Automatic semantic labelling of 3D city models

Merwin Rook
m.rook@student.tudelft.nl

January 7, 2016

1 Introduction

Geographic Information Systems (GIS) have developed into sophisticated systems for maintaining and analysing spatial and thematic information of spatial objects (Stoter and Zlatanova, 2003). A GIS operates with the largest scope of these objects: spatially and semantically, their relationships and the means to analyse these components (Zlatanova, 2000). Geographical information however, is still largely presented in a two dimensional (2D) field, as two-dimensional geo-information is available in large amounts, at different scales and covering many application domains (Stoter and Van Oosterom, 2002). In the last years, the need for three dimensional (3D) information is rapidly increasing, as 2D GIS has shown its limitations in a lot of applications, like: noise prediction models (noise spreads out in 3D), water flood models, air pollution models, geological models and real estate market analysis. Other disciplines that can benefit from 3D geo-information are: 3D urban planning, environmental monitoring, telecommunications, public rescue operations or landscape planning (Stoter and Zlatanova, 2003; Gröger and Plümer, 2012). New sensing technologies, like laser scanning and photogrammetry, create 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) and once the developments in 3D GIS provide a compatible functionality and performance, the spatial information services will evolve into the third dimension (Zlatanova, 2000).

Spatial information of the urban environment is considered the most complex data in geospatial information (Yanbing et al., 2007). The lack of availability of thematic and semantic meaning of (parts of) these 3D models is still a big limiting factor for the increasing number of applications that make use of these models (Diakité et al., 2014). As many 3D city models are available as a collection of surfaces, their geometry is unstructured and as has no meaning. This means that these models do not contain semantics. Semantics can be described as: the branch of linguistics and logic concerned with meaning (Oxford Dictionaries, 2015). The lack of semantics in these models leads to a decrease in the usability of these 3D models. For example: the geometry of a building is not differentiable from the geometry of a road, while geometries within the same object, for example: roof, windows and walls, have no semantic meaning. While such models may still be valuable in visualisation, their use for GIS purposes is hindered by the lack of semantics (Brodeur, 2012).

The concept of semantic enrichment, i.e. adding of semantic information to the geometry, is necessary for creating additional value of the dataset and is therefore crucial to create 3D city models that meet the requirements of relevant applications (Henn et al., 2012). Therefore, this research aims at semantically enriching 3D models, to increase the usability of these models for GIS analysis. The goal is therefore to research and develop a method to automatically enrich a virtual 3D city model with semantic information. Figure 1 depicts the recognition of the different building parts, for example roof, window, wall and building ground surface. Which this research aims at to automatically recognise and label with their semantic meaning.

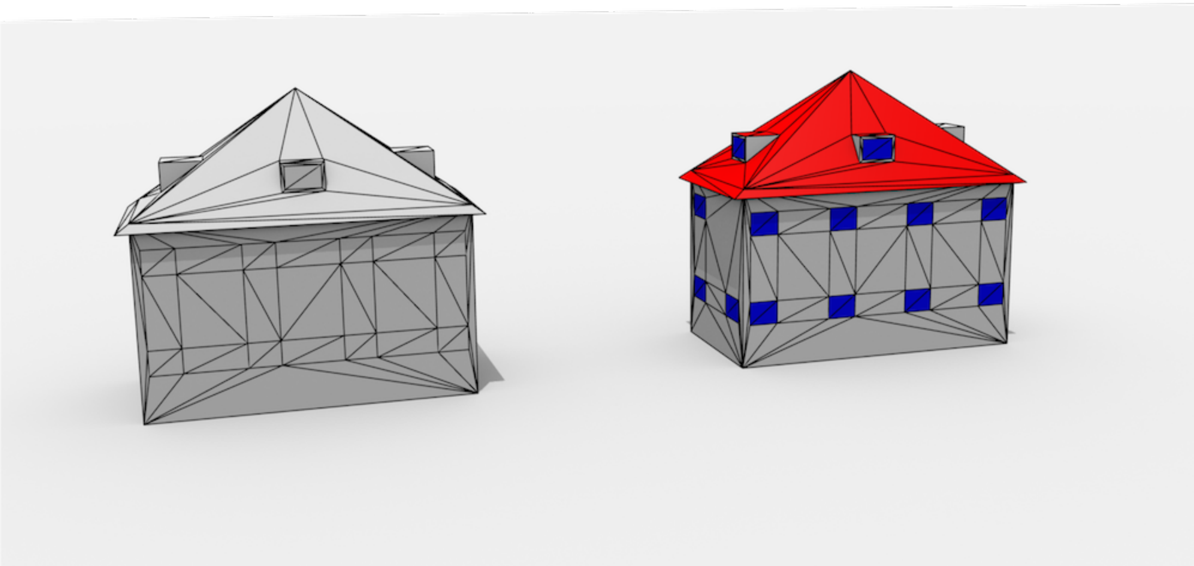

Figure 1: Recognising different building parts in a virtual 3D model

2 Scientific relevance

Current research mainly focuses on the construction of 3D city models from point data, captured by LiDAR (light detection and ranging)(Lafarge and Mallet, 2012; Poullis and You, 2009), mostly creating urban 3D models without semantic information. This research aims at bridging the gap in the lack of research that currently exists in enriching these 3D city models. Adding semantics 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, as stated by (Döllner et al., 2006, p. 2): "Virtual 3D city model creation needs to be based on automatic and semiautomatic acquisition methods wherever possible", depicting the importance of automating the process. Although some research in creating semantically rich 3D models has been carried out (Verdie et al., 2015; Xiong et al., 2013), research in creating or 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.

Next to geometrical models created from sensor data, other sources of information are getting integrated into GIS. For example, CAD models, that are currently integrated into GIS and vica versa (Mommers, 2015; de Laat and van Berlo, 2011). As a lot of information is stored as CAD models, what creates major possibilities in integrating this data, in order to create urban scenes (Stoter and Zlatanova, 2003). This integration needs more advanced integration of semantics, therefore these semantics must first be added to this GIS data (de Laat and van Berlo, 2011). Other sources of 3D city models are 3D modelling software, e.g. Esri's CityEngine. The usability of these models can be highly increased if semantic information is added to the geometry, as the effects of the changes in the environment, as modelled in the software, can be simulated and calculated.

3 Related work and literature research

The following chapter will provide an overview of already existing works. The first section introduces the generation of 3D city models, followed up by an literature research on semantics and themes and the role of these concepts in the generation of CityGML and the level of detail of these 3D city models. This chapter ends with a description of possible classification techniques, that can be useful in this research.

3.1 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 3D laser scanners 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. These raw points are not directly used, due to the huge size of the models, and the very low level of abstraction. Therefore, 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). The increase in availability of this data has triggered the extensive increase of the use of these 3D models for analysis and visualization (Previtali et al., 2014; Stoter and Zlatanova, 2003). A growing number of applications even rely on 3D city information (Stadler and Kolbe, 2007; Stoter and Zlatanova, 2003; Gröger and Plümer, 2012).

3.2 Semantics

Most of these applications require semantic information (Stadler and Kolbe, 2007), currently semantically rich models are still not largely available and research in creating or enriching existing 3D city models is almost non-existing and holds many scientific and software opportunities (Biljecki and Arroyo Ohori, 2015). 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). The semantic web is an extension of the internet protocol, which contains meaningful information about the data that machines can understand. This understanding about the meaning of data offers new possibilities in linking and processing data. So for the semantic web to function, computers must have access to structured collections of information and sets of inference rules, that they can use to conduct automated reasoning (Berners-Lee et al., 2001).

3.3 Semantics for the purpose of spatial analysis

For geographical data, semantics are important for a number of reasons:

- 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 model of houses is merged with a digital surface model. They describe a process where different datasets can be merged with the use of semantics, for example: the building ground surface of the model of a house should be connected to the surface, whereby the geometry of parts of the house and the terrain surface must be separately accessible.

- Semantics are used in analysis, 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 used to simulate the impact of events in the real world environment.

- Data harmonisation can be done more effective if semantic information is available (van Oosterom and Zlatanova, 2008). A good example of these cases is the INSPIRE framework, wich should make it possible to combine spatial data and services from different sources (INSPIRE, 2013).

- Finally, semantics are recognized as one of the most important features that separates virtual 3D models, used for only visualization, from models employed in spatial analyses (Biljecki et al., 2014).

Biljecki et al. (2015b) researched the current utilisation of 3D city models. In their research, they point out the importance of semantics in use cases of 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 this 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 floorspace surface.

3.4 Semantic interoperability

In order to be able to use data from several sources or combine seperate datasets, (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 as one of the main issues in the GIS field (Kolbe et al., 2005). Where semantic interoperability plays a major role, as 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 te standard CityGML, as CityGML perfectly 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 in the development of CityGML, which is explained in the next section.

3.5 CityGML

CityGML is a specification or standard that is a concrete and application specific data format for geographical data and its semantics. The structures, aggregations and taxonomies in the standard allow advanced analysis for the applications earlier described (Gröger and Plümer, 2012). The data format is an open, independent standard of the International Organization for Standardization (ISO). The standard can be extended and specified for a specific application domain, and explicitly supports simple and complex 3D geometry and topology (Kolbe et al., 2005). The format consists of two hierarchies: the semantic and the geometrical, in which the corresponding objects are linked by relationships (Stadler and Kolbe, 2007).

CityGML 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 holds geometrical, semantical and relational aspects of 3D city models. The following paragraphs will give a brief overview about the structure of CityGML.

CityGML is not limited to only storing buildings, but holds all relevant features in an urban scene. The features, or objects, are organized into 13 modules, or themes, which can be combined as needed for a specific application. The CityGML class taxonomy distinguishes between different objects, buildings and other man-made objects, waterbodies and vegetation, but also roads and other transportation facilities. On its thematic level, CityGML 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 (Gerhard Gröger, 2012). These spatial and semantic properties of defined objects are structured in five different LoD (Kolbe et al., 2005; Gröger and Plümer, 2012).

3.6 Level of Detail

The LoD in 3D city modelling serves as a specification-related instruction for the acquisition, modelling, generalisation and exchange of spatial data. The term, or definition, LoD is very incoherent. Biljecki et al. (2014) therefore redefined LoD of a 3D city model as: "the degree of its adherence to its corresponding subset of reality". In other words, how close the virtual representation visually reflects the actual real-world scene.

Five levels of LoD's are defined in the CityGML standard and will be used in this research. These five levels offer a clear and straightforward distinction and are used in related research (Boeters et al., 2015; Biljecki et al., 2014).

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: LoD 3 models are models that 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).

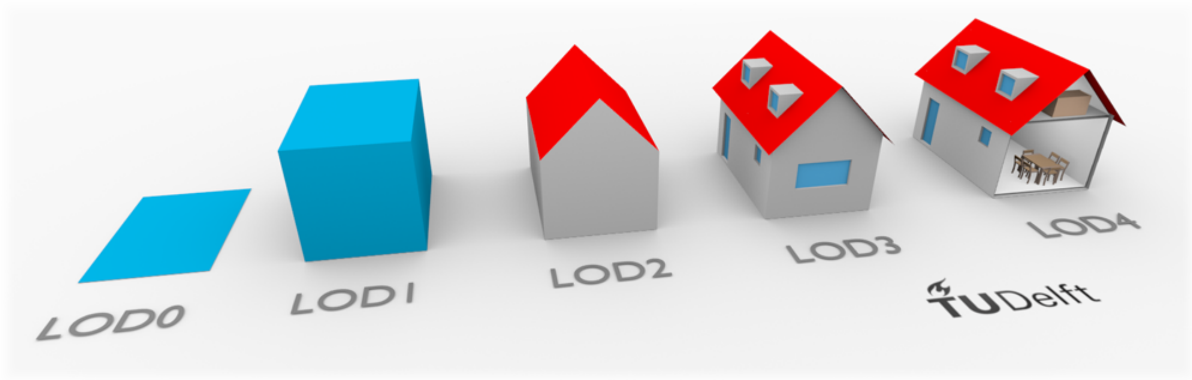

Figure 2: Different levels of LOD (F. Biljecki, 2016)

The concept of LoD plays a central role in this research, as the LoD is a very decisive matter in the presence and the geometrical properties of certain semantic classes in a 3D urban scene. For example, if a model is LoD 1, roofs are always flat, while the same roofs represented in LoD 2 have a different geometry, as these can be sloping. A detailed description of the semantic classes that will be classified in this research, are explained in section 5.

3.7 Related work in the semantic enrichment of 3D data

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

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.
- 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.

Henn et al. (2012) researched a method to classify buildings, in LoD 1, by their building type, whereby a 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 hospital, 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 requiere 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. Here fore, they use the Combinatorial Map (C-Map) data structure. 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: façade, wall, ground floor, ceiling, windows and doors. Focussing on indoor and outdoor semantic classes.

3.8 Classification methods

Different classification methods exist for the labelling of polygons, points and clusters of points. This section describes some of these methods that might be interesting for this research.

Support Vector Machine (SVM) A SVM is a supervised learning algorithm, whereby the aim is to automatically find regularities and patterns in data (Henn et al., 2012). The SVM uses training samples to assign a class to a feature. These training samples are non-linearly mapped to a high dimensional feature space. The SVM finds a hyper-plane, a linear decision surface, which divides the set of training data in a way where all the points with the same label are on the same side of the hyper-plane. The basic principle is that the SVM finds the most optimal hyper-plane in a high dimensional feature space (Cortes and Vapnik, 1995). An example of a linear decision space is given in figure 3. Whereby the dots represent the sample data, which subdivide the decision-space in two.

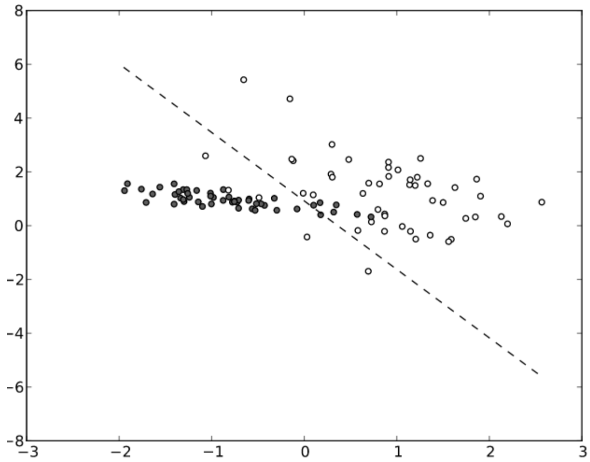

Figure 3: Linear SVM classifier, source: Wikipedia

Decision Tree 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 the feature (Maimon and Rokach, 2005).

Bayesian probability The bayesian probability method is used to evaluate the probability of a hypothesis. The Bayesian probability specifies some prior probability, which is then updated in the light of new, relevant data (evidence). The Bayesian interpretation provides a standard, pre-defined set of procedures and formulae to perform a calculation. As in Bayesian probability, a weight is given to each variable (or attribute) and added up or multiplied by a pre-defined factor. The end score assigns the actual class to the pre-defined region, segment (Wikipedia, 2015b).

4 Research objectives

This chapter defines the research question and its scope. Thereby clarifying decisions in creating this research plan. The chapter begins with defining the research question, followed by the purpose statement and a clarification of what semantic classes will be sought for in the data. It ends with a description of problems that are expected to be encountered.

4.1 Research question

In the course of this Geomatics Msc thesis, the goal is to automatically enrich a 3D city model with semantic information, in order to support spatial analyses which require them. This research therefore aims at answering the following question:

"How can a 3D city model be semantically enriched automatically?"

To bring a sufficient 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?

4.2 Research scope and purpose statement

The goal is to develop an automatic workflow that takes a 3D virtual city model as input, and turns out the same geometry, that is structured in a way that geometry that shares a semantic meaning, is stored together. The following semantic classes will therefore be identified.

- Terrain surface, containing all ground surfaces that are not part of a building.
- Roof surface
- Wall surface
- Building ground surface, which is the terrain surface that is part of a building.
- Openings, which include walls and windows

This selection of classes is based on the CityGML class taxonomy, as explained in the chapter related work and literature research, section 3.6. Thereby, these semantic classes are expected to be detectable in the 3D city models, by using the methods proposed later in this research. The research aims less at labelling geometries, or collections of geometries, with thematic information, thematic information as explained in the subsection 3.5. Thereby, most available 3D data models only hold buildings, which limits the possibility to test and develop the algorithm.

4.3 Semantic classes per LoD

Depending on the LoD of the dataset, differentsemantic classes will be relevant in this process, depending on the LoD of the dataset. The LoD of a 3D city model, as described in the literature research, defines the geometric properties of a dataset. As, for example, LoD 0 only holds flat surfaces, LoD 1 models only have flat roofs and LoD 2 can also hold sloping roofs. The following paragraph will therefore define what semantic properties will be added, depending on the LoD.

LoD 0: Models with LoD0 will not be considered in this research.

LoD 1: In models with LoD1, the labelling process will aim at adding semantics to buildings only.

LoD 2: In models with LoD2, the labelling process will aim at adding semantics to: terrain surfaces, walls, roofs and building ground surfaces.

LoD 3: In models with LoD3, the labelling process will aim at adding semantics to: terrain surfaces, individual walls, roofs and building ground surfaces. Thereby, openings and dormers will be detected.

LoD 4: Models with LoD4 will be left untouched in this research, as labelling indoor geometries is outside the scope of this research.

4.4 Research requirements

The functional and non-functional requirements are:

- The process must create valid results in flat, as well as in models where the terrain is sloping.
- The algorithm will only take files witch are stored in wavefronts object format. This formats can be opened, stored, and processed as ascii. Thereby, the formal and simple definition of the geometric objects, containing vertices, edges and faces, is unambiguous.
- The algorithm will only work on models which hold a valid geometry.

4.5 Expected difficulties and risks

This section elaborates on the possible difficulties and risks in the research. These risks and difficulties have to be encountered or avoided.

- There is little research on the semantic labelling of polygon meshes. Most research focusses on classification of points in a point cloud. This lack of research means that there are currently few researched techniques available, meaning that the methods have to be developed in this research.
- The little availability of 3D city models, what leads to less testing of the developed algorithm.
- The availability of valid models, which means the availability of models that have a valid geometry. Ledoux (2013) researched the validation of solids, thereby giving different examples of valid and invalid solids. For example, a solid is invalid when different solids overlap and 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.

5 Methods

The following chapter provides the methodology on which the labelling process is done. First, the data structure that is used to label the 3D city model is described. Second, different methods, in which triangle attributes are calculated, will be explained. Third, a method is given to find the local neighbourhoods of the triangles.

5.1 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 (?). 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.

Vertex data:

v Geometric vertices

vt Texture vertices

vn Vertex normals

vp Parameter space vertices

Polygonal faces

f Face

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. An example of an object file is shown in Figure 4.

```
# 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
```

Figure 4: Simple object file, Source: Fileformat.info (2015)

5.2 Workflow

Figure 5 gives an overview of the labelling process, where the different steps in the process are visualised in a schema. The third step, Processing and labelling, is further differentiated in Figure 6. This image is used as a guide for the different processing steps that are going to be taken, the decision tree of the technical workflow is used to structure the rest of this chapter.

Figure 5: schema of the workflow

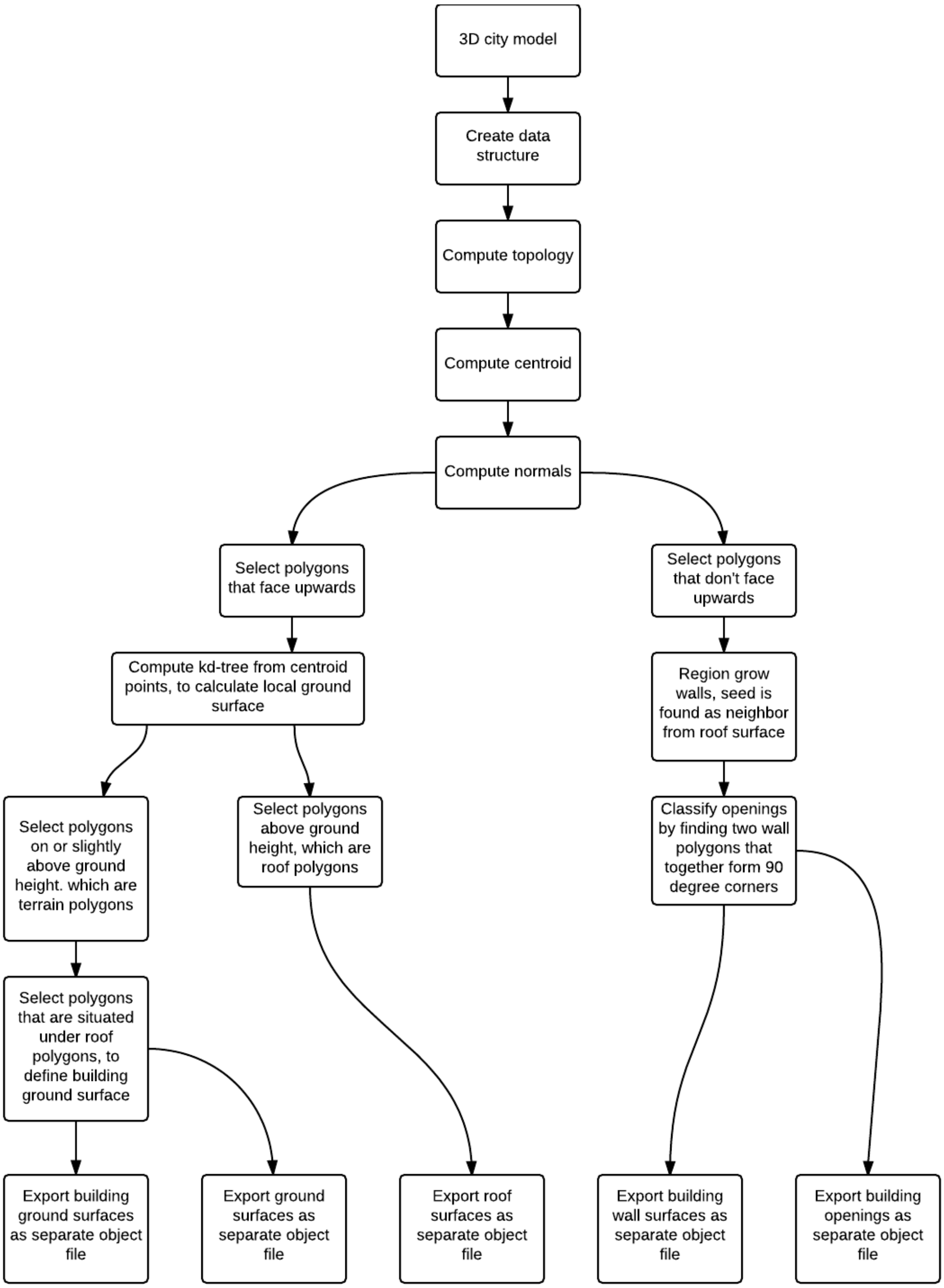

Figure 6: Decision tree of the technical workflow

5.3 Datastructure and triangle attributes

The datastructure is created by opening the file with Python. Next, all lines of the file will be iterated over. Dependent on the first letter of the line, an instance of a point, which only holds the X, Y and Z coordinates, or of a triangle is created. Triangles hold multiple attributes, these attributes are used in the labelling process. The following sections describe these attributes and their use.

5.3.1 Labels

During the labelling process, different steps are taken that distinguish the triangle on certain geometrical values. This distinction is stored as an attribute of the triangle instance.

5.3.2 Topology

As the triangles in wavefront object files are defined by the vertices in that same file, a topology can be created by only utilising this information. In the Oxford Dictionaries (2015), topology is defined as: "The study of geometrical properties and spatial relations unaffected by the continuous change of shape or size of figures". In this case, a topology gives the spatial relationship between the different triangles in the 3D city model. In order to retrieve the topology, all points are iterated over, creating a temporary dictionary, that stores all points as a key, while adding all triangles that. Later this dictionary is used to create a triangle attribute, which stores all neighbouring, or adjacent, triangles.

5.3.3 Centroid

A centroid is calculated for every triangle. These centroids are later used to find nearby triangles, by making use of a kd-tree, which will later be explained.

A triangle is created out of three points: $P_1 = (x_1, y_1, z_1)$, $P_2 = (x_2, y_2, z_2)$ and $P_3 = (x_3, y_3, z_3)$. Then, the centroid (C) of a triangle is calculated as:

$C_x = (X_1 + X_2 + X_3)/3$
$C_y = (Y_1 + Y_2 + Y_3)/3$
$C_z = (Z_1 + Z_2 + Z_3)/3$

5.3.4 Surface normals

A normal is a line or 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, 2015a) as depicted in figure 7. The normals of the triangles will be used to extract the roof, terrain and floor levels, which normals all point upwards.

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 $n = (P_2 - P_1) \times (P_3 - P_1)$. In matrix form, this makes:

$$\text{n=det}\left(\begin{bmatrix} i & j & k \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \end{bmatrix}\right) = \begin{pmatrix} (y_2 - y_1)(z_3 - z_1) - (y_3 - y_1)(z_2 - z_1) \\ (z_2 - z_1)(x_3 - x_1) - (x_2 - x_1)(z_3 - z_1) \\ (x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1) \end{pmatrix}$$

5.4 Select upward facing polygons to create local neighbourhoods, in order to find the terrain surface height

The next step in the labelling process is to find all triangles that face upwards. The triangles that face upwards can be either roof, terrain, building ground surface, or floor/ceiling. These triangles will be further differentiated by using the local neighbourhood. These are created

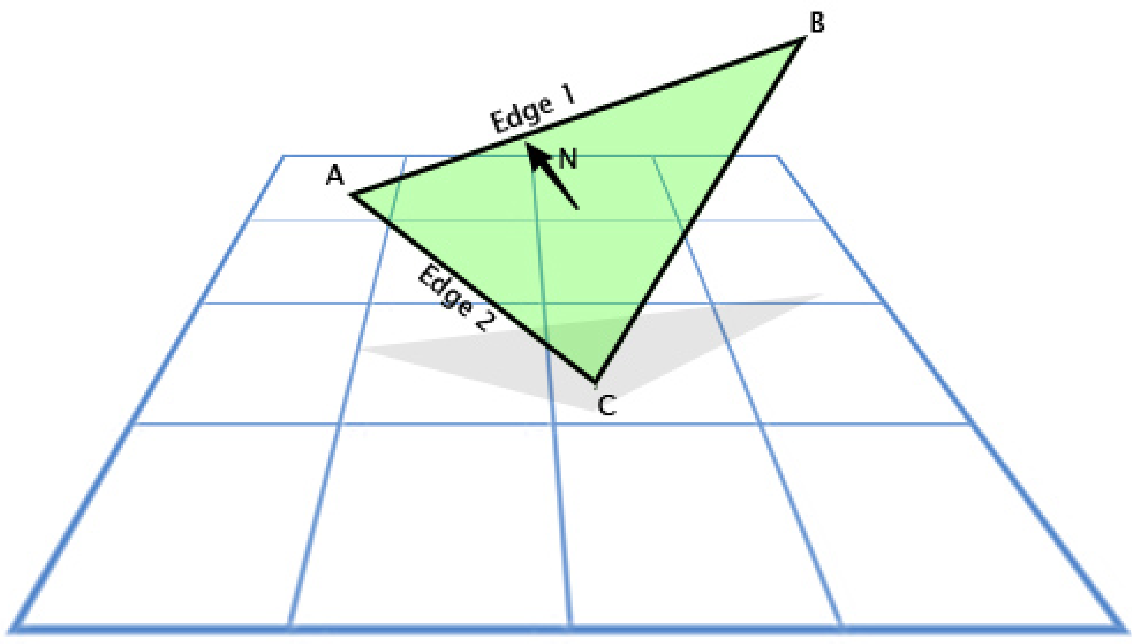

Figure 7: Normal of a triangle, Source: blog.wolfire.com

to find local terrain surfaces. As one of the aims of this research is to create a robust work flow, that also works on 3D city models of hilly environments, a local terrain surface must be defined for every subregion of the 3D city model. In order to do so, a kd-tree is used to find nearby triangles. This step is referred to in the project planning (Figure 9) as step 2.

kd-tree The kd-tree, a main memory data structure, is a first option for the indexing of point clouds (van Oosterom, 1999). The kD-tree recursively divides the space using a root-leaf structure. The root corresponds to the complete spatial area and the leafs represent the resulting areas of the split. If an area is not split any further it becomes an end-leaf. In turn, the area is split on the X- and Y- axis. For the 3D case, the Z-axis is also included.

In the KD-tree, that forms the basis for finding the terrain surface, only the X and Y axis are used and take the triangle's centroids as input. This KD-tree is than used to fit a surface to the selected polygons. Next, the created surface is used to classify the roof and terrain polygons. This step is referred to in the project planning (Figure 9) as step 3.

5.4.1 Find polygons aligned under roof surfaces

After the classification of the roof and terrain polygons, a new distinction is made in the just classified terrain polygons: building ground surfaces are extracted, which form the floor in buildings, are distinguished from the terrain. These polygons are aligned under a roof surface, what is therefore used as a classification rule. This step is referred to in the project planning (Figure 9) as step 4.

5.4.2 Region grow walls

Next, neighbouring polygons of roofs are sought, which are called seeds, that are not orientated upwards and situated below the roof height. These polygons are used to region grow the wall, whereby, by using the earlier created topology, neighbouring triangles are iterated over, while checked if they share the same orientation as the seed. This step is referred to in the project planning as step 5.

5.4.3 Find openings

The created walls are iterated over, while checking if they hold a 90 degree angle. If they do, it's neighbours, if they share an edge, will be checked if they also hold a 90 degree angle. If these neighbouring polygons together form a square or a rectangle (figure 8), the two polygons will be classified as opening. This step is referred to in the project planning (Figure 9) as step 6.

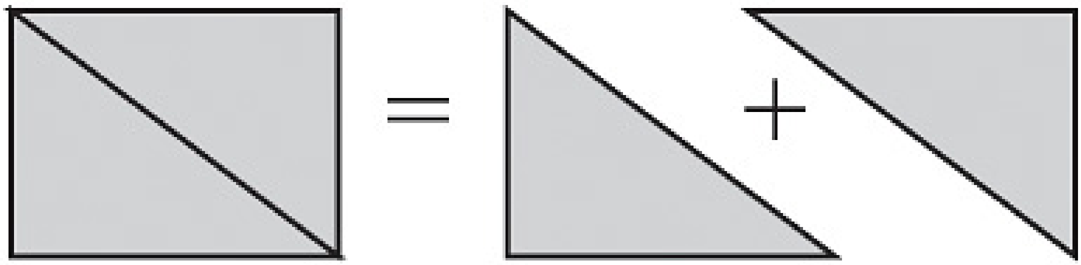

Figure 8: Two triangles, forming a rectangle

5.4.4 Check accuracy of the classification process

As a last step, the results of the classification process will be checked. In this step, a method will be developed to test the accuracy of the classification process. This step is referred to in the project planning (Figure 9) as step 7.

6 Schedule

The schedule for this master thesis is given in Figure 9. The chart gives a clear overview of the planning of the project. Every task should be done in the time that is stated in this chart.

6.1 GANTT chart

ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Prepare P2	1	2	1	2	75%
P2 Presentation	2	1	2	0	0%
Step 2	3	1	0	0	75%
Step 3	4	1	0	0	0%
Step 4	5	1	0	0	0%
Step 5	6	1	0	0	0%
Step 6	7	1	0	0	0%
Step 7	8	1	0	0	0%
Optimise	9	5	0	0	0%
P3 presentation	14	1	0	0	0%
Write final Thesis	15	15	0	0	0%
Visualize results	17	5	0	0	0%
Results evaluation	18	1	0	0	0%
P4 presentation	19	3	0	0	0%
P5 presentation	28	3	0	0	0%

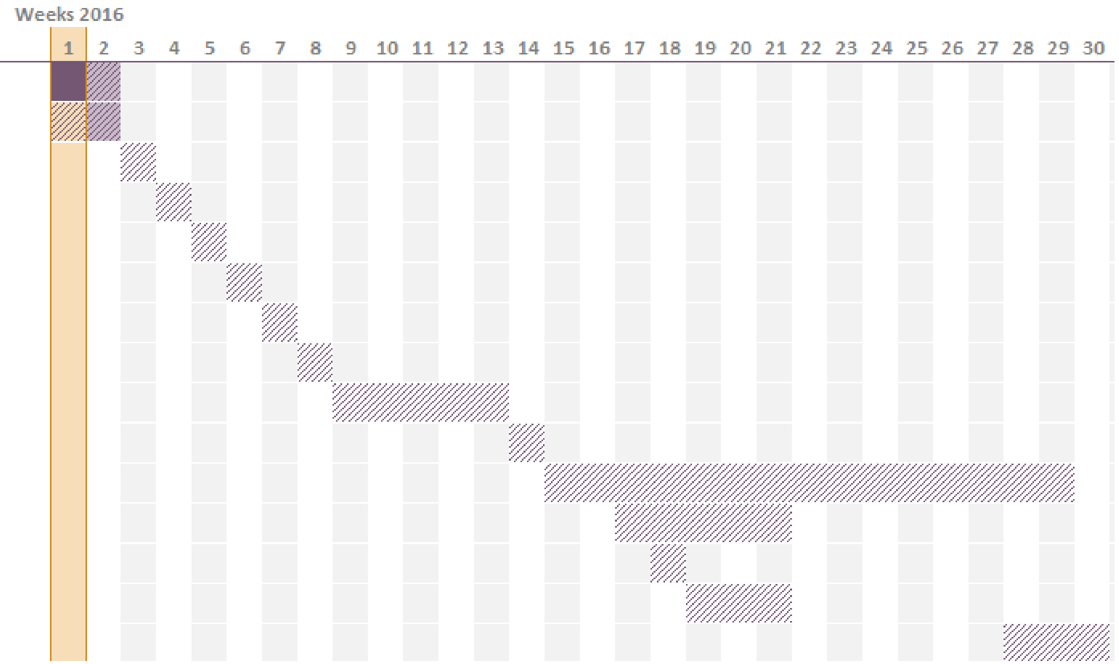

Figure 9: GANTT chart

7 Tools and Data

The following chapter gives a brief overview of what tools and data are going to be used in the project.

7.1 Test datasets

The 3D city models that are being used are described in this paragraph.

Paris dataset The Paris dataset (Figure 10) is a dataset with LoD 3, but does not contain openings. Next to buildings, the datasets holds street furniture, a side walk and a road. The dataset is downloaded from: http://tf3dm.com, where datasets are downloadable for free.

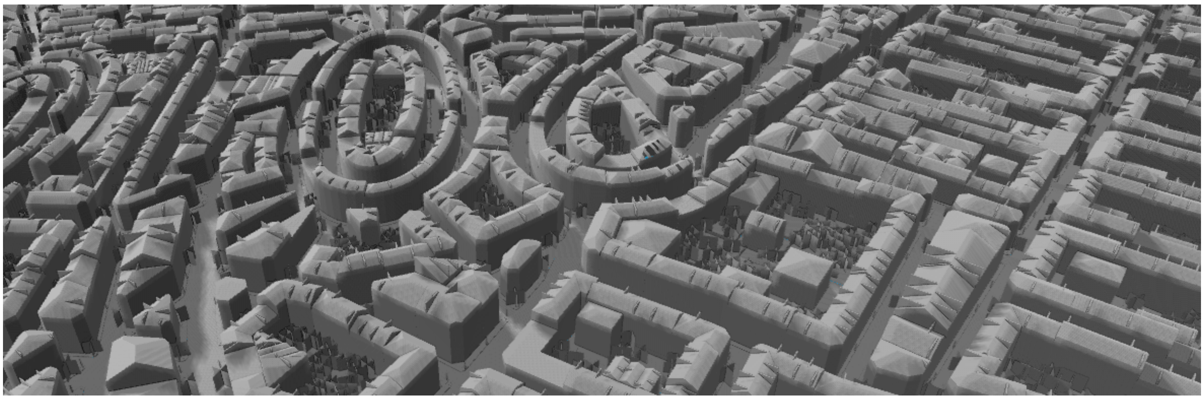

Figure 10: Paris Dataset

Waldbruecke dataset The second dataset that will be used, is a 3D model of small villages in Waldbruecke in the state of Baden-Wuerttemberg in Germany (Figure11). The model holds houses in LoD 1 and LoD 2. The buildings with a different LoD will be extracted, so 3D city models are created with one consistent LoD. The dataset is freely accessible on the CityGML website (CityGML, 2016).

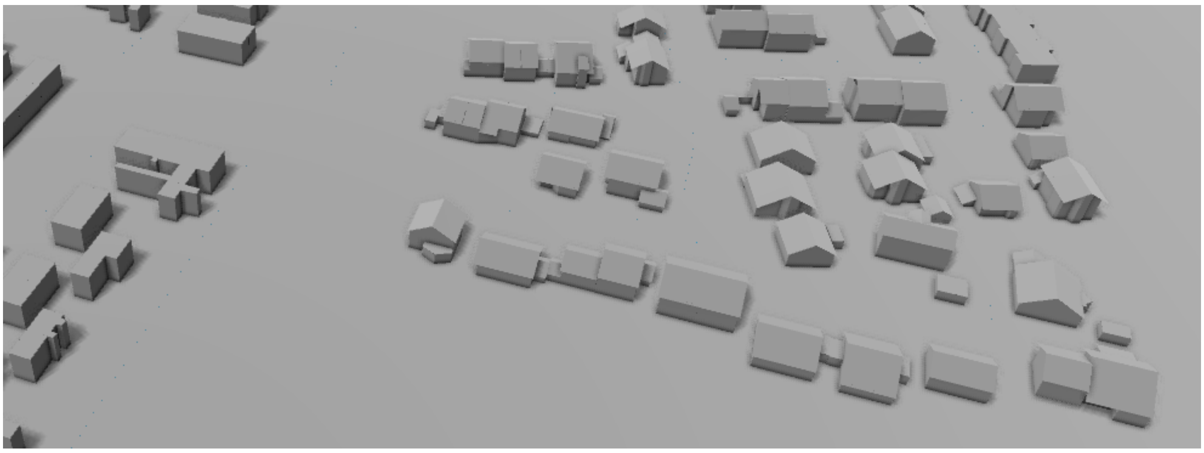

Figure 11: Waldbruecke dataset

Delft dataset The third dataset (Figure 12) is a model with LoD 2 of a part of the city of Delft. This dataset does not hold a ground surface, although the building blocks hold a building ground surface. The dataset is created by Hugo Ledoux, assistant professor at the 3D geo-information group at the TU Delft.

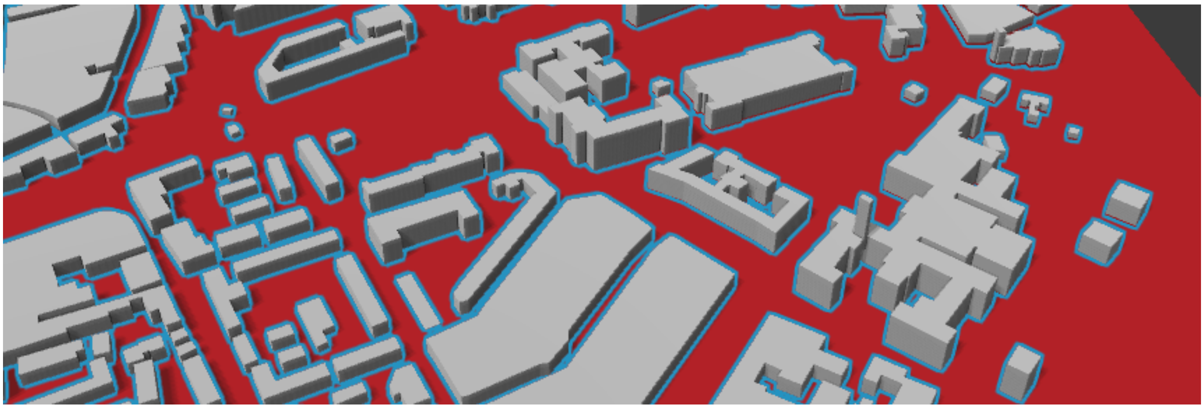

Figure 12: Delft Dataset

7.2 Tools

The following tools are going to be used in this research.

- Python, is used to write the labelling algorithm in.
- Meshlab, is used to visualise intermediate results.
- Blender, is used to visualise the end results.

In this research, only the geometry of the models will be used. The geometry of the model will be stored, processed and written from and to an Wavefronts object file format.

7.3 Mentors

The mentors in this Msc Thesis project are Filip Biljecki and Abdoulaye Diakité.

References

Tim Berners-Lee, James Hendler, Ora Lassila, et al. The semantic web. 2001.

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

Filip Biljecki, Hugo Ledoux, Jantien Stoter, and Junqiao Zhao. Formalisation of the level of detail in 3d city modelling. *Computers, Environment and Urban Systems*, 48:1–15, 2014.

Filip Biljecki, Gerard BM Heuvelink, Hugo Ledoux, and Jantien Stoter. 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, 2015a.

Filip Biljecki, Jantien Stoter, Hugo Ledoux, Sisi Zlatanova, and Arzu Çöltekin. Applications of 3d city models: State of the art review. *ISPRS International Journal of Geo-Information*, 4(4): 2842–2889, 2015b.

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

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

CityGML. Citygml datasets, 2016. URL `http://www.citygml.org/?id=1539`.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

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

Abdoulaye Abou Diakité, Guillaume Damiand, and Gilles Gesquière. Automatic semantic labelling of 3d buildings based on geometric and topological information. In *3DGeoInfo 2014*, 2014.

Jürgen Döllner, Thomas H Kolbe, Falko Liecke, Takis Sgouros, and Karin Teichmann. The virtual 3d city model of berlin-managing, integrating, and communicating complex urban information. In *Proceedings of the 25th Urban Data Management Symposium UDMS*, volume 2006, 2006.

J. Stoter F. Biljecki, H. Ledoux. An improved lod specification for 3d building models. *Computers, Environment and Urban Systems, Under review*, 2016.

Fileformat.info. Wavefront obj file format summary, 2015. URL `http://www.fileformat.info/format/wavefrontobj/egff.htm`.

Claus Nagel Karl-Heinz Häfele Gerhard Gröger, Thomas H. Kolbe. Ogc city geography markup language (citygml) encoding standard. Technical Report OGC 12-019, Open Geospatial Consortium, 2012. URL `https://portal.opengeospatial.org/files/?artifact_id=47842`.

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

André Henn, Christoph Römer, Gerhard Gröger, and Lutz Plümer. Automatic classification of building types in 3d city models. *GeoInformatica*, 16(2):281–306, 2012.

INSPIRE. D2.5 inspire generic conceptual model, version 3.4rc3. Technical Report OGC 12-019, INSPIRE, 2013. URL `http://inspire.ec.europa.eu/documents/Data_Specifications/D2.5_v3.4rc3.pdf`.

Thomas H Kolbe, Gerhard Gröger, and Lutz Plümer. Citygml: Interoperable access to 3d city models. In *Geo-information for disaster management*, pages 883–899. Springer, 2005.

Florent Lafarge and Clément Mallet. Creating large-scale city models from 3d-point clouds: a robust approach with hybrid representation. *International journal of computer vision*, 99(1): 69–85, 2012.

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

Marc-O Löwner, Joachim Benner, Gerhard Gröger, and Karl-Heinz Häfele. 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, 2013.

Oded Maimon and Lior Rokach. *Data mining and knowledge discovery handbook*, volume 2. 2005.

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

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

Charalambos Poullis and Shi You. Automatic reconstruction of cities from remote sensor data. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2775–2782. IEEE, 2009.

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

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

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

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

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

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

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

Peter van Oosterom and Sisi Zlatanova. *Creating Spatial Information Infrastructures: Towards the Spatial Semantic Web*. CRC Press, 2008.

Peter Van Oosterom, Siyka Zlatanova, and Elfriede Fendel. *Geo-information for disaster management*. Springer Science & Business Media, 2006.

Yannick Verdie, Florent Lafarge, and Pierre Alliez. Lod generation for urban scenes. Technical report, Association for Computing Machinery, 2015.

Vivek Verma, Rakesh Kumar, and Stephen Hsu. 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, 2006.

Wikipedia. Normal (geometry), 2015a. URL `https://en.wikipedia.org/wiki/Normal_(geometry)`.

Wikipedia. Bayesian probability, 2015b. URL `https://en.wikipedia.org/wiki/Bayesian_probability`.

Xuehan Xiong, Antonio Adan, Burcu Akinci, and Daniel Huber. Automatic creation of semantically rich 3d building models from laser scanner data. *Automation in Construction*, 31: 325–337, 2013.

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

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