns out the same geometry which is semantically enriched in a CityGML format, dependent on the LoD.

1.2.1 Level of Detail

The semantic, spatial and geometrical properties of the building model are structured in five different LoDs (Kolbe et al., 2005; Gröger and Plümer, 2012). The concept of LoD describes how close the virtual representation reflects the actual real-world scene, and includes the spatio-semantic coherence (Stadler and Kolbe, 2007). Five levels of LoD are defined in the CityGML standard and will be used in this research. In Figure 1.2, the different levels of LoD are visualized. These five levels offer a clear and straightforward distinction and are used in related research (Boeters et al., 2015; Biljecki et al., 2014). CityGML specifies the following LoDs

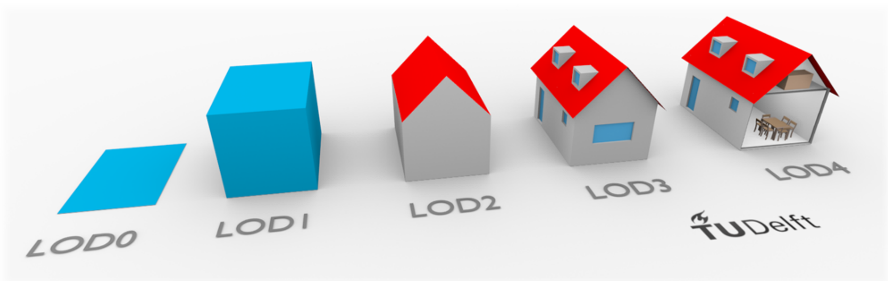


Figure 1.2: Different LoDs (Biljecki et al., 2016)

- **LoD 0:** 2.5D building footprints and/or roof edge polygons (Boeters et al., 2015; Biljecki et al., 2014). A possible application for LoD 0 is density or distance calculation for fire precautions or land tenure visualisation (Löwner et al., 2013).

- **LoD 1:** Extruded footprints (prismatic models) (Boeters et al., 2015; Biljecki et al., 2014), represented as a block model. In other words, a vertical extruded solid, without any semantic structuring. Possible applications for these models are noise mapping approaches or real volume calculations in flood modelling applications (Löwner et al., 2013).

- **LoD 2**: Simple models with differentiated roof structures (Boeters et al., 2015; Biljecki et al., 2014). The outer surfaces can be differentiated by the class BoundarySurface. These surfaces can be individually labeled with semantics like WallSurface, RoofSurface, GroundSurface, etc. Chimneys, Dormers and Balconies may be associated to a building in LoD 2 using the class BuildingInstallation (Löwner et al., 2013). A possible use case for these models is the calculation of the potential for solar energy (Biljecki et al., 2015a).

- **LoD 3**: Detailed architectural models with openings such as windows and doors (Boeters et al., 2015; Biljecki et al., 2014). In LoD 3, the building is represented by a geometrically detailed outer shell. Compared to LoD 2, the class Opening is added, which consists out of windows and doors (Löwner et al., 2013). LoD 3 models are used in, for example, heat transmission analysis (Biljecki et al., 2015a).

- **LoD 4**: These models also contain detailed indoor geometries of buildings (Boeters et al., 2015; Biljecki et al., 2014). Whereby interior structures are represented as Room, which may enhanced by the attributes class, function and usage (Löwner et al., 2013).

1.3 SEMANTIC CLASSES PER LOD

The concept of LoD plays a central role in this research, as the LoD is the decisive matter in the presence of semantic classes in a 3D urban scene. In other words, different semantic classes will be relevant in the classification process, depending on the LoD of the 3D city model that is processed. The following paragraph will therefore define what semantic properties will be added, depending on the LoD.

LoD 0: Models with LoD 0 will not be considered in this research.

LoD 1: In models with LoD 1, the labelling process will aim at adding thematic information to Buildings only. This means clustering the different Building components together, forming the thematic CityGML class AbstractBuilding.

LoD 2: In models with LoD 2, the labelling process will aim at adding thematic information to Buildings and, if present, the terrain. This means clustering the different Building components together, forming a the CityGML class Building. Thereby, adding semantic information to the semantic components of the Building: Roofsurface, WallSurface, GroundSurface, OuterFloorSurface and OuterCeilingSurface.

LoD 3: Models with LoD 3 will not be classified in this research, because of the higher complexity of the model.

LoD 4: Models with LoD 4 will be left untouched in this research, as labelling indoor geometries is outside the scope of this research.

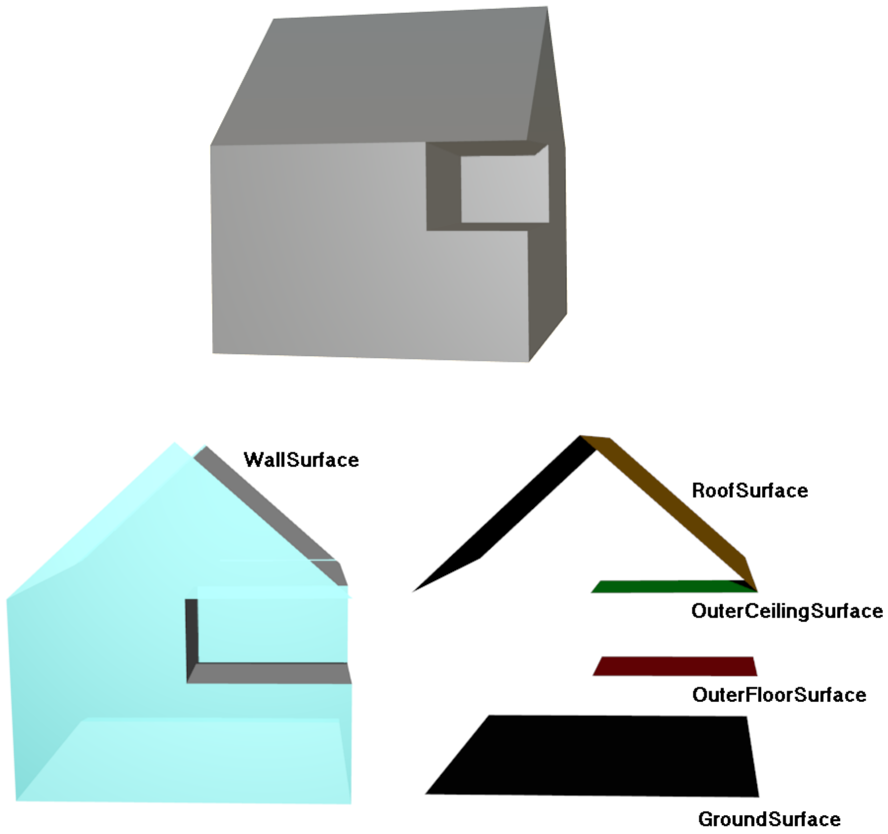


Figure 1.3: Visualisation of all semantic classes in a LoD2 model in the CityGML standard. Source: OGC (2012, p. 70).

The decision to focus on models with LoD 1 and 2 is based on the bigger availability of models with this LoD. Thereby, models with LoD three or higher are considered to be too complex to fully automate the semantic labelling process in this stage.

1.4 CONTRIBUTIONS

In the course of this research, multiple classification techniques are developed and tested. These techniques are developed through a process of incremental design, where the best practices from earlier tested methods are selected and used in the development of the end product. One of the tested methods is presented in Rook et al. (2016). This thesis presents the best practice of all the different tested methods, which does not include the works in Rook et al. (2016). The paper is published during the writing of this master thesis.

The central concepts in this developed methodology are: first, the construction of a spatial index, whereby difficulties with geometrical invalidities such as double stored vertices and small gaps are overcome. Second, an approach which reconstructs the single buildings in the 3D city model. Third, two approaches to enrich the geometries in a single building with semantic information. To do so, this paper presents a logic, based on extended semantic class definitions in CityGML, which allows automatic semantic labelling to create a semantically rich CityGML dataset from a polygon mesh.

1.5 OVERVIEW OF THE THESIS

The research is structured in the following way. This chapter functions as an introduction to the research. Chapter 2 gives an overview of relevant works of others. Next, chapter 3 describes the challenges that have to be overcome. The methods are explained in chapter 4. The implementation of the proposed methods are chronologically described in chapter 5. This chapter chronologically describes the development of the methodology that is described in the previous chapter. This methodology is developed through a process of incremental design. Where the methods are developed from scratch, using the related work as inspiration.

2 RELATED WORK AND LITERATURE RESEARCH

This chapter will provide an overview of already existing works which are relevant for this research and starts with a description of the different techniques that are used to create 3D city models. Next the concept and use of semantics in spatial and non-spatial information science is explained, followed by an introduction to CityGML. The following section elaborates on works which focus on semantically enriching 3D data. Next, different classification techniques and means to analyse the accuracy are explained. This chapter concludes with spatial data handling techniques.

2.1 THE CREATION OF 3D CITY MODELS

Developments in massive 3D data acquisition made it possible to create dense 3D data from the human environment (Diakité et al., 2014; Stoter and Zlatanova, 2003). Different techniques are developed to capture the human environment. Photogrammetry and Light detection and ranging (LiDAR), are currently the most used techniques. 3D laser scan data usually consists out of a collection of points, holding an X, Y and Z coordinate with additional attributes like colour or return intensity. To create a 3D city model, the points are used to create vectorized models, whereby the point geometry is converted to edges and faces, representing the sensed environment (Previtali et al., 2014). Another method to create 3D city models with LiDAR, is to combine the points with other data. Vosselman et al. (2001) describes how LoD 2 city models can be generated from ground plans combined with LiDAR. Takase et al. (2003) describe how LiDAR is combined with topographic maps and aerial images to automatically create textured 3D city models. In these techniques, the 2D polygons are given a third dimension that is computed from the LiDAR points that intersect with the polygon.

Remondino and El-Hakim (2006) gives an overview of available solutions that are used in the generation of 3D models from terrestrial and aerial images. In these methods, a building is photographed from different angles. These images are then used to compute a 3D coordinate for the features, or clusters of pixels, that appear in at least two images. The increase in the availability of this data has triggered an extensive increase of the use of these 3D models for analysis and visualization (Previtali et al., 2014; Stoter and Zlatanova, 2003).

2.2 SEMANTICS IN 3D CITY MODELS

The increasing availability of these models triggers the development of 3D GIS. However, 3D GIS applications require semantic information (Stadler and Kolbe, 2007). Before getting into more detail about semantics in geographical data, first the term semantics will be explained.

Semantics, in the sense of data, is best explained by Tim Berners-Lee's concept of the semantic web (Berners-Lee et al., 2001). He explains that the semantic web is an extension of the internet protocol, which also contains meaningful information about the data, or what it represents, that machines can understand. This "understanding" offers new possibilities in linking and processing data, such as automated reasoning. But for the semantic web to function, computers must have access to structured information about what the data actually represents. This way, a computer can infer rules or use logic to conduct automated reasoning (Berners-Lee et al., 2001).

In the scope of this research, the definition of semantic information is limited to what a surface represent in the real world. For example a roof or a wall.

2.3 SEMANTICS FOR THE PURPOSE OF SPATIAL ANALYSIS

For geographical data, semantics are important for many applications, some examples are:

- **Data integration** Stadler and Kolbe (2007) define the relation between semantics and geometry and describe how semantics in geographical data can reduce the ambiguities for geometric integration, which means merging different datasets into one. For example, when a 3D building models are merged with a digital surface model. They describe a process where different datasets can be merged with the use of semantics, for example: the ground surface of a building model of a house should be connected to the terrain. In order to do so, the geometry of parts of the house and the terrain surface must be separately accessible.

- **Data harmonisation** Data harmonisation is the process of creating consistency in data, in order to allow the unification of different datasets. This process can be done more effective if semantic information is available (van Oosterom and Zlatanova, 2008). A good example of data harmonisation is the INSPIRE framework, which should make it possible to combine spatial data and services from different sources (INSPIRE, 2013).

- **Real world simulations** Vosselman et al. (2001) identifies an increasing interest for 3D city models by urban planners and the telecommunication industry. For example, to simulate the view from a certain location in the city or to compute the behaviour of communication signals in an urban environment.

- **Spatial analysis** Finally, semantics are recognized as one of the most important features that separate virtual 3D models, used for visualization only, from models employed in spatial analyses (Biljecki et al., 2014). As models without semantic information may still be valuable for visualisation and other purposes, their full potential in a GIS is hindered by the lack of semantics (Brodeur, 2012). For example in flood modelling or disaster management (Van Oosterom et al., 2006). Where semantics are used to model the effect of rising water levels and to simulate the impact of events in the real world environment, or to determine the number of affected households by a flood.

Models without semantic information do not allow 3D analyses, because it is not possible to identify the surfaces of interest, e.g. roof surfaces to estimate the solar irradiation on them, or walls to calculate the total facade area.

Biljecki et al. (2015b) researched the current use and utilisation of 3D city models. In their research, they point out the importance of semantics in the different use cases of semantically rich 3D city models. Therefore they categorised 29 use cases. Some of these use cases are:

- **Noise mapping and visualisation** The use of semantics in the propagation of noise in urban environments. In this case, the use of semantics can lead to more accurate and precise predictions and a better assessment of the consequences of noise.
- **Emergency response**, where semantics can for example be used to determine the best position for the deployment of ladder trucks, whereby windows and doors must be distinguishable from other building features.
- **(Indoor) navigation and route visualisation**, where the path-finding algorithm uses semantics to create a topology of the building, as for example where doors are situated.
- **Legal or commercial real estate assessment**, where 3D city models are used to automatically determine the floorspace surface.

2.4 CITYGML AND SEMANTIC INTEROPERABILITY

In order to combine data from different sources, semantic interoperability is required. Interoperability is defined as the ability of computer systems or software to exchange and make use of information (Oxford Dictionaries, 2015). The lack of data heterogeneity is considered to be one of the main issues in the GIS field (Kolbe et al., 2005), because the lack of data heterogeneity hinders the interoperability of data.

Semantic interoperability presumes common definitions of objects, attributes, and their relationships, dependent on a specific domain (Kolbe et al., 2005), crucial for data integration. Semantic interoperability for geographical data is therefore a central issue in the development of the standard CityGML, as CityGML fits into the concept of the Spatial data infrastructure (SDI), which is expected to become more important in the future (Gröger and Plümer, 2012). Semantic interoperability is therefore a key issue for the development of CityGML and therefore plays a central role in this research, as it gives the decisive framework for the different thematic and semantic labels that are added in the labelling process. CityGML is a standard for geographical data and its semantic information, it defines the structure, the aggregations and the taxonomies of the data. The standard is open and independent and specifies the spatial and semantic aspects of the 3D city model. In the standard, the data is ontologically structured, and allows advanced analysis (Gröger and Plümer, 2012), exchange and storage of data.

In CityGML, semantically rich 3D city models have, besides the spatial and graphical aspects, an ontological structure including a thematic class, attributes, and the interrelationships between the two. This way, CityGML can be extended and specified for a specific application domain, because it explicitly supports simple and complex 3D geometry (Kolbe et al., 2005), while relating semantic information to the spatial objects and their geometry (Stadler and Kolbe, 2007). This section elaborates on the thematic and semantic class taxonomy of the CityGML standard.

2.4.1 Thematic information

CityGML is not limited to only storing buildings, but holds all relevant features in an urban environment. On the thematic level, the CityGML class taxonomy defines classes and relations for the most relevant topographic objects in cities and regional models, comprising built structures, elevation, vegetation, water bodies, city furniture, and more. The thematic model of the CityGML standard is based on a hierarchical decomposition of geographical objects, that depends on the LoD of the 3D city model. This way, urban features can be combined to create a dataset which holds the needed information for a specific application (Stadler et al., 2009).

The Building model

The building model is the most detailed thematic class of CityGML. It allows for the representation of thematic and spatial aspects of buildings, building parts and installations in four levels of detail (Fan et al., 2009).

The central class of the building model is the AbstractBuilding. From this central class, the two classes Building and BuildingPart are derived. These classes have a composite relationship, as a Building can contain BuildingParts or one or more Buildings. BuildingParts form the semantic classes of the building model, the presence of this semantic information depends on the LoD of the model. For example, a building may be assigned a solid geometry in LoD1. In LoD2 the building is further decomposed into surfaces, like a WallSurface and a RoofSurface. These associated geometries, or semantic classes, should again refer to the AbstractBuilding they are part of, because they form the outer shell of the building (OGC, 2012).

A Building and its BuildingParts share (Stadler et al., 2009), or inherit (OGC, 2012) attributes from the AbstractBuilding class, like a creation data or a reference that points to the same object in another dataset. Thereby, attributes like year of construction, usage or function are provided for buildings, and inherited by the building parts (Stadler et al., 2009).

The Building class is one of the two subclasses of AbstractBuilding. If a building only consists of one (homo-geneous) part, the class AbstractBuilding shall be used. A building composed of structural segments, differing in for example the number of storeys or when they have a different roof type, the building has to be separated into one building having one or more additional BuildingParts (OGC, 2012, p. 65). This is visualised in Figure 2.1.

A fully coherent CityGML dataset has the advantage that each geometry object ‘knows’ what thematic role it plays and that each thematic feature ‘knows’ its location and spatial extent (Kolbe, 2009). Thereby, the rich thematic and semantic information can be used for thematic queries, analyses, or simulations (Stadler et al., 2009).

Figure 2.1: Examples of buildings consisting of one and two building parts. Source: OGC (2012, p. 65).

2.4.2 Semantic classes in the CityGML Building model

The thematic building class comprises different semantic classes. The classes in a LoD 2 model are depicted in Figure 2.2 and 2.3. This section elaborates on the geometric properties and modelling rules of these classes:

- **GroundSurface** A GroundSurface is an exterior, lower boundary surface of a Building, BuildingPart or BuildingInstallation (SIG3D, 2015), also referred to as the ground plate of a building and is congruent with the buildings footprint. The surface normal of the ground plate points downwards (OGC, 2012).

- **WallSurface** A wallSurface is an exterior, lateral boundary surface of a Building, Building part or BuildingInstallation. A wall is a vertical construction that bounds the internal space. The WallSurfaces should generally lie in the horizontal, up to 45 degrees (SIG3D, 2015).

- **BuildingInstallation** A BuildingInstallation is a permanently installed part of the buildings outer shell, which is an accessory for the building structure, including loggia, dormer, etc and is not a mandatory class. Doors and windows are modelled by their corresponding CityGML classes and are not part of the BuildingInstallation (SIG3D, 2015).

- **RoofSurface** A RoofSurface is a exterior, upper boundary surface of a Building, BuildingPart or BuildingInstallation. A roof encloses a building from above. The normals of a roof should generally lie upwards (SIG3D, 2015).

- **OuterFloorSurface** The OuterFloorSurface is the Exterior, upper boundary surface of a Building, BuildingPart or BuildingInstallation which is not a roof. The normals of an outer floor surface should generally be vertical and directing upwards (SIG3D, 2015; OGC, 2012). An example is the floor of a loggia (OGC, 2012).

- **OuterCeilingSurface** A OuterCeilingSurface is a exterior, lower boundary surface of a Building, BuildingPart or BuildingInstallation against the outer space (SIG3D, 2015). Examples are the visible part of the ceiling of a loggia or the ceiling of a passage. The normals of an outer ceiling surface should generally be vertical and directing downwards (OGC, 2012).

- **ClosureSurface** Additionally, buildings with open sides, like a barn or a hangar, are virtually closed. These features are closed to be able to compute their volume. The surface which closes these features is called a ClosureSurface.

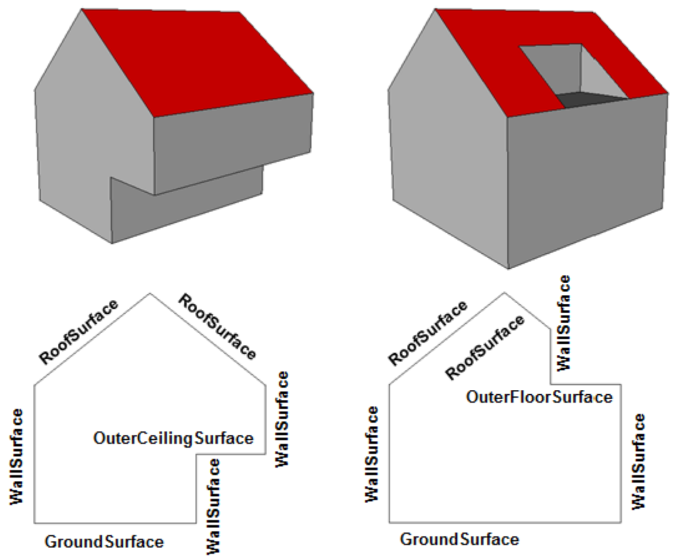


Figure 2.2: Semantic BoundarySurfaces classes of a LoD2 model in the CityGML standard (1). Source: OGC (2012, p. 70).

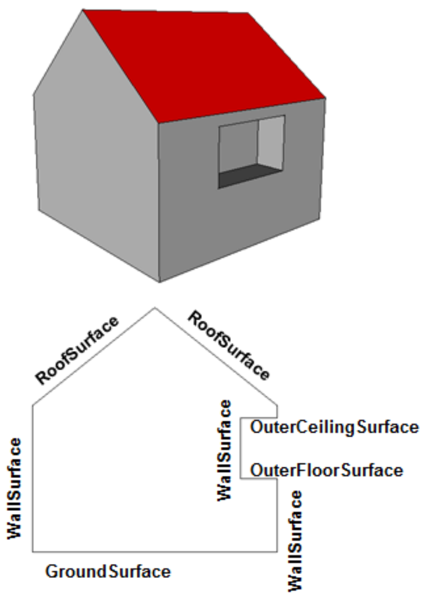


Figure 2.3: Semantic BoundarySurfaces classes of a LoD2 model in the CityGML standard (2). Source: OGC (2012, p. 70).

2.5 RELATED WORK IN THE SEMANTIC ENRICHMENT OF 3D DATA

This section summarises the efforts in the field of semantic enrichment and classification of spatial and non-spatial data.

2.5.1 Semantically enriching 3D city and building models

Enriching 3D city models with semantics has been researched in different fields with varying methods. In most cases, semantics are manually added to these models. Some research in semantically labelling 3D city models exists. This section shortly elaborates on these research efforts.

Henn et al. (2012) researched a method to classify buildings, in LoD 1, by their building type, whereby a Support Vector Machine (SVM) was developed. A SVM is a supervised learning algorithm, whereby the aim is to automatically find regularities and patterns in data. In the research by Henn et al. (2012), seven classes of building types, which are typical for urban development in Germany, where classified. Some of the different classes are: detached and semi-detached buildings, terraced buildings and villas. The classification is purely done on geometrical properties, such as length, width (the shortest edge), footprint area or the volume of the building. Second, the feature space consists out of measures that reflect the complexity of the building, like the number of right angles and vertices of the footprint. Some building types are thereby defined by their construction, compared to their neighbouring buildings. As for example, the terraced buildings are a part of building blocks that consist out of at least three buildings. Third, infrastructural features are used, based on an assumption that certain types of infrastructural institutions agglomerate in certain city districts, whereby the feature is calculated as the distance from the building to the infrastructural institution, like hospitals, stations or schools. SVM algorithms use a set of training data, that define the feature space for the classification process. In the case of Henn et al. (2012), whole streets, where one class of buildings frequently appears, are, after a cleaning process, used as training data. This selection of training data does require knowledge about the scene that is to be classified. The algorithm classified the buildings accurately in over 90% of the time.

Verdie et al. (2015) created a workflow that produces a semantically rich 3D city model from a triangular mesh, created by a Multi-View System (MVS). In their framework, the input data is a raw triangle surface mesh. The classification step relies on a MRF, in order to distinguish between four classes: ground, trees, façade and roof. The method is unsupervised and only uses geometric attributes. The following logic defines their work best: the ground class is characterized by locally planar surfaces, that are located below the other classes. Trees have curved surfaces. Façades are vertical surfaces, that are adjacent to roofs, another class, which are composed of planar surfaces. In the research, no single triangles are used in the classification process. Instead, super-facets are used, that are sets of connected triangles with the same characteristics.

Diakité et al. (2014) propose a method that is based on a propagation method that is directed by heuristic rules, in order to retrieve semantics of the building components. The approach takes vector data as input, where the Combinatorial Map (C-Map) data structure is used to reconstruct the topological relations. The C-Map is a edge-centered data structure, that, in 3D, describes an object by it's vertices, edges and faces. The basic element of a C-Map is the dart, which is part of each incident cell, meaning that two cells are incident, if one belongs to the boundary of the other. The process entirely relies on a method of heuristic rules, which combines topological and geometrical criteria, which gives the flexibility to define as much rules as desired, whereby only geometry is used. The different semantic classes are: facade, wall, ground, floor and roof.

In the work of Biljecki and Arroyo Ohori (2015) a conversion method is described, where a 3D city model is converted from CityGML to Wavefronts object format and the other way around. In the conversion from object to CityGML, the triangles are enriched with semantic information. This classification is solely based on the orientation of the surface normal. For instance, a surface whose normal is horizontal is most likely a wall. Thematic information is captured by utilising the information from a spatial index, which allows easy retrieval of triangles which share an edge. These relationships are used to create clusters of triangles, which together form distinct groups of faces that represent a single object.

Boeters et al. (2015) propose a method to automatically generate indoor geometries, based on existing CityGML LoD2 exterior geometries. In their research, the same approach as Biljecki and Arroyo Ohori (2015) is used for the classification of the exterior geometries. These surfaces are classified as RoofSurface, WallSurface or FloorSurface. The classification is based on a method that computes the pitch angle of a triangle, computed with the normal vector of the triangle. The classification thresholds for the different classes is visualised in Figure 2.4.

2.5.2 Semantically enriching point cloud data

While this research does not focus on point clouds, they are of interest to this research, because they are also 3D data. The focus on the classification of point clouds does limit the applicability on 3D city models.

Pu et al. (2006) look for seven urban classes in a point cloud, by using the properties of clustered segments of points in that point cloud. These classes are floor, wall, window, roof, door, extrusion and intrusion. The distinguishing features of these classes are:

- The size of the segment, as walls, windows and doors can be easily distinguished from other features by the size of the clustered segment.
- The position, because certain features appear only in a certain position. For example, windows and doors are always on walls, while roofs are always on top of walls.
- The orientation, as walls and roofs can be distinguished by their direction.

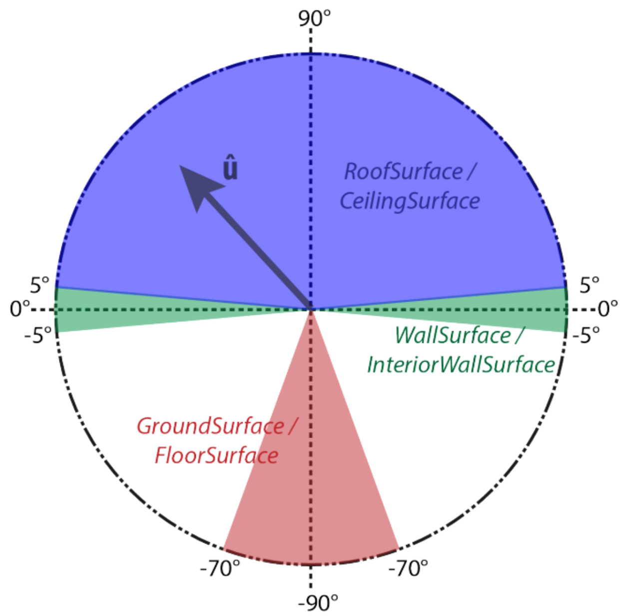


Figure 2.4: The angles in the figure show which angle results in which surface type. Source: Boeters et al. (2015, p. 13).

- Topology, as building features have certain topology relations with other features or for example, the terrain.
- Last, miscellaneous constraints, that includes other information, for example, point density, as windows have lower point density because glass reflects fewer laser pulses.

Next, Pu et al. (2006) describe the importance of the order in which the regions are assigned a class, as some feature recognition is based on other feature recognition. For example, terrain and walls are detected first, but the recognition of walls first needs the recognition of terrain, while extrusion and intrusion features need wall features. Therefore the order of feature recognition is: ground, wall, window, roof, door, extrusion and intrusion.

Waldhauser et al. (2014) aimed at developing models to automatically classify ground cover and soil types from airborne LiDAR. The focus of this research was to find a new, fast and reliable algorithm for the classification of point clouds, which can minimize the manual checking and correction. Here fore, they used the logic of supervised machine learning, whereby the focus is on decision trees. A number of 11 classes is ought to be recognized and classified in the point cloud. The decision tree seeks to partition the entire feature space of a dataset, one variable at the time. Hereby they state, that the larger the training dataset, the better the classification will be. One of the main point attributes they used to do the partitioning in the decision tree, is the echo of the LiDAR point, also called the number of return.

Niemeyer et al. (2014) addresses the task of contextual classification of airborne LiDAR. Here fore they integrate a random forest classifier into a conditional random field, without using any external information like aerial imagery. What distinguished this research from other point cloud classification approaches, is that Niemeyer et al. (2014) not only utilises the information of the points, or the points region into account, but also uses the labels of the surrounding neighbourhood. The different point region attributes the classification is based on, are the following: intensity, ratio of echo number, height above DTM, variance of point elevations in a sphere of radiance r. The ratio of point density in a sphere of radiance r, eigenvalue based features in a sphere of radiance r, point density in a sphere of radius r and the variation of intensity, omnivariance, planarity, anisotropy, sphericity, point density, number of returns, mean curvature and gaussian curvature in a sphere of radius r.

Khoshelham and Díaz-Vilariño (2014) present an approach for 3D indoor modelling from point clouds, based on a shape grammar. The research demonstrates that interior spaces can be modelled by iteratively placing, connecting and merging cuboid shapes. Here fore, the parameters and the sequence of grammar rules are learned automatically from a point cloud. Using the shape grammar, an indoor environment is modelled as a configuration of parametrized spaces. Such a model contains semantics like height and volume of the spaces and their topological relationships.

Pittarello and De Faveri (2006) present an outline for automatic semantic building model reconstruction from preprocessed point clouds. In this process, the building reconstruction is a multi step process in which each step corresponds to a different level of detail and uses a specific set of symbols, production rules and semantic rules. The steps are embedded in a control structure, which determines the next step. The process is largely based on the topological relations of the different surfaces.

2.5.3 Semantically labelling non-environmental 3D data

Kalogerakis et al. (2010) present a data-driven approach to simultaneous segmentation and labelling of parts in 3D meshes. The meshes in this research vary from shapes that represent humans to everyday objects like a vase or sunglasses. The classification function uses training data from a collection of labelled meshes. They define the problem of mesh part recognition as a problem of optimizing a Conditional random field (CRF).

2.6 CLASSIFICATION METHODS

The aim of this research is to classify three dimensional surfaces. Different methods exist in the classification of spatial and non-spatial data. This section describes the methods that are used in this research.

2.6.1 Decision tree

Decision tree leaining is a widely used and practical method. The method suits best for classification problems with conclusive and decisive classes (Shalev-Shwartz and Ben-David, 2014). The classification problem of the geometrical data in this research satisfies this condition, as the classes are well defined and explicit. A decision tree classifies instances by sorting these instances down a tree from the root to a leaf node. This leaf nodes provides the classification of the instance, where each node in the tree specifies a test on one of the instance its attribute (Mitchell, 1997). The decision tree uses a tree as a predictive model, whereby observations of a feature lead to the conclusion about this feature. It uses a classification scheme to do so, a hierarchical structure that is accompanied by descriptive information. Algorithms to create a decision tree work top-down, eventually classifying the features. This classification scheme assigns a class to each feature (Maimon and Rokach, 2005).

2.6.2 Propositional logic

Propositional logic is the branch of logic that studies ways to join or modify statements to form more complicated propositions or sentences. Thereby, also the logical relationships and properties that are derived from this method of combining or altering statements is part of propositional logic (Internet Encyclopedia of Philosophy, 2016). In other words, propositional logic deals with logical relationships between propositions, which are statements and assertions, taken as a whole. This means that the fundamental unit of analysis is the whole proposition, which can only be True or False.

2.6.3 Heuristic rules

Heuristics stands for strategies that use available and accessible information to control or improve problem-solving processes or decisions by humans or in man-machine interaction (Pearl, 1984). In heuristics, the use of the general knowledge, or knowledge gained by experience is used in the reasoning to get to a decision or, in this case, to do a classification. Wikipedia (2015b) provides a simple and clear definition of heuristic techniques, or simply called a heuristic: "A Heuristic, is any approach to problem solving, learning, or discovery that employs a practical method not guaranteed to be optimal or perfect, but sufficient for the immediate goals. Where finding an optimal solution is impossible or impractical, heuristic methods can be used to speed up the process of finding a satisfactory solution". Heuristic rules are used by Diakité et al. (2014) to semantically label a 3D model of a house.

2.7 INTERPRETING THE THEMATIC AND SEMANTIC CLASSIFICATION ACCURACY

Stehman (1997) states that the comparison of classification algorithms is a complex and open problem, because of three main reasons. First, performance can be defined in many ways, for example: accuracy, speed, or readability. Second, an appropriate tool is necessary to quantify the performance definition. Third, a consistent method is needed to compare the measured values in a correct and consistent way. The confusion matrix provides an answer to all these three difficulties.

The confusion matrix is a specific table layout that allows visualization of the performance of an algorithm, and provides an adequate solution for all the three above mentioned problems. In the confusion matrix, each column represents the instances in a predicted class while each row represents the instances in the actual class. The confusion matrix makes it easy to see if the system is confusing two classes, in other words: mislabelling an instance (Wikipedia, 2015a). The most popular measure for classification accuracy and the overall success rate is defined as the trace of the confusion matrix, where the total number of rightly classified instances is divided by the total number classified instances. This measure is multi-class, symmetrical, and ranges from perfect misclassification to perfect classification (Stehman, 1997).

2.8 SPATIAL DATA HANDLING

3D city models consist out of points, lines and polygons, which all are types of spatial data. The multiple dimensions of spatial data makes the processing more demanding in terms of computing power and therefore time. In order to efficiently handle the spatial data, spatial indexing and spatial access methods are used. The main purpose of spatial access methods is to support efficient spatial selection. Examples of these are nearest neighbour queries of spatial objects or other spatial analysis as topological relationship calculations. These access methods are needed because most spatial data is unstructured, irregularly distributed and large. A spatial access method needs to take spatial indexing and clustering techniques of the data into account. Without spatial indexing, computing tasks with geometric or geographical data take much more time. For example, in searching for the nearest neighbours, every spatial object has to be iterated over to calculate the distance to every other object, leading to longer query or computing time which is unacceptable in practice for interactive users and many applications. Therefore spatial indexes are required (van Oosterom, 1999).

2.8.1 kd-tree

The spatial index method used in this research is the kd-tree. The kd-tree is a space-partitioning data structure to organize points in a k-dimensional space. The kd-tree is a binary tree, in which every non-leaf node splits, with the use of a hyperplane, the space in two, creating half-spaces. The points on the left of this hyperplane are represented by the left sub-tree, the points on the right by the right sub-tree. The hyperplane splits the space in turns on all different axis (Wikipedia, 2015c). This space partitioning creates a tree data structure, that is used to efficiently iterate over the data.

2.8.2 Region Growing

Vosselman et al. (2004) describe different automatic processing techniques to extract information from point clouds. To do so, the paper gives an overview over different techniques for the extraction of surfaces from point clouds. These smooth surfaces are mostly extracted by clustering neighbouring points together, that share similar properties, like, for example, the orientation of normal vector of the point and its neighbourhood. This way, a point cloud is grouped into multiple regions, which are clusters of points. The most useful technique for this research that is shortly described, is the region growing approach.

This region growing approach is used by Rovers et al. (2015) to add semantic information to point clouds. In this case, colour and geometry are used to group the similar points in the point cloud together. Next, the groups, or clusters, of points are used in a supervised classification method in order to add semantics. The same concept is used in image segmentation problems for computer vision (Hoover et al., 1996), or where objects are ought to be recognised in photographs, like in Google's and Facebooks image recognition challenge ILSVR (ImageNet, 2015). In these cases, not points, but pixels are grouped together.

3 CHALLENGES

In the course of this research, a number of challenges are identified, partially from the work of related researchers, partly from preliminary experiments. This chapter elaborates on these different challenges, which have to be overcome in the development of the methodology.

3.1 COMPLEXITY OF THE SEMANTIC CLASSES AND THE LACK OF DEFINITION

The definitions of semantic classes in CityGML are not restrictive, which allows a lot of freedom in the modelling of the human environment. A good example is the number of possibilities to model the underground structure or the foundation of a building, visualised in Figure 3.1, which shows three valid ways to model the cellar and foundation of a house.

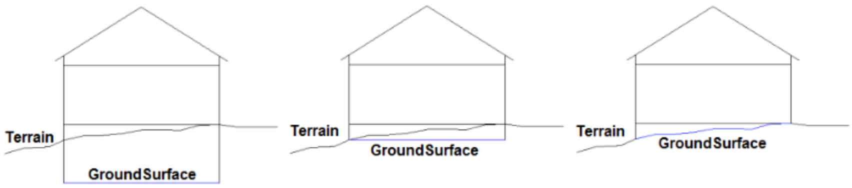


Figure 3.1: Example of the modelling of a Buildings underground structure and the terrain intersection. Source: SIG3D (2015).

The freedom CityGML offers gives way to model complex environments in a very realistic way, while still being a valid model. A downside however, is that this freedom limits the possibility to infer classification rules on the surfaces in the 3D city model. OGC (2012) does prescribe the orientation of the surface normals for the different classes. But the possibilities to utilise this information in this research is also limited, which is further explained in the next paragraph.

3.2 LIMITED APPLICABILITY OF UTILISING THE SURFACE NORMAL

The normal of the surfaces in the 3D city model play an important role in the classification of surfaces in CityGML, as a fair share of semantic information can be inferred just by analysing the orientation of the surface (OGC, 2012). Also the classification by Biljecki and Arroyo Ohori (2015), as described in section 2.4.1, solely depends on the surface normals. In their research, the 3D city models where converted from CityGML to OBJ. The CityGML standard defines rules for the orientation of the different surfaces, what creates opportunities to store and later label the surfaces with this information.

However, a robust classification solely based on the normals of the triangles is not possible in this research, because the orientation of the normal can be correct in 3D city models which originally where stored as CityGML, this is not necessarily the case in models from other sources.

Thereby, even in CityGML it is impossible to distinguish between the classes RoofSurface and OuterFloorSurface, where both the normals point upwards, and the classes GroundSurface and OuterFloorSurface, where the surface normals point downwards. These classes can therefore not be detected by only utilising the surface normals. In figure 1.3, all the semantic classes are visualised, this figure shows that the RoofSurface, OuterCeilingSurface, OuterFloorSurface and GroundSurface all have normals that point up- or downwards and cannot be classified by only utilising this information. Therefore, a method has to be devised where a class assignment depends on the relationship with surrounding and other present semantic classes.

3.3 TOPOLOGY AND SPATIAL INDEX

To retrieve the spatial relationship of a triangle with other triangles, and their semantic class, topological relations have to be recovered to obtain additional insights that may hint at the semantics of a surface. Biljecki and Arroyo Ohori (2015) referred to a topology as a spatial index. After inspecting the test models (to be introduced later), it turned out that these topological relations are not always directly retrievable. Some models contain double vertices, where others hold gaps between adjacent triangles. Other cases which cause missing topological relationships are floating roofs, or roofs that are not connected to a wall. The availability of models which hold a valid geometry is researched by Ledoux (2013), who investigated the validation of solids, giving different examples of valid and invalid solids. For example, a solid is invalid when they overlap another solid or when they invade each others' space. Another example of a valid triangle is when two adjacent triangles share the same points and edge, if not, the creation of the topology will be impossible or much harder since real-world models are virtually never error-free.

3.4 SEMANTIC CONTENT AND LOD DETECTION

To correctly classify all the semantic classes in the 3D city model, the algorithm must first recognise the content of the different semantic classes. This *scan* is required because the labelling process should automatically realise what classes it has to classify. For instance, among the different selected models that are used to test the algorithm, some models have a terrain, while others do not. Therefore, some buildings in the different models have a BuildingInstallation which represent dormers or chimneys, while other models only contain roofs, walls, together forming a Building. These features have to be recognised, in order to make a valid classification. The generated output also depends on the LoD of the 3D city model, i.e. it makes no sense to classify semantic boundary surfaces in an LoD 1 dataset, as explained in section 1.3. Therefore, the LoD of the model should also be detected, as different LoDs contain different semantic classes. Automatically detecting the LoD of the model is also one of the main challenges in this problem.

3.5 LACK OF THEMATIC DEFINITION BUILDING AND ABSTRACTBUILDING

The datasets which are used share no consistency in the geometric aggregation of the classes Building and AbstractBuilding. Or, as explained in the OGC (2012) standard : "CityGML allows many different alternatives for modelling. This is an obstacle in the validation process, because it is not unambiguously defined what validity actually means without further specification". For example, the elements Building and BuildingPart can be modelled in three different ways: as a single solid, a composite solid or as one single multi surface geometry. All the three options are valid in the CityGML standard (OGC, 2012). Thereby, the aggregation of Buildings and BuildingParts is not only based on geometrical properties and can therefore not be aggregated by geometrical properties only. Figure 3.2 depicts a case which shows the challenge of recognizing and aggregating different Buildings into one AbstractBuilding. The Buildings in this single model can be aggregated as one AbstractBuilding, but can also be stored separately. Both are correct.

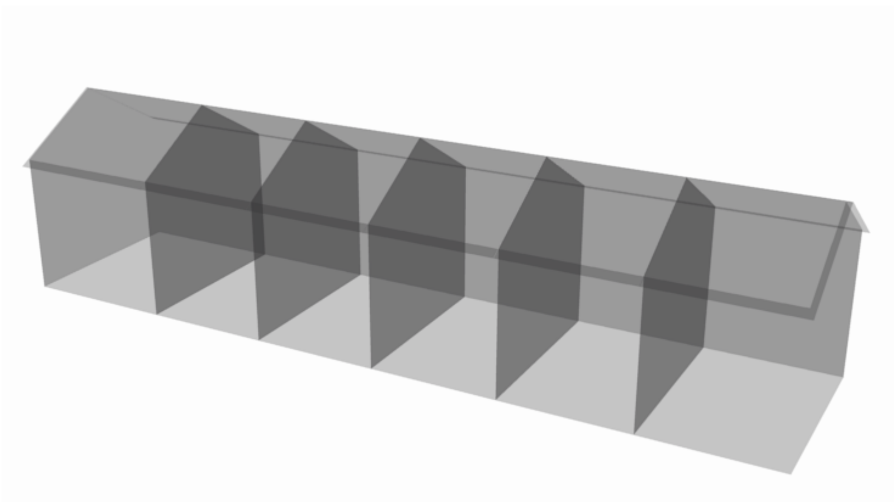


Figure 3.2: Example of an aggregation of BuildingParts into one Building. Data source: (Rotterdam municipality, 2015)

This obstacle makes it hard, or impossible, to reconstruct the thematic aggregations and information from the original 3D city model.

4 METHODS

To enrich a 3D city model with semantic and thematic information, the just proposed challenges, see section 3, have to be overcome. The methodology to overcome these challenges and to enrich a 3D city model with semantic and thematic information is explained in this chapter. This methodology is structured in a number of steps, which are illustrated in Figure 4.1. These steps function as guide through the methodology proposal. In step 5: Semantic classification, two approaches are proposed, which focus on a different part of the defined challenges. The process takes a triangulated polygon mesh as input. Figure 4.2 displays a screenshot of a polygon soup.

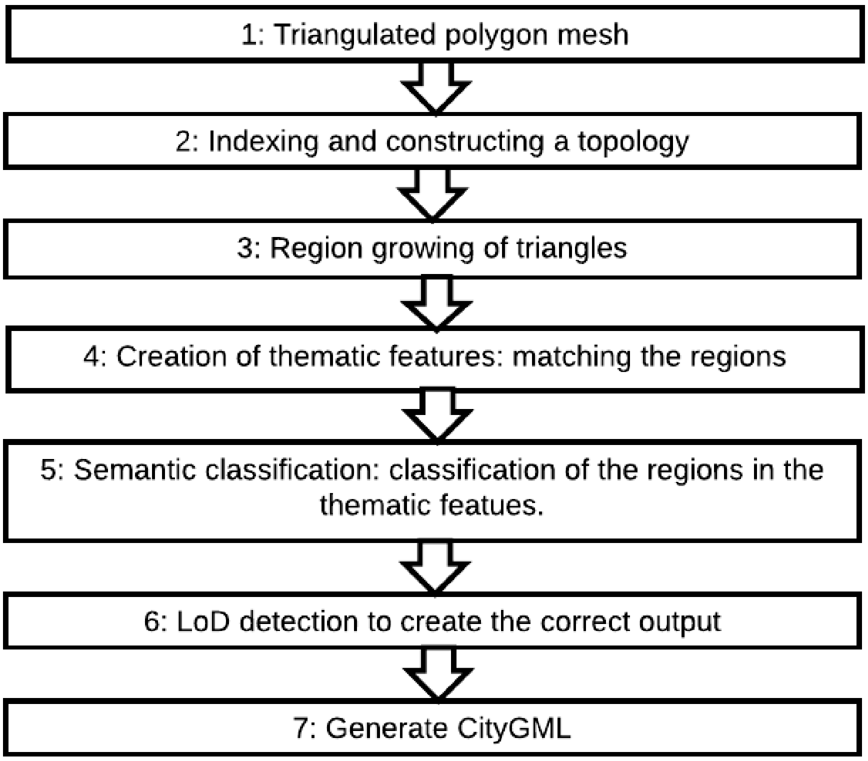


Figure 4.1: Different steps in the workflow

Introduce two semantic labelling approaches.

4.1 INDEXING AND CONSTRUCTING A TOPOLOGY

The first step is the spatial indexing of the triangles in the model. With this index, adjacent spatial features of an object can be easily retrieved. This is necessary to cluster the triangles into regions, in order to exploit adjacency

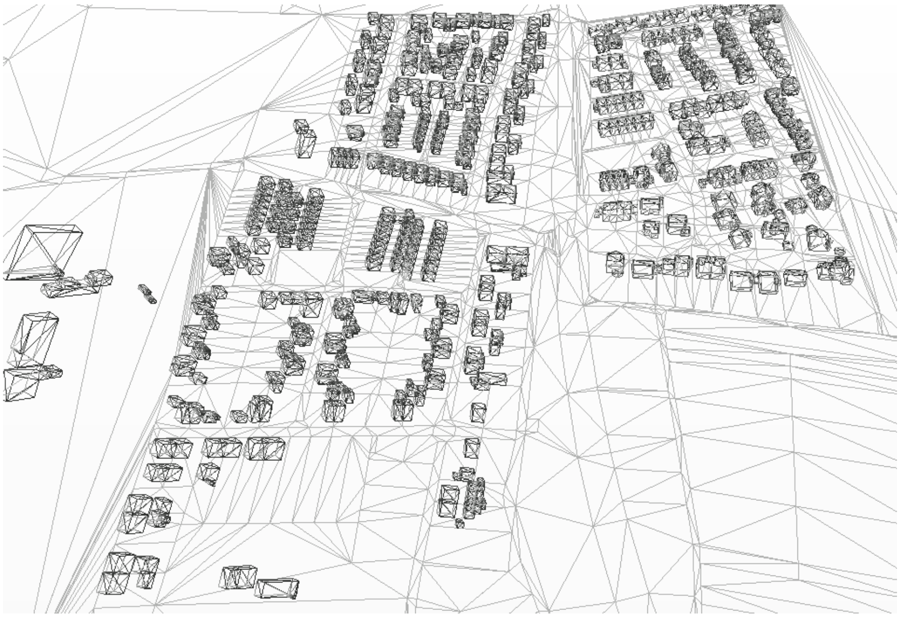


Figure 4.2: Screenshot of the semantically unlabelled Waldbrucke city model. Data source: (CityGML, 2016)

relationships of the triangles and to recompose the single Building entities, which are the next steps in the labelling process (Figure 4.1). A spatial index is also used in the research of Biljecki and Arroyo Ohori (2015), who used the spatial index to group the triangles that formed connected components and together represent individual buildings. Although the method to compute the index is different, the goal of the indexing is the same.

In the research of Biljecki and Arroyo Ohori (2015) the edges are indexed. Next, these edges are matched to detect neighbouring triangles, which are triangles that share an edge. In this research, the triangles are indexed on a relationship where two triangles share one vertex, instead of an edge. An example is given in Figure 4.3, where the triangles in example 'A' share two vertices, which together form one edge. While in example 'B', two neighbouring triangles only share one vertex and therefore not an edge. The proposed methodology also indexes these triangles as neighbours.

This different approach allows an improved reconstruction of semantic and thematic features, because some buildings are composed out of different parts which not necessarily share an edge, but can also share one single vertex. This is visualised in Figure 4.4, where a building has an annex on its roof, which is part of the building. This annex is not connected to the rest of a building by an edge, but by one single vertex.

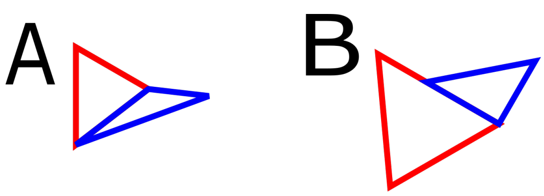


Figure 4.3: Example A: two triangles share an edge. Example B: two triangles share a vertex.

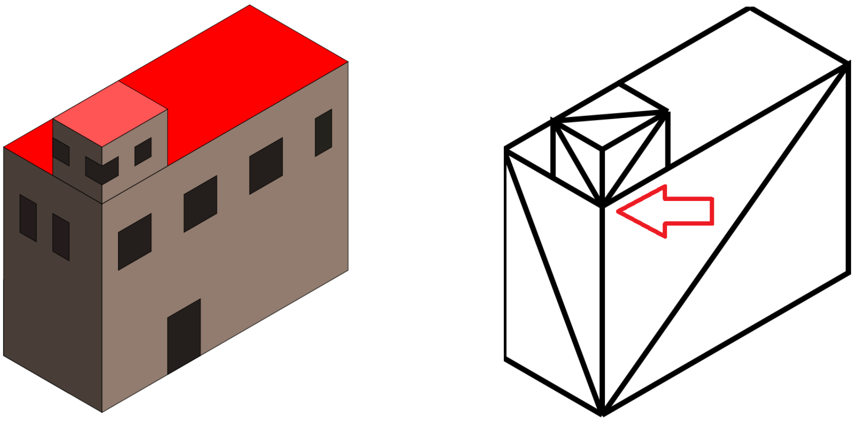


Figure 4.4: A building, of which its parts are connected by one single vertex.

As described in section 3.3, some datasets contain double stored vertices or adjacent triangles separated by a hole, also referred to as sliver polygons. To handle and restore all these different cases, a one fits all method has to be devised, that: one, snaps the vertices that are very close together. Two, corrects the errors that are caused by double stored vertices.

The proposed method reconstructs the topology through the use of a kd-tree, which is often used in region growing algorithms for point clouds. This kd-tree takes the coordinates of all the vertices as input, and returns a list for every vertex with its k nearest neighbours and the distance to those neighbours, as explained in the implementation chapter, section 5.3.1. To snap and to detect double stored vertices, a threshold is calculated by selecting 250 random edges from the city model, and multiplying the length of the 50st smallest edge by 0.01. This threshold is used as a measure to identify vertices that should be snapped or amended. The kd-tree is used as follows.

After the construction of the kd-tree all vertices are iterated over, while checking the distance to its 5 nearest neighbours. In cases where the distance is zero or below the set threshold, a neighbouring relationship between two vertices is stored. Figure 4.5 visualises this process, in which vertex N_1 is stored as a neighbour of vertex P, the vertices N_2 and N_3 are not stored as neighbours, because their distance to vertex P is bigger than the threshold.

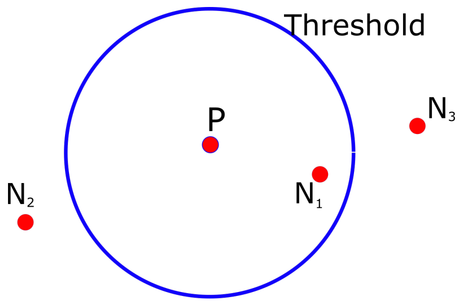


Figure 4.5: Vertex P, its 3 nearest neighbours (N_1, N_2, N_3) and the threshold that is used to reconstruct the topology.

This relationship is later used to retrieve the adjacency relationships of the triangles. In other words, every triangles is composed out of three vertices. The neighbour relationships between the triangles are retrieved by identifying all the triangles which contain one of the neighbours of the vertex the triangle is composed of. This way, the problem of gaps is overcome.

4.2 REGION GROWING OF TRIANGLES

The input 3D city models that the method seeks to enrich come as a triangulated mesh. To reconstruct single semantic features, which together form one surface and represent one semantic class, regions are grown. A region is a cluster of adjacent triangles which hold corresponding geometrical properties. Using regions instead of individual triangles has one main advantage: it gives way to exploit the topological relations between those regions, in order to aggregate the different semantic classes into single building features. For example, recognising the WallSurfaces, by exploiting their adjacency to a roof surface, facilitates the storage of this relationship. This relationship can later be used to create the individual thematic AbstractBuilding entities and to label the surfaces with a semantic class. This approach is used by Niemeyer et al. (2014), as explained in the related works, section 2.4.2, who exploits information about surrounding clusters of points to assign a class to another region of points in a point cloud.

The region growing algorithm is explained in Algorithm 4.1. The geometrical property that is used to cluster the triangles is based on the pitch angle of the triangle, computed with the surface normal (explained in section 5.4.2), for which a threshold is created. This threshold separates the triangles in two groups: triangles which represent a WallSurface and triangles which don't. The threshold is set as a range from zero to five degrees and is also used in related research (Biljecki and Arroyo Ohori, 2015). This is further discussed in section 4.7, where the properties of the different semantic classes are explained.

The threshold that is set creates two collections of triangles: one with triangles which represent a WallSurface and one with triangles which represent another semantic class, these collections are processed individually in the region growing algorithm. This creates two sets of regions: WallSurfaces and non-Wallsurfaces.

Algorithm 4.1: Region grow (L, T, τ)

Input: A list L of triangle class instances T, a seed point triangle τ, and a list of neighbouring triangles KNN.

Output: A list of regions (R) containing a collection of neighbouring triangles which share similar geometrical properties.

while *length of KNN* > 0 **do**
  for T *in* KNN **do**
    if *T holds similar geometrical properties as seed point triangle τ* **then**
      Add T to region (R); Add neighbours of T to KNN
    else
      continue

The result of the region growing algorithm is visualised earlier in Figure 1.3. This image shows that different surfaces are partitioned and that one single region represents one semantic class and represents one complete surface. These regions function as the input for the next step, which assembles the different regions into single building entities. The distinction between the WallSurfaces and non-Wallsurfaces is of great importance for the next step.

4.3 THEMATIC CLASSIFICATION: BUILDING RECONSTRUCTION

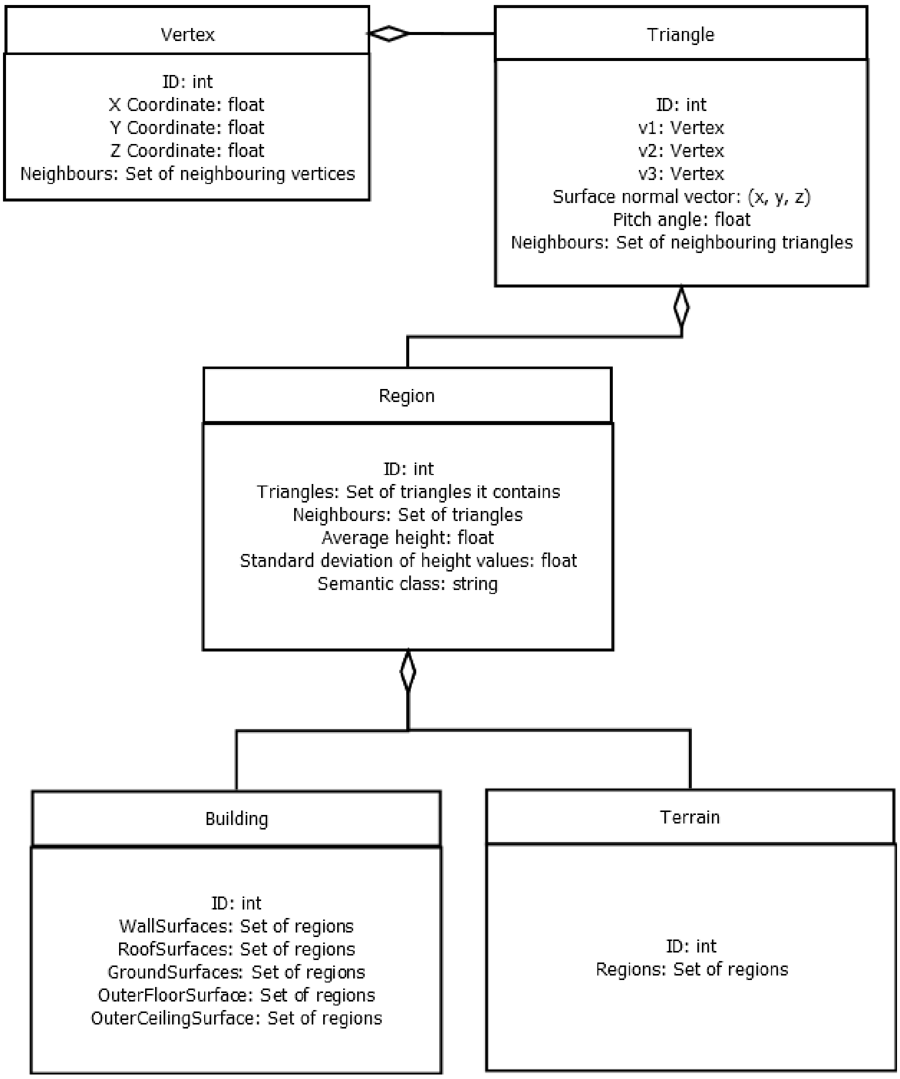


Figure 4.6: Data structure of the algorithm.

After the completion of the region growing, the regions, which individually represent a semantic class and together form a thematic entity, are clustered together into single building entities. This facilitates the creation and recognition of thematic building features, needed to create a CityGML dataset. A single building, in this case, is formed by a region of wall triangles, with all connected non-wall regions. Therefore, a building can be defined as a set of connected triangles, which is separated from other sets of connected triangles in the model by empty space or the terrain. A UML diagram of the data structure is depicted in Figure 4.6.

In this process, the WallSurfaces function as the base class for the reconstruction of the buildings. In other words, the WallSurface are the central class, the non-WallSurfaces regions are matched and connected to these wall regions.

More complex buildings however, as shown in Figure 4.7, contain multiple WallSurfaces and non-Wallsurfaces. This means that matching WallSurfaces to non-WallSurfaces once is not sufficient, because that will lead to multiple single building entities. This is best explained with the example in Figure 4.7. In this Figure, the red, green and blue WallSurface regions will be recognised as one single building, if the different regions are only matched once, while these WallSurfaces are in fact part of a bigger building.

In order to reconstruct such buildings, all WallSurfaces and non-Wallsurfaces have to be added to the same building until the building contains all surfaces which are connected to one of its parts. More concrete, for the building in Figure 4.7 this means, that the building is not complete until the blue, red and green WallSurfaces and all connected non-WallSurfaces are part of the building. Therefore, the reconstruction of the buildings is done through a process which is described in Algorithm 4.2.

The regions are matched by exploiting the earlier created spatial index. These relationships are exploited by comparing the neighbours of all the triangles of the WallSurfaces with the triangle IDs in the non-Wallsurfaces set.

Algorithm 4.2: Compose buildings out of surfaces

Input: A table which stores all the connections between a WallSurface and its neighbouring non-WallSurfaces, a table which stores the connection between a non-WallSurface and all neighbouring WallSurfaces
Output: A single building entity.

Select a WallSurface W

while *set of non-WallSurfaces K which are connected to $W > 0$* **do**
    for *all connected non-WallSurfaces K_n in K* **do**
        Add connected non-WallSurface K_n to the building entity
        for *All WallSurfaces U that are connected to the just added non-WallSurface K_n* **do**
            Add connected WallSurface U_n to the building entity
            for *all non-WallSurfaces G which are connected to the just added WallSurface U_n* **do**
                Add non-wallSurface G_n to set K

4.3.1 Detecting the terrain

The presence of a terrain however, leads to multiple buildings being assembled as one. This is caused by the fact that these buildings will be connected to the same non-Wall region, which in this case represents the terrain.

To prevent this, the user must define if the 3D city model has a terrain or not. If the user states that the city model has a terrain, the regions which contain terrain triangles are detected. This is done by, first, checking if a non-WallSurface region contains more than 25% of all non-Wall triangles. If so, the region is labelled as terrain. Second, if a non-WallSurface region is connected to more than four WallSurface regions, it will also be labelled as terrain. Theses measures are set after experiments with different city models. Next, the neighbouring regions of the regions which are defined as terrain will be filtered out during the reconstruction of the buildings. This means that in step 6 of algorithm 4.2, the WallSurfaces, which are connected to the terrain, will not be added to the building entity.

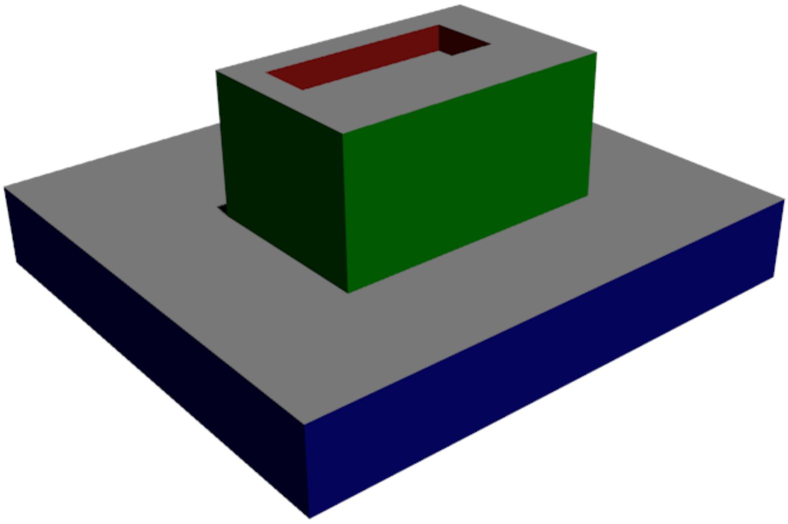


Figure 4.7: A building which is composed of multiple WallSurface regions, depicted in red, green and blue

The outcome of the matching process is shown in Figure 4.8, in which each individual building is given a different colour. These single buildings later function as input for the classification of the semantic classes.

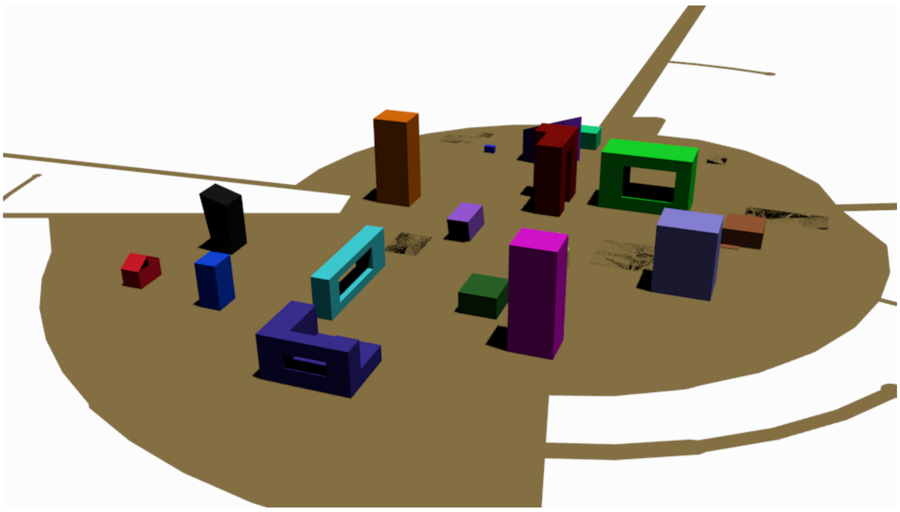


Figure 4.8: 3D city model, where the triangles which form a building are clustered.

4.4 SEMANTIC CLASSIFICATION

The coming two sections describe the methods on which the regions of the individual building entities are assigned a semantic class. Multiple approaches are tested in the course of this research, two of which are presented in this report. Both approaches take the recomposed buildings, which are created in the previous step, section 4.3, as input and exploit the separation of the WallSurfaces and the non-WallSurfaces that is created in the previous step.

The first approach, named the two class approach, is described in the following section and is based on a best practice of various tested methods that aimed to automatic semantically label freely available 3D CityGML models from the internet. Therefore, it is important to notice that the goal of developing this methodology is to semantically label an existing 3D city model which only contains the semantic classes RoofSurface, GroundSurface, WallSurface and, additionally, the terrain.

The second approach, named the comprehensive approach, aims at additionally recognising the classes OuterFloorSurface and OuterCeilingSurface. This specific topic has not been researched before. Thereby, the usability of this methodology will not be limited to semantically enriching an existing 3D city model, but is also an explorative approach to automatic semantically label a 3D city model during its creation, where additionally the classes OuterFloorSurface and OuterCeilingSurface can be labelled. A more in depth explanation follows in section 4.7. In both approaches, the first semantic class that is labelled is the WallSurface.

4.5 SEMANTIC CLASSIFICATION: THE WALLSURFACES

The regions which contain the WallSurfaces are already separated from the regions that do not represent a WallSurface during the region growing. This selection of triangles is based on the pitch angle of the triangle, which is computed with the normal vector of the triangle, described in section 5.4.2 and visualised in Figure 4.9. To make a selection, a threshold is set, which labels the triangles with a pitch angle between -5 and 5 degrees as a WallSurface. This is visualised in Figure 2.4.

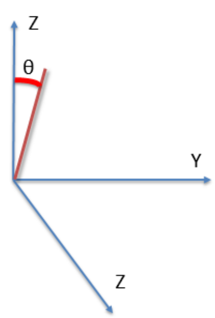


Figure 4.9: Pitch angle in the 3D coordinate system.

4.6 SEMANTIC CLASSIFICATION: TWO CLASS APPROACH

As explained above, the two class approach aims at distinguishing between the WallSurfaces, the RoofSurfaces and the GroundSurfaces only. To label the non-WallSurfaces in the reconstructed buildings as either RoofSurface or GroundSurface, the height properties of the WallSurfaces are exploited. For each WallSurface in the building, the height value of the lowest vertex in the region and the standard deviation of the height values of all vertices are calculated. Next, the lowest average height value of the WallSurfaces in the building is selected, and a threshold is created by adding one tenth of the standard deviation to this lowest average height value.

This threshold is used to classify the non-Wall regions. The non-WallSurface regions which have a minimum height that is lower than the threshold are labelled as GroundSurface. Regions which have a minimum height which is higher than the threshold are labelled as Roofsurface. This very simple approach seemed sufficient for most models that are available online. In section 6, the accuracy of this implementation is given.

Finally, the terrain triangles are separated from the GroundSurfaces. To do so, the assumption is made that a RoofSurface is always situated above a GroundSurface. This provides possibilities to separate the terrain from the GroundSurface, by checking if a GroundSurface is alligned under a RoofSurface. If a triangle, which is part of a GroundSurface region, is not alligned under a RoofSurface, it will be labelled as terrain. A further in depth explanation is given in section 5.7.

4.7 SEMANTIC CLASSIFICATION: THE COMPREHENSIVE APPROACH

This second implementation is an explorative approach to label a 3D city model with, additionally, the classes OuterCeilingSurface and OuterFloorSurface. This semantic labelling is merely based on the definitions of the different semantic classes.

In the CityGML standard however, the semantic classes are insufficiently defined. For example, a RoofSurface is defined as: "The major roof parts of a building or building part are expressed by the class RoofSurface." (OGC, 2012, p.72). While a GroundSurface is defined as: "The ground plate of a building or building part is modelled by the class GroundSurface. The polygon defining the ground plate is congruent with the building's footprint. However, the surface normal of the ground plate is pointing downwards" (OGC, 2012, p.70).

These not so strict definitions give producers of 3D city models a lot of freedom in the creation of the model. This, however, also has a downside, because it leaves much freedom in the interpretation of the modelling rules. Or, as described by Biljecki et al. (2016), who faces the same difficulties in the specification of LoD in (OGC, 2012): "from a geometric point of view, the five LODs are insufficient", and "their specification is ambiguous." The same applies to the definitions of the semantic classes, because, as mentioned in section 3.2, the classification of 3D city models from other sources does not allow labelling solely based on the normals of the surfaces. Therefore, the definitions in CityGML must be extended. These new definitions are defined in the coming section in which the methodology for the semantic labelling is defined.

This methodology is based on a number of definitions, statements, and arguments. In the Internet Encyclopedia of Philosophy (2016), a statement is defined as: "a declarative sentence, or part of a sentence, that is capable of having a truth-value, such as being true or false". These statements are necessary to make classification rules, because the CityGML standard does not provide enough rules to infer automated classification. Therefore, heuristic rules are used to come to the statements.

Out of these different statements, arguments are formulated. An argument is a series of statements linked by logical inferences (Internet Encyclopedia of Philosophy, 2016), or as explained in Oxford Dictionaries (2015): "a reason or set of reasons given in support of an idea, action or theory". A definition is a statement of the exact meaning of a word (Oxford Dictionaries, 2015). To come to a logic that allows the semantic labelling of the surfaces, first a set of definitions is given. These are needed to create unambiguous statements and arguments.

4.7.1 Definitions

First, before the statements that are used in the classification process can be introduced, the following terms need to be defined:

- An object, denoted X, is a feature in the 3D euclidean space denoted R^3. An object can represent a part of, or an entire building. Therefore, an object can be defined as: a data construct that provides a description of anything present in the scene.

- A region, denoted R, is a set of triangles that share similar geometric properties. A single region always represents one semantic class.

- A building is a set of connected regions, and comprises at least a RoofSurface, a GroundSurface and one WallSurface.

4.7.2 Statements

Next, a set of statements is given, from which a number of arguments are formulated. The statements are given in the coming section, introduced by a description, the origin and the reasoning behind the statement. In the introduction of the statements, a distinction is made, wherein the semantic classes are discussed separately in order to give a better overview. The statements are based on the definitions of the semantic classes in the CityGML standard and related works, or are logically derived from these or other definitions or heuristic rules. This section elaborates on these statements per semantic class.

WALLSURFACES The CityGML standard states that: "All parts of the building facade belonging to the outer building shell can be modelled by the class WallSurface" (OGC, 2012, p. 71). A Facade is in Oxford Dictionaries (2015) defined as: "The principal front of a building, that faces on to a street or open space". Therefore it can be stated that a WallSurface is the exterior, lateral boundary surface of a building and a vertical construction, as proposed in (SIG3D, 2015).

Statement 1: A WallSurface is an exterior, lateral boundary surface of a building and is always a vertical construction.

SIG3D (2015) states that the normals of the wall surfaces should generally lie in the horizontal, up to 45 degrees (SIG3D, 2015). This definition limits the possibilities in the proposed classification algorithm, where the WallSurfaces function as a central class to later classify other classes. Therefore the threshold used by Boeters et al. (2015) and Biljecki and Arroyo Ohori (2015) is used, which states that the angles of the normals of the wall surfaces should generally be between 0 and 5 degrees. Section 5.4.2 further elaborates on the calculation of this threshold. This threshold is used in the creation of two sets: WallSurfaces and non-WallSurface, which is used in the region growing, section 4.2.

Statement 2: The pitch angle of triangle, based on the normal vector, of the WallSurfaces should lie in the horizontal, up to 5 degrees.

ROOFSURFACE The third statement comes from merging the descriptions of SIG3D (2015) and OGC (2012). The truth of the statement is confirmed by the definition of a roof in Oxford Dictionaries (2015), which states that a roof is: "the structure forming the upper covering of a building or vehicle". Meaning that all surfaces coloured red in Figure 4.10 can only be a RoofSurface. This statement implies that a surface which closes a building from above is always a RoofSurface, also when it only encloses a smaller part of the building.

Statement 3: A RoofSurface is the upper boundary surface of a Building: a RoofSurface encloses a building from above.

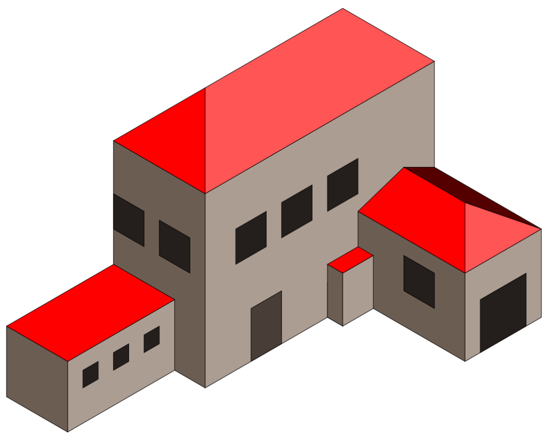


Figure 4.10: Nn example of a building where roof surfaces are coloured red.

GROUNDSURFACE Statement 4 originates from the definition of a GroundSurface in (OGC, 2012) and (SIG3D, 2015). This rule implies that the GroundSurface of a builing is congruent with the buildings footprint and functions as the foundation of the building. Figure 4.11 gives some examples of possible GroundSurfaces.

Statement 4: A GroundSurface is the lower boundary surface of a Building which encloses the building from below. The GroundSurface functions as the ground plate or foundation of the building and is congruent with the buildings footprint: a GroundSurface encloses the building from below

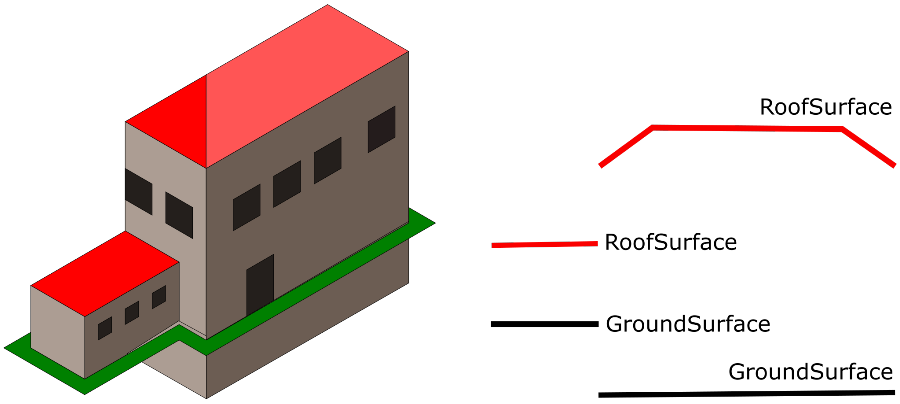


Figure 4.11: GroundSurfaces of a Building.

OUTERCEILINGSURFACE AND OUTERFLOORSURFACE Statements 5 and 6 are a unification of the definitions for a OuterFloorSurface and a OuterCeilingSurface in (OGC, 2012) and (SIG3D, 2015).

Statement 5: A OuterFloorSurface is an upper boundary surface of a Building, which is not a roof.

Statement 6: A OuterCeilingSurface is the lower boundary surface of a Building, which is not a floor.

These two statements imply that a OuterFloorSurface and a OuterCeilingSurface only can occur when they are not a RoofSurface, which is the upper part of a building which closes a (part of a) building from above. Or are not a GroundSurface, which encloses the building from below and functions as the foundation of the building.

4.7.3 Arguments

An argument is a series of statements linked by logical inferences (Internet Encyclopedia of Philosophy, 2016), based on the described statements. An argument must comprise with, and may not contradict with the statements. This section gives the arguments that, together with the statements, function as classification rules.

ARGUMENT 1: **A RoofSurface cannot be alligned under, or above another RoofSurface.** A RoofSuface always encloses a building from above (Statement 3). Therefore a RoofSurface cannot be alligned under or above another RoofSurface. Because in that case, one of the overlapping RoofSurfaces would no longer be the upper boundary or covering surface, which is contradictory to statement 3.

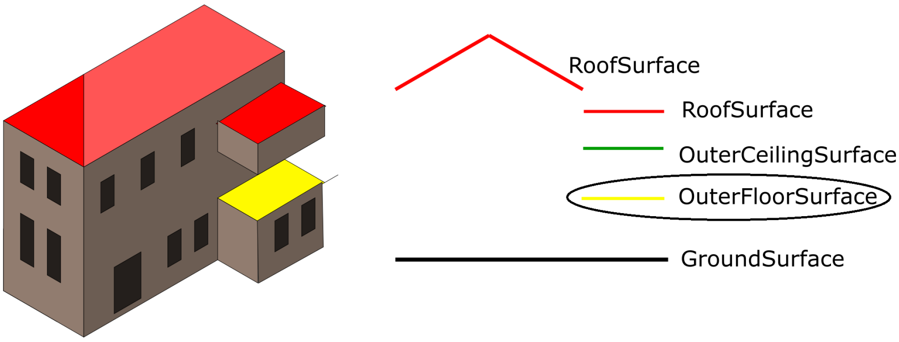


Figure 4.12: A WallSurface, OuterCeilingSurfaces and OuterFloorSurfaces are always situated between the GroundSurface and a RoofSurface.

ARGUMENT 2: **A GroundSurface cannot be alligned under another GroundSurface.** A GroundSurface cannot be alligned under, or above another GroundSurface, because a GroundSurface is always the lower boundary surface of a Building (Statement 4). Therefore a GroundSurface cannot be alligned under or above another GroundSurface. Because in that case, one of the overlapping GroundSurface would no longer be the lower boundary or ground plate of the building, which is contradictory to statement 4.

ARGUMENT 3: **A WallSurface, OuterCeilingSurfaces and OuterFloorSurfaces are always situated between the GroundSurface and a RoofSurface.** The RoofSurface encloses a building from above, while a GroundSurface functions as the ground plate of the building. Therefore, the surfaces that represent a WallSurface, OuterCeilingSurface or a OuterFloorSurface cannot be a upper or lower boundary surface and will always be situated between the GroundSurface and the RoofSurface. Figure 4.12 visualizes this argument.

ARGUMENT 4: **An OuterFloorSurface can only be present when a OuterCeilingSurface is above this surface.** If there is no OuterCeilingSurface, a OuterFloorSurface cannot be present. Because without a OuterCeilingSurface, a possible OuterFloorSurface will be a RoofSurface, because it will be the upper closing surface of a part of the building. In Figure 4.13, a Roofsurface is encircled. This surface will be a OuterFloorSurface, which is encircled in Figure 4.12, if a OuterCeilingSurface is situated above it. This argument is contradictory with examples given by SIG3D (2015), where a number of examples are given. These examples are visualised in Figure 4.14. In the scope of this research, with the goal to automatically enrich a 3D city model, such surfaces will be labelled as RoofSurface, because both examples enclose a part of the building from above.

ARGUMENT 5: **A surface can only represent one semantic class.** A surface can only represent one semantic class, because of the following reasons: it cannot be a WallSurface and one of the other classes at the same time; this is prevented by the threshold. A RoofSurface cannot be a RoofSurface and a GroundSurface at the same time, because it cannot be the upper and lower boundary surface. Last, it can only be a OuterCeilingSurface or an OuterRoofSurface when a surface is not a RoofSurface or a GroundSurface. Thereby, it cannot be a OuterCeilingSurface and a OuterRoofSurface at the

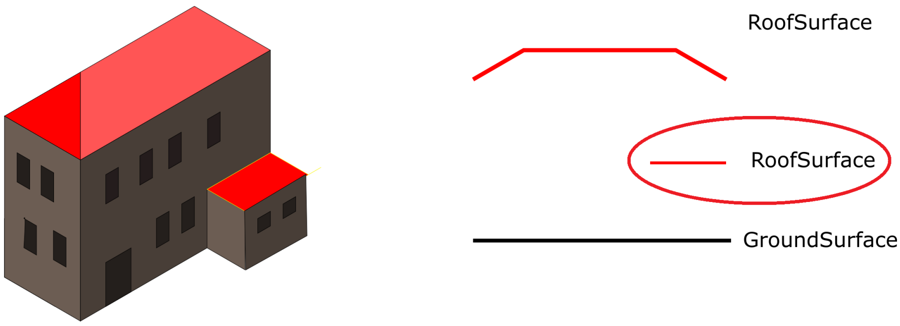


Figure 4.13: An example of a RoofSuface, which can contradict with examples in (SIG3D, 2015).

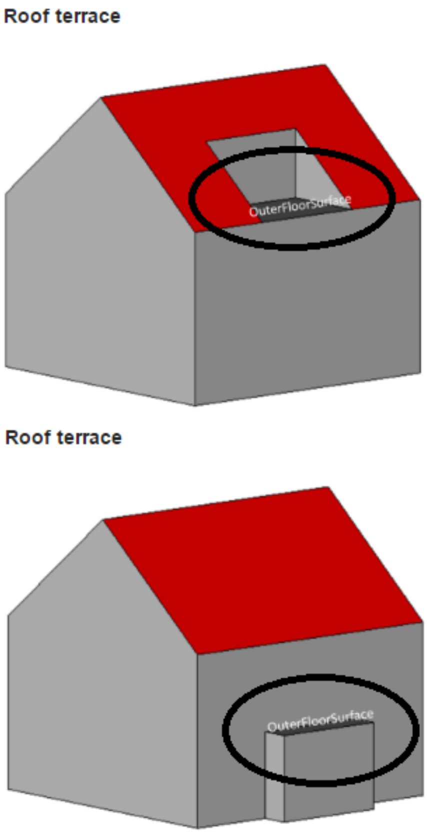


Figure 4.14: Source: (SIG3D, 2015)

same time, because a surface cannot be the upper and lower boundary at the same time.

4.8 DECISION TREE CLASSIFICATION

The above described geometrical properties of the semantic classes, in relation to the other classes in a building, are used to classify the surfaces. This classification is fully based on classification decisions which are derived from the statements and arguments that are just explained. These decisions are embedded in a decision tree. Decision tree learning is a widely used and practical method that works best for classification problems with conclusive and decisive classes (Mitchell, 1997). The classification problem in this research satisfies this condition, as the classes are well defined and explicit. A decision tree classifies instances by sorting these instances down a tree, where the end node, a leaf, assigns a semantic class (Mitchell, 1997).

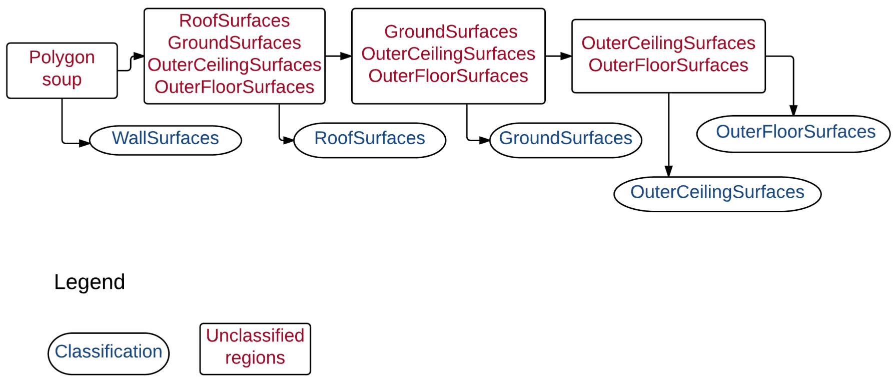


Figure 4.15: Decision tree of our method.

The proposed decision tree is visualised in Figure 4.15, which shows the order on which the semantic classes are recognized in the classification process. The decision tree takes the single buildings, as explained in section 4.3, as input and classifies the regions in this building by a set of rules. These rules are derived from the statements and arguments just explained.

In the process, a classification of one single region automatically classifies all triangles within that region. This section describes the order on which the regions are classified, this order is of major importance and is inspired by the work of Pu et al. (2006) who used this technique in point cloud classification, described in section 2.4.2.

4.8.1 Step 1: the classification of the WallSurfaces

The order, which is presented in Figure 4.15, recognizes the different semantic classes step by step. The first class that is recognised is the WallSurface, which is the central class in this classification process. The WallSurface is the class that is most easily detectable, because the WallSurface is the vertical structure of a building. The Walls are already recognised in the region growing algorithm and include all triangles of which the pitch angle of the normal is between -5 and 5. This is further discussed in section 5.4.2.

4.8.2 Step 2: the classification of the RoofSurfaces

The WallSurface of a Building is always situated between the RoofSurface and a GroundSurface (Argument 3). This argument is exploited to recognise the RoofSurface, which is the highest non-WallSurface of the building (Statement 3). Next, other roof surfaces will be recognised by exploiting argument 1, which states that RoofSurfaces cannot overlap each other. To do so, the second highest region in the building will be checked for overlap with the just recognised RoofSurfac, which is the highest surface of the building. The algorithm keeps on checking for RoofSurfaces, until a non-RoofSurface is recognised. If a non-RoofSurface is recognised, the algorithm starts to look for another semantic class: the GroundSurface.

4.8.3 Step 3: the classification of the GroundSurfaces

The method to find the GroundSurface is similar to the method to find the RoofSurface. The difference is that for finding the GroundSurface the lowest region of the building is detected, exploiting Statement 4. Next, other GroundSurfaces will be recognised by exploiting argument 2, which states that GroundSurfaces cannot overlap each other. To do so, the second lowest region in the building will be checked for overlap with the just recognised GroundSurface. The algorithm keeps on checking for GroundSurfaces, until a non-GroundSurface is recognised.

4.8.4 Step 4: Distinguishing OuterCeilingSurface and the OuterFloorSurface

In this fourth step, the still unclassified regions are labelled. Here fore, the algorithm selects the next highest region iteratively, working its way down. The possible classes for the remaining unclassified triangles are reduced to three in this step. Although the algorithm already aimed for identifying the RoofSurfaces, some RoofSurfaces can still be in the collection of unclassified triangles. This is visualized in Figure 3, in which the different steps are visualised. This image shows that after identifying the RoofSurfaces in step 2, there can still be RoofSurfaces left. Therefore, before the algorithm checks if a surface is a OuterCeilingSurface or a OuterFloorSurface, it will first check if it is not dealing with another RoofSurface or GroundSurface.

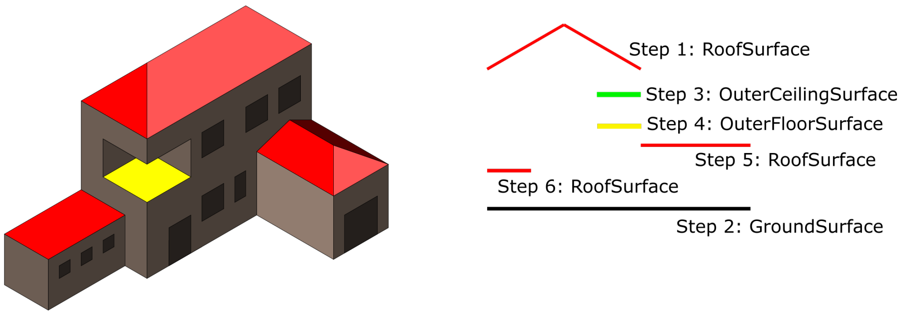


Figure 4.16: Order on which the regions are assigned a class.

If the Surface is not a RoofSurface, it can only be a OuterCeilingSurface or a OuterFloorSurface. To distinguish between these two classes, first the OuterCeilingSurface is ought to be recognised because a OuterFloorSurface cannot be present without a OuterCeilingSurface being situated above it (Argument 4). By taking the highest region, a OuterCeilingSurface can be detected. When a surface is detected which is alligned under a OuterCeilingSurface, it will be classified as OuterFloorSurface.

4.8.5 Step 5: Filter out terrain

The last step is the separation of the terrain triangles from the GroundSurface triangles, which occur if the model is stored with a terrainSurface. To do so, the assumption is made that a RoofSurface is always situated above a

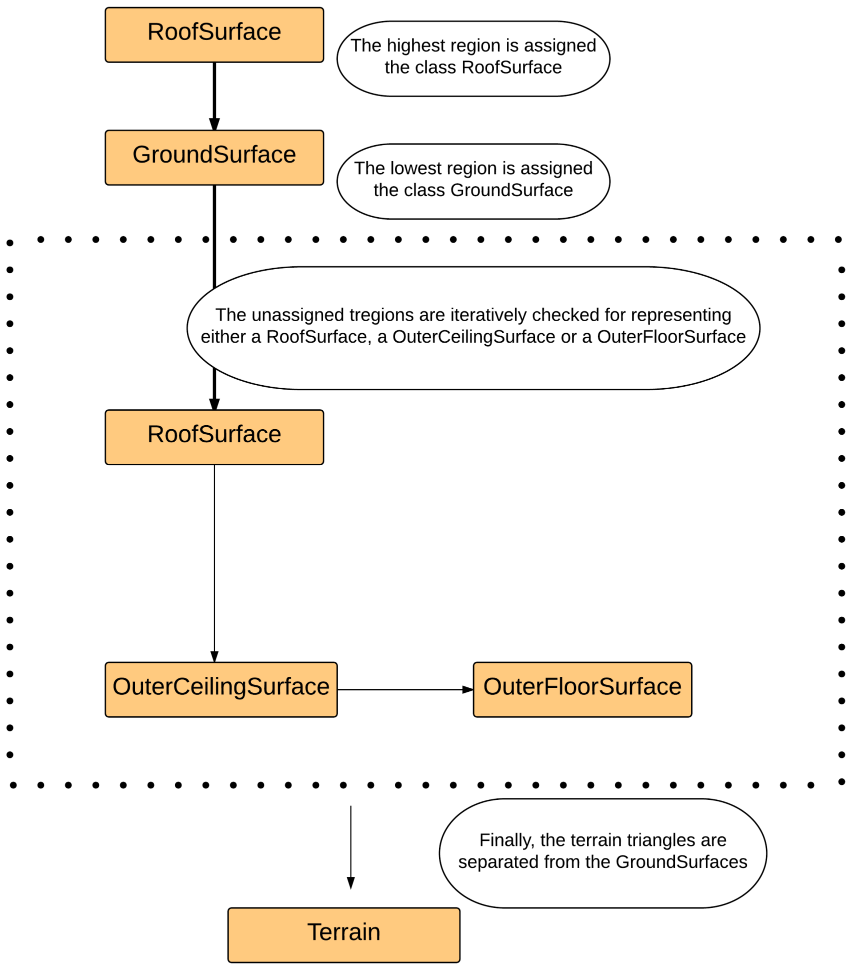


Figure 4.17: Schema of the order on which the regions are assigned a semantic class

GroundSurface, providing possibilities to separate the TerrainSurface from the GroundSurface by checking if a GroundSurface is alligned under a RoofSurface. If a triangle which is part of a GroundSurface region is not alligned under a RoofSurface, it will be labelled as terrain.

The different steps of the workflow are depicted in Figure 4.17.

4.9 DISTINGUISHING BETWEEN CELLARS AND OUTERCEILINGSURFACES

CityGML also allows the modelling of cellars. Cellars are situated below the terrain and should be assigned the class GroundSurface, causing possible height differences in the ground plate of a building. This creates another difficulty in the distinction between a GroundSurface and an OuterCeilingSurface. This problem is visualised in Figure 4.18 in which the already proposed methodology should correctly recognise both GroundSurfaces. While in Figure 4.19, the proposed methodology will classify the OuterCeilingSurface as GroundSurface, because it does not overlap a GroundSurface.

To catch one of these exceptions, two solutions are proposed.

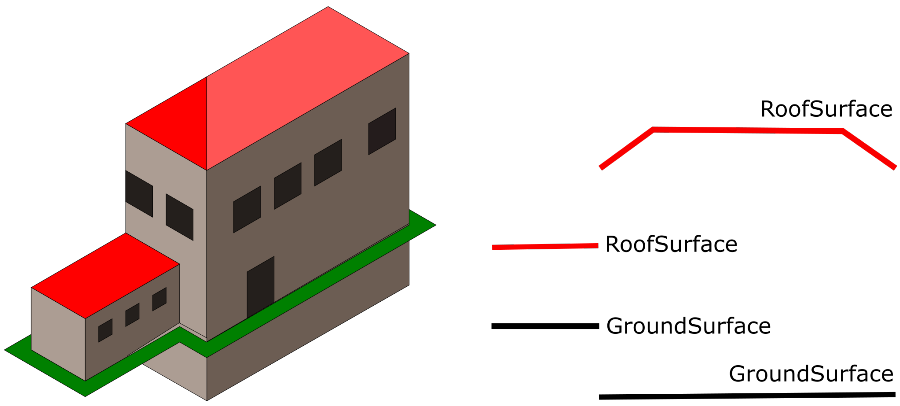


Figure 4.18: GroundSurface.

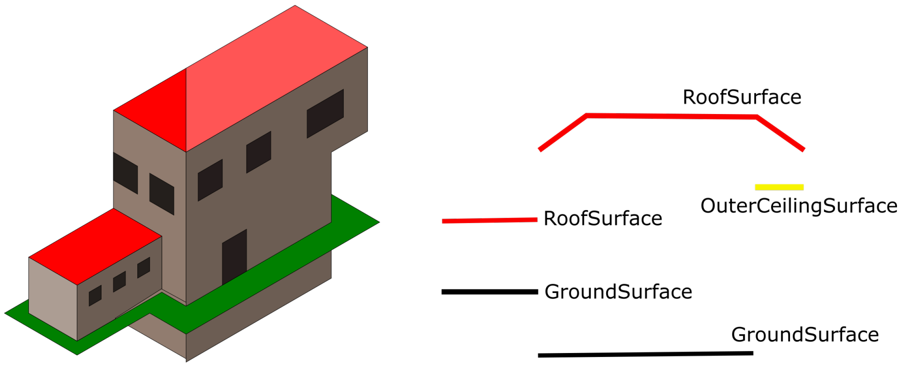


Figure 4.19: OuterCeilingSurface.

First, by assuming that a 3D city model does not hold any cellars and has one single flat ground plate, which allows that the lowest region of the building is the only present GroundSurface. In other words, only the lowest region will be assigned the class GroundSurface. All regions above this lowest region will not be assigned the class GroundSurface.

Second, when a terrain surface or information about the terrain height is present, the original proposed methodology can be used with additional a threshold that is set by the terrain height. This terrain height will determine if the OuterCeilingSurface in Figure 4.19 will be classified as OuterCeilingsurface or GroundSurface. Because if this particular surface is situated above the terrain, it cannot be a GroundSurface. When it is situated below the terrain, it can only be a GroundSurface.

4.10 LOD DETECTION

The LoD of the CityGML file the algorithm should return is dependent on the LoD of the 3D city model that functions as input. In order to detect the LoD of the 3D city model, multiple properties of the buildings can be used. These all require the detection of semantic classes.

In the first place, the normal of the triangles in the RoofSurfaces is checked, as used by Biljecki and Arroyo Ohori (2015). The normals of the RoofSurfaces in a 3D city model with LoD1 should all point in the same direction, while in a model with LoD2 this is not necessarily the case.

Additionally, the presence of the semantic classes OuterCeiling and OuterFloor can be used to distinguish between LoD1 and LoD2. Because a model with LoD 1 can not hold one of these two classes.

5 IMPLEMENTATION

This chapter describes the implementation of the just proposed methodology. This chapter follows the order of the workflow, which is visualised in Figure 4.1. This chapter starts with a short explanation about the decisions that are made in this experimental design:

First, in section 4.9, it is explained that the implementation of the methodology depends on the presence of cellars and/or knowledge about the height of the surrounding terrain. In order to test the proposed methodology, the assumption is made that the 3D city model does not hold any cellars and that the height of the terrain is unknown. This implies that the GroundSurface of a building is formed by one single ground plate which is represented by one single region.

Second, one of the research goals is to detect the LoD of the city model, in order to generate CityGML. The implementation however, allows a user to define the LoD of the CityGML file that is to be generated. If no LoD is given, the algorithm automatically detects the LoD. Other parameters the user can define are explained in the following section.

5.1 INPUT PARAMETERS

The implementation takes four user defined input parameters. These parameters are:

- **The city model** The 3D city model that is processed.
- **LoD** The LoD of the 3D city model, which defines the LoD of the CityGML file the algorithm generates. If the LoD is not specified by the user, the algorithm automatically detects the LoD.
- **Orientation** The orientation of the 3D city model, which states which axis defines the height in the model. This is necessary, because in some 3D city models, the height is defined by the Y-axis, while in others it is defined by the Z-axis.
- **Presence of a terrain** The presence of a terrain is a limiting factor in the reconstruction of the building entities. This is explained in section 3. Therefore, the presence of a terrain is defined by the user. The methods to detect the terrain are presented in section 4.3.

5.2 INPUT DATA

The labelling process takes a triangulated polygon mesh as input. Therefore, the triangle is the main component of the algorithm. Working with triangles has some advantages: the shapes simplicity allows simple and unambiguous computations. Thereby, most semantically unlabelled models come as polygon meshes. In Figure 4.2, an image of a triangulated mesh is given.

5.2.1 Wavefronts object format

The algorithm, that will be developed, will use Wavefront object files as input. This data format is used to store and exchange geometric objects, composed of lines, polygons, and free-form curves and surfaces. Next to geometry, colours and texture can also be stored in the object format (Wikipedia, 2015e). In practice, many 3D city models are stored in object file format and the format has been used in GIS applications (Biljecki and Arroyo Ohori, 2015). Because only faces (i.e. triangles) and vertices will be used, only these data types will be described. Faces are formed by a set of points. These points are connected in the order they are stored, forming lines. The faces are created by connecting these lines, as shown below.

Simple Wavefront file

```
v 0.0 0.0 0.0
v 0.0 1.0 0.0
v 1.0 0.0 0.0
f 1 2 3
```

5.3 CONSTRUCTING THE SPATIAL INDEX

The next step involves the construction of a spatial index. The construction of the index is discussed in depth in section 4.1. This section elaborates on the tools that are used to construct the spatial index.

5.3.1 Scipy spatial kd-tree

The kd-tree as a spatial indexing tool is explained in section 2.7.1. This section provides a short overview of the working of the Scipy spatial (SciPy, 2015) module only, because it plays a major role in this research. The scipy spatial module (SciPy, 2015), takes al individual vertex coordinates (x, y and z) as input and returns two matrices. The first matrix holds all distances from point x to its k nearest neighbours. The second matrix holds all index numbers of its k nearest neighbours. The index numbers of the points in the input matrix refer to the index numbers of the output.

5.4 REGION GROWING OF TRIANGLES

The second step in the thematic and semantic labelling process is the process of region growing. These regions individually represent a semantic class and are composed out of a set of neighbouring triangles with a similar orientation. This orientation threshold functions to distinguish between the WallSurfaces and the non-WallSurfaces. To do so, first the surface normal is computed.

```
[[ 0.  0.  1.]          [[  0 477 572]
 [ 0.  1.  1.]           [  1 286 963]
 [ 0.  1.  1.]           [  2 863 348]
 ...,                    ...,
 [ 0.  1.  1.]           [997 147 495]
 [ 0.  0.  1.]           [260 998 504]
 [ 0.  0.  1.]]          [442 999 605]]
```

Figure 5.1: Return of the Scipy spatial (SciPy, 2015) module kd-tree. Left the distance to the k nearest neighbours, right the index number of the k (3) nearest neighbours

5.4.1 Surface normals

A surface normal is a line or a vector that is perpendicular to a given object. In the three-dimensional case a normal to a surface is a vector that is perpendicular to the tangent plane to that surface (Wikipedia, 2015d) as depicted in figure 5.2.

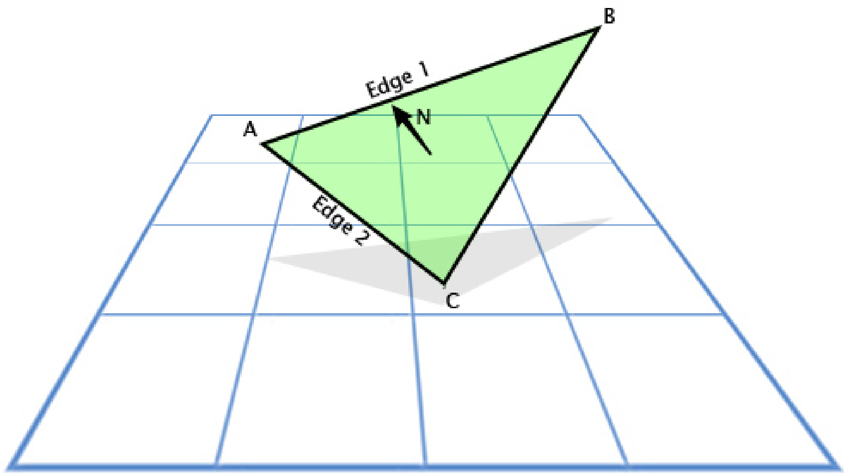


Figure 5.2: Normal of a triangle, Source: blog.wolfire.com

The normals are computed in the following way (Rust, 2015). If $P_1 = (x_1, y_1, z_1)$ and $P_2 = (x_2, y_2, z_2)$ and $P_3 = (x_3, y_3, z_3)$ form the triangle. The normal vector, to the triangle with these three points as its vertices, is given by the cross product $N = (P_2 - P_1) \times (P_3 - P_1)$. The matrix form is worked out in 5.1, which is equal to 5.2.

$$N \begin{bmatrix} A_X \\ A_Y \\ A_Z \end{bmatrix} = \left(\begin{bmatrix} i & j & k \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \end{bmatrix} \right) \tag{5.1}$$

$$N \begin{bmatrix} A_X \\ A_Y \\ A_Z \end{bmatrix} = \begin{pmatrix} (y_2 - y_1)(z_3 - z_1) - (y_3 - y_1)(z_2 - z_1) \\ (z_2 - z_1)(x_3 - x_1) - (x_2 - x_1)(z_3 - z_1) \\ (x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1) \end{pmatrix} \tag{5.2}$$

Resulting in the normal vector N. Next, the normal vector is normalised. This normalisation is worked out in equation 5.3

$$
\begin{aligned}
D &= \sqrt{(A_X * A_X) + (A_Y * A_Y) + (A_Z * A_Z)} \\
N_x &= A_X / D \\
N_y &= A_Y / D \\
N_z &= A_Z / D
\end{aligned}
\tag{5.3}
$$

Whereby N_x, N_y and N_z is the normalized vector and D is the magnitude or length.

5.4.2 The pitch angle

The characteristic on which the WallSurface triangles and Non-WallSurface triangles are selected is the pitch angle of the triangle with the z-axis of the reference frame. The angle of the normalized surface normal with the z axis, θ, corresponding to a pitch motion, is computed with equation 5.4, as used by Boeters et al. (2015).

$$\theta = \arcsin \frac{N_z}{D} = \arcsin \left(\frac{N_z}{\sqrt{N_x^2 + N_y^2 + N_z^2}} \right) \tag{5.4}$$

Next, two sets are created. One with WallSurface triangles and one with non-WallSurface triangles. Both sets function as input for a region growing algorithm as explained in algorithm 4.1 and are processed individually. A region therefore can only consist of triangles which represent a WallSurface or triangles which do not represent a WallSurface.

5.5 CREATION OF THEMATIC FEATURES

After the regions are grown, the next step involves the matching of the regions which contain WallSurfaces with the adjacent regions which contain the Non-Wallsurfaces. This way, the single building features are reconstructed.

The matching of the surfaces is based on the stored neighbour relationship that is computed during the creation of the index. These adjacency relationships were computed for all single triangles. These relationships are copied to the region instances, which contain all the neighbouring triangles of all the triangles in that particular region. The actual matching is done through comparing the neighbours of the WallSurface region with the triangles in the non-WallSurface regions and the other way around. Next, to match the regions instead of the individual triangles, a temporary table is created which stores the relationship between a triangle and the region which it is part of. This is explained in depth in section 4.3.

5.6 semantic classification

As explained at the beginning of this chapter, two approaches are tested in this implementation. The reconstructed buildings, which are created in the previous step, function as input for both approaches. These single building entities are processed one by one, meaning that the semantic class of a surface of a building is independent from class assignments in other building entities.

Section 4.4 describes two different approaches to label the non-WallSurfaces. The classification in the two class approach is solely based on setting a threshold, which represents a maximum GroundSurface height, and which than classifies the non-WallSurfaces by checking if they are situated below or above this threshold. The whole approach is described in detail in section 4.6 and is therefore not further explained in this section.

The semantic labelling methodology in the comprehensive approach, section 4.7, is based on the mutual relationship of the different non-WallSurface regions in one single building entity. These building entities at least contain one region with WallSurfaces and two or more regions with non-WallSurfaces.

The methodology in this experimental design is based on the assumption that the 3D city model does not hold any cellars and that the height of the terrain is unknown. This is already mentioned at the beginning of this chapter and discussed in section 4.9.

In the process, the highest non-WallSurface region is automatically assigned the class RoofSurface, while the lowest region is automatically assigned the class GroundSurface. Next, as visualised in Figure 4.17, the unassigned regions are iteratively assigned a class. This class assignment is based on an exclusion process for the different classes, as described in section 4.8.

This exclusion is solely based on the spatial relationship of the triangles in the region that gets assigned a semantic class, with the triangles in the regions that are already assigned a class. More concrete, as explained in sections 4.7.3 and 4.8, the class assignment is based on an overlap relationship between the regions that gets assigned a semantic class and regions that are already assigned a semantic class. This is further explained by using the example in Figure 5.3. In this example, the highest region is selected and classified as a RoofSurface. The lowest region is classified as a GroundSurface. Next, the third surface will be assigned a semantic class. This third surface can only represent a RoofSurface or a OuterCeilingSurface, based on the assumptions made at the beginning of this chapter. To determine which class is assigned to this region, an overlap relationship between this surface and the other surfaces in the building has to be checked for. To do so, the triangles are converted from 3D to 2D by ignoring the height dimension in the processing.

Next, a overlap relationship, which is either True of False, is defined for every triangle in the region that gets classified, with every triangle in the region that is already assigned a semantic class. In the example in Figure 5.3, this means that every triangle in the OuterCeilingSurface is checked for its spatial relationship with every triangle in the RoofSurface.

Three possible relationships are possible: disjoint, meet and overlap. The 'disjoint' relationship includes all two triangles that have no touching or overlapping relationship. The 'meet'relationship includes all relationships between two triangles that either share a point or an edge, or where the point of one triangle touches an edge of the other triangle. The 'overlap' relationship includes all cases where one triangle invades the space of the other triangle. Some examples are given in Figure 5.5.

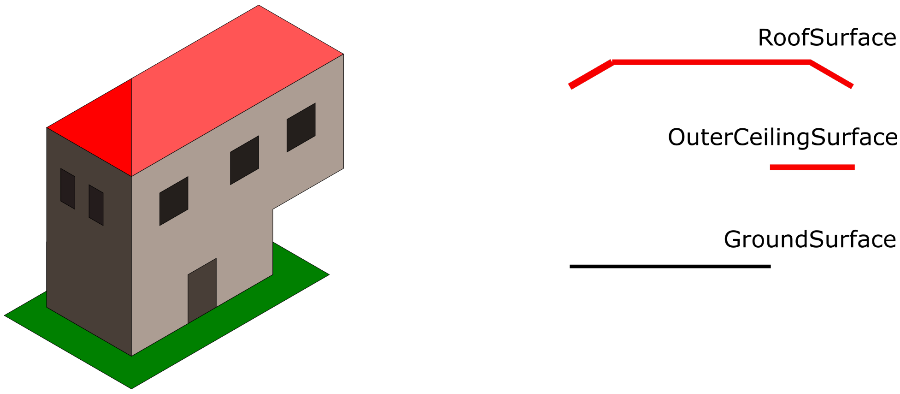


Figure 5.3: Semantic classification, example one

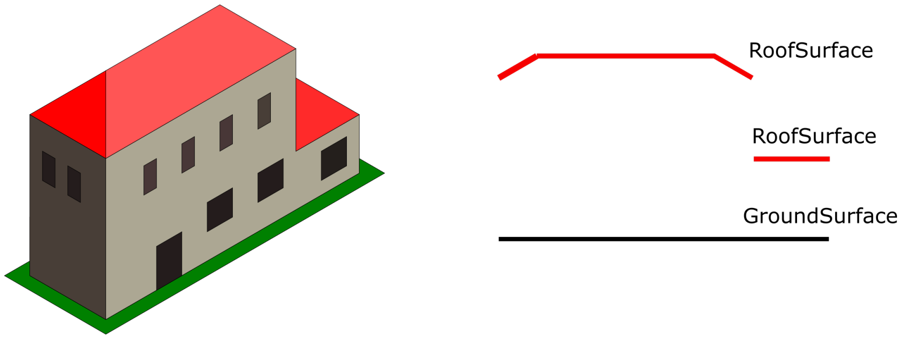


Figure 5.4: Semantic classification, example two

To determine if two regions do not overlap, none of the checked spatial relationships between the triangles of the two regions can be overlap. If one of the triangles does overlap with one of the triangles in the other region, the relationship between the regions is set as overlapping. In the example in Figure 5.3, the OuterCeilingSurface has an overlap relationship with the RoofSurface, and is therefore classified as OuterCeilingSurface. While in Figure 5.4, the RoofSurface does not overlap with the other RoofSurface, and is therefore classified as RoofSurface. The classification rules for all semantic classes are given in section 4.8. The Python package shapely is used to compute the spatial relationships between the triangles.

5.7 DIFFERENTIATING BETWEEN THE TERRAIN AND GROUNDSURFACE

In the last step of classification process, the terrain, if present, is separated from the GroundSurfaces. This method is used in implementation one and two. This is necessary, because the triangles which form the terrain are in most cases part of a region which also contains GroundSurfaces. To distinguish between these two classes, the assumption is made that a GroundSurface triangle is always aligned under a RoofSurface region. Here fore, every triangle which is part of GroundSurface region is processed individually, in

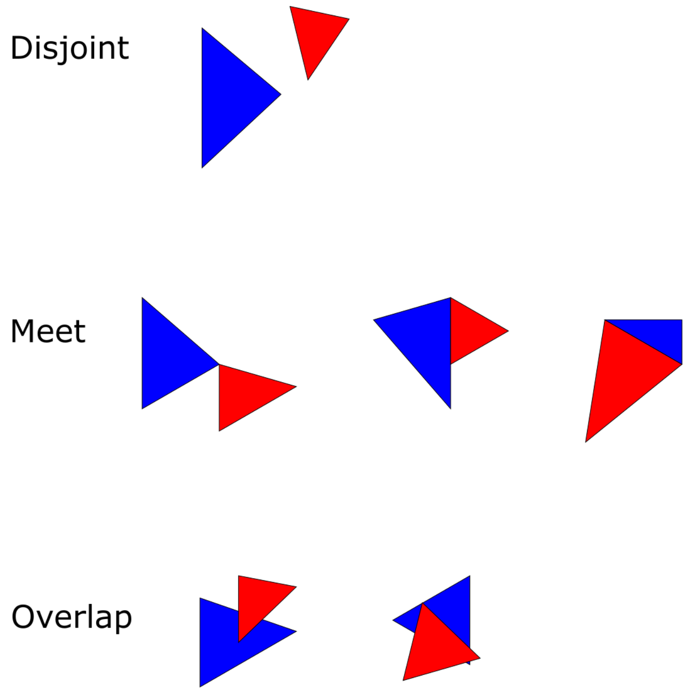


Figure 5.5: Examples of the 3 categories of topological triangle relationships

which every vertex of the GroundSurface triangle is checked on whether it is not disjoint with all RoofSurface triangles in that region. The relationship between the vertex and the triangle is disjoint if the vertex does not intersect at all with the triangle or its boundary. In other words, the GroundSurface will be labelled as terrain if one of its vertices is disjoint with all triangles which form the RoofSurface.

This method however, is very sensitive with regards to floating point precision errors. Therefore, disjoint relationships where detected in cases where this is incorrect. To catch these errors, every vertex that is disjoint from a triangle is additionally checked for being on the line of that particular triangle. This checking must be done in a way that leaves room for floating point precision errors. This method is explained in the following paragraph.

POINT ON LINE Every triangle consists out of three vertices A, B and C. In order to check if vertex P is on one of the lines AB, BC or CA, all three lines have to be processed individually. Figure 5.6 gives an example of a case where point P is checked to be on line AB or not. The method is given in Algorithm 5.1. The distance function is this algorithm is given in Equation 5.5.

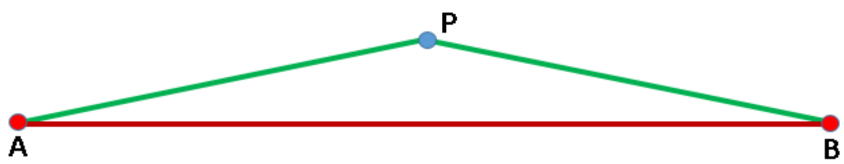


Figure 5.6: Example to check whether point $P1$ is on line AB

Algorithm 5.1: POINT ON LINE ALGORITHM

Input: Line AB, defined by point A and B. Point P
Output: True if point P is on line AB, False if not

if *Distance(A,P) + Distance(B,P) < Distance(A,B) * 1.02 and Distance(A,P) + Distance(B,P) > Distance(A,B) * 0.98* **then**
  Return True
else
  Return False

$$Distance(P1, P2) = \sqrt{(P1x - P2x)^2 + (P1y - P2y)^2} \quad (5.5)$$

5.8 OUTPUT IN CITYGML

After assigning a class to every triangle, the CityGML file is created. In CityGML, every building is stored as a separate <cityObjectMember> and gets assigned a unique ID: <gml:id="buildingID">. For models with LoD 2, every triangle which is part of a region, or a MultiSurface in CityGML, is given a semantic class: <bldg:"GroundSurface">. Every triangle which is part of this surface is assigned a unique ID: <gml:Polygon gml:id="PolygonID">". The code that is used to create a CityGML file with LoD 2 is given below in section 5.8.1, in which the above described id's and the geometry is coloured red.

Generating CityGML models with LoD 1 is slightly different, and is given section 5.8.2. The difference between the two, is that the surfaces in models with LoD 2 are given a semantic class. As already explained in chapter 4.7.3; CityGML defines the orientation of the surface by its normal. The semantic labelling therefore allows the correction of the normals of the surfaces in the model, while generating the CityGML file.

5.8.1 Generating CityGML with level of detail 2

```
<cityObjectMember>
  <bldg:Building gml:id="buildingID">
   <bldg:boundedBy>
    <bldg:"GroundSurface">
     <bldg:lod2MultiSurface>
      <gml:MultiSurface>
       <gml:surfaceMember>
        <gml:Polygon gml:id="PolygonID">
         <gml:exterior>
          <gml:LinearRing>
           <gml:posList>
           Vertex1 Vertex2 Vertex3
           </gml:posList>
          </gml:LinearRing>
         </gml:exterior>
        </gml:Polygon>
       </gml:surfaceMember>
```

```
          </gml:MultiSurface>
          <gml:MultiSurface>
            <gml:surfaceMember>
              ...

              </gml:surfaceMember>
          </gml:MultiSurface>
        </bldg:lod2MultiSurface>
      </bldg:”GroundSurface”>
    </bldg:boundedBy>
    <bldg:boundedBy>
  <bldg:”WallSurface”>
  ...

  </bldg:Building>
</cityObjectMember>
```

5.8.2 Generating CityGML with level of detail 1

```
<cityObjectMember>
  <bldg:Building gml:id=”buildingID”>
    <bldg:lod1MultiSurface>
      <gml:MultiSurface>
        <gml:surfaceMember>
          <gml:Polygon gml:id=”PolygonID”>
            <gml:exterior>
              <gml:LinearRing>
                <gml:posList>
                Vertex1 Vertex2 Vertex3
                </gml:posList>
              </gml:LinearRing>
            </gml:exterior>
          </gml:Polygon>
          <gml:Polygon gml:id=”PolygonID”>
            <gml:exterior>
            ...

          </gml:surfaceMember>
      </gml:MultiSurface>
    <bldg:lod1MultiSurface>
  </bldg:Building>
 </cityObjectMember>
```

The CityGML file that is generated does not hold a terrain.

6 RESULTS AND ANALYSIS

This chapter gives and reflects on the results of the implemented methodologies. To test these, different 3D city models are used.

6.1 TEST DATASETS

The 3D city models that are being used in this research are described in this section. The models are freely accessible online, and are selected because they are stored in CityGML and therefore already contain semantics. These models are converted to Wavefronts object format, where all information, except the vertices and faces, is deleted. Creating a polygon mesh, or a soup of polygons that is used to test the proposed classification methods. After the classification, the original semantic information from the CityGML file is used te evaluate the accuracy of the developed methods.

WALDBRÜCKE DATASET The first dataset that will be used, is a 3D model of a small village in Waldbruecke in Germany (Figure 6.1). The model holds small houses and apartment buildings in LoD 1 and LoD 2 and does not hold any complex structures. The terrain of the model is completely flat. The dataset is freely accessible on the CityGML website (CityGML, 2016).

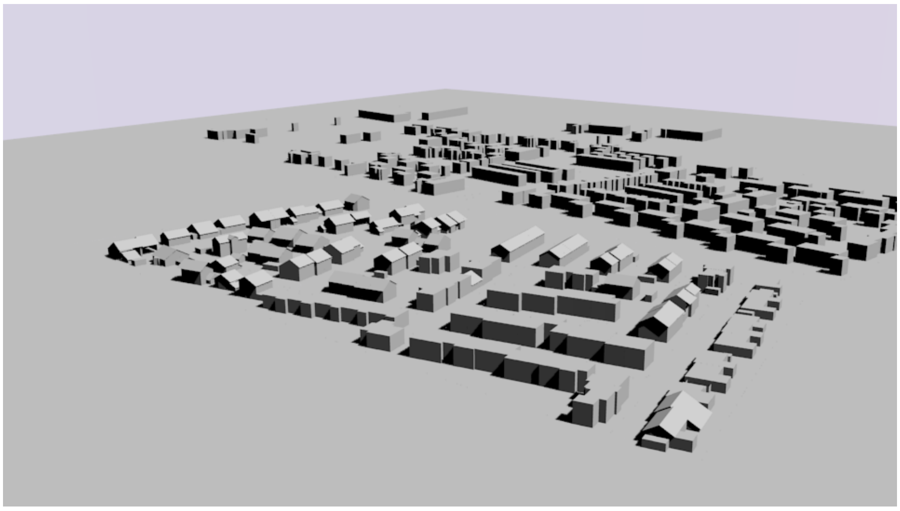


Figure 6.1: Waldbruecke dataset

ROTTERDAM The second dataset is a model of the neighbourhood Kleinpolder in Rotterdam (Figure 6.2). The model holds houses and apartment buildings in LoD 2. The model only holds buildings and has no terrain. The dataset is freely accessible on the website of the municipality of Rotterdam (Rotterdam municipality, 2015).

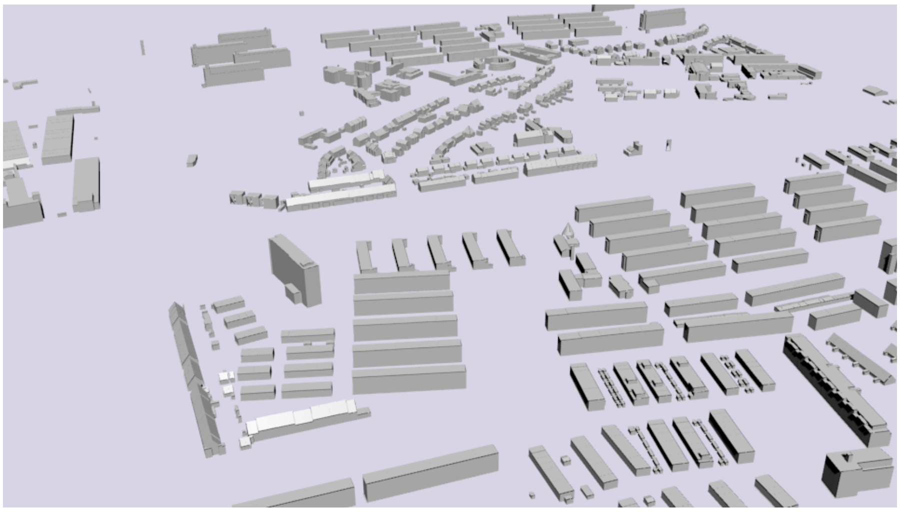


Figure 6.2: Rotterdam dataset

SWITZERLAND LV 95 The Switzerland LV 95 model is a model without a terrain and only holds buildings with LoD 1 and 2. The houses in the model are situated in the mountains, what distinguishes this model from the other models. The dataset is a test product from of the Swiss topographic office and is freely accessible on their website (Swisstopo, 2015).

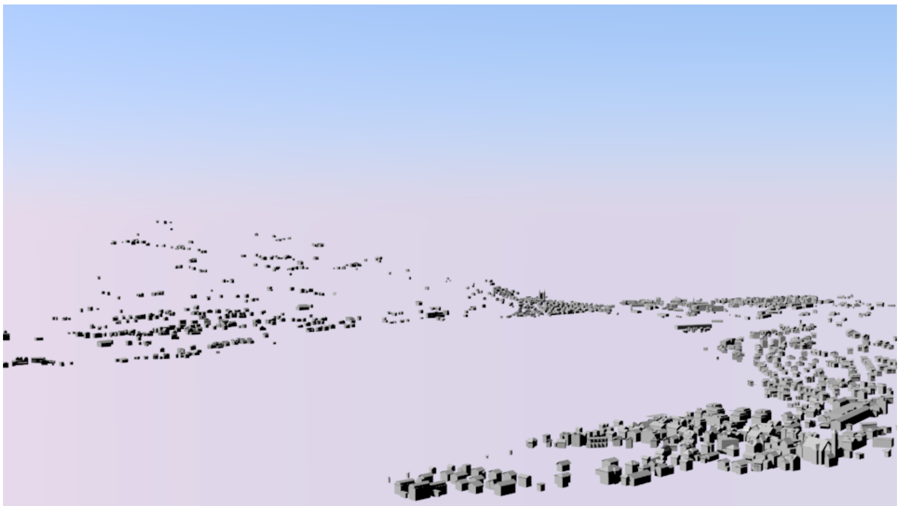


Figure 6.3: Switzerland dataset

NEW YORK The fourth dataset (Figure 6.4) is a model with LoD1 of Manhatten, New York. The dataset is free to download on the website of the University of Munich (Kolbe et al., 2015). The dataset only holds a road network as terrain polygons. The models contains low and high buildings, which makes it an interesting case for testing the proposed methodology.

MONTREAL The dataset from the city of Montreal is freely available on the website of City of Montreal (2015). The dataset only contains buildings and is of LoD 2.

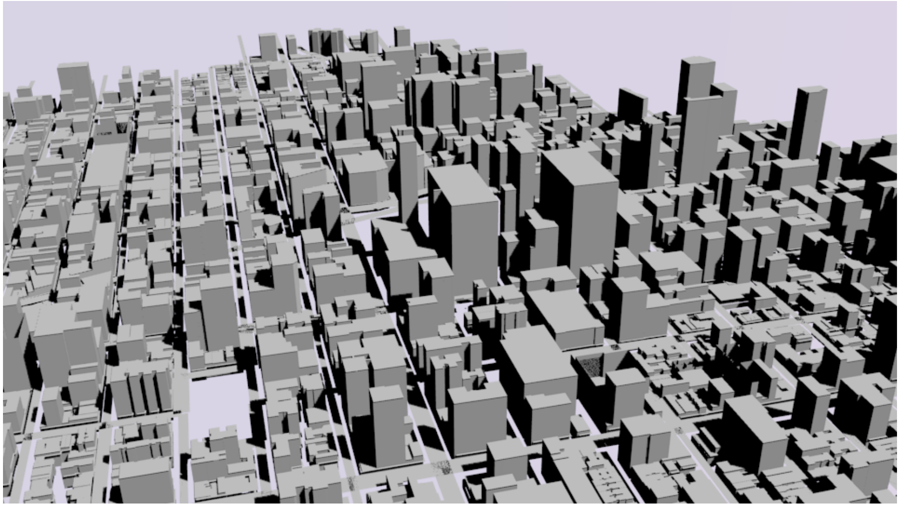


Figure 6.4: New York Dataset

TEST DATASETS CREATED WITH ARC CITYENGINE Different test datasets are created with CityEngine, a program that is developed to create 3D city models. These models are used to test the comprehensive approach, because no models could be found that contain the semantic classes OuterCeilingSurface and OuterFloorSurface.

6.2 ACCURACY ASSESSMENT

The following sections elaborate on the accuracy of the labelling process. The accuracy is assessed by comparing the semantic class of the original CityGML dataset with the class of the classification algorithm. It is important to notice that the original dataset can also contain errors.

In this section, first the thematic labelling and the detection of the LoD is given. The limitations of the methods, which are logically derived from the accuracy measures, are given in the next section 6.6. Therefore, the limitations are not discussed in this section.

6.2.1 Thematic labelling

Table 6.1 shows that the number of detected buildings in the labelling process deviates from the number of buildings in the original datasets. This is caused by different building entities which are connected by one or more vertices or edges. In section 3.5, the lack of definition of the thematic entities in the CityGML standard is discussed. With this discussion in mind, the number of buildings that are detected by the labelling algorithm cannot be considered as necessarily wrong. Please note that the part of the algorithm which recomposes the buildings is the same for both thematic labelling approaches. Therefore they are not discussed separately.

Table 6.2 gives the number of classified and unclassified triangles. Triangles which are not classified are mainly part of a region which do not represent a wall. These regions are not matched to a wall region, therefore, these regions are not part of a building and are left unclassified causing unclassified triangles being an effect of an error in the thematic aggregation.

Table 6.1: 3D city models used for testing the performance of the method.

Name	Size (Buildings)	LoD	Reconstructed buildings
Rotterdam	1544	2	507
Montreal	384	2	191
Switzerland	3151	2	2218
Waldbruecke	606	2	273
New York	approx 1M	1	276
CityEngine 1	6	2	6
CityEngine 2	7	2	7

Table 6.2: Classification matrix: the total number of triangles, the number of classified and the number of unclassified triangles

	triangles in the model	classified	unclassified
Rotterdam	57581	57533	48
Montreal	57581	57578	3
Switzerland	135389	135227	162
Waldbrucke	12157	12152	5
New York	103552	103556	0
CityEngine 1	204	204	0
CityEngine 2	276	276	0

Visual inspection of the models, like the New York model, as described in sections 6.4.5 and 6.4.3, also shows that some buildings have missing parts.

The detection of the terrain is visually checked for accuracy. Most models which hold a terrain, like the New York and the Waldbrucke model, have their terrain stored as, for example, a road network. Therefore, these measures are not shown in the different tables. The labelling is checked for accuracy nonetheless. In all tested models, the terrain, which represent all non-building features, are labelled correct.

6.2.2 LoD detection

Table 6.1 shows the LoDs that are detected. All the detected LoDs are correct.

6.3 SEMANTIC LABELLING

The following section describe the results and the accuracy of the semantic classification. In this section, a short description of the overall results are given. The following two sections describe the accuracy in more detail, distinguishing the two class approach and the comprehensive approach.

The results of the two class approach for the Switzerland, the Montreal and the Rotterdam dataset with the two class approach are around 99%. Some errors occur because of missing neighbour relationships, leading to unclassified triangles as discussed in the previous section.

The results of the comprehensive approach differ. For the models that are made in CityEngine, the results are a 100% or 97% accurate. Other models made in CityEngine, which are no described in this report, where made on a different scale in CityEngine and showed less good result because of more prevalent floating point precision errors. This is further discussed in the next section: 6.6. Also the Rotterdam dataset is classified with the

Table 6.3: Classification accuracy Rotterdam dataset in percentages

	WallSurfaces	RoofSurfaces	GroundSurfaces
WallSurfaces	57.3	0.2	0
RoofSurfaces	0	24.7	0
GroundSurfaces	0	0	17.8

Table 6.4: Classification accuracy Montreal dataset in percentages

	WallSurfaces	RoofSurfaces	GroundSurfaces
WallSurfaces	52.5	0	0
RoofSurfaces	0	8.5	0.2
GroundSurfaces	0	0	38.8

methods in the comprehensive approach. The classification on this dataset is less accurate than the results from the two class approach, which can be explained by the higher sensitivity to float precision errors.

6.4 ACCURACY ASSESSMENT OF THE TWO CLASS APPROACH

6.4.1 Rotterdam dataset

In this paragraph, the results of the labelling process of the Rotterdam dataset are given. The total number of buildings in the dataset is 1544, while the labelling process detected 507 buildings. The total number of unclassified triangle is 48. Inspection of the model revealed that these 48 triangles are part of a single polygon, which is not connected to any other polygons. Therefore, this surface is not identified as a (part of a) building and therefore ignored by in the labelling process. Table 6.3 shows the omissions and commissions. The table shows that 88 surfaces are labelled as a WallSurface, while these surfaces should be labelled as RoofSurface. This discrepancy can be explained by the orientation of these surfaces, because these surfaces represent beams which support the roof. Therefore these surfaces are not oriented upwards, but fall within the threshold that identifies the RoofSurfaces. Overall, the 99.7 % of the triangles are given the correct label, with a kappa coefficient of 0.99.

6.4.2 Montreal dataset

The Montreal model holds 57581 triangles in total. A number of three triangles are unclassified. These triangles appear spread out over the model and are either a WallSurface or a RoofSurface. The accuracy of the semantic labelling is presented in table 6.4, overall the classification accuracy on this model is 99.8 % with a kapa coefficient of 0.99. In the Montreal dataset, 191 buildings are detected. The original CityGML file holds 384 buildings. The labelled dataset is visualised in Figure 6.5.

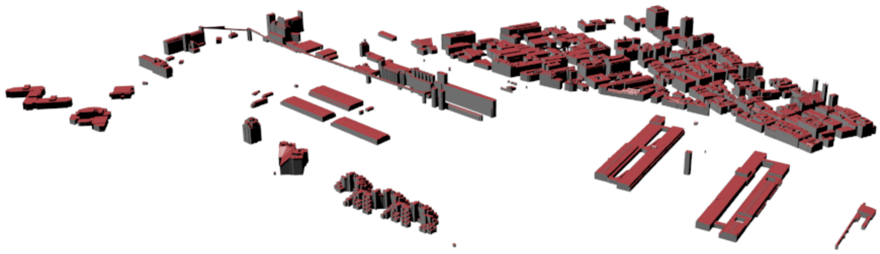


Figure 6.5: Visualisation of the labelled Montreal dataset.

Table 6.5: Classification accuracy Switzerland dataset in percentages

	Wall	Roof	Ground	
WallSurfaces	40.1	0.1	0.002	
RoofSurfaces	0.1	48.4	1.3	
GroundSurfaces	0.008	0	9.96	

6.4.3 Switzerland dataset

The total number of unclassified triangles is 162 out of a total of 135389. These unclassified triangles are not clustered together and do not belong to the same building. Therefore, it seems evident that these triangles are not classified because they, or the region they are part of, are not connected to a set of walls and therefore to a building. This is most likely caused by the higher level of detail of the model, which leads to a lower threshold in the indexing of the model and therefore in the matching of the regions. Table 6.5 shows the accuracy of the semantic labelling. Before this table was created, the triangles which hold the class BuildingInstallation in the original file are filtered out. A number of 2054 BuildingInstallation triangles are labelled as WallSurface and a number of 1431 triangles are labelled as RoofSurface. The class BuildingInstallation is discussed in section 2.3.2, and can either represent RoofSurfaces or WallSurfaces. These triangles are not checked for accuracy any further.

The total number of buildings in the original dataset is 3151, the algorithm detected 2218 buildings. The labelled 3D city model is visualised in Figure 6.6.

6.4.4 Waldbrucke dataset

The original Waldbrucke dataset holds some deviancies. Some buildings in the Waldbrucke model have a GroundSurface, while others don't. This is visualised in Figure 6.7 and Figure 6.8. In which Figure 6.8 shows that the buildings with a sloping roof are not closed, while their WallSurfaces stick through the terrain surface. Thereby, the original CityGML file of Waldbrucke does not hold the right semantic classes and can therefore not be used to validate the labelling process. Therefore, the model is validated through visual inspection.

A total of 5 triangles are unclassified in the labelling of the Waldbrucke data. These triangles are most likely unlabelled because the regions they are part of are not matched to a WallSurface.

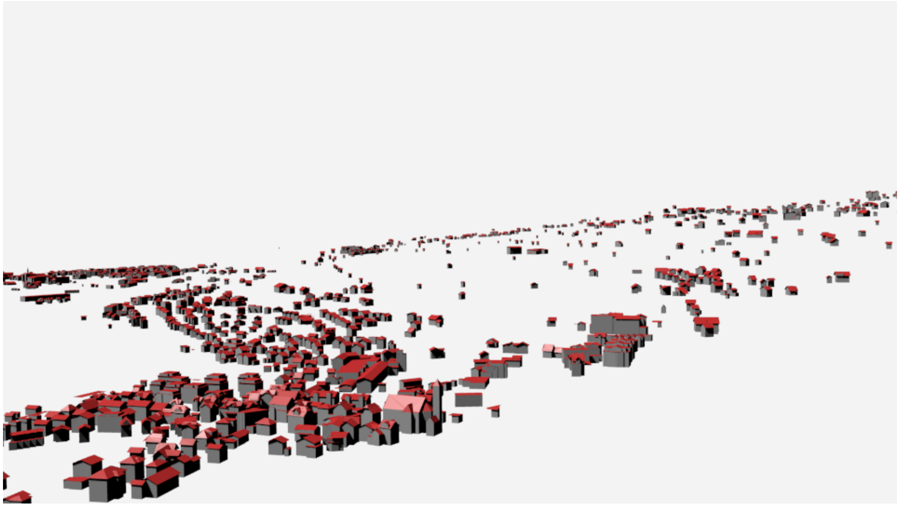


Figure 6.6: Visualisation of the labelled Switzerland dataset.

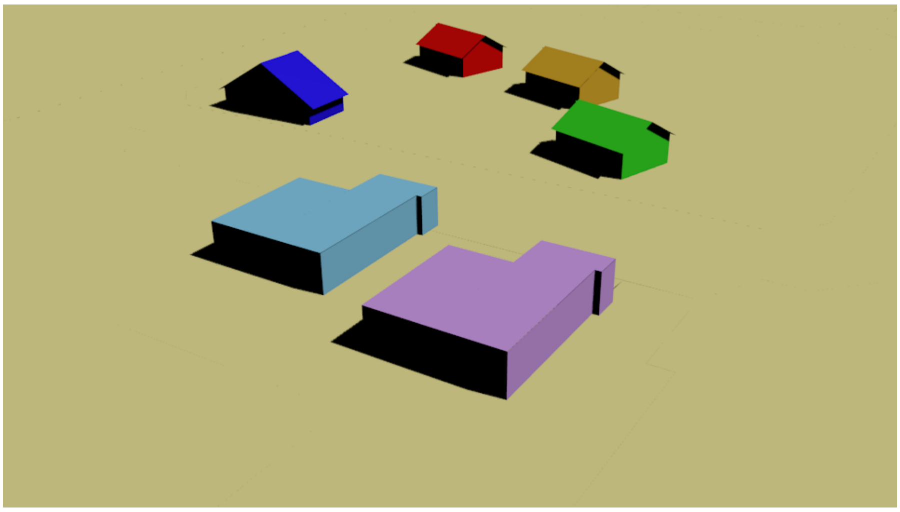


Figure 6.7: Buildings from the Waldbrucke dataset seen from above.

The algorithm detected 273 buildings, the original model holds 606 buildings. Two of these buildings are missing a RoofSurface and are part of the set unlabelled triangles described above. The terrain in this model is one closed surface.

6.4.5 New York dataset

The New York dataset is a 3D city model with LoD 1. The original dataset contains 103552 triangles, the labelled dataset contains 103556 triangles.

For this model, no ground truth for the semantic classification is available, because it is a model with LoD 1. Therefore, the accuracy of the labelling process on this model is only inspected visually. The visual inspection showed that some buildings are missing GroundSurfaces and/or RoofSurfaces. Five of these cases are detected. The terrain surface, which in this model represent a road network and is not connected to any WallSurface, is labelled correct. The number of detected buildings is 276.

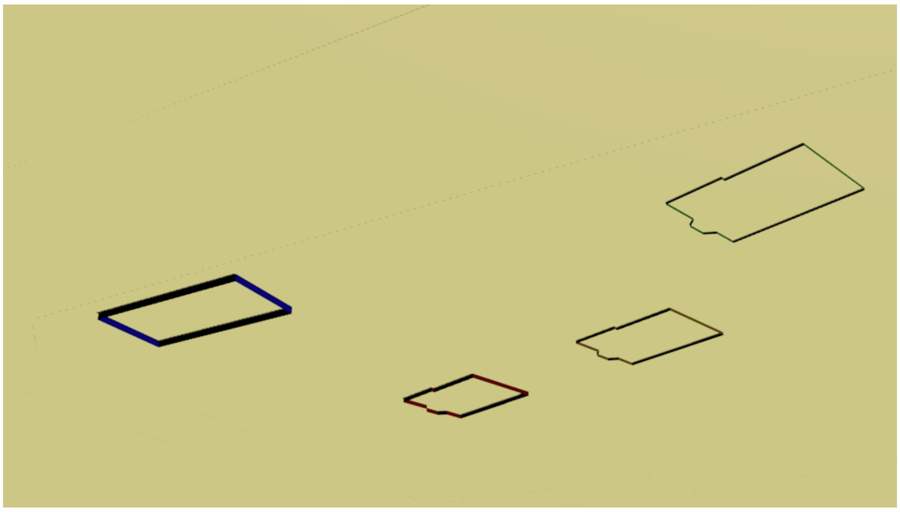


Figure 6.8: Buildings from the Waldbrucke dataset seen from below.

The fact that some buildings are missing a GroundSurface and/or a RoofSurface, while the total number of triangles in the automatically labelled dataset is higher than the number of triangles in the original dataset, points on an error in the reconstruction of the single buildings. This is most likely caused by WallSurfaces that are not correctly matched to non-WallSurfaces. Leading to a single building being stored as multiple buildings, but in parts, while some surfaces of this building appear in multiple parts of this building.

This is most likely caused by floating point precision errors in the data, where errors occur in the part of the algorithm which separates the terrain from the GroundsSurfaces. Or by the matching algorithm which connects the non-WallSurfaces tot the WallSurfaces.

6.5 ACCURACY ASSESSMENT OF THE COMPREHENSIVE APPROACH

This section gives the accuracy assessment of the comprehensive approach. The results are given with the idea to proof that the approach does work, but also to display its current limitations. The results, and especially the methods current limitations, are further discussed in section 6.6. The second approach is tested with a variety of models. The first test models are created in CityEngine to test the methodology. These 3D city models are most suitable for testing, because they contain very little floating point precision errors. These errors highly effect the results of the methodology, which is further elaborated in section 6.6.6.

6.5.1 CityEngine Model 1

The fist model that is used to test the methodology holds 7 buildings, which are composed out of 204 triangles. All triangles are labelled in the labelling process. Figure 6.9 is a visualisation of the labelled 3D city model seen from above, 6.10 is a visualisation of the same model seen from below. These

Table 6.6: Classification accuracy CityEngine model 1 in percentages

	Wall	Roof	Ground	Ceiling	OutRoof
Wall	69.6	0	0	0	0
Roof	0	12.7	0	0	0
Ground	0	2.9	6.9	0	0
Ceiling	0	0	0	6.9	0
OutRoof	0	0	0	0	2.9

images, together with table 6.6, show that all 204 surfaces are assigned the correct semantic class. Also the right number of buildings is detected.

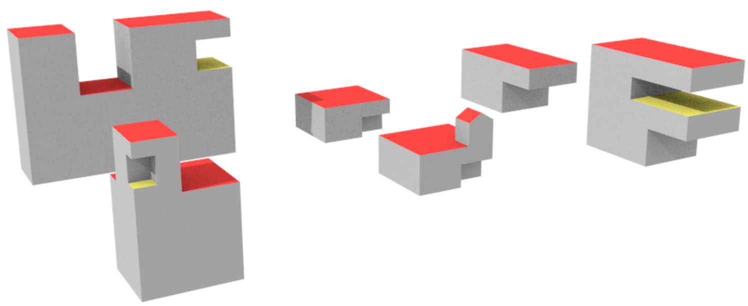


Figure 6.9: Labelled CityEngine model 1, seen from above.

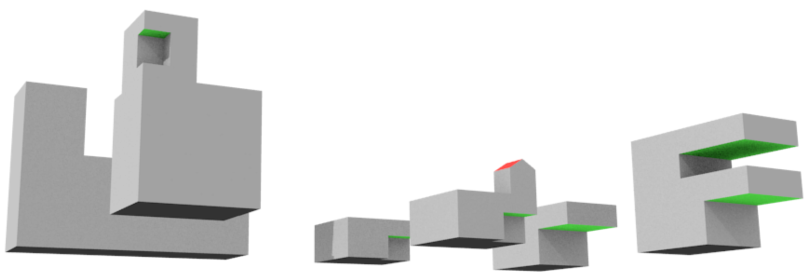


Figure 6.10: Labelled CityEngine model 1, seen from below.

6.5.2 CityEngine Model 2

The second tested model contains seven buildings and is composed out of 276 triangles. In the labelling process, seven buildings where detected,

Table 6.7: Classification accuracy CityEngine model 2 in percentages

	Wall	Roof	Ground	Ceiling	OutFloor
Wall	60.9	0	0	0	0
Roof	0	27.5	0	0	0
Ground	0	2.2	5	0	0
Ceiling	0	0	0	2.2	0
OutFloor	0	0	0	0	2.2

Table 6.8: Classification accuracy Rotterdam in percentages

	Wall	Roof	Ground
Wall	57.3	0.15	0
Roof	0	21.86	0
Ground	0	0	17.7
Ceiling	0	2.8	0
OutFloor	0	0.1	0

which is the correct number. The labelled city model is visualised in Figure 6.11. Table 6.7 gives the assigned labels per class. Overall, the algorithm assigned the correct class to 97 % of the triangles in this model.

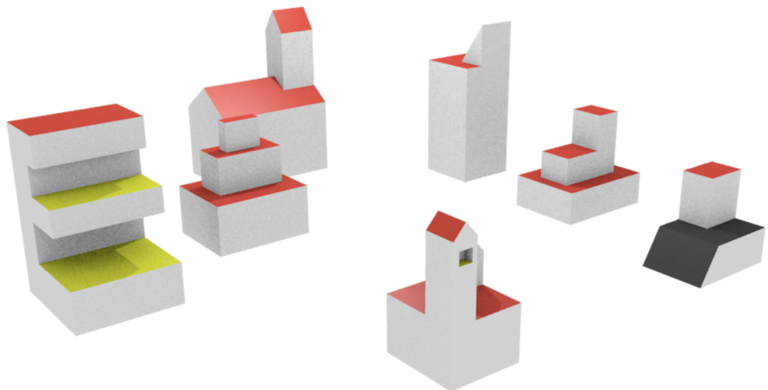


Figure 6.11: Labelled CityEngine model 2.

The errors in the classification, Table 6.7, are discussed in section 6.6.

6.5.3 Rotterdam dataset labelled with the comprehensive approach

The results of the processing of the Rotterdam city model with the method in the comprehensive approach are given in Table 6.8.

The results show that the classification accuracy is considerably lower than in two class approach, which can be found in 6.3. This is caused by floating point precision errors, which will be further discussed in section 6.6

6.6 LIMITATIONS OF THE METHODOLOGY

This section discusses the results and the limitations of the proposed methods. In this discussion, the focus is on the comprehensive approach. Before discussing the limitations into more depth, the following has to be explained.

Various methods have been tested on a wide range of models. During this testing, the researcher realised that there is no one fits all approach. In other words, this methodology cannot be translated in an algorithm which can take any 3D city model from any source as input and output a thematically and semantically labelled 3D city model. This is because of the following reasons:

Presence of other (undefined) objects than buildings or terrain, which makes the labelling process over complex. Currently, the algorithm classifies every surface as a being part of either a building or the terrain. Thereby, objects other than the terrain or a building are not modelled in a consistent way, creating difficulties in the filtering of these objects. An example is given in Figure 6.12. In this figure, one can clearly see that the terrain in the centre of the village is labelled as a RoofSurface, which is coloured red. This is caused by the canal around the village, which has a vertical ridge. This ridge is identified as a WallSurface, causing the algorithm to recognise a building and the labelling of a RoofSurface.

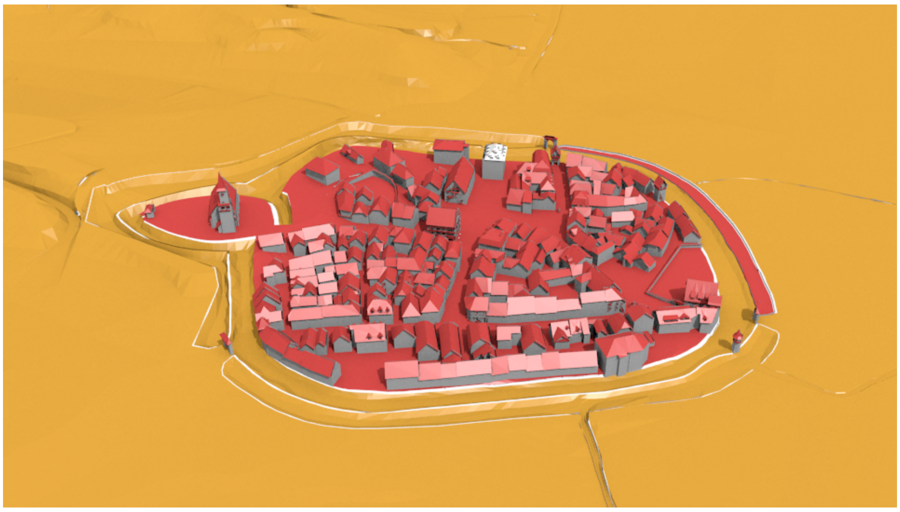


Figure 6.12: Labelled city model of Ettenheim. The model is labelled with the two class approach.

Incomplete buildings and missing surfaces, which are buildings without, for example, a GroundSurface. These missing surfaces make the creation of a set of rules, that can be used in the labelling process, overcomplex.

Incorrect geometries, which is discussed in the expected challenges, section 3.3. Causing surfaces being left unmatched to a building, as discussed in the accuracy assessment, section 6.2.1.

Next, a number of limitations are given. These limitations are further discussed here. In this discussion, the focus is on the comprehensive approach, the first four limitations (sections 6.6.1 until 6.6.4) are applicable to both approaches.

6.6.1 Matching of surfaces and building parts

In some models, the algorithm did not match the different regions correctly. This has already been discussed in section 6.2.1. These missing connections between surfaces are either caused by inaccuracies in the data, as described in section 3.3, or by the method used to construct a topology and a spatial index. These 'missing connections' did not occur in all models, and are therefore most likely caused by inaccuracies in the data.

Thereby, a threshold is used in the construction of the index. This threshold is calculated by randomly selecting edges from the city model. Therefore, the threshold varies per input model and can even vary if one model is processed multiple times. Therefore, the threshold is not universally valid. Future work should focus on the calculation of a universally valid method.

6.6.2 One region can represent multiple classes

Section 4.7.1 states that one single region only can represent one semantic class. The current region growing algorithm however, only distinguishes between triangles which represent a wall and triangles which represent another class. Therefore, in the current implementation, one region can represent multiple classes, which should, if in line with the theory, be subdivided into multiple regions. Figure 6.13 gives an example of two regions which are grown as one region, but should be subdivided in two regions. This building is part of the model in Figure 6.11. This building is the only error in the classification in this particular model.

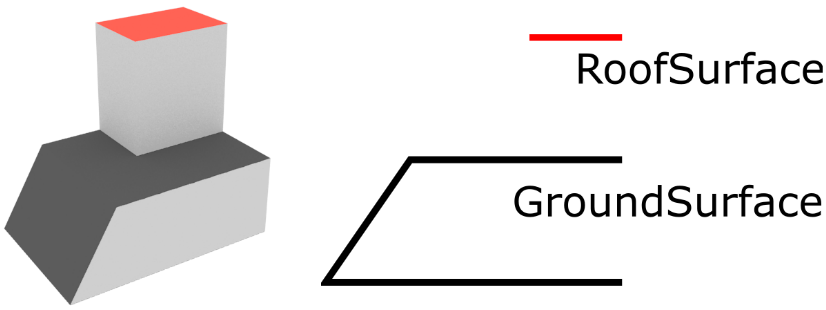


Figure 6.13: Current defect of the region growing algorithm.

6.6.3 Distinguishing between the terrain and the GroundSurface

In both approaches, the terrain is separated from the GroundSurface by making the assumption that the GroundSurface is always aligned under the RoofSurface. Because of the threshold of 5 degrees, that is set to distinguish between the WallSurfaces and the Non-WallSurfaces, a GroundSurface can be be aligned under a WallSurface, while not being aligned under a RoofSurface. Therefore, to distinguish between the GroundSurface and the terrain, the assumption: 'a GroundSurface is always aligned under a RoofSurface', should be extended with WallSurfaces.

6.6.4 Recognition of regions which contain the terrain

In the current implementation, the presence of the terrain is given as user input. The triangles that represent the terrain are recognised by the properties of the region it is part of. In the next step, in which adjacent regions are matched, the recognition of the terrain regions function as input and is used to limit further matching. This way, it is prevented that multiple buildings, which are all connected to a particular terrain region, are recognised as a single building.

The properties on which a region that contains terrain triangles is recognised are the number of WallSurface regions it is connected to, for which a threshold of four is set, and the percentage, 25%, of the total non-WallSurfaces triangles that the region contains (section 4.3). These thresholds however, are set by experimenting on a set of test data sets. These measures proved sufficient in most models, but can differ largely if used on other models. Therefore, it is advised, before implementing this approach, to experiment with the data in order to recompute these thresholds.

The presence of a terrain is therefore a very limiting factor, because the likeliness of false positives during the detection of the terrain becomes bigger if the 3D city model becomes more complex. For example, if the RoofSurface of a building is connected to more than four annexes that are part of the building, the algorithm will detect a terrain, while it is a RoofSurface. Therefore, after testing approaches to automatically detect the terrain, it was decided to let the user define the presence of a terrain.

The following limitations are only applicable to the comprehensive approach.

6.6.5 Methodology only works on models till LoD 2.2

To classify the OuterCeiling and OuterFloorSurfaces, an extension of the definitions of the semantic classes is proposed. This is described in section 4.7. These extensions allow automatic semantic classification. This methodology however, is limited by properties of the 3D city model. These properties are best described by the extended definitions of Biljecki et al. (2016), which, for LoD 2, are visualised in Figure 6.14. In the visualisation of LoD 2.3, one can see that the roofs have an overhang over another RoofSurface. How this is a limiting factor is explained with the use of Figure 6.15.

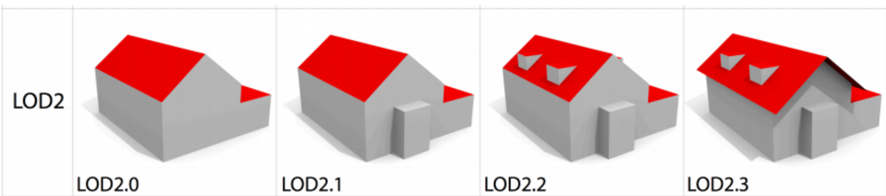


Figure 6.14: Surface labelled as OuterCeilingSurface, Source: (Biljecki et al., 2016)

In Figure 6.15, a RoofSurface is classified as a OuterCeilingSurface because the RoofSurface in this figure overlaps the OuterCeilingSurface. This

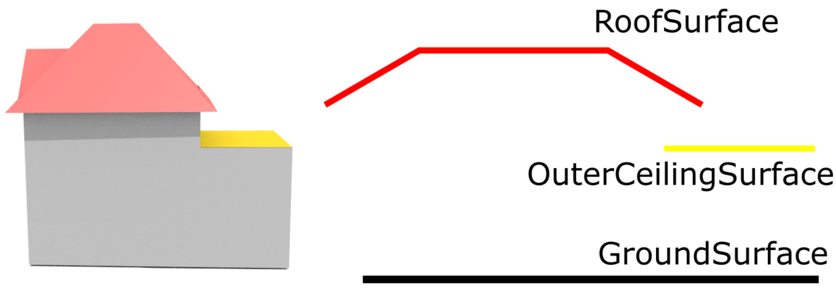


Figure 6.15: Limitation of the methodology, caused by the detail in LoD 2.3. Source: (Swisstopo, 2015)

overlap in the relationship leads to a recognition of a OuterCeilingSurface, while it is in fact a RoofSurface. This is a major limitation for the comprehensive approach.

6.6.6 Precision errors

Most classification errors of the comprehensive approach on the Rotterdam dataset, as given in Table 6.8, occur because of precision errors in the data. These errors do not occur in the two class approach, Table 6.3. These errors occur in the detection of overlap relationships, on which the comprehensive approach is based. These errors occur because of floating point precision errors. The comprehensive approach is therefore much more sensitive to such errors.

6.6.7 Complete and partly overlapping surfaces

In the classification of the different semantic classes, an overlap relationship is used to come to a classification. This overlap relationship however, can only be positive or negative. The classification for a single surface can therefore also be only positive or negative.

This classification method however, does in some cases not comprise entirely with the extended definitions of the semantic classes, which form the basis for the proposed methodology. For a RoofSurface, the extended definition states: "A RoofSurface is the upper boundary surface of a Building: a roof encloses a Building from above". Surfaces which are only partly overlapped by another surface should, according to the definition, be subdivided into two surfaces with both a different class. This is further explained with the example that is given in Figure 6.16.

In Figure 6.16, the OuterCeilingSurface, coloured green, is partly a RoofSurface, because it us the upper boundary of a part of the building which encloses the building from above. This is visualised in the same Figure, below the model. The right example is the same surface, which is subdivided in two surfaces that either represent a RoofSurface and a OuterCeilingSurface. Only after this subdivision, all semantic classes comply with the extended definitions.

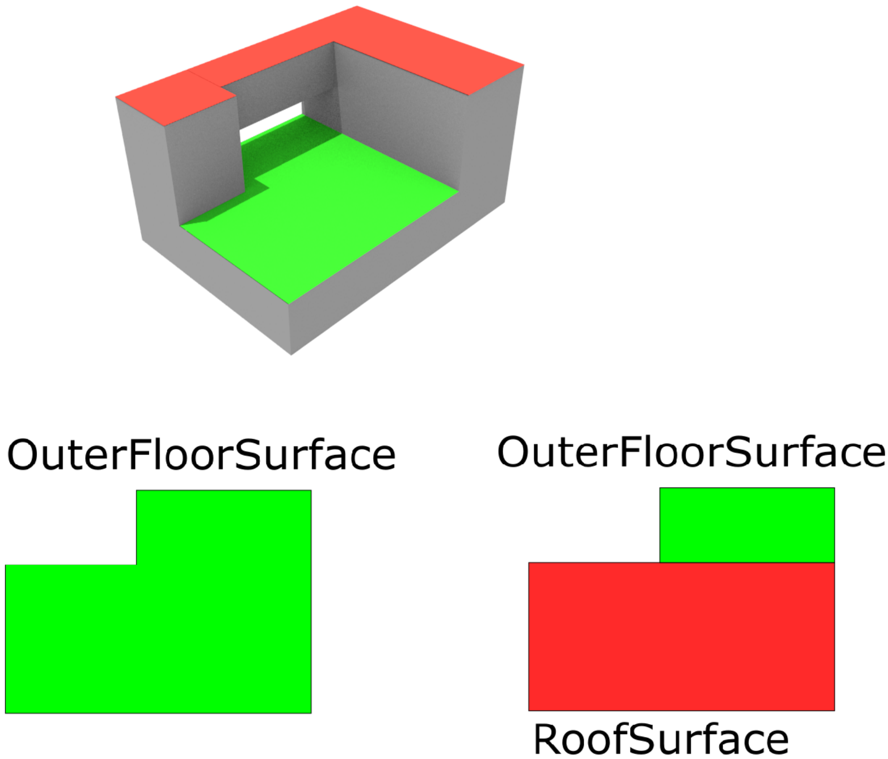


Figure 6.16: An case which demands the subdivision of a surface.

7 CONCLUSIONS

In order to answer the main research question: **"What is needed to automatically enrich a LoD 1 or 2 3D city model with thematic and semantic information as defined in CityGML, by only utilising the models geometry?"**, first the sub questions will be answered.

HOW IS THE LOD OF THE 3D CITY MODEL DETECTABLE? The LoD of a 3D city model is detectable by the orientation of the surface normals of the RoofSurfaces, as already proposed by Biljecki and Arroyo Ohori (2015). Thereby, the presence of the semantic classes OuterCeilingSurface and OuterFloorSurface is proved to be a sufficient parameter in the recognition of semantic classes. These methods however only help in distinguishing between the LoDs 1 and 2. Thereby, if all buildings in a 3D city model with LoD 2 have flat roofs and the buildings do not hold the classes OuterCeilingSurface and OuterFloorSurface, the current implementation will detect the wrong LoD.

WHAT SEMANTIC AND THEMATIC CLASSES CAN BE DISTINGUISHED BY ONLY USING GEOMETRIC PROPERTIES, DEPENDENT ON THE LOD? The semantic classes WallSurface, RoofSurface, GroundSurface, OuterCeilingSurface and OuterFloorSurface can be distinguished by structuring the geometry of the individual buildings in the 3D city model. Here fore, the definitions of the semantic classes in CityGML have to be extended, which is described in the methodology, section 4.7. The following points described these extensions briefly:

First, the assumption is made that WallSurfaces have a pitch angle, which is computer by the triangles normal vector, which is smaller than 5 degrees. Second, it is stated that the GroundSurface encloses the building from below and that it forms the ground plate of a building. Third, the definition of the RoofSurface is extended with the fact that it encloses a building from above.

The used methodology to distinguish between the different semantic classes is as follows. The set threshold for the WallSurface creates two collections, first triangles that fall within this threshold: the WallSurfaces and triangles that don't: the triangles that represent all other classes. These two collections function as input for a region growing algorithm. The regions which represent a WallSurface, are than used to creating single buildings. Next, a topologic structure is used to match all other regions to these Wallsurfaces. Next, the highest and lowest region in the building entity is assigned the classes RoofSurface and GroundSurface. The unclassified regions, which can represent a RoofSurface, GroundSurface, OuterCeilingSurface or a OuterFloorSurface, are distinguishable by computing the overlap relationships between the different surfaces in one single building model. The classes BuildingInstallation and ClosureSurface cannot be distinguished. These classes however, are not obligatory to create a valid CityGML file and are therefore not researched.

In addition to the recognition of these semantic classes, the theme's Building and the terrain can be distinguished. Here fore, the assumption is made that the GroundSurface of a building is aligned under the other parts of a building, e.g. the RoofSurface. This method proved sufficient, but requires user input which defines the presence of the terrain. Further possible differentiation of the terrain surface has not been researched.

How can these geometric properties be used in the classification of the 3D city model? The method used to come to a classification is based on a number of geometric properties. First, the relative height of the surfaces; the highest surface, or region, is automatically assigned the class RoofSurface. The lowest surface is automatically assigned the class GroundSurface. The other surfaces in the building are labelled by computing an overlap relationship between the different surfaces in a single building. In other words, the surfaces are labelled with a semantic class, if the surface is overlapped, or not, by an already classified surface.

Can methods established in remote sensing, e.g. classification of point clouds be used? The main method that is used and established in the processing of remote sensing data is the concept of region growing. This method clusters adjacent triangles together which have a pitch angle that falls within the set threshold. The regions are a central concept in this research. Next to region growing, the kd-tree that is used in the construction of the spatial index, also referred to as the topology, originates from the processing of point clouds that are created in the field of remote sensing. Thereby, the idea used in the semantic labelling of a region, which uses the geometry of surrounding regions, also originates from point cloud processing techniques (section 2.4.2).

How can the thematic features, e.g. buildings, be recomposed? The buildings are recomposed with a methodology that clusters adjacent spatial features together in regions. Next, these regions are matched to its adjacent regions, forming single building entities. In this matching process, the central class is the WallSurface; the WallSurfaces are used to recompose the single buildings. This means that the assumption is made that one single building is composed by a set of triangles which are all connected. The terrain is still a limiting factor in this process, because a terrain is, in most cases, connected to all, or multiple buildings. To prevent the aggregation of multiple buildings into one entity, some thresholds are set. These thresholds are described in section 4.3, the limitations of this method is discussed in 6.6.4.

How accurate is the classification process? The accuracy assessment in chapter 6 showed that an accuracy of 100% is possible. This accuracy does depend on a number of factors, especially for the comprehensive approach. First, the data, because float precision errors are a big limiting factor in the labelling process. This causes errors in the overlap relationship calculation. Also, in some models, some surfaces are not matched to a building entity. This is most likely caused by float precision errors which are present in the data.

Second, to fully comply with the extended definitions of the semantic classes as proposed, some surfaces, regions, should be subdivided, where each part gets assigned a different class. This is discussed in section 6.6.7. These cases are not included in the 100% accuracy that is mentioned at the beginning of this paragraph. The 100% classification accuracy is only achieved in one model.

Now, the research question: **"How to automatically enrich a LoD 1 or 2 3D city model with thematic and semantic information as defined in CityGML, by only utilising the models geometry?"** is answered.

Automatic recognition of thematic features from 3D city models is possible by clustering adjacent triangles together. First, a region growing algorithm is used to recreate surfaces, in which the WallSurfaces are grown independently from all the other classes. Second, the WallSurfaces are used to recreate the single buildings, by utilising adjacency relations of single triangles. These techniques, to recompose the thematic features and to recognise the terrain, depend on a number of thresholds. These threshold do vary and depend on the input data. The results of the thematic labelling showed that the number of re composed buildings in the labelled test datasets do differ from the number of buildings in the original files.

Beacuse of the lack of definition of building entities in CityGML, the re composition cannot be verified by comparing the labelled dataset with a ground truth. In other words, the definition of a single building in CityGML gives the possibility to aggregate one single building and its parts in different ways, which can all be valid.

In this research, two approaches are tested which aim at semantically classifying the surfaces in a building. The first approach, the two class approach, only distinguishes between the semantic classes WallSurface, RoofSurface and GroundSurface and uses to relative height of the surfaces to come to a classification.

The second approach, the comprehensive approach, aims at the automatic enrichment of 3D city models with additional the classes OuterCeilingSurface and OuterFloorSurface. The definitions of these classes however, as defined in the CityGML standard, do not allow automatic semantic labelling. Therefore these definitions must be extended. These extensions are based on heuristic rules. These extensions are: first, the setting of a threshold that defines a maximum slope for WallSurfaces. Second, an extension of the definition for RoofSurfaces, which states that a roof encloses a building from above. Third, extending the definition of a GroundSurface with the rule that it encloses a building from below.

The extensions are used to create a logic which allows the recognition of the semantic classes by their relative height and the interdependent overlap relations of the different surfaces, which are represented by a region, in one single building.

Both approaches are developed with in mind the goals that are described above. The two class approach works best in the labelling of existing 3D city models which do not hold the classes OuterCeilingSurface and OuterFloorSurface. The comprehensive approach additionally recognises these classes too.

The accuracy of the labelling processes highly depends on the input data. For the two class approach, an accuracy around 99% can be achieved. For the comprehensive approach, a 100% accuracy is achieved where the input data does not contain float precision errors and where the buildings in the model are not too complex.

The thematic and semantic classification leverages the usability of the data, as it can be used in an extended range of applications which require the added information.

8 FUTURE WORK

Out of this research, a methodology is proposed, which is based on an extension of the definitions of the semantic classes in CityGML, that allows automatic semantic en thematic enrichment of 3D city models. The complexity of the problem is described in section 3. The challenges described in this section are, to some extend, overcome but not completely solved. Therefore, the challenges described in section 3 can be interpreted as future work.

Next, this chapter describes other challenges that are derived from the limitations described in section 6.6: Limitations .

8.1 EXTENDED DEFINITIONS CITYGML

As described in the conclusion, automatic semantic labelling of 3D city models requires an extension of the definitions of the semantic classes in CityGML. Per semantic class, the extension of the definition is proposed as:

- **WallSurface** A WallSurface encloses a building from above and is congruent with the ground plate of a building.
- **GroundSurface** A GroundSurface encloses a building from below.
- **WallSurface** The pitch angle of the normal vector of a WallSurface triangle cannot exceed 5 degrees.

In this thesis, it is demonstrated that by extending the definitions, automatic semantic labelling is possible by exploiting classification rules that are inferred from a logic that is based on the above described extensions. Thereby, an extra extension of the definition of the GroundSurface should be discussed: SIG3D (2015) advises to model the terrain as a flat plane, OGC (2012) does currently not specify this.

For future work in automatic semantic classification of CityGML, a number or remarks are given in the following paragraphs. These are learned during this research.

THE SEMANTIC CLASS BUILDINGINSTALLATION The semantic class BuildingInstallation in CityGML is defined as: ”The BuildingInstallation class is used for building elements like balconies, chimneys, dormers or outer stairs, strongly affecting the outer appearance of a building” and ”A BuildingInstallation is an outer component of a building which has not the significance of a BuildingPart, but which strongly affects the outer characteristic of the building. Examples are chimneys, stairs, antennas, balconies or attached roofs above stairs and paths” (OGC, 2012, p, 64). This definition is neither inclusive or exclusive, meaning that is does not include or exclude surfaces. Dormers and chimneys, for example, can be stored as one BuildingInstallation as well as RoofSurfaces and WallSurfaces. This creates ambiguity between models, where in one model dormers and chimneys are classified as BuildingInstallation, while in others these surfaces have a different class.

Therefore the following extension or description for the class BuildingInstallation can be discussed, to overcome this ambiguity: "A BuildingInstallation is a part of the building which volume does not add up to the volume of a building". This means that buildingparts which are attached to, but do not enclose the internal space of a building are classified as BuildingInstallation. This definition does not allow dormers or chimneys to be classified as BuildingInstallation, but includes all external parts that are not part of the main structure.

DEFINING THE SEMANTIC CLASS OPENINGS IN LOD 3 MODELS The format CityGML can, until now, be used and processed in a limited number of software packages. Boeters et al. (2015) demonstrated that the processing, or the adding of information to the dataset can be done by first converting the data to another format and later transform the data back to CityGML. The surface normal proofed to be a sufficient matter in this transformation.

In order to process models with LoD 3 in the same way, a technique to distinguish the windows could come helpful. Therefore, the possibilities to define the surface normal of Windows and Doors in the opposite direction of the surface normals of its surrounding surfaces should be researched, in order to be able to distinguish these features on their surface normal. Another possibility is to store the geometry with surfaces of the class Opening double, but in different directions. This way, the surface normal of this class points in two directions and is easily distinguishable.

Currently, few measures can be used to classify the semantic class Opening in a polygon soup. A measure that can be useful is symmetry, because most windows are represented as a rectangle, which is composed of two identical triangles. This method will fall short however, if a window has a circular or unusual shape.

THEMATIC FEATURES All thematic features are present in the model are, in this research, labelled as Building or Terrain. CityGML defines much more thematic features. An example is the thematic class Bridge. In the description of this class, different semantic classes are described. The main function of a bridge however, is not very well defined and leaves room for interpretation. In Oxford Dictionaries (2015), a bridge is defined as: "A structure carrying a road, path, railway, etc. across a river, road, or other obstacle". By using this definition, a bridge can be automatically detected or generated if two features of the class TransportationComplex, like Tracktype, RoadType or WaterBody, cross. An example of a bridge were a TrackType and RoadType cross is given in Figure 8.1

Before implementing the proposed changes, possible side effects should be discussed. In this discussion, the trade off between creating models which represent the built environment in the most realistic way and the necessity and possibility to automate the 3D modelling process should be central.

My personal opinion is that automation should be the central concept in the creation of 3D city models for the use in a GIS. This means that the thematic and semantic classes should be defined in a way that allows the formulation of a logic that can be translated into machine readable commands.

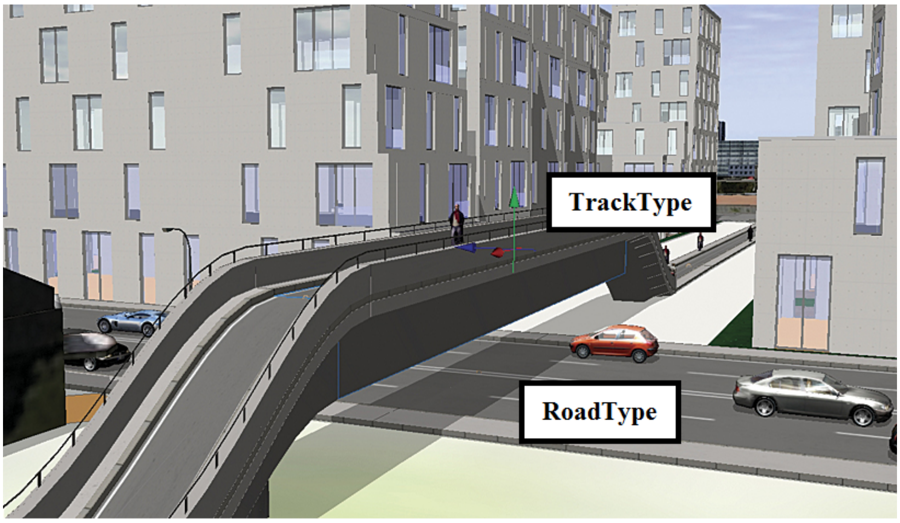


Figure 8.1: 3D Scene were a TrackType feature and a RoadType feature cross. Source: esri.com

8.2 THE DETECTION OF OVERLAP RELATIONSHIPS WITH POLYGONS INSTEAD OF TRIANGLES

Because the city models that functioned as input in this research are all triangulated meshes, the overlap relationships have to be calculated with triangle-triangle intersection methods. This is extensively described in section 5.6.

Future work should therefore focus on testing this approach with polygon meshes. Working with triangles is in this approach, not only computationally less efficient than working with non-triangulated surfaces, the computations also become more ambiguous.

8.3 MORE CONSTRAINTS IN THE REGION GROWING OF THE REGIONS

Limitation 6.6.2 describes that some regions do contain triangles with a different semantic class. This is a limiting factor in the current implementation. Therefore, future work should focus on developing more constraints in the region growing algorithm, which prevents that one region can contain triangles from a different semantic class.

8.4 TESTING THE POSSIBILITY TO COMBINE THE DUTCH BAG AND BGT DATASETS

The possibility to automatically semantically enrich a 3D city model may create new possibilities to combine different datasets in the creation of a 3D city model. This is explained with an example where two datasets from the Netherlands are used: the BAG and the BGT.

Goorman (2010) did research on two key registers on the spatial data administration in Zwolle, the Netherlands, and explains that two topographic datasets are created: the BAG and the BGT.

A difference between the two is the kind of geometry that is stored. For the BGT, the ground-level building geometry is captured, while for the BAG register, the choice was made to record the shape of a building as seen from above ('top-down') (Goorman, 2010). This difference is shown in Figure 8.2, which contains two screenshots. One from a building in the BAG and the other of the same building in the BGT. Figure 8.3 is a photograph from the buildings mapped in Figure 8.2.

Figure 8.2: The Bag and the BGT of the same area in Delft

Figure 8.3: Example of the buildings which overlap in Figure 8.2

Future research should focus on combining these two datasets and data from aerial and terrestrial laser scanners to create a semantically rich 3D city model. Because with the extraction of different height values from the LiDAR data, it should be possible to compute the height of these different buildings, allowing the creation of a 3D city model with overhanging buildings. A problem with combining these two datasets is that the geometry between both datasets may have an offset of around one meter (Goorman, 2010).

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

BIBLIOGRAPHY

Berners-Lee, T., Hendler, J., Lassila, O., et al. (2001). The semantic web.

Biljecki, F. and Arroyo Ohori, K. (2015). Automatic semantic-preserving conversion between obj and citygml. In *Eurographics Workshop on Urban Data Modelling and Visualisation, Delft (The Netherlands)*. Eurographics.

Biljecki, F., Heuvelink, G. B., Ledoux, H., and Stoter, J. (2015a). Propagation of positional error in 3d gis: estimation of the solar irradiation of building roofs. *International Journal of Geographical Information Science*, 29(12):2269–2294.

Biljecki, F., Ledoux, H., and Stoter, J. (2016). An improved lod specification for 3d building models. *Computers, Environment and Urban Systems*, 59:25–37.

Biljecki, F., Ledoux, H., Stoter, J., and Zhao, J. (2014). Formalisation of the level of detail in 3d city modelling. *Computers, Environment and Urban Systems*, 48:1–15.

Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., and Çöltekin, A. (2015b). Applications of 3d city models: State of the art review. *ISPRS International Journal of Geo-Information*, 4(4):2842–2889.

Boeters, R., Arroyo Ohori, K., Biljecki, F., and Zlatanova, S. (2015). Automatically enhancing citygml lod2 models with a corresponding indoor geometry. *International Journal of Geographical Information Science*, 29(12):2248–2268.

Brodeur, J. (2012). Geosemantic interoperability and the geospatial semantic web. In *Springer Handbook of Geographic Information*, pages 291–310. Springer.

City of Montreal (2015). portail donnees ouvertes.

CityGML (2016). Citygml datasets.

de Laat, R. and van Berlo, L. (2011). Integration of bim and gis: The development of the citygml geobim extension. In *Advances in 3D geo-information sciences*, pages 211–225. Springer.

Diakité, A. A., Damiand, G., and Gesquière, G. (2014). Automatic semantic labelling of 3d buildings based on geometric and topological information. In *3DGeoInfo 2014*.

Fan, H., Meng, L., and Jahnke, M. (2009). Generalization of 3d buildings modelled by citygml. In *Advances in Giscience*, pages 387–405. Springer.

Goorman, N. (2010). Bag & bgt: Spatial key registers-compatibility and municipal use in zwolle.

Gröger, G. and Plümer, L. (2012). Citygml–interoperable semantic 3d city models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 71:12–33.

Henn, A., Römer, C., Gröger, G., and Plümer, L. (2012). Automatic classification of building types in 3d city models. *GeoInformatica*, 16(2):281–306.

Hoover, A., Jean-Baptiste, G., Jiang, X., Flynn, P. J., Bunke, H., Goldgof, D. B., Bowyer, K., Eggert, D. W., Fitzgibbon, A., and Fisher, R. B. (1996). An experimental comparison of range image segmentation algorithms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(7):673–689.

ImageNet (2015). Large scale visual recognition challenge 2014 (ilsvrc2014).

INSPIRE (2013). D2.5 inspire generic conceptual model, version 3.4rc3. Technical Report OGC 12-019, INSPIRE.

Internet Encyclopedia of Philosophy, O. (2016). Propositional logic. http://www.iep.utm.edu/prop-log/. Accessed: 2016-08-087.

Kalogerakis, E., Hertzmann, A., and Singh, K. (2010). Learning 3d mesh segmentation and labeling. *ACM Transactions on Graphics (TOG)*, 29(4):102.

Khoshelham, K. and Díaz-Vilariño, L. (2014). 3d modelling of interior spaces: Learning the language of indoor architecture. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(5):321.

Kolbe, T. H. (2009). Representing and exchanging 3d city models with citygml. In *3D geo-information sciences*, pages 15–31. Springer.

Kolbe, T. H., Burger, B., and Cantzler, B. (2015). Citygml goes to broadway. In *Photogrammetric Week*, volume 15, pages 343–356.

Kolbe, T. H., Gröger, G., and Plümer, L. (2005). Citygml: Interoperable access to 3d city models. In *Geo-information for disaster management*, pages 883–899. Springer.

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

Löwner, M.-O., Benner, J., Gröger, G., and Häfele, K.-H. (2013). New concepts for structuring 3d city models–an extended level of detail concept for citygml buildings. In *Computational Science and Its Applications–ICCSA 2013*, pages 466–480. Springer.

Maimon, O. and Rokach, L. (2005). *Data mining and knowledge discovery handbook*, volume 2.

Mitchell, T. M. (1997). Machine learning. *Burr Ridge, IL: McGraw Hill*, 45:37.

Mommers, B. (2015). De crossover revolutie tussen bim en gis. http://www.slideshare.net/Stumico/140424-stumico-gis-en-bim-bram-mommers. Accessed: 2015-12-12.

Niemeyer, J., Rottensteiner, F., and Soergel, U. (2014). Contextual classification of lidar data and building object detection in urban areas. *ISPRS journal of photogrammetry and remote sensing*, 87:152–165.

OGC (2012). Ogc city geography markup language (citygml) encoding standard 2.0. Technical Report OGC 12-019, Open Geospatial Consortium.

Oxford Dictionaries, O. (2015). Oxford dictionaries. http://www.oxforddictionaries.com/definition/english/semantics. Accessed: 2015-12-12.

Pearl, J. (1984). Heuristics: intelligent search strategies for computer problem solving.

Pittarello, F. and De Faveri, A. (2006). Semantic description of 3d environments: a proposal based on web standards. In *Proceedings of the eleventh international conference on 3D web technology*, pages 85–95. ACM.

Previtali, M., Barazzetti, L., Brumana, R., Cuca, B., Oreni, D., Roncoroni, F., and Scaioni, M. (2014). Automatic façade modelling using point cloud data for energy-efficient retrofitting. *Applied Geomatics*, 6(2):95–113.

Pu, S., Vosselman, G., et al. (2006). Automatic extraction of building features from terrestrial laser scanning. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(5):25–27.

Remondino, F. and El-Hakim, S. (2006). Image-based 3d modeling: a review.

Rook, M., Biljecki, F., and Diakité, A. (2016). Towards automatic semantic labelling of 3d city models. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., IV-2/W1*, pages 23–30.

Rotterdam municipality (2015). Links rotterdam 3d.

Rovers, A., De Vreede, I., Rook, M., Psomadaki, S., Nagelkerke, T., Quak, W., Van der Spek, S., Beers, B., Voute, R., and Verbree, E. (2015). Semantically enriching point clouds: The case of street levels.

Rust, D. (2015). How to find surface normal of a triangle. Mathematics Stack Exchange. URL:http://math.stackexchange.com/q/307155 (version: 2013-08-29).

SciPy (2015). Spatial algorithms and data structures.

Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge University Press.

SIG3D (2015). Modeling guide for 3d objects - part 2: Modeling of buildings (lod1, lod2, lod3).

Stadler, A. and Kolbe, T. H. (2007). Spatio-semantic coherence in the integration of 3d city models. In *Proceedings of the 5th International Symposium on Spatial Data Quality, Enschede*.

Stadler, A., Nagel, C., König, G., and Kolbe, T. H. (2009). Making interoperability persistent: A 3d geo database based on citygml. In *3D Geo-Information Sciences*, pages 175–192. Springer.

Stehman, S. V. (1997). Selecting and interpreting measures of thematic classification accuracy. *Remote sensing of Environment*, 62(1):77–89.

Stoter, J. and Van Oosterom, P. (2002). Incorporating 3d geo-objects into a 2d geo-dbms. In *Proceedings FIG, ACSM/ASPRS, Washington DC, April 19-26, 2002*.

Stoter, J. and Zlatanova, S. (2003). 3d gis, where are we standing? In *ISPRS Joint Workshop on'Spatial, Temporal and multi-dimensional data modelling and analysis', Québec, October, 2003*.

Swisstopo (2015). swissbuildings3d 2.0.

Takase, Y., Sho, N., Sone, A., and Shimiya, K. (2003). Automatic generation of 3d city models and related applications. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34:5.

van Oosterom, P. (1999). Spatial access methods. *Geographical information systems*, 1:385–400.

van Oosterom, P. and Zlatanova, S. (2008). *Creating Spatial Information Infrastructures: Towards the Spatial Semantic Web*. CRC Press.

Van Oosterom, P., Zlatanova, S., and Fendel, E. (2006). *Geo-information for disaster management*. Springer Science & Business Media.

Verdie, Y., Lafarge, F., and Alliez, P. (2015). Lod generation for urban scenes. Technical report, Association for Computing Machinery.

Verma, V., Kumar, R., and Hsu, S. (2006). 3d building detection and modeling from aerial lidar data. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2213–2220. IEEE.

Vosselman, G., Dijkman, S., et al. (2001). 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.

Vosselman, G., Gorte, B. G., Sithole, G., and Rabbani, T. (2004). Recognising structure in laser scanner point clouds. *International archives of photogrammetry, remote sensing and spatial information sciences*, 46(8):33–38.

Waldhauser, C., Hochreiter, R., Otepka, J., Pfeifer, N., Ghuffar, S., Korzeniowska, K., and Wagner, G. (2014). Automated classification of airborne laser scanning point clouds. In *Solving Computationally Expensive Engineering Problems*, pages 269–292. Springer.

Wikipedia (2015a). Confusion matrix.

Wikipedia (2015b). Heuristic.

Wikipedia (2015c). kd-tree.

Wikipedia (2015d). Normal (geometry).

Wikipedia (2015e). Wavefront .obj file.

Xiong, X., Adan, A., Akinci, B., and Huber, D. (2013). Automatic creation of semantically rich 3d building models from laser scanner data. *Automation in Construction*, 31:325–337.

Yanbing, W., Lixin, W., Wenzhong, S., and Xiaomeng, L. (2007). On 3d gis spatial modeling. In *Proceedings of the ISPRS Workshop on Updating Geospatial Databases with Imagery and the 5th ISPRS Workshop on DMGISs, Urumchi, Xinjiang, China*, pages 237–240. Citeseer.

Zlatanova, S. (2000). *3D GIS for urban development*. International Inst. for Aerospace Survey and Earth Sciences (ITC).

A REFLECTION

This reflection is a mandatory part of this thesis. The aim of this reflection is to look back and see if the proposed approaches work and to understand the “how” and ”why”, and, subsequently to learn from this. In other words, this chapter reflects on the different processes of the conducted research.

The reflection is organised in the following way. First, the relationship between the research and the wider social context is discussed. Second, the relationship between the research and the field of Geomatics is elaborated on. Third, the relationship between the methodical line of approach of the Master Geomatics and the method in this framework is discussed. This chapter ends with a discussion about how and why the proposed methodology did work, and to what extend. Thereby, the process and planning of the research are discussed.

A.1 MY RESEARCH IN ITS SOCIAL CONTEXT

In the past years there has been a growing interest for 3D city models. These city models are used in 3D GIS, but also function as the backbone for ’the city of things’, in where the city is monitored and managed by a wide range of sensors. A central concept in these new developments, and therefore in this research, is the standard CityGML. CityGML allows for integrated storage, processing and exchange of these city models. Thereby, the CityGML standard is easily adaptable to a certain use case, while it still allows the storage of attribute data, which is information about the different city objects. More important, the standard is fastly becoming the de facto standard for 3D city models.

This research focusses on the automatic enrichment of the geometry in a 3D city model with thematic and semantic information. In other words, the proposed methodology takes an unstructured set of triangles, which represents a city, and structures this geometry in single buildings, while adding semantic information to the triangles in these buildings. Semantic information, in this case, means what a surface represents, for example a WallSurface, a GroundSurface or a OuterCeilingSurface. 5 Different classes are distinguished. This semantic information is crucial in the use of 3D city models and is needed to create city models that are fit for use.

A.2 MY RESEARCH IN THE FIELD OF GEOMATICS

The growing demand for 3D city models is triggered by new developments in remote sensing techniques. These techniques are LiDAR and photogrammetry, and allow the capturing of 3D data of the human environment. These techniques create huge datasets, which are, among other things, used to create 3D city models.

A central concept in the field of Geomatics is the automation of data processing techniques. Automated data processing, in this case, means computer guided extraction of information from large (combined) datasets. This automation process allows the up-scaling of techniques, creating economic benefits and increased access to up to date data and, in the scope of this research, an improved understanding of the built environment in a machine readable format. This research aimed at creating a methodology that allows automatic classification of 3D city models.

A.3 THE METHODICAL LINE OF APPROACH

The MSc Geomatics focusses on the management of geographical data. The methodical line of approach, in this sense, is merely based on automating these data management processes. The methodology that is created during this research is based on this line of approach. This is done with the following in mind.

The way humans perceive the physical reality around them is based on complex reasoning, the social and cultural context, earlier experiences and/or human perception and knowledge. A computer simply needs commands. The approach taken in this research is to define commands a computer can understand, in order to interpret data in a way that humans do it. This is done by extending the current definitions in the CityGML data with heuristics rules, that, in the scope of this research, are straightforward rules or knowledge gained from the human perception about its living environment.

A.4 RESULTS

The results showed that by defining these rules, a 3D city model can be automatically enriched with semantic and thematic information. With clean input data, a 100% accuracy in the semantic classification can be achieved.

A.5 LESSONS LEARNED

The most important lessons I have learned in respect to writing this thesis are described in the coming sections.

The biggest challenge for me was to focus on the methodological approach, as I was completely focussed on delivering a product instead of a piece or research. The realisation that my focus was on the wrong aspect came after writing the research paper, which forced me to restate and overthink the problem statement, the challenges and the focus. Thereby, I neglected the scientific perspective, whereby I focussed on finding a solution as soon as possible.

A.6 PROCESS AND PLANNING

The planning of the research conducted has not always been optimal. Especially at the beginning, the lack of time and a underestimation of the amount of work that was still ahead led to a postponement of the first P4 deadline.

This also brought new possibilities. First the extra time was used to write a paper. This paper has already been accepted and will be published in the coming weeks. This paper helped to restructure the whole research and concept. Next, another methodology was developed, with in mind the lessons from the paper and the earlier developments.

COLOPHON

This document was typeset using LaTeX. The document layout was generated using the `arsclassica` package by Lorenzo Pantieri, which is an adaption of the original `classicthesis` package from André Miede.