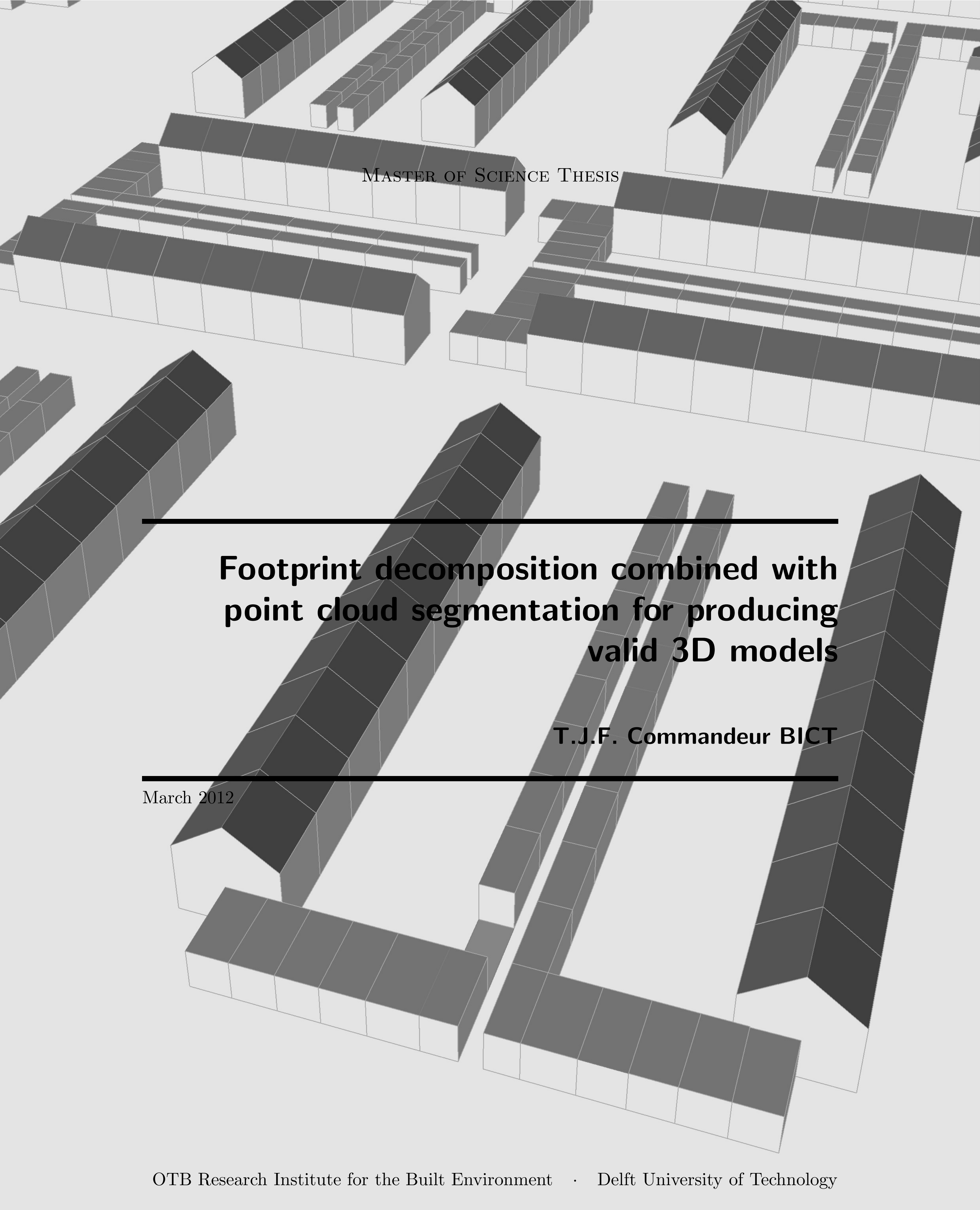


Master of Science Thesis

Footprint decomposition combined with point cloud segmentation for producing valid 3D models

T.J.F. Commandeur BICT

March 2012

OTB Research Institute for the Built Environment · Delft University of Technology

TU Delft

OTB Research Institute for the Built Environment

Delft University of Technology

Footprint decomposition combined with point cloud segmentation for producing valid 3D models

Master of Science Thesis

For obtaining the degree of Master of Science in Geomatics
at Delft University of Technology

T.J.F. Commandeur BICT

March 2012

OTB Research Institute for the Built Environment · Delft University of Technology

Delft University of Technology

Delft University Of Technology
Department Of GIS Technology

The undersigned hereby certify that they have read and recommend to the OTB Research Institute for the Built Environment for acceptance a thesis entitled **"Footprint decomposition combined with point cloud segmentation for producing valid 3D models"** by **T.J.F. Commandeur BICT** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: March 2012

Head of department: ______________________________
prof. dr. ir. P.J.M. van Oosterom

Supervisor: ______________________________
dr. H. Ledoux

Reader: ______________________________
dr. ir. B. Gorte

Reader: ______________________________
ir. E.J. Vriend (iDelft)

Abstract

For the creation of three-dimensional (3D) city models, extrusion of building footprints is widely applied. Extrusion results in block shaped buildings. The main problem with these 3D models is that they do not represent height differences within a building. This thesis presents a method to improve building footprints by subdivision into parts describing height differences and roof shapes. These improved building footprints are used for creating 3D models by separate extrusion of each of these parts. Applying this method results in 3D building models with multiple heights. Several subjects are discussed in this thesis: generalization, decomposition, segmentation, 3D geometry reconstruction and their validation.

This research is based on the combination of two existing methods, one for generalization and decomposition of building footprints, and one for segmentation of point clouds. These methods are extended and problems are solved by including: (1) direction of lines, (2) best fitting line, (3) adjacent buildings, (4) preservation of holes, (5) handling of slant lines, (6) quality statistics and (7) validation. The existing generalization method creates topology errors which are reduced by including knowledge from adjacent buildings.

Decomposed building footprints are merged based on the segmented point cloud. The resulting decomposition is a subdivision describing multiple roof shapes and height jumps, based on linear features in the building footprint. Reconstruction of the actual 3D geometry is performed by extrusion of the decomposition cells. The resulting reconstruction is a Level of Detail 1 (LoD) model with height differences. To test the usability of the improved footprints for reconstruction of a LoD2 model, parametric shape fitting is applied. These tests are executed using a small set of roof shapes proving the usability. All reconstructed 3D building geometries are validated using Oracle Spatial 11g.

In order to analyze, test and improve the developed algorithms, a prototype is implemented in C++. This prototype is tested intensively with several real-world data sets. Results from these tests are proving proper functioning of the developed method and support the conclusions.

Acknowledgements

I wish to express my upmost gratitude to the people who directly, or indirectly helped me to complete this thesis. First of all I would like to thank my supervisors prof. dr. ir. Peter van Oosterom and dr. Hugo Ledoux for their time and guidance. Hugo helped me a lot with his insights and knowledge on the research topic. Also thanks go out to dr. ir. Ben Gorte for supplying me with helpful comments on the report. Dr. ir. Friso Penninga has made available his support by supplying me a 3D model of Den Haag for comparison.

It is a pleasure to thank the company iDelft, especially Erik Vriend, for providing office space, an inspiring setting and their much needed support. The time spent at their office has provided me a better acquaintance with the world of Geomatics combined with a nice time and many interesting conversations.

I am indebted to my friends Mark and Martin for inspiring me to start the Master Geomatics which I, by finishing this thesis, now have successfully completed. Them and other friends have been an unbelievable support in any possible way. I also want to show my gratefulness for the luxurious Wednesdays which where the highlight of my week.

My deepest gratitude is owed to my parents, brother, sisters and other family for always being there for me. Without any doubt they have given me the trust, strength and support needed to fulfil this thesis. Special thoughts go out to my grandfather, who I know would have been more then proud of me for obtaining my Master degree.

The last two years at the Delft University of Technology have been an extraordinary experience. It showed me that the possibilities are endless if you are willing to work for it. As Confucius once said: *"By three methods we may learn wisdom: First, by reflection, which is noblest; second, by imitation, which is easiest; and third by experience, which is the bitterest"*.

Delft, The Netherlands
March 2012

T.J.F. Commandeur BICT

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

3D	Three-dimensional
AHN2	Actueel Hoogtebestand Nederland versie 2 (*National Height model of the Netherlands version 2*)
ASCII	American Standard Code for Information Interchange
BAG	Basisregistraties Adressen en Gebouwen (*Basic registrations Addresses and Buildings*)
DSM	Digital Surface Model
DTM	Digital Terrain Model
GEOS	Geometry Engine Open Source
GIS	Geographic Information System
ISO	International Organization for Standardization
k-NN	k-Nearest Neighbor
LiDAR	Light Detection and Ranging
LoD	Level of Detail
OGR	OGR Simple Features Library
RAG	Region Adjacency Graph
RANSAC	Random sampling consensus
SQL	Structured Query Language
TIN	Triangular Irregular Network

Chapter 1

Introduction

Three-dimensional (3D) city modeling is the creation, storage, maintenance and analysis of urban objects in a digital representation. Created 3D city models are for example used to calculate volumes of buildings [Köninger & Bartel, 1998] which are useful for cadastres, noise pollution calculations [Schulte & Coors, 2008] and in emergency response [Lee & Zlatanova, 2008]. Besides calculations, 3D city models are also used for visual representation of an urban environment.

Creating digital city models by hand is a complex and time consuming task. When reconstructing areas with up to 100 buildings this still is a doable task. When areas are becoming larger, like complete cities, the amount of work per building becomes problematic. When this process is automated, both processing time and labor are decreased. However, after more than two decades of research on the automatic reconstruction of 3D city models, there still not seems to be a complete solution.

Existing approaches try to achieve an automated reconstruction process which solves for all cases. Buildings can be complex structures which are difficult to model, especially in an automated process. Some claim their approach solves for up to 90% of the built environment after visual inspection [Taillandier, 2005]. Statements on the percentage of reconstruction cannot be compared directly throughout publications since a 'solved building' depends on the inspector. Buildings that are not reconstructed during the automated process can be automatically flagged for manual editing. Besides not being reconstructed, a building can also be reconstructed incorrectly. A quality parameter can be appointed to the building to represent the level of certainty.

This chapter starts by giving an introduction and motivation of the problem in Section 1.1 followed by the objectives of this thesis research in Section 1.2. Afterwards, the scope of this thesis is summarized in Section 1.3 and finally in Section 1.4 the structure for the rest of this document is uncovered.

1.1 Motivation

The company iDelft is manufacturing various products with Geo-information including 3D city models. The starting point of this master thesis research is the request from iDelft to design and implement an automatic 3D building reconstruction algorithm which outputs valid 3D geometries [Ledoux et al., 2009; Kazar et al., 2008]. This request is based on findings of shortcomings with available methods; not having generalization, buildings containing multiple heights and the handling of height jumps. Examples of these shortcomings are given in the following.

Different 3D model representations, called Level of Detail (LoD), are described in CityGML, the international standard for representing and storing 3D city models [Gröger et al., 2008]. The LoD describes the accuracy and detail of the created model. The definition used within this thesis for LoD1 is the extrusion of building footprints -explained in the next paragraph- to a calculated height, producing "block-shaped" polyhedra describing buildings with flat roofs. For LoD2 the definition is buildings with walls, roof shapes and semantics (attributes) describing the models properties. Examples of LoD1 to LoD3 can be seen in Figure 1.1. LoD3 describes architectural models which have detailed roof and wall structures including doors, windows and balconies.

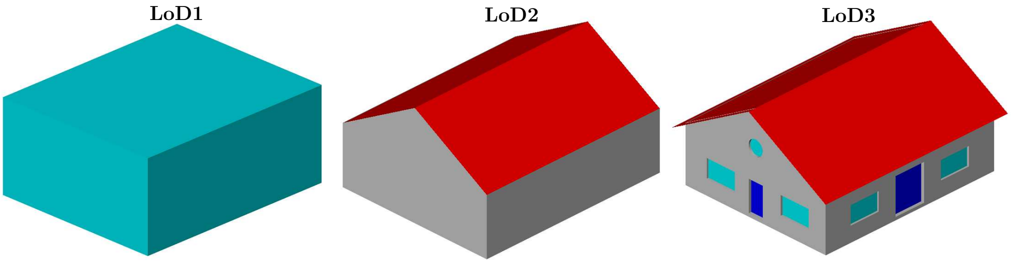


Figure 1.1: Different Level of Detail models, from LoD 1 to LoD 3. Figure from Häfele [2010].

Building footprints describe the outline of a building. Various methods are available for obtaining building footprints, like geodetic measurements on location but also aerial photography. The biggest difference between the various sources is the decision of what is considered the outline of a building. The one source describes the walls of a building as the footprint in contrast to others where roof outlines are considered the footprint of a building. An example of building footprints is shown in Figure 1.2 where these are overlaid on top of an aerial photograph. This example directly shows the difference between the two, this is because they are different representations but also since the aerial photograph is not corrected for distortion. The definition of a building footprint used throughout this thesis is that it describes the outline of the roof.

For the creation of a LoD1 city model, one of the simplest ways is extrusion of building footprints as shown by Ledoux & Meijers [2011]. However building footprints often describe complex buildings, where complex is a building consisting of multiple heights and/or roof shapes. When an extrusion is performed on such a building footprint, result would not be satisfactory. When a building has multiple roof heights, some parts would be consistently too low and others too high due to averaging the multiple heights. The same problem arises when the building consists of multiple roof shapes, a gabled roof of a

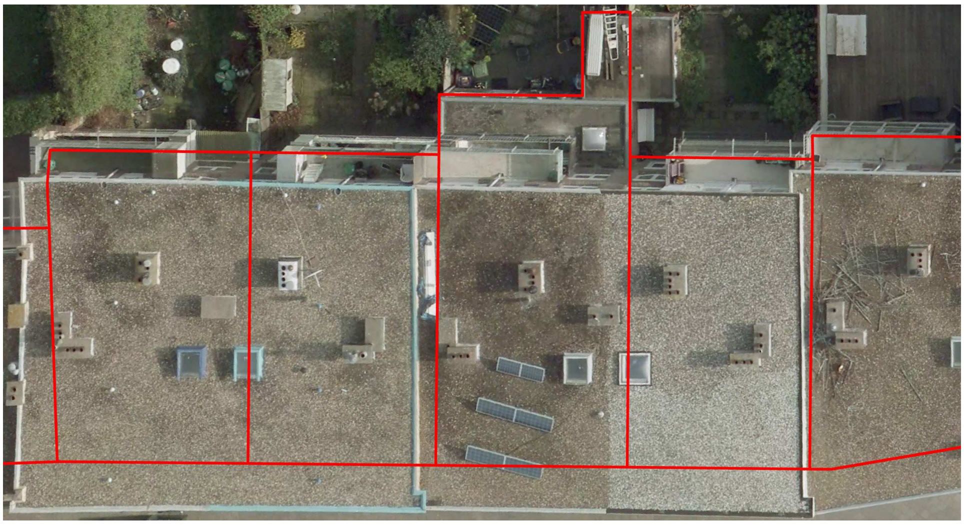


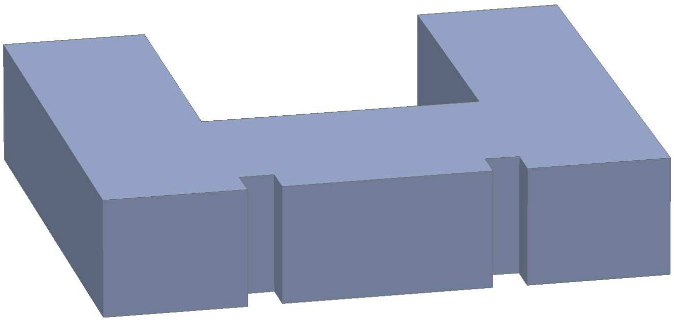
Figure 1.2: Building footprints overlaid on an aerial photograph, illustrating the building outline. Data is courtesy of the municipality of Den Haag.

house with the flat roof of a garage. Most complex building shapes can be created from a combination of simple roof shapes. When a building footprint is split given these multiple shapes or heights and extrusion is applied to each part of the footprint, the end results improve. An example of the improvement using split building footprints together with an aerial photo of the building can be seen in Figure 1.3. Within this thesis the splitting of the building footprints is attempted which gives results as shown in Figure 1.3b.

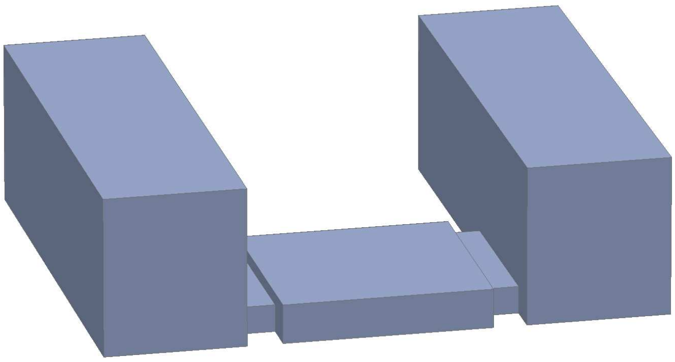
Building footprints can include details which influence the decomposition in a negative matter, therefore generalization is needed. However careless generalization can introduce gaps or overlap between building that where originally touching which has to be prevented. Besides the need for generalization within the context of decomposition, the requirements of the 3D city model could also be to obtain a generalized reconstruction with less details. The research of Kada & Luo [2006] shows that generalization and decomposition of building footprints can be combined using line equations. The side effect of this method is the lack of preserving topology. Within this thesis research this is attempted to be solved by extending the generalization process with the inclusion of adjacent buildings.

The generalization of building footprints results in output footprints different than the original. To identify the impact of the changes in the footprints, quality statistics are used. After the calculation of the quality statistics these are combined into a single quality parameter. Using this total quality parameter, building footprints can be automatically flagged to identify potential unusable footprints for the 3D reconstruction process. These flags exclude the buildings from further processing if necessary or marking for visual inspection or manual editing. This improves the automated process since incorrect 3D reconstructions are automatically flagged and do not need to be found by hand.

Oude Elberink & Vosselman [2009] state that reconstructing a building based on *a priori* models, like parametric shape fitting, lacks flexibility compared to reconstruction based on data. A combination of model based and data driven approaches is best for a reconstruction method. Segmentation of the point cloud data can improve the parametric shape fitting and decomposition as mentioned in the future work of Kada & McKinley [2009] where Oude Elberink & Vosselman [2009] points out that decomposition of the building footprints improves the segmentation process. Given these conclusions this thesis investigates the combination of two existing methods, generalization and decomposition as described by Kada & McKinley [2009] together with the segmentation of point clouds as described by Oude Elberink & Vosselman [2009].



(a) Reconstruction using extrusion of an untouched building footprint



(b) Reconstruction using extrusion of a split building footprint

(c) Bird's eye view of OTB Research Institute. The imagery is copyrighted by Picometry International and Microsoft (2012)

Figure 1.3: 3D reconstruction of a building and its bird's eye view.

1.2 Objectives

This thesis concentrates on the automatic process for the reconstruction of 3D city models. The reconstruction is based on building footprints and point clouds, which can come from various sources with varying quality and accuracy. The algorithm needs to be flexible thus able to process these different kinds of data sources. The algorithm should be able to process large data sets, it should be implemented in such a way that it is scalable. This means that the reconstruction is not limited to a maximum number of buildings.

The main research question of this thesis is:
Does the combination of decomposition and point cloud segmentation improve building footprints for the creation of valid 3D geometries?

In order to answer this research question the main objective is:

- Design and implement a building footprint generalization and decomposition algorithm in 2D combined with point cloud segmentation for the creation of valid 3D geometries.

To be able to complete the main research objective, it is subdivided into sub-research objectives:

1. Study the available approaches on footprint decomposition, footprint generalization and parametric shape fitting.
2. Compare available methods for footprint decomposition and generalization.
3. Analyze results from the chosen methods and identify their problem cases.
4. Design and implement a method for footprint decomposition and generalization combined with point cloud segmentation.
5. Design and implement a method for creating valid 3D geometry and validate using Oracle spatial.
6. Test the designed method for the usability with parametric shape fitting.
7. Perform quality assessment of the created model.
8. Compare the prototype results with existing model from another method.

To accomplish these sub-research objectives, decomposition is used for splitting building footprints to improve the creation of LoD1 models. The improvements describe multiple heights or roofs within a single building. The aim is creating a decomposition which divide the building footprint into cells that form these multiple heights. Since decomposition is creating more subdivisions then there are roof shapes, cells need to be merged. Merging the cells is done by using knowledge from the segmented point cloud. The output of the 3D reconstruction is geometrically validated to test the functioning of the algorithm. Additionally it is tested if the created decomposition is usable for the application of parametric shape fitting to create a LoD2 model by using the segmented point cloud.

The building footprints are necessary for application of the segmented point cloud. If a segment is not known to be part of a building it can describe other objects like trees or cars. The identification of the kind of object a segment describes without external information is impossible [Oude Elberink, 2010]. Hence building footprints are used to identify point cloud segments as belonging to a building and include them in the reconstruction.

The algorithm is analyzed, tested and improved by implementing it in C++ using external libraries. The used libraries are the OGR Simple Features Library (OGR)[1] for input and output and Geometry Engine Open Source (GEOS)[2] for some geometric operations. The implementation is tested intensively with several data sets of different sources and accuracies.

1.3 Scope

The generalization and decomposition of building footprints combined with the segmentation of point clouds in order to produce 3D models, is the main subject of this thesis research. The creation of a prototype is used for the validation and testing of developed ideas. The acquisition of data and therewith the creation of building footprints are not dealt with in this thesis.

The generalization and the decomposition of building footprints is completely covered and reconstructed where the point cloud segmentation is covered in theory and an existing implementation is used. These two approaches are combined for the improvement of the footprint decomposition. Adjacency knowledge of the segments from the point cloud is not used neither covered.

The output of the created algorithm is a LoD1 3D city model with valid geometries. Several experiments are done for testing the usability of the decomposition with parametric shape fitting using the segmented point cloud. This results in a LoD2 model by implementing detection and geometric reconstruction of a limited set of roof shapes.

Not everything is reconstructed from literature, improvements are made by including additional parts. Some additions are: including adjacent buildings in the generalization process, handling of slant lines in the decomposition, filtering created segments and correcting roofs using adjacent buildings.

Things that are not covered within this thesis research:

- Decomposition of circular elements.
- Detection and fitting of curved surfaces.
- Overhanging roofs.
- Fitting of building geometries within the terrain.
- Adjacency knowledge from segments.

The scope of this research is simplified into a flow diagram and shown in Figure 1.4.

[1]Information on OGR can be found at http://www.gdal.org/ogr
[2]Information on GEOS can be found at http://geos.osgeo.org

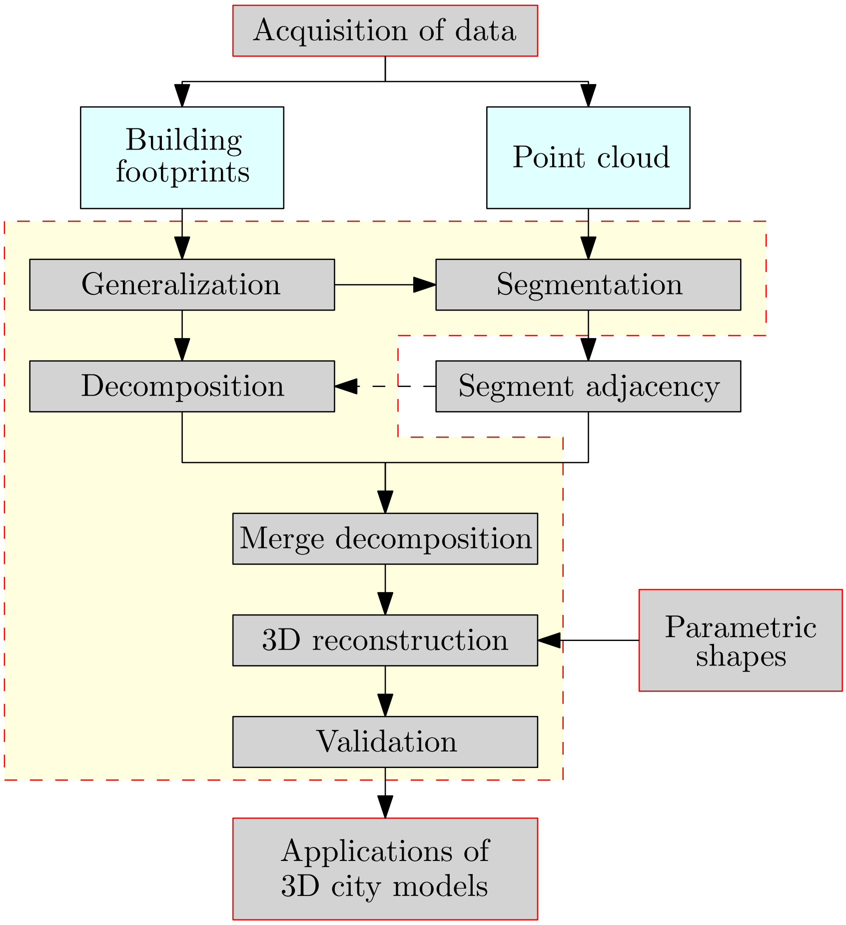


Figure 1.4: Scope of this thesis research, yellow selection, related to the possible complete workflow.

1.4 Outline

The structure of this research is based on the workflow of the reconstruction process as shown previously in the Scope and structured as follows:

Chapter 2 describes the related work in the context of 3D city modeling specifically on the reconstruction of 3D building models. From this related work some observations are made and conclusions are drawn on which the further research is based.

Chapter 3 covers the generalization and decomposition of building footprints using line equations. The consideration of adjacent buildings when generalizing building footprints is explained including the improvements. Besides this, several quality parameters are described which are used for identifying the generalization quality of the decomposed building footprint.

Chapter 4 explains the segmentation of a point cloud into planar faces. The used segmentation algorithm is described including its parameters. Also extra filtering options are provided for the use within this thesis research to exclude outliers.

Chapter 5 presents the combination of both the decomposition and segmentation by introducing merging of the decomposed footprints based on the created segmentation. From the merging of the decomposition the actual geometric reconstruction is covered in two parts, LoD1 using extrusion and LoD2 reconstruction using a limited amount of roof shapes to show the usability for parametric shape fitting. This chapter also covers the constraints on valid geometries and the validation using Oracle Spatial.

Chapter 6 explains the implementation of the algorithm into the prototype and gives more technical details of the solution. The experiments executed for determining the most optimal workflow of the algorithm are discussed here together with the results of the quality parameters and the validation. Also the data sets that are used for testing and experiments are explained and their differences are shown. In between the created data sets are compared to a 3D model from an existing method.

Finally Chapter 7 outlines the conclusions drawn from this research, recommendations based on the conclusions and future work related to this thesis.

Chapter 2

Related Work

This chapter gives an overview of the research in the area of 3D city modeling, particularly the reconstruction of 3D building models. From this research a selection is highlighted which has relevance to this thesis.

In the last two decades there has been a lot of research about 3D city modeling and automation of the building reconstruction process, as described in the research overview of Haala & Kada [2010]. Most approaches are based on aerial photography, building footprints and/or point clouds. Haala & Kada [2010] categorize the methods into three categories: (1) reconstruction based on Digital Surface Model (DSM) simplification, (2) segmentation of point clouds and (3) parametric shapes. All three categories use point clouds to determine heights of buildings. Point clouds are acquired from different sources like Light Detection and Ranging (LiDAR) or image matching in stereo photography otherwise referred to as photogrammetry [Hirschmüller & Bucher, 2010]. A short summary of what these three categories describe is explained in the following.

Digital Surface Model simplification These methods use a meshed DSM which contains high detail data. The DSM does not describes the 'bare surface' but only other features like houses, trees, cars etc. This DSM is a meshed surface described by a (TIN). The approach is to simplify the mesh and thereby creating different LoDs for the DSM as presented by Wahl et al. [2008]. As they describe there are various ways to simplify such a mesh. One method is clustering edges or vertices to reduce their numbers. These methods rely on topology constraints to assure the original building structure is kept. It also relies on geometric criteria which are difficult to identify. The incorrect identification of these criteria, combined with the possibility of incomplete data result in incorrect simplifications.

Segmentation of point clouds Segments are groups of points, combined based on their relationship like corresponding surface and normal direction. The segments can be created with a large number of approaches like 3D Hough transform, region growing etc., all described by Haala & Kada [2010]. In Figure 2.1 an example of point cloud segmentation is shown. When the segmentation is done the next step is feature recognition, creating topological understanding of the relation between segments. The topology between the segments is commonly described in a region adjacency graph (RAG). The segments are described by nodes which are connected when they share a common border [Oude Elberink, 2009]. An approach using target graph matching for typical roof patterns to describe the connectivity of segments is presented by Oude Elberink & Vosselman [2009] while an approach with Hough transform using LiDAR point clouds and building footprints is described by Vosselman & Suveg [2001].

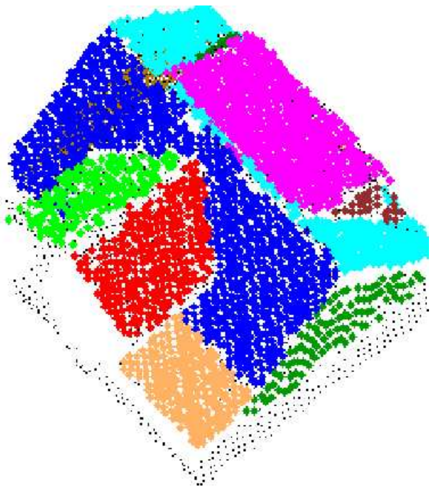


Figure 2.1: Segmentation of a building point cloud. Figure from Dorninger & Pfeifer [2008].

Parametric shape fitting Parametric shapes describe models of common roof types which are fitted on top of the footprint of a building, some common roof types are shown in Figure 2.2. A 3D point cloud is overlaid on top of the fitted roof to check if it is the correct shape. Not all parametric shapes can be placed on top of arbitrary shaped footprints. A footprint can describe multiple roof shapes like a house with a gabled and a garage with a flat roof. Therefore extruding the building to a generalized eaves height can be performed to create the walls of a building model, where the roof shape is found by parametric shape fitting [Durupt & Taillandier, 2006]. Another approach is to make a decomposition of the footprint to fit roof shapes onto every decomposed part [Kada & McKinley, 2009]. Decomposition is completely subdividing a footprint into a set of non-overlapping cells.

Existing implementations Some existing methods explained in the coming sections describe the semi- or fully-automatic production of 3D city models. The categories as described before are all covered by these methods. Some of them are a combination of these categories trying to overcome shortcomings of the one with the strengths of another. Finally an outline of generalization approaches is given.

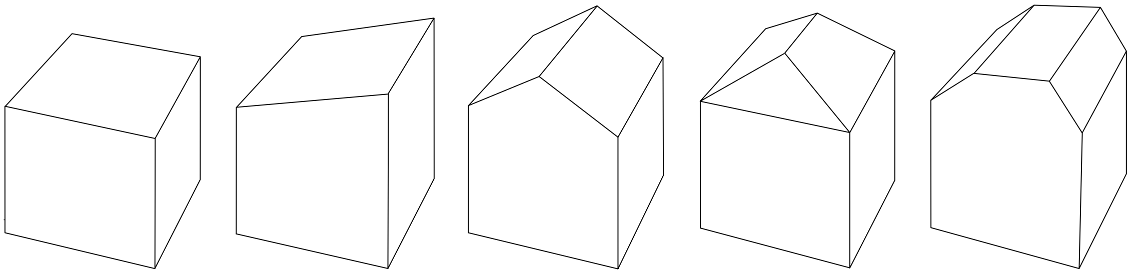


Figure 2.2: Flat, shed, gabled, hipped and Berliner roof shape. Figure from Kada & McKinley [2009].

2.1 Parametric Shape fitting using a Digital Surface Model

For the reconstruction of a LoD2 model of a city, the method of Taillandier [2005] uses parametric shapes which are derived from the building footprint and a DSM for the computation of the exact shape parameters and correlation of the shape.

The building footprint is extruded to a generalized eaves height and assumed horizontal. For each segment of the eaves a plane is derived going through this segment and orthogonal to it. This results in a set of 3D planes intersecting the building footprint as shown in Figure 2.3. From the 3D planes a graph is produced by intersecting all planes. By exhaustive search a result set is found of possible roof shapes, the parametric shapes. This result set is filtered by removing models which for example have a face with a small area or two edges with a small angle.

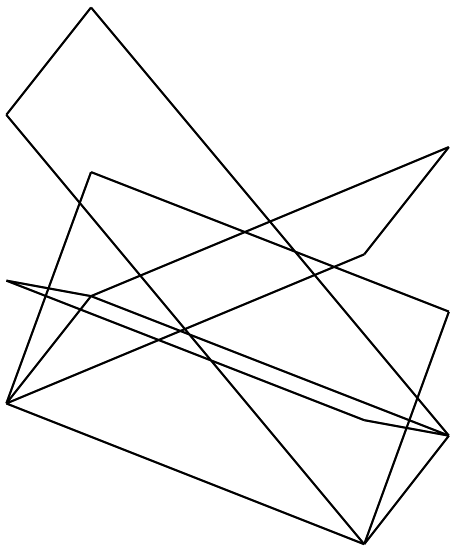


Figure 2.3: Roof planes derived from the building footprint. Figure from Taillandier [2005].

Correlation of the resulting models is done with a DSM constructed from high resolution imagery. The exact parameters for the models, the slope and the altitude, are computed by minimization on the correlation DSM. With these parameters a correlation score based on each pixel of the DSM is calculated for each model. The model with the highest score is fitted on the extruded eaves.

With the goal of improving this method Durupt & Taillandier [2006] describe a different form of the 3D plane extraction. Since the constraints of the original plane extraction have a great influence and forcing symmetry they obtain the planes using a random sampling consensus (RANSAC) algorithm. The exact parameters of the models are extracted with the RANSAC algorithm before utilizing the exhaustive search. By using this algorithm they obtain better results as well as asymmetric shapes.

The results with the original approach are about 85% with an acceptance improvement of about 4% when using the RANSAC algorithm, giving a total of 89%. The approaches perform best on symmetric roof shapes with generalized eaves and a centered ridge. Especially when there are height discontinuities or complex shapes the algorithm fails.

2.2 Cell Decomposition for Parametric Shape fitting

A method for splitting building footprints, based on cell decomposition taken from Solid modeling, is proposed by Kada & McKinley [2009]. After the decomposition of the footprints parametric shape fitting is applied given a list of *a priori* models. The decomposition of a building footprint can be seen in Figure 2.4.

Since the complexity and details of building footprints can be high, a generalized form of the building outline is made. The generalization removes small details from the footprint that could be problematic for the decomposition and shape fitting. This generalized footprint is used to create a decomposition of nonintersecting rectangular cells. The cells are rectangular since parallelism and right angles are forced. The decomposition result is used to compare the parametric roof shapes with the point cloud. For every decomposition cell the points that fall within are selected and the best fitting parametric shape is computed based on these points. Different roof shapes are defined *a priori*, some examples are already given in Figure 2.2. A roof shape is selected for every decomposed part of the footprint which are glued together to form the complete roof structure for the building. The roof is then placed on top of the original footprint for the final result.

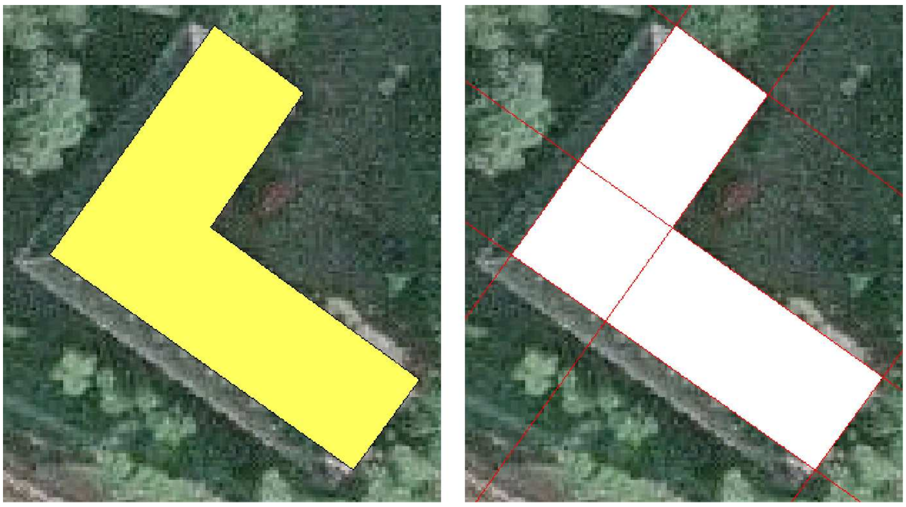


Figure 2.4: Decomposition of a building footprint. Figures from Kada & McKinley [2009].

2.3 Footprint improvement using Split and Merge

The creation of a building model, using the method described by Durupt & Taillandier [2006] and explained before, fails when having complex shapes or height discontinuities. In order to improve this method Vallet et al. [2011] is proposing a split and merge of the building footprint before reconstruction. Since only splitting a footprint could result in too many small areas a merge is necessary to keep the number of resulting parts as low as possible.

From the footprint a set of main directions is calculated by clustering the angles of the boundary. The number of main directions is kept as small as possible by using elimination like summed length smaller then a given threshold. Both the vertical and horizontal gradients of the DSM are used to find plausible split lines. An energy function is used to compute if a line is actually a split line. The footprint is recursively split according to lines of minimal energy and which are parallel to one of the main directions. Since the splitting creates cuts that are not needed, because of local height discontinuities, a merging step is done.

The merging of the footprint is performed with a greedy algorithm. All possible merges are calculated together with their merging scores. A queue is created based on the height of the score. The merge with the highest score is performed and all merges related to the merged parts are removed from the queue. All merges related to the new part are calculated, as well as their scores, and added to the queue.

An example of a split and a merged building footprint using this method is shown in Figure 2.5. Even though this method is proved using one reconstruction method, Vallet et al. [2011] state that this method is usable for different reconstruction methods. Since the results are a decomposition of the original footprint, and not a complete reconstruction, this can indeed be used as input for other methods.

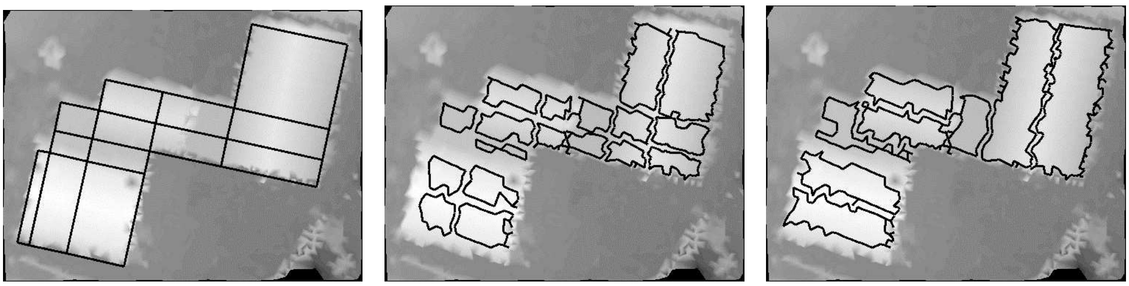
Figure 2.5: Split result (left) and merge result (right). Figures from Vallet et al. [2011].

2.4 3D Hough Transform

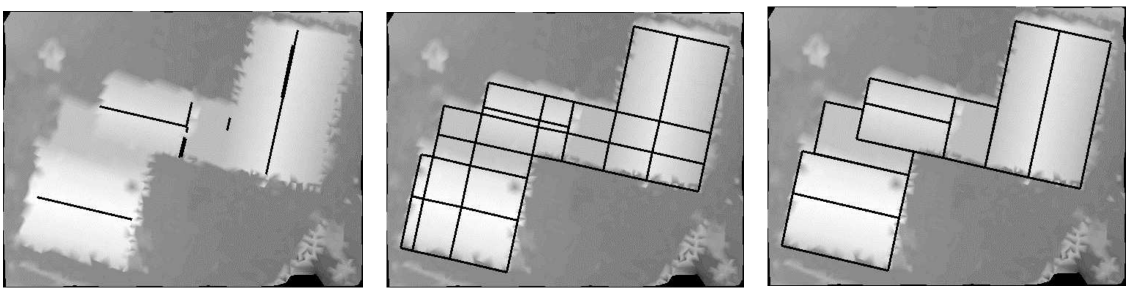
The automatic detection of roof faces by analyzing point clouds can be performed in multiple ways. The method as described by Vosselman & Suveg [2001] uses the well known Hough transform extended to a 3D transformation [Vosselman, 1999]. In order to improve the detection of roof face outlines, the building footprint is used.

The building footprints are split into rectangles since a lot of buildings can be modeled using a union of multiple simple roof shapes based on a rectangular area. Splitting of the footprints is performed by the extension of the footprint lines that intersect in concave corners. Not many of the created rectangles correspond directly to height jumps or ridge heights. The 3D Hough transform is therefore used to search for missing intersection lines or height jumps.

The 3D Hough transform produces better results when the number of points per plane is large compared to the number of planes to be extracted which should be small. For this reason the 3D Hough transform is applied to the points within every rectangular part of the footprint created by the splitting. The detected faces are merged when they belong to the same plane. The detection of intersection lines from the detected faces is as simple as testing if the faces are located near the intersection line. Height jumps are given the heuristic that they are parallel to one of the boundaries of its segment. If the height difference is significant this line is used to split the segment. An example of the steps can be seen in Figure 2.6.



(a) Split building footprint (b) Planes from 3D Hough transform (c) Planes merged



(d) Intersections and height jumps (e) Split footprint with intersections and height jumps (f) Final result

Figure 2.6: Roof reconstruction based on 3D Hough transform. Figures from Vosselman & Suveg [2001].

The reconstruction of roof shapes by using this methods seems promising, however Vosselman & Suveg [2001] state that the reconstruction works correct for simple buildings with flat or gable roofs but fails on more detailed larger buildings. Most of the errors in the models are due to dealing with the segments separately instead of the complete building at once. The Hough transform does not detect faces when the number of points is very small. This is not only because of small faces but also with missing point cloud information like low reflection or water on horizontal faces. Given these factors it shows that the method is relying heavily on the point cloud which introduces problems when the data is incomplete.

2.5 Target Graph matching

As described by Oude Elberink [2008] there are multiple problems for building reconstruction from detailed point clouds. On the one hand there is the data driven problems and on the other hand there are model driven problems. The data driven problems are that of lack of points due to low reflectance or no reflectance from water on horizontal roofs. In the absence of data the reconstruction fails because of missing faces or intersection lines. Using a model driven approach combines valid building shaped to construct a 3D model. The reconstruction is bound to the building shapes described, complex buildings could therefore be to complicated to reconstruct.

An integration of data driven and model driven approaches is developed by Oude Elberink [2009]. The data driven approach gives the freedom to model everything available from the data, given the knowledge of model driven approaches to rule out illogical combinations and solve for missing data. The integration is performed by target based graph matching which is matching the features found from the data to the described models.

An input LiDAR point cloud is segmented using a surface growing algorithm based on a 3D Hough transformation and least squares fitting of points. From the found segments intersection lines and height jumps are derived. These lines features are labeled accordingly since they describe neighboring relations of segments. Given the intersection lines and height jumps, a topology graph is created describing the relationship between segments, see Figure 2.7. The roof topology graph is matched to a list of target graphs which are described by the model. The matches are combined into a resulting topological description of the roof which is combined with the building footprint and adjusted accordingly.

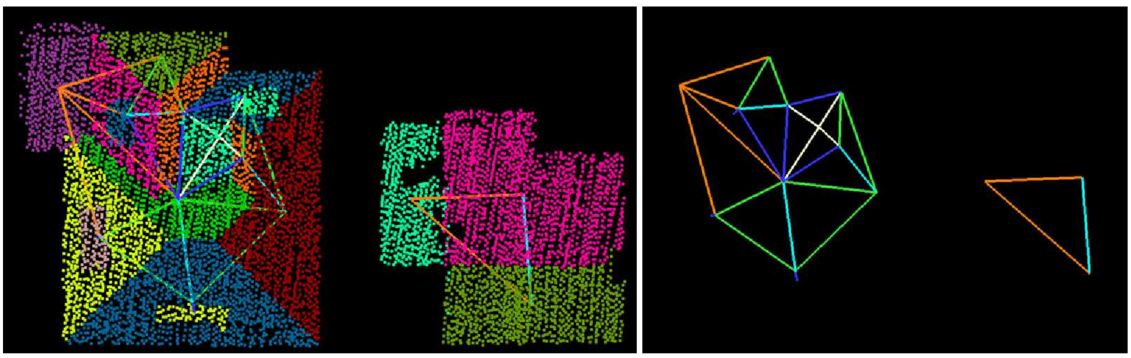


Figure 2.7: Roof topology graph with and without segments (top view). Figure from Oude Elberink & Vosselman [2009].

The method produces building models from which about 20% have incomplete matches between the found segments and the target graphs. However from the total amount of buildings only 8% fails to be fitted to the original point cloud. The incomplete matches are thoroughly described by Oude Elberink & Vosselman [2009] and in 40% of the cases this is due to missing segments to complete a target match. The missing segments are related to the segmentation parameters, altering these parameters could solve for the incomplete matches but also introduce problems in other parts of the model.

2.6 Generalization of building footprints

The generalization of building footprints is the creation of less detailed and a more abstract representation of a geographic data set. It is based on reducing the number of lines which describe the building outline. An example of building footprint generalization is shown in Figure 2.8. The line reduction can be performed in several ways like removing intermediate vertices, averaging lines and creating shortcuts. The selection of which lines to combine and replace by a generalized line is the basis of generalization.

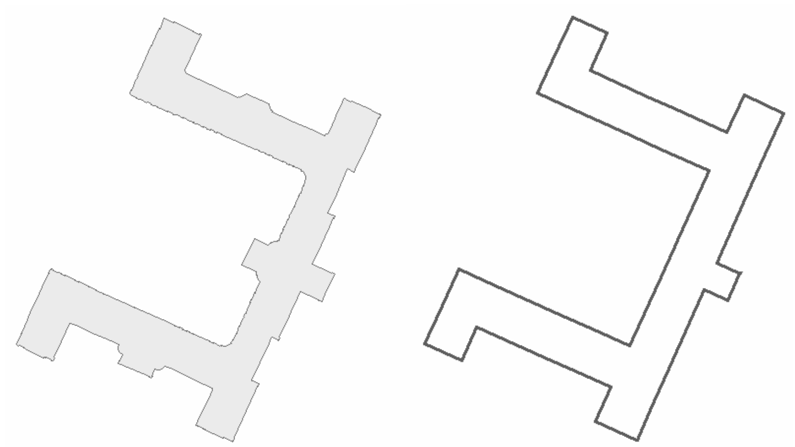


Figure 2.8: Original building (left) and generalized building (right). Figures from Kada & Luo [2006].

The removal of intermediate vertices is done by selecting the minimum subsequence of the original vertices and using the outer vertices to create the generalized line. This method is based on a distance threshold between the original and generalized line [Douglas & Peucker, 1973]. Line generalization is not usable for building footprint generalization since the original vertices are uses. Sometimes a new vertex must be introduced for the creation of a generalized line. The generalization should therefor be based on edges rather then vertices. Besides selecting lines by a distance threshold, other methods are available which use angles [Chen et al., 2005], areas [Bose et al., 2006] or topological relations [Berg et al., 1998].

The described criteria for the selection of generalizable lines, when combined, have proven to produce useful results. In the following sections three generalization methods are described which use -a selection of- these criteria for building footprint generalization.

2.6.1 Generalization of building footprints using half-spaces

The method of Kada & Luo [2006] describes generalization of building footprints using half-spaces. Simply put a half-space is a line which divides space in half. All lines within the building footprint are used as half-space primitives which together result in the same outline, Figure 2.9 top. The number of half-spaces need to be reduced to obtain a generalized result, Figure 2.9 bottom.

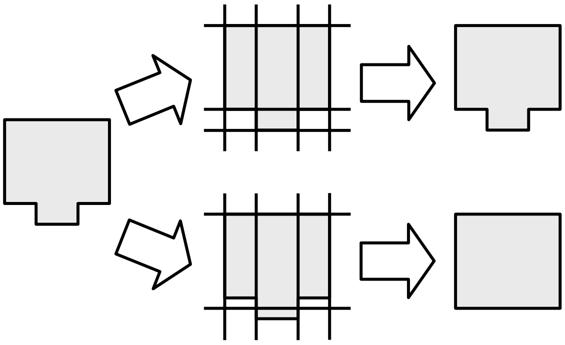


Figure 2.9: Original building (left), cell decomposition (middle) and result (right). Figures from Kada & Luo [2006].

The half-spaces are created by aligning them with one or more lines on the same straight line. The half-space primitives are then merged based on buffers and pairwise comparison of the straight lines. If the lines fall within a given distance threshold and are almost parallel, the half-spaces are merged. All short lines that are completely covered by the buffer are also added without considering their orientation. This process is repeated for all line segments until nothing is left to be merged. From the created buffers new half-spaces are calculated which result in approximation of that part of the original footprint.

The created half-spaces are used to split space, since splitting infinite space is unsuitable two times the buildings bounding box is used. When this space is split the cells belonging to the building need to be identified. A grid of 2D points is distributed over the minimum bounding rectangle of the cell where the points outside are discarded. The fraction of these points inside the original and the total number inside the cell is calculated, when this value is below 50% the cell is discarded. When all cells are traversed, the generalized footprint is created by performing a union on the remaining cells, see Figure 2.10.

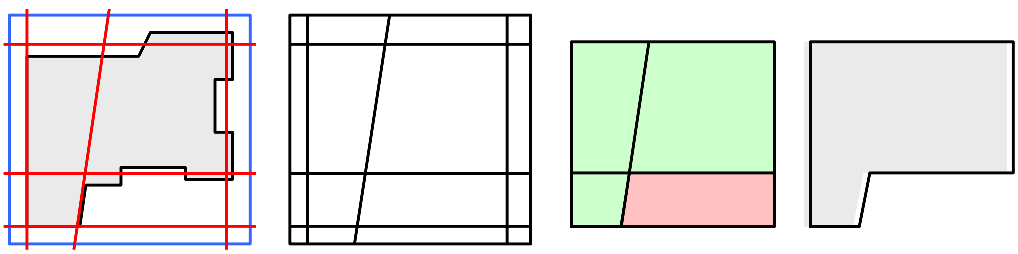


Figure 2.10: From left to right: Building ground plan with red half-space primitives, the resulting cell decomposition, discarded non-building cells (red) and the generalized building footprint. Figures from Kada & Luo [2006].

2.6.2 Optimal and topologically safe simplification of building footprints

For the generalization of building footprints, as needed for decomposition, an approach is described by Haunert & Wolff [2010]. The approach is using an integer program with constraints to generalize footprints. Their generalization is based on line simplification since, as Haunert & Wolff [2010] note, line simplification has been approached by optimization in contrast to building generalization.

For the input building footprint some constraints are set together with cost measures. The cost measures are area, regularity and similarity. The area cost means that a local simplification should have as little impact as possible on the area described by a footprint. The regularity describes that regular shapes meaning angles of 90 degrees are preferred. Opposite to the regularity there is similarity meaning the preservation of the original shape and line directions. These three cost measures are combined using a weighted sum to find the most optimal solution for the generalization.

For the constraints four requirements are given:

1. Each line in the generalized footprint should correspond to an original line in the input, which means the generalized line should intersect with the original line.
2. The sequence of corresponding generalized lines is a subsequence of the original input.
3. The Hausdorff distance between the generalized line and the original line should not exceed ϵ.
4. Each pair of different and non-consecutive generalized lines do not intersect.

When a generalized footprint satisfies the four requirements the result is a *feasible solution*. The next problem is to find the feasible solution with the minimum number of lines using the cost measures. The created generalized lines should be feasible but besides that also combinations of generalized lines have to be feasible. For testing the feasibility of multiple generalized lines a directed graph is used describing all possible generalizations. The selection of a generalized line using this approach is shown in Figure 2.11.

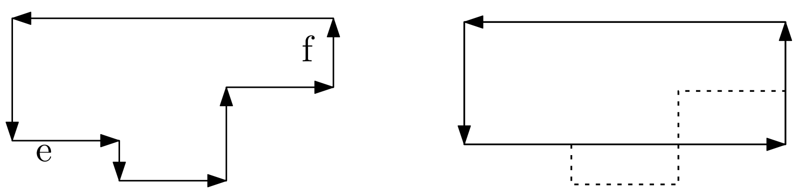


Figure 2.11: Original building (left) and result of selecting generalized lines (right). Figures from Haunert & Wolff [2010].

2.6.3 Simplification of building footprints based on edge-moves

A new method on simplification of building footprints is given by Buchin et al. [2011]. This method describes the use of edge-moves for the simplification. An edge-move is a local operator preserving the orientation of the edge, see Figure 2.12. The authors claim that edge-moves can always be performed on a non-convex polygon while preserving area, orientation and topology. The edge-move operates on three consecutive edges of which the vertices of the middle are moved along the consecutive lines of the other two. While moving the vertices, the original orientation of the middle edge is preserved.

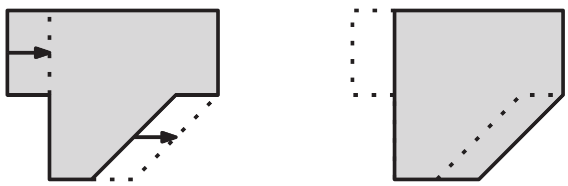


Figure 2.12: Two edge-moves (left) and the result (right). Figures from Buchin et al. [2011].

Edge-moves can either remove or add area to the polygon, hence two edge-moves, one adding and one removing area, are combined so the area and orientation are preserved. Angular restrictions are included to reduce the number of directions in the simplified polygons, while preserving the area, based on the creation of so called *staircases*. Also wall squaring for buildings is included which means restoring the right angle corners in man made structures. The wall squaring is obtained by imposing angular restrictions based on the buildings main orientation.

The preservation of topology is obtained by performing edge-moves on what is stated as a *subdivision*. What this means is the combination of multiple adjacent building footprints into a single component where shared walls are described by inner edges. The edge-moves are evaluated as being valid for both parts of the *subdivision*. The increase in area is always compensated by decreasing with another edge-move.

Besides their polygon simplification an extension of this method is given for urban-area generalization. An interesting notion from Buchin et al. [2011] is that wall squaring, which they explain for simplification purposes, is not used in their extension for generalization. Figure 2.13 shows that when using wall squaring it could cause misalignment which reduces the impression of a street running in between the buildings.

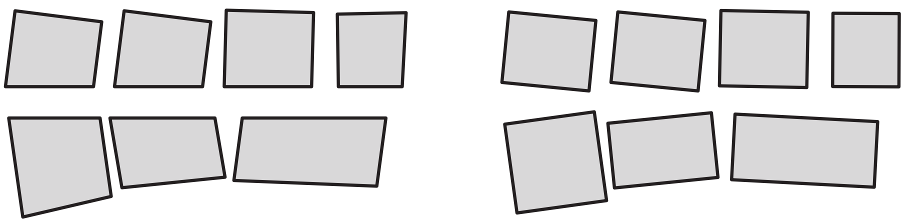


Figure 2.13: Original building outlines giving the impression of a street (left) and misaligned walls reducing this impression (right). Figures from Buchin et al. [2011].

2.7 Observations from existing methods

Some conclusions can be given from the related work:

- Complex building shapes can be reconstructed from a combination of simple shapes and connecting shapes.
- Creating a reconstruction only based on information gathered from data gives incorrect results due to missing or misinterpreted data.
- Creating a reconstruction by fitting pre-described shapes onto data, the resulting reconstruction is limited by the described shapes or a combination of them.
- Rectangular building footprints without small details are needed for split/decomposition approaches, if the footprints have small details then generalization is needed.
- Building footprint generalization has not been optimized yet, nonetheless a lot of research is done.
- Building generalization needs to be topology preserving for the creation of 3D city models.
- Splitting of building footprints sometimes creates subdivisions which are not needed, merging of the subdivision is needed to improve the results.
- Segmentation of a point cloud can be used for the detection of height jumps and ridge lines.

Noticeable from the methods is that the subdivision -or decomposition- of building footprints is used in multiple approaches. From this one can assume that it is an important aspect of generation of 3D city models. From the results of the splitting methods it it clear that one of the problems is that it generates too many subdivisions. To solve for problems originating from this, merging of the cells is proposed by Vosselman & Suveg [2001] and Vallet et al. [2011]. The aim of decomposition is to create a subdivision of the building footprint for which every part describes a single roof shape.

A request of iDelft is the generalization of building footprints for the creation of 3D city models with multiple details. For some applications the building footprint data sets they have contain to many details which they want removed. Besides building footprints can contain -almost- straight line segments described by multiple vertices. These unwanted features are removed by generalization resulting in a cleaner building footprint data set as wanted by iDelft.

Since building footprints can consist of small details which influence the decomposition in a negative matter, generalization is needed. However generalization can introduce gaps or overlap between neighbor buildings that where originally touching. When producing a 3D city model, results with gaps or overlap between buildings are incorrect. To avoid such problems topology preserving generalization is needed as proposed by Haunert & Wolff [2010] and Buchin et al. [2011].

The research of Kada & Luo [2006] shows that generalization and decomposition of building footprints can be combined into a single step using line equations. That research is extended by Kada & McKinley [2009] for the use of parametric shape fitting on the decomposition cells. The generalization approach is proved usable for the creation of 3D city models. The side effect of this method is the lack of preserving topology. Preserving of topology can be solved by extending the generalization process with the inclusion of adjacent buildings.

Based on the findings of Oude Elberink & Vosselman [2009] one can conclude that creating a building reconstruction based on models, like parametric shape fitting, lacks flexibility compared to reconstruction based on data. A combination of model based and data driven approaches is best for a reconstruction method. Given this and the need for merging unneeded subdivisions created by decomposition, the segmentation of point clouds as described by Oude Elberink & Vosselman [2009] can be used. The segmentation can be used to improve the decomposition by merging subdivisions based on knowledge obtained from the segments.

Based on these conclusions this thesis investigates the combination of two existing methods, generalization and decomposition as described by Kada & McKinley [2009] together with the segmentation of point clouds as described by Oude Elberink & Vosselman [2009]. The future work of both point out that their method can be improved by using a method similar to that described by the other.

From the perspective of the roof shape detection of Oude Elberink & Vosselman [2009], height jumps and multiple roof shapes are detected from the adjacency knowledge of the segments. Their results show that especially finding height jumps is problematic. Within this thesis research the segments are not used for detecting height jumps and multiple roof shapes but are to be detected by the decomposition. The shape and symmetry of a building footprint contains many clues on the location of these cases.

An in deep description of the generalization and decomposition based on the method of Kada & McKinley [2009] is given in Chapter 3, while the point cloud segmentation method of Oude Elberink & Vosselman [2009] is thoroughly described in Chapter 4.

Chapter 3

Generalization and decomposition using line equations

This chapter concentrates on the generalization of building footprints and their decomposition using line equations. As described in Chapter 1, generalization and decomposition of building footprints is the start for the generation of 3D building models.

City models do not always need to be of high detail, sometimes it is even useful to use less detailed building models as for example noise pollution modeling [Billen & Zlatanova, 2003]. In this case generalization of building footprints is wanted to make the noise calculations less complex. Generalization is also needed when using decomposition of building footprints since it removes small features which can introduce artifacts in the decomposition. Besides introducing artifacts using decomposition without generalization will result in over decomposition. An example for the need of generalization for decomposition is shown in Figure 3.1.

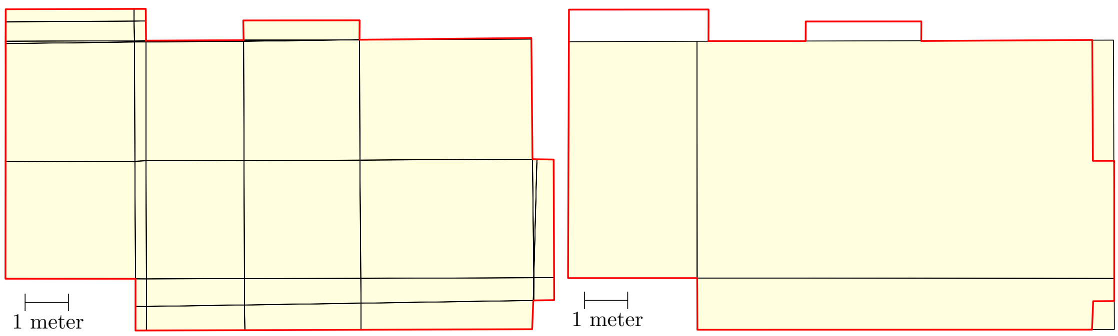


Figure 3.1: Decomposition of a building footprint without (left) and with (right) generalization of 1m where the red outline is the original.

When using decomposition for roof shape fitting as described by Kada & McKinley [2009], the generalization is less restrictive since the detected roof shapes are corrected according to the extruded original footprint. This means that the building outline of the created building is identical to the original building footprint. However when creating generalized buildings as this thesis investigates, the generalization needs to be topologically correct, which is much more strict then the method described by Kada & McKinley [2009]. When buildings touch in the original footprints they also need to touch in the generalized footprint, as also buildings cannot overlap after generalization. The restrictions on this kind of generalization are more severe [Kada & Luo, 2006; Haunert & Wolff, 2010]. Improvements on the building footprint generalization approach of Kada & Luo [2006] are described by Peter et al. [2008] for preserving main building shape and facade lines. Using parts of this method, some topological errors can be prevented.

The method of Kada & McKinley [2009] is described in several publications [Kada & Luo, 2006; Kada & McKinley, 2009; Kada, 2010]. Using this information, the generalization and decomposition methods are reconstructed. Some specifics about this method could not be found like the exact steps taken for the generalization and have been recreated while adding improvements. Within this research many different tries to reconstruct the generalization steps and its order have been taken. In this chapter the generalization and decomposition are explained as found best suiting for the 3D building reconstruction. Other possibilities that have been experimented with are described in Chapter 6.

The steps needed for the generalization and decomposition process are shown in Figure 3.2. This flowchart describes the order of the steps within the generalization and the decomposition process. Each block in the flowchart is covered in this chapter. The grey blocks with the red outline are the input and output data, where the other grey blocks are the generalization as covered in Section 3.1 and the decomposition which is covered in Section 3.2. The cyan and yellow blocks are the sub-steps of respectively the generalization and decomposition. The colors of these blocks are corresponding to the stage of the real data examples shown throughout this chapter. The blue block of the Adjacent buildings shows the location of the insertion of knowledge from adjacent buildings as explained in Section 3.1.4.

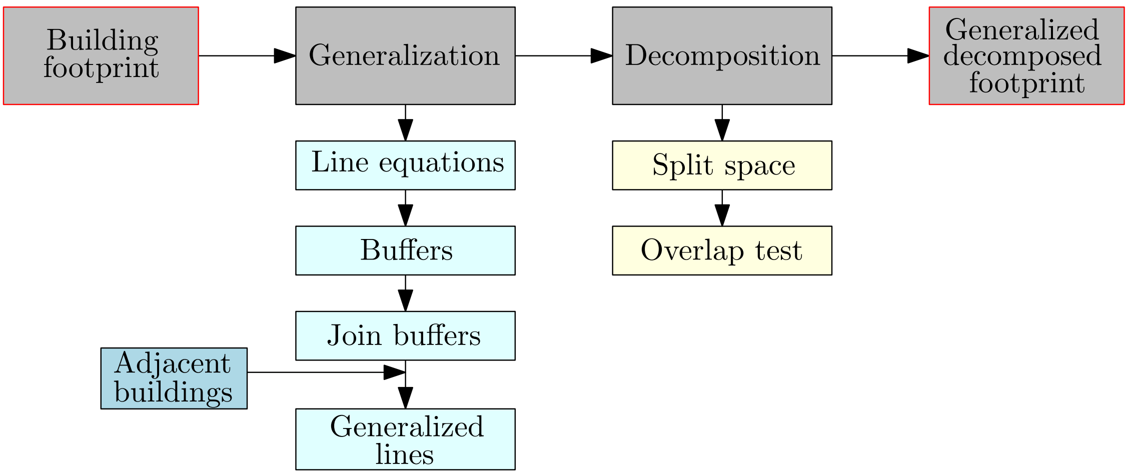


Figure 3.2: Generalization and decomposition process.

3.1 Generalization using line equations

The generalization of a building footprint is performed by combining lines which are near parallel and within a given distance. In order to calculate these properties the polygon lines are translated into equations. The input of the generalization is a set of polygons which are split into lines and stored as equations. The lines are grouped so that the resulting number of generalized lines is smaller then the original.

Lines are grouped into partitions containing a buffer B and two sets of lines L_P and L_{NP}, $P = (B, L_P, L_{NP})$. The buffer B is describing the area between two lines $Ax+By+C_{min} = 0$ and $Ax + By + C_{max} = 0$. The set of lines L_P contains the parallel lines within the buffer B, where L_{NP} contains all other lines within the buffer. The distance between C_{min} and C_{max} needs to be below the generalization threshold. Besides the distance, the angle between lines is also of high importance. In order not to create too many lines with small direction differences, lines are said to be parallel when they have a small angle difference. An example of a partition is given in Figure 3.3 where the dashed lines describe the buffer, the green lines the content of L_P and orange line the content of L_{NP}.

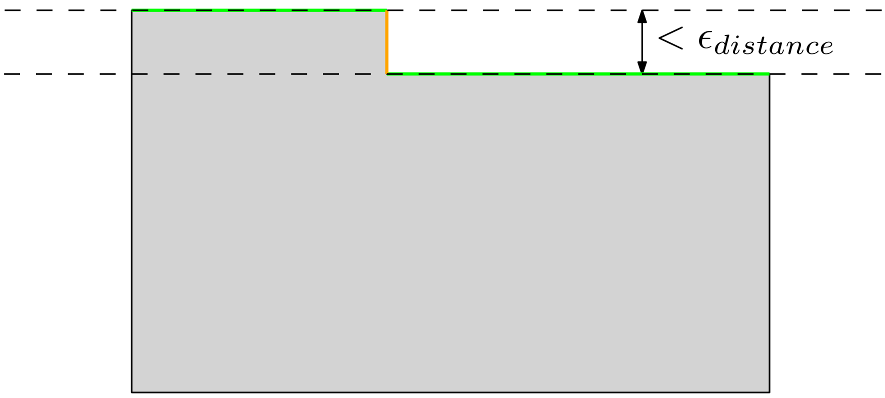


Figure 3.3: Building footprint with buffer lines (dashed) describing a generalization partition. Green lines are parallel marked lines and in orange an included line with arbitrary direction.

When no more partitions can be formed, the generalized lines are created from weighted average of its parallel lines, using their length as weight. The lines within L_{NP} are not needed here as they do not contain additional information. The calculation of the generalized line can be performed using different approaches as described in Section 3.1.3.

The research described by Kada [2010] shows the generalization of 3D buildings using the generalization of planes. This thesis is focussed on the generalization of 2D building footprints using the generalization of lines. The equations given in Kada [2010] use the normal vector of planes to be compared with the direction vector of the buffer. However for the line equations, the direction vector given by the line equations are used. Due to this the equations in the following sections are transformed from the 3D plane calculations using normal vectors into 2D line calculations using direction vectors.

3.1.1 Transformation from polygons to line equations

Polygons are described by rings of points where the last point is the same as the first. Polygons consists out of one outer ring and zero or more inner rings. The outer ring describes the outer extends of the polygon, where inner rings describe holes. In deep explanation of handling holes is described in Section 3.4.

Lines can be constructed by taking pairs of consecutive points like the first and second, second and third points and so on. The lines are thus described by two points which can be converted into a line equation. The conversion from the coordinates to an equation depends on the used projection of the input polygons. An example using a Cartesian coordinate system is shown in Equation (3.1).

$$
\begin{aligned}
&Ax + By = C \qquad (3.1)\\
&\text{where,}\\
&A = y_1 - y_2,\\
&B = x_2 - x_1,\\
&C = x_2 * y_1 - x_1 * y_2
\end{aligned}
$$

Besides the line equation, the length of the line is stored also. The length is simply calculated as shown in Equation (3.2).

$$length = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \qquad (3.2)$$

3.1.2 Joining lines using buffers

At the start of the algorithm for every line a buffer is created from the calculated line equation. The buffers are sorted starting with the one of highest importance. The buffers are merged if possible starting from highest importance to lower importance, based on two properties: angles and distance. Both properties are explained below. Two distinct terms are used for the combination of lines. When lines are found to be parallel they can be 'joined', as explained below in Joining buffers based on angles. Lines that have arbitrary directions but completely fall within a buffer of higher importance can be 'included', as explained in Including buffers based on distance.

Joining buffers based on angles

Joining buffers based on angles is important to overcome the possibility of over fragmentation since polygon lines are not always perfectly parallel. Two buffers are tested if they can be joined based on parallel angles. All points p in both buffers should be within the distance threshold $\epsilon_{distance}$ and the buffers within the angle threshold ϵ_{angle}. Two buffers can be joined when Equations (3.3) and (3.4) are satisfied, where C_{min} and C_{max} are calculated as shown in Equation (3.5).

$$|C_{max} - C_{min}| < \epsilon_{distance} \qquad (3.3)$$

$$\left| \arctan(\frac{-A_1}{B_1}) - \arctan(\frac{-A_2}{B_2}) \right| < \epsilon_{angle} \tag{3.4}$$

$$\begin{aligned} \forall p \in L_P \cup L_{NP} : C_{min} &= \min(Ap_x + Bp_y) \\ \forall p \in L_P \cup L_{NP} : C_{max} &= \max(Ap_x + Bp_y) \end{aligned} \tag{3.5}$$

When both these equations are satisfied the buffers can be joined. The two sets of lines L_P and L_{NP} from buffer B_2 are merged with the ones from buffer B_1 as shown in Equation (3.6). New A_{new} and B_{new} values are calculated using Equation (3.7) and the new C_{min} and C_{max} using Equation (3.5) by inserting the A_{new} and B_{new} values.

$$\begin{aligned} L_{P_{new}} &= L_{P_1} \cup L_{P_2} \\ L_{NP_{new}} &= L_{NP_1} \cup L_{NP_2} \end{aligned} \tag{3.6}$$

$$\begin{aligned} A_{new} &= \frac{\sum_{i=0}^{n} length(L_{P_i})A}{\sum_{i=0}^{n} length(L_{P_i})} \\ B_{new} &= \frac{\sum_{i=0}^{n} length(L_{P_i})B}{\sum_{i=0}^{n} length(L_{P_i})} \end{aligned} \tag{3.7}$$

To explain the requirements that have to be met before lines can be joined, Figure 3.4 visualizes Equations (3.3) and (3.4). Here the green lines are found to be parallel if both these equations are satisfied.

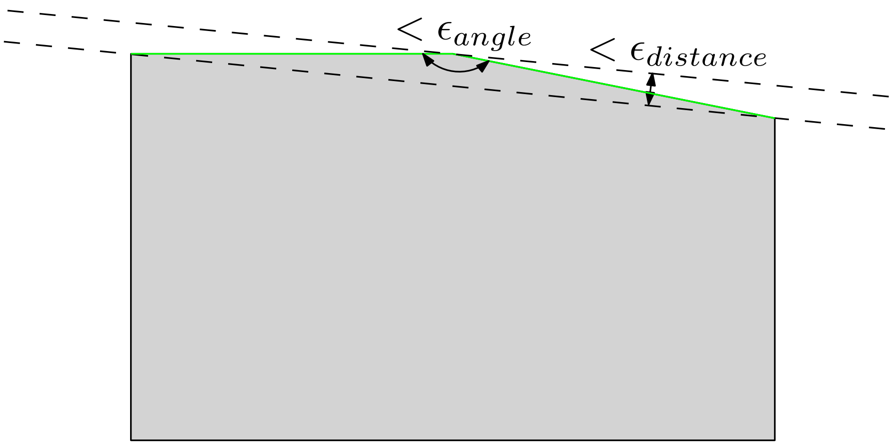


Figure 3.4: Building footprint with buffer lines (dashed) describing a generalization partition. The direction of the partition is based on the weighted average of the contained lines. Green lines are found as parallel lines.

Including buffers based on distance

A buffer B_2 with arbitrary direction could be included in buffer B_1 of higher importance if the points it contains fall completely within buffer B_1. This means that Equation (3.8) must be satisfied in order to include buffer B_2 with buffer B_1.

$$\forall p \in L_P \cup L_{NP} : \min(Ap_x + Bp_y) \geq C_{min} \wedge \max(Ap_x + Bp_y) \leq C_{max} \tag{3.8}$$

If buffer B_2 can be included with buffer B_1 its lines will be merged into the set of lines L_{NP} because these may not change the orientation of the buffer B_1. The buffers are merged as shown in Equation (3.9). Since L_P does not change the values for A and B stay unchanged as well. Only the values for C_{min} and C_{max} are recalculated using Equation (3.5).

$$L_{NP_{new}} = L_{NP_1} \cup L_{N_2} \cup L_{NP_2} \tag{3.9}$$

In Figure 3.3 an example is shown of a line includable within a buffer of higher importance. The line shown in orange can be included within the buffer that is created by joining the green lines in an earlier step.

Importance of line direction

The direction of lines is important when deciding which to join based on parallelism. The generalization method of Kada & Luo [2006] does not mention line direction at all. Sometimes lines can be joined giving multiple solutions as shown in Figure 3.5. Two possible solutions are the dashed and the dotted buffer. The dashed buffer is the result of joining the red and the orange line, which would also include the small green line. The lines are joined due to the red line being the longest, however the orange line has its orientation in the opposite direction compared to the red line. The dotted buffer is the result when joining the green lines and including the orange line, this would be the case if the direction of the lines is taken into account and given a higher priority.

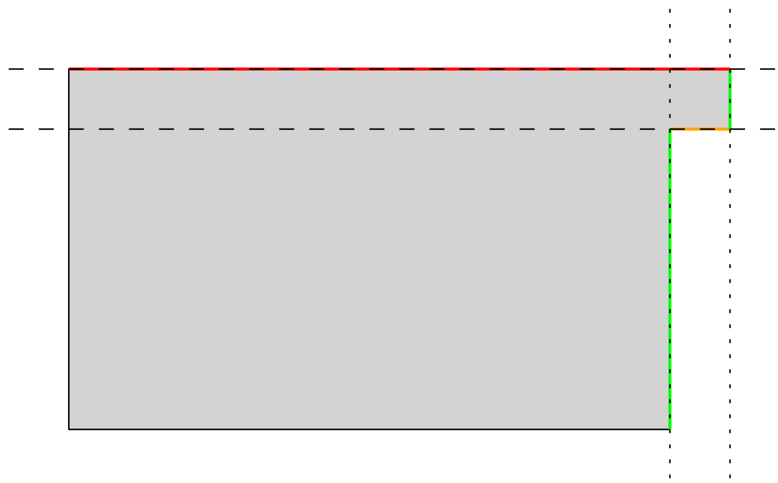


Figure 3.5: Building footprint of two possible buffers with the same line (orange). Dashed buffer is the result when red and orange lines are joined, and the dotted when the orange line is included with the green lines.

3.1.3 Generalized line calculation

When there are no more buffers left to be created, the final equations for the lines are calculated in order to create the new generalized lines. The equation for the generalized line is only depending on the parallel lines described by the buffer, which are stored within L_P. During the earlier process of joining the parallel lines the values for A and B are already updated, which leaves only the determination of the C value. The determination can be performed by averaging given all parallel lines weighted by their length, but it can also be set equal to the line with the highest priority.

Approach 1: Weighted average lines

The approach as described by Kada & Luo [2006] is based on averaging the buffered lines both their orientation and location. A similar approach is described in Kada [2010]. However this is, as described before, for the generalization of 3D buildings using 3D planes. This approach uses the area of the planes as weight for the orientation and location of the generalized plane. For this research this can be translated into the length of the lines being used as weight. Since the orientation of the buffer is already calculated and explained in the previous, only the location is left to solve. In Equation (3.10) the adjusted version from the original weighted average equation by Kada [2010] can be found.

$$\forall p \in L_P : C_{new} = \frac{\sum_{i=0}^{n} length(L_{P_i})(Ap_x + Bp_y)}{\sum_{i=0}^{n} length(L_{P_i})} \tag{3.10}$$

An example using this weighted average approach is shown in Figure 3.6. The green lines are joined since they are parallel while the orange line completely falls within the buffer created by joining the green lines and is thus included. The generalized line resulting from the recalculation of the orientation and location is shown in red.

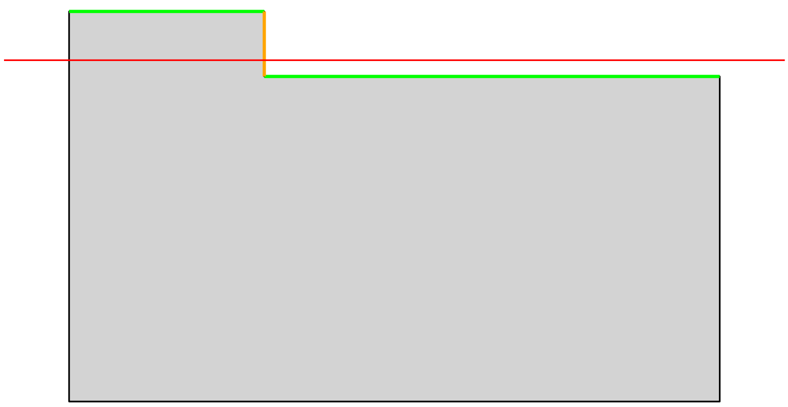


Figure 3.6: Building footprint with generalized line (red). The green lines are parallel thus joined where the orange line is included with these, resulting in the red line which location is the weighted average of the green lines.

Using the previously described generalization combined with the average line calculation a generalization is performed on an input data set. More information on the data sets used can be found in Chapter 6. Some examples of the resulting generalization are shown in Figure 3.7. Within these figures the original polygons are described by the red line with the points in black, the resulting generalization is shown with the black outline and filled cyan.

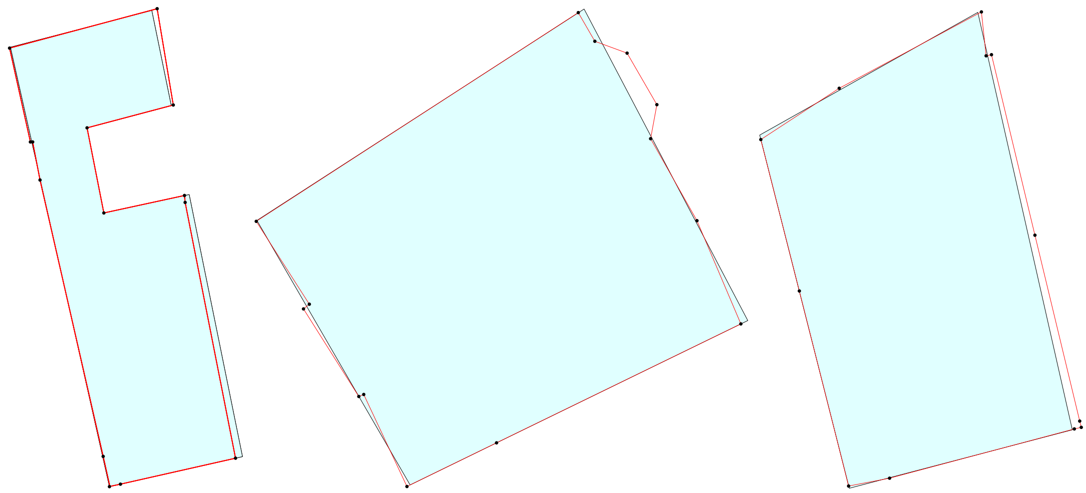


Figure 3.7: Original building outlines (red line with black points) overlaid on generalized footprints (1m) using weighted average lines.

Approach 2: Best fitting lines

Based on quality evaluations a new approach improving that of Kada & Luo [2006] is developed by Peter et al. [2008] for preserving facade and original footprint lines. Especially when a block of connecting buildings is processed they should not overlap or introduce gaps after generalization, further reading on this in Section 3.1.4. This approach is implemented in such a way it can be applied after the generalization as an subsequent step. The lines of the generalized footprint are adjusted to coincide with the main building lines of the original footprint. Selection of the main building lines is similar to the merging using the line buffers as described before. Selection of the best fitting main line is performed using the weight of the original line and the distance between the generalized and original one.

Given these observations the adjustments of the lines can be performed directly when calculating the new location of the generalized line, no subsequent step is needed. The best fitting location of the line is that of the line with the highest weight in the buffer. Since the calculation as described by Peter et al. [2008] is not useful in this case a new one has been formulated. The calculation of the generalized line location is shown in Equation (3.11).

$$\forall p \in \max(length(L_P)) : C_{new} = Ap_x + Bp_y \tag{3.11}$$

The result of the best fitting approach is that generalized lines are at the same location of one of the original lines. In Figure 3.8 the location of the generalized line, the red line, is based on the best fitting location thus the line with the highest weight. The selection of the best fitting line for the generalization retains most of the original structure and facade lines of a building, compared to using the weighted average.

Figure 3.8: Building footprint with generalized line (red). The green lines are parallel thus joined where the orange line is included with these, resulting in the red line which location is equal to the green line with the highest priority.

Using the best fitting line method, the results resemble more of the original shape of buildings. In Figure 3.9 one can see the results of the generalization using this method. In this figure the red line with the black points resemble the original footprint where the black outline filled with cyan resembles the generalized shape. These examples show that small features are removed correctly and also the extra nodes are gone.

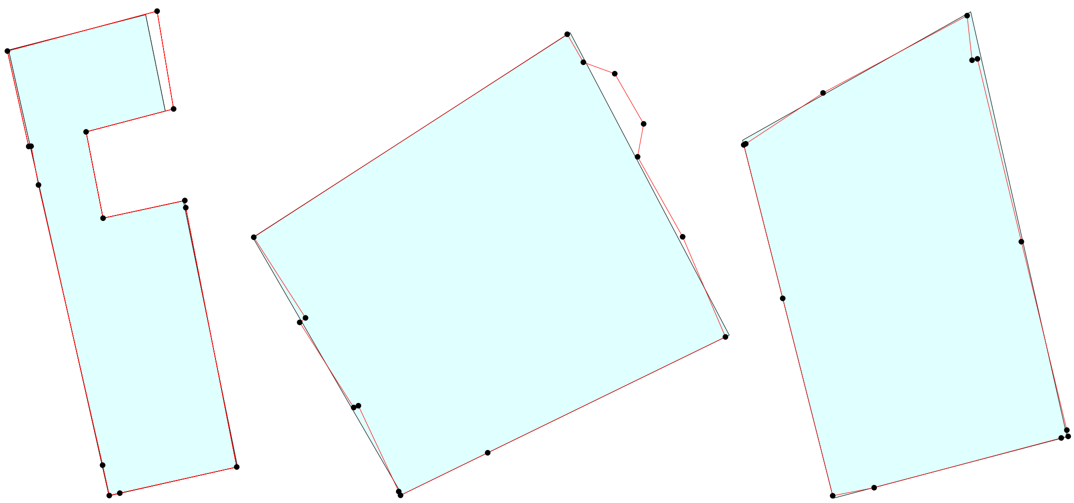


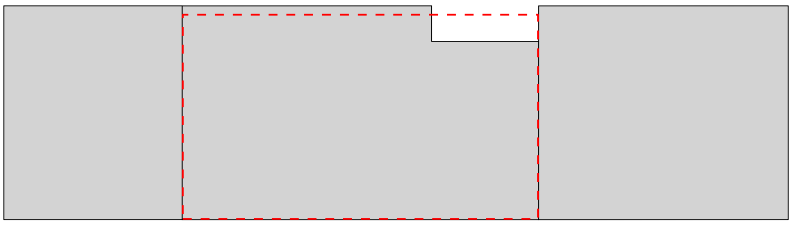
Figure 3.9: Original building outlines (red line with black points) overlaid on generalized footprints (1m) using best fitting lines.

3.1.4 Including adjacent buildings

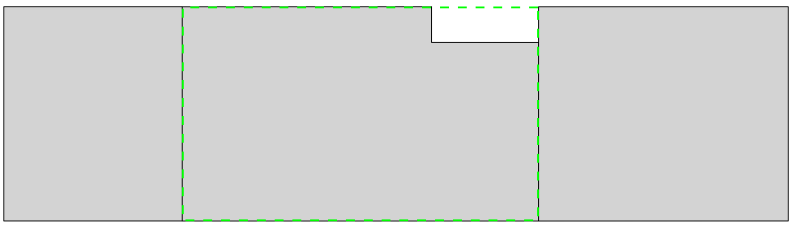
The previous sections describe the generalization of buildings as single structures. When looking at the surroundings many buildings are build with one or more outside walls connected to other buildings. Therefore there is a need of introducing the adjacent buildings into the generalization. Using adjacent buildings is also noticed by Peter et al. [2008]. In the following some problems are stated which can be solved by taking into account adjacent buildings whereafter the usage of adjacent buildings is explained.

Lost common facade lines

Buildings which are build in rows or blocks mostly have common characteristics like facades and roof shapes. When generalizing a building without using knowledge from the surroundings, these common characteristics could be lost. For example facade lines that are common for buildings should be kept for the recognition of buildings. Figure 3.10a shows what happens if the adjacent buildings are not used in the analysis, the common facade is lost by the generalization. The lost common facade line can be solved when using the best fitting line calculation which is shown in Figure 3.10b.



(a) Lost common facade line

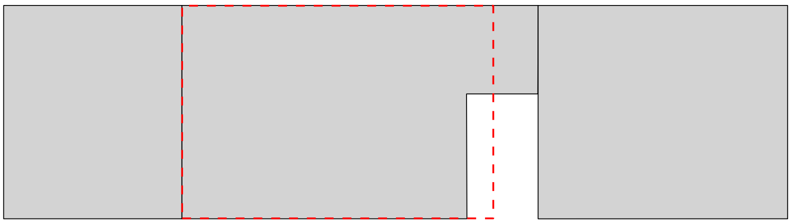


(b) Corrected lost common facade line

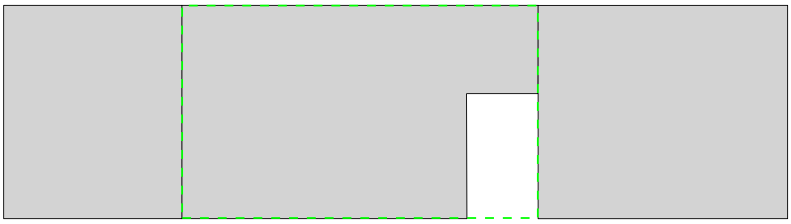
Figure 3.10: Building with lost common facade line due to generalization and corrected by best fit.

Gaps

Buildings that are adjacent in the original footprint can be separated during generalization since footprints are considered independently. Gaps that emerge from this separation are influencing the usability of the model. When the model is for example used for a noise study, the gaps will introduce incorrect results. When taking into account wall lines from adjacent buildings these can contribute to the weight of the lines. Thus all collinear edges contribute to the weight of that main building line. The additional weight should be enough for the generalized wall line to shift to that of the adjacent building. Figure 3.11a shows gaps emerging from generalization whereas Figure 3.11b shows the gaps solved using the best fitting line calculation.



(a) Gap between buildings



(b) No gap between buildings

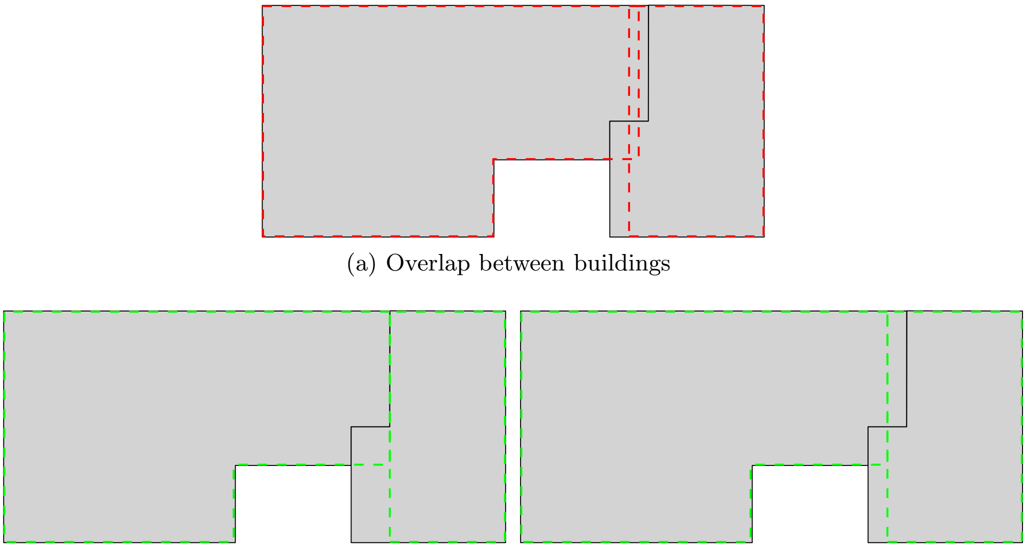
Figure 3.11: Building with a gap between adjacent building due to generalization and corrected by best fit.

Overlap

A problem more difficult to solve is that of overlap between buildings. When buildings are generalized as described in the previous overlap between them could occur, as shown in Figure 3.12a. Overlap could be detected and models altered accordingly, however this requires knowledge of previously generalized buildings or simultaneously generalization of buildings.

Overlap of buildings can be solve in numerous ways. The smallest building can be adjusted to the larger building by shifting the conflicting lines as shown in Figure 3.12b. However the larger building can also be adjusted to the smaller one. If the problem is a common line, this line can be adjusted so that both buildings are altered by taking the average of both lines, an example is shown in Figure 3.12c. A third solution could be restoring the original lines of both buildings. Attributes describing the importance of the buildings could also be used for deciding which building to adjust for the overlap.

All of the described problems are found in data sets. Some examples solving the lost facade lines, overlap and gaps are shown in Figure 3.13. The examples are taken from real data sets and are generalized using the created algorithm and the best fitting line method.

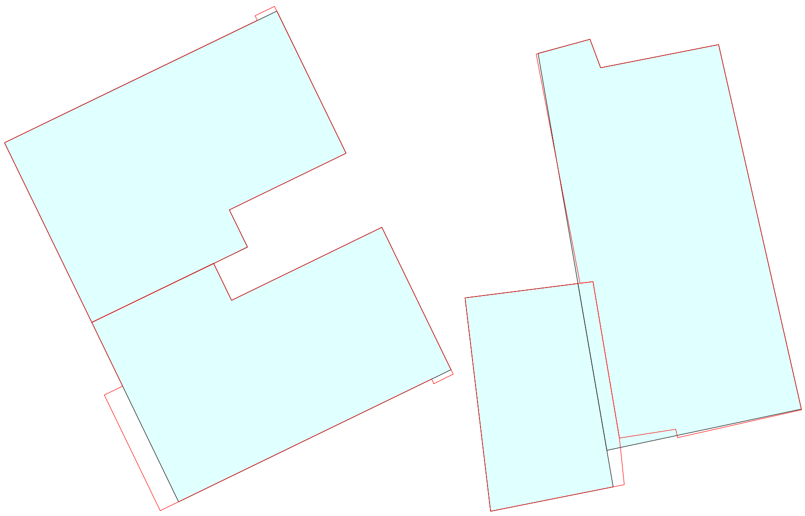


(a) Overlap between buildings

(b) Possible overlap solution

(c) Possible overlap solution

Figure 3.12: Buildings which overlap the adjacent building due to generalization (top) and multiple solutions (bottom).



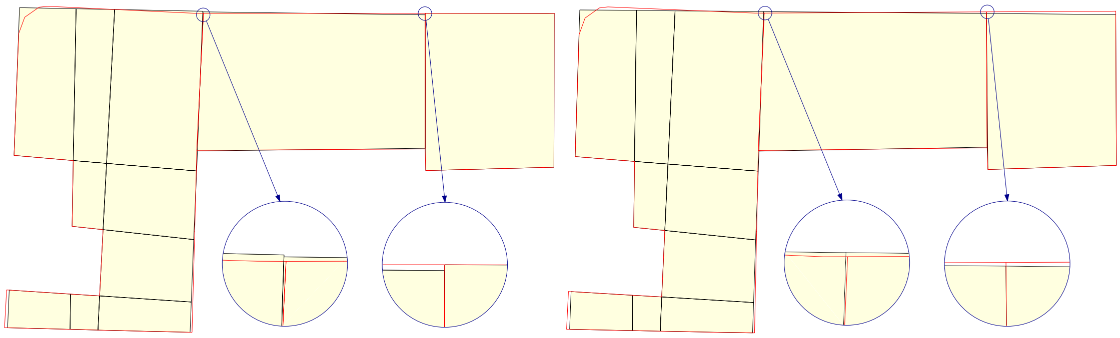
(a) Lost common facades solved

(b) Overlap solved

Figure 3.13: Generalization results using real data and problems solved using best fitting line. Original (red outline) overlaid on the generalized (cyan) footprint.

Using adjacent buildings

When taking into account the adjacent buildings the examples given earlier can be solved. However when only taking into account the directly adjacent buildings, facades can for example have slight direction changes. An example is shown in Figure 3.14a. The problems exist since more then just the adjacent buildings can contribute to a building, even the buildings adjacent to that can. The solution to this is using a complete block of buildings. A block of buildings is the group that is adjacent, but also adjacent to the adjacent buildings and so on. An example of what is considered adjacent buildings is shown in Figure 3.15 where the solid red outline is the selected building and the red dotted outlined are the adjacent buildings. The definition of a building block is also shown in Figure 3.15 where all buildings in the figure describe the block. The result acquired when using the complete building block can be seen in Figure 3.14b.



(a) Including adjacent buildings (b) Including complete block

Figure 3.14: Generalization results using adjacent buildings. Original (red outline) overlaid on the generalized (yellow) footprint.

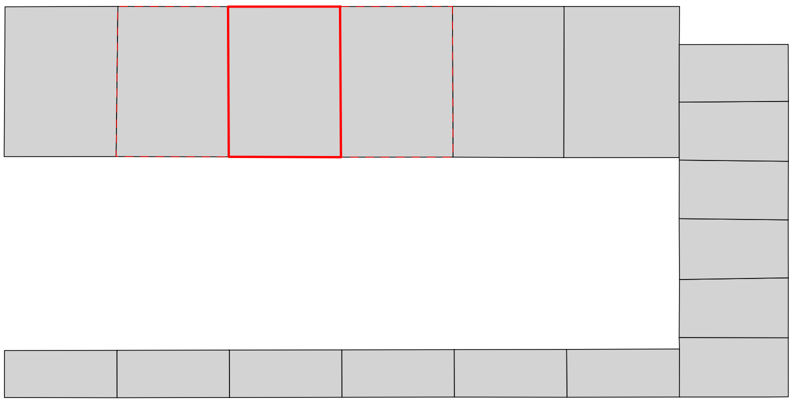


Figure 3.15: Definition of adjacent buildings and building blocks, solid red is the selected building, red dashed are adjacent and all describe the building block

In the generalization the adjacent buildings are included after creating the partitions of the building as shown in Figure 3.2. The adjacent buildings are transformed into line equations as described for the building itself. The adjacent building lines should apply to the same rules as the normal building lines in order to be inserted into a partition of the building. However there is another condition it should satisfy too, the adjacent building line should be within the generalization distance of a line within the partition. This is an important condition, when this is not used the building block will be generalized as a whole.

3.2 Decomposition using generalized line equations

Decomposition of building footprints means splitting it into multiple nonintersecting cells. This can be done in multiple ways like fitting rectangular shapes or extending existing lines. A decomposition method using line extension based on equations is described by Kada & Luo [2006]. Their research combines the previously described generalization with decomposition.

After the generalization is complete, a set of generalized line equations is left. Using these line equations, the decomposition can be created directly. The line equations are used to split 2D space followed by calculating the overlap of the created cells with the original footprint. Cells with a low overlap are removed which leaves only the cells belonging to the building.

3.2.1 Creating decomposition cells

The decomposition is thus performed by splitting space with the generalized line equations using brute force. Since splitting infinite 2D space is not practical, two times the bounding box of the original footprint is used. The creation of this two times bounding box is carried out by calculating the width and height of the normal bounding box and doubling its size in both dimensions.

The two times bounding box is intersected given the line equations and the intersection points are used to split. Line for line the space is divided into two until all lines are processed. The result is a large box containing the generalized, decomposed building footprint but also extra cells which still have to be removed as shown in Figure 3.16.

3.2.2 Overlap test

Since within the decomposition process the created cells cannot be identified as belonging to the building or not there is the need for such an identification step. The method of Kada & Luo [2006] suggests an overlap test. The cells are overlaid on top of the original footprint and its intersection is computed. The area of the decomposition cell D and the intersection with the original O is then calculated as shown in Equation (3.12). Since the generated cells outside of the building are fairly large, due to using the two times bounding box, the overlap with the original is low. A value of 50% overlap is used since the low overlap of the large cells outside the building. When the overlap is lower then 50% the cell is removed, leaving only building cells. The split double bounding box with the selected cells is shown in Figure 3.16.

$$\frac{Area(D \cap O)}{Area(D)} > 0.5 \tag{3.12}$$

3.2.3 Results of the decomposition

Using the double bounding box to split space using the generalized line equations and incorporating the overlap test for the determination of cells being part of the building,

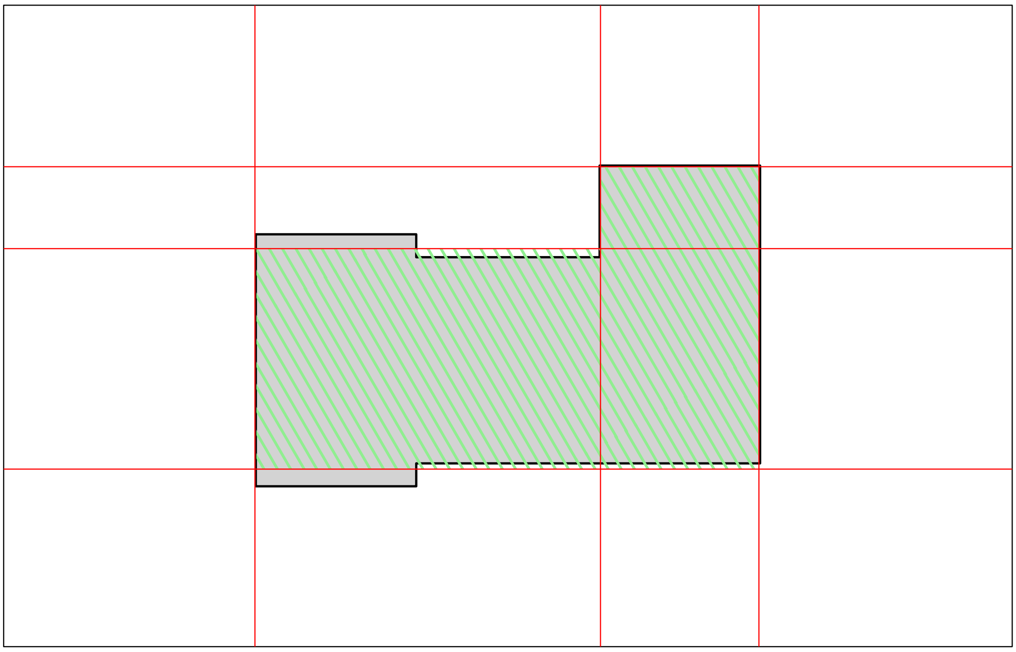


Figure 3.16: Two times bounding box of the original footprint split using generalized line equations, with the selected cells resulting from the overlap test.

a decomposition of the generalized footprint is acquired. The result of these steps is a generalized building footprint split into multiple decomposition cells. The end result of the decomposition is shown in Figure 3.17 where the dashed red line denotes the original contour. During the decomposition multiple quality statistics are computed which are explained in Section 3.6.

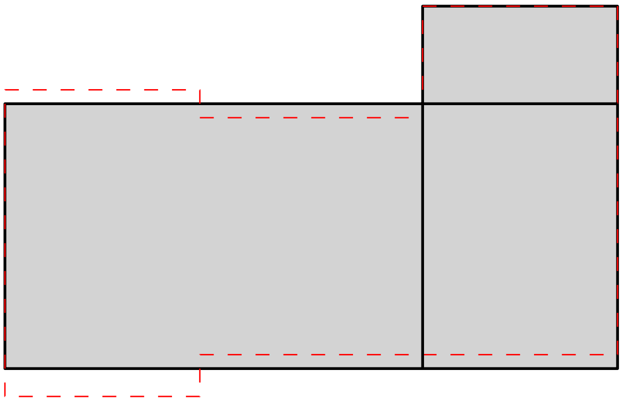


Figure 3.17: Decomposition result (black lines filled with grey) overlaid with the original outline (red dashed).

When using the created decomposition algorithm on a real-life data set, results are as expected. The decomposition divides the building footprints in multiple cells given the set of infinite generalized lines. In Figure 3.18 some examples are given, the blue lines are the resulting lines from the generalization where the black lines are the decomposition lines splitting the footprint into multiple cells.

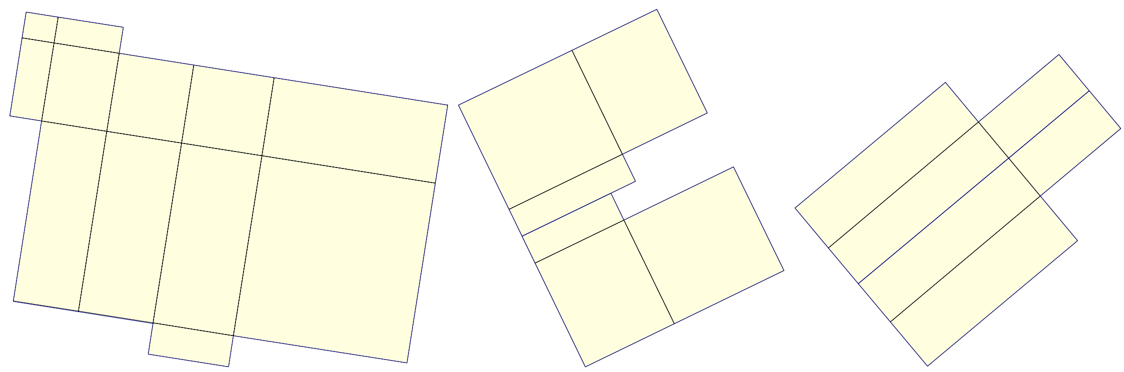
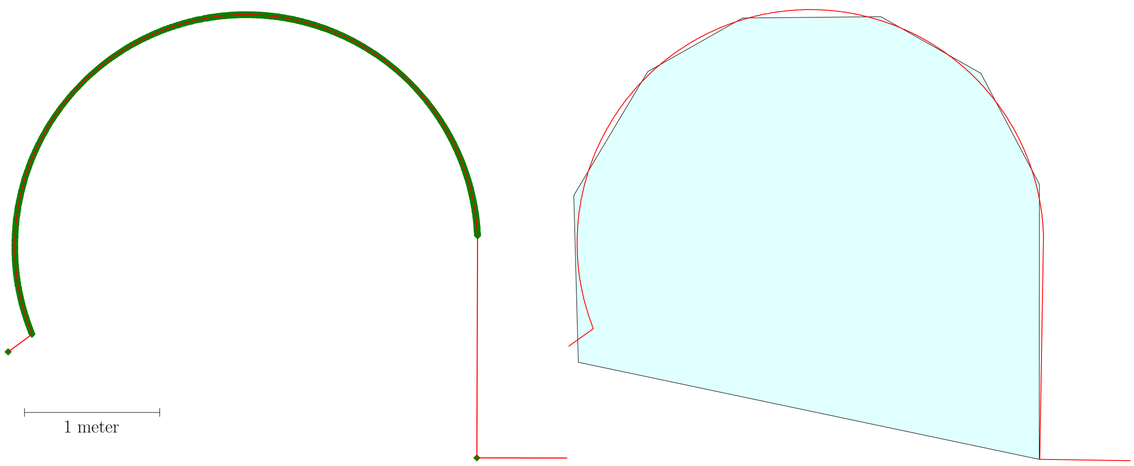


Figure 3.18: Decomposition results using real data (blue being the building outlines).

3.3 Circular Elements

Building footprints are sometimes describing elements that are of circular shape. Since the storage of curves is not always possible in Geographic Information System (GIS) data structures, circular elements are discretized into straight line segments. These circular elements are thus described by a large amount of nodes. The discretized circular element in Figure 3.19a is described by about 1000 nodes of about 1 cm distance.

When circular elements are generalized, in the same manner as described before, results are as shown in Figure 3.19b. The created shape, shown in black, shows that the generalization of the circular elements does not produce problems.



(a) Input nodes (green squares) with resulting line (red)

(b) Generalization result (black) compared to original line (red)

Figure 3.19: Circular elements in building footprints, discretization and generalization.

If the decomposition of circular elements is inspected it shows a lot of slant decomposition lines, see Figure 3.20. The line equations created by the circular elements are introducing undesirable decomposition lines. When using such a decomposition the extrusion will be disturbed by the many small cells.

Since the decomposition results are of high importance for the subsequent steps, the circular elements in building footprints render the results of these decompositions unusable for further processing. A solution could be an initial check of the building footprint for the presence of circular elements and processing these separate from the rest of the footprint.

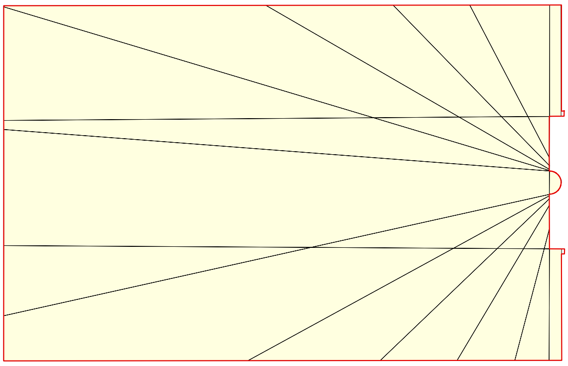
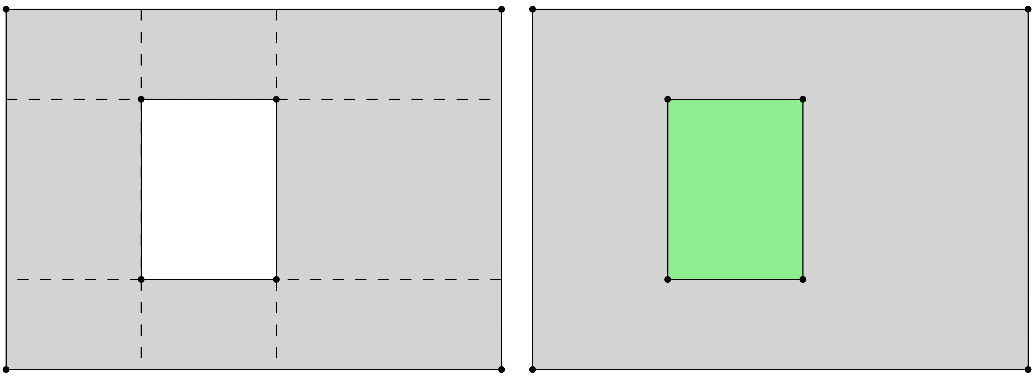


Figure 3.20: Decomposition result with circular elements (black) overlaid with original contour (red).

3.4 Holes and Islands

Building footprints are able to contain holes like inner yards. These holes are inner rings of the polygon describing the building footprint. By inserting these inner rings into the generalization the algorithm is able to create generalization results including holes. For the decomposition and the overlap test these holes are includable into the normal generalization without introducing problems for the end result. Figure 3.21a shows an example of a building with a hole where the dashed lines are the results of the decomposition.



(a) Polygon with an inner ring creating a hole and decomposition lines (dashed)

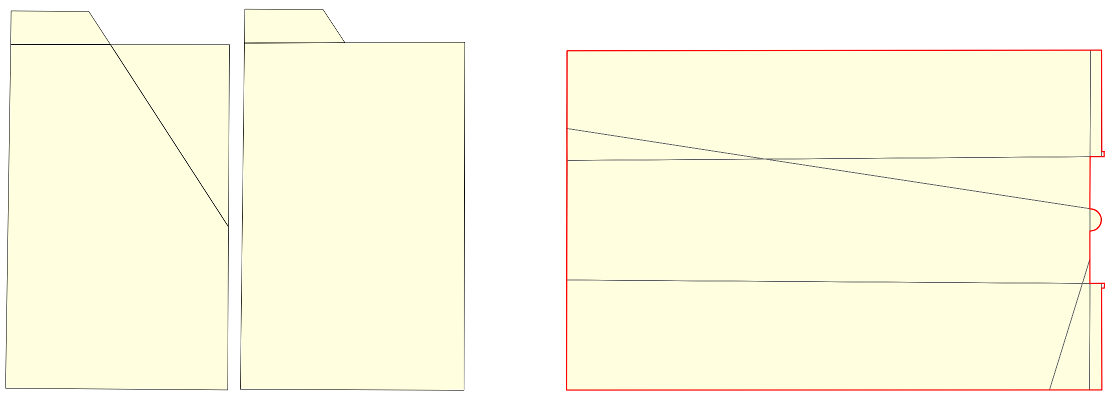
(b) Polygon with an inner ring which is filled with another polygon creating an island

Figure 3.21: Building footprint with inner rings creating a hole or an island.

Besides holes there can also be islands covering the holes of other polygons. The presence of islands in building footprints is less common then holes, however they do exist. These island do not change the functioning of the generalization, they are generalized identical to the holes they fill, due to the way they are generalized. An example of an island is shown in Figure 3.21b.

3.5 Slant lines

The handling of slant lines is not covered in the publications about cell decomposition [Kada & Luo, 2006; Kada & McKinley, 2009; Kada, 2010; Peter et al., 2008]. Slant lines are lines that do not lay in the main direction of a building, but for example have an angle between 30 and 60 degrees. When a building shape is convex, all angles are smaller then 180 degrees, these lines do not introduce artifacts but they do when a building shape is concave, thus has lines with an angle larger then 180 degrees. The slant lines introduce unwanted decomposition lines within the building. An example of artifacts due to slant lines can be seen in Figure 3.22a.



(a) Decomposition artifacts due to slant lines

(b) Decomposition with slant lines splitting only covered cells

(c) Decomposition improvement on circular elements compared to Figure 3.20

Figure 3.22: Decomposition improvements of building footprints with slant lines.

The best solution to solve the decomposition artifacts due to slant lines is to split only the decomposition cells covered by a part of the line. First all the lines have to be marked if they are in the main direction of the building. For this the main direction of the building has to be calculated. This is done by choosing the line with the highest count of right angles with other lines. Angles are said to be right angles if they are 90 degrees with a variance of the angle threshold set for the generalization.

Now the main direction is known, the lines that are in this direction or perpendicular to it are used to split the complete two times bounding box. After these are processed the slant lines are. The cells that are overlapped by the slant line are identified and these are split, leaving the cells that are not overlapped untouched. The improvements generated by cutting the slant lines last is shown in Figure 3.22b.

Most of the artifacts created by the circular elements are solve by using this as well. The improvement on circular element in Figure 3.20 is shown in Figure 3.22c. The two slant lines that are left are within the angle threshold thus are considered parallel to the main direction.

3.6 Quality Assessment

The automatic generalization and decomposition of building footprints is explained but since the generalization is intended to change the footprint, there is the need for quality assessment to identify the impact. Research on the assessment of generalization of building footprints is described in numerous publications [Podolskaya et al., 2008; Filippovska et al., 2008, 2009; Filippovska & Kada, 2010; Kada et al., 2009]. The quality metrics used within this thesis research are described below.

After calculating these values they can be combined using Euclidean distance between quality parameters of the original and generalized footprints. This is applicable since the quality parameters are all in the range from 0 to 1. The Euclidian distance give a better differentiation between these quality parameters (QP) then just a sum, see Equation (3.13). Using this total quality (TQ) value the building footprints can be flagged to identify potential unusable footprints. These flags can then be used to exclude the buildings from further processing if necessary or marking for visual inspection. Since the decomposition introduces additional points and multiple cells, a union needs to be performed before being able to calculate the generalization statistics.

$$TQ = \sqrt{\sum_{i=1}^{n} QP_i^2} \tag{3.13}$$

3.6.1 Point count

Counting the points that are part of the input polygon and output polygon gives an indication of the impact the generalization has on the shape of the footprint. Since the input is a polygon the number of points can be computed by simply counting the number of points it consists of.

For the generalized polygon this is a bit more difficult since the decomposition introduces extra points as shown in Figure 3.23. To calculate the number of points for the decomposed polygon with multiple cells a union is performed which results in a single polygon but still consists of multiple points per line. To solve this a geometric simplification can be performed with an extremely small distance since the lines are already perfectly parallel. The result is the total number of points of the generalized shape, less points being better.

3.6.2 Contour trueness

The contour of a building is the outline of the footprint. The total amount of deviation between the original and generalized contour lines shows the change in shape of the building. Since the generalization process tends to arrange lines in parallel arrangement the overlap will return no results due to small variations. To accomplish the overlap of

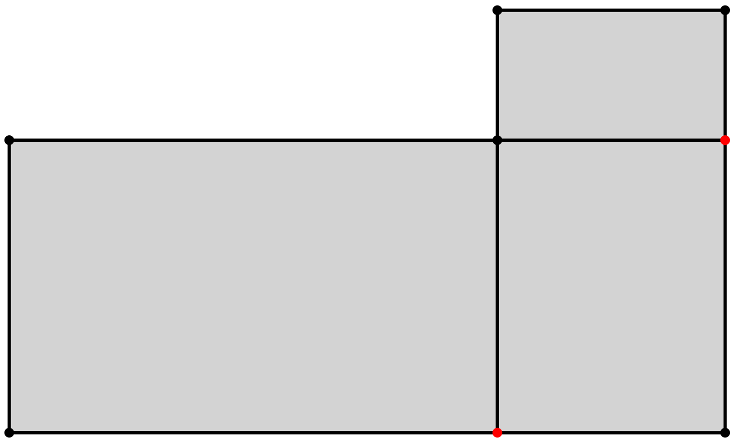


Figure 3.23: Extra points (red) introduced by decomposition.

slightly changed lines a small buffer is put around the original footprint O. This buffer is then intersected with the generalized contour lines G to compute the contour trueness of the generalization. Since the use of a buffer creates slightly better results the buffer should either be created with flat endings or the intersection should be modified to cancel out the negative effect. The contour trueness is calculated using Equation (3.14).

$$CT = \frac{length(G \cap Buffer(O))}{length(O)} \tag{3.14}$$

3.6.3 Symmetric difference

The symmetric difference of original and generalized footprints is described by a combination of intrusion and extrusion. The intrusion of a footprint is the difference in area which is described by the original but not the generalized footprint, thus the parts that are deleted. For extrusion it is the opposite, this is described by the generalized but not the original footprint, thus the added parts. Both the intrusion and the extrusion are calculated using set theory as shown in Equations (3.15) and (3.16).

$$I(O) = O \setminus G \tag{3.15}$$

$$E(O) = G \setminus O \tag{3.16}$$

A visual representation of the intrusion and extrusion can be seen in Figure 3.24. The original outline (red dashed) overlaid on top of the generalized footprint. The green selection is the intrusion where the blue selection is the extrusion of the footprint resulting from the generalization. The combination of both metrics is the symmetric difference as shown in Equation (3.17).

$$SD = 1 - \frac{Area(I(O) + E(O))}{Area(O)} = 1 - \frac{Area((O \setminus G) + (G \setminus O))}{Area(O)} \tag{3.17}$$

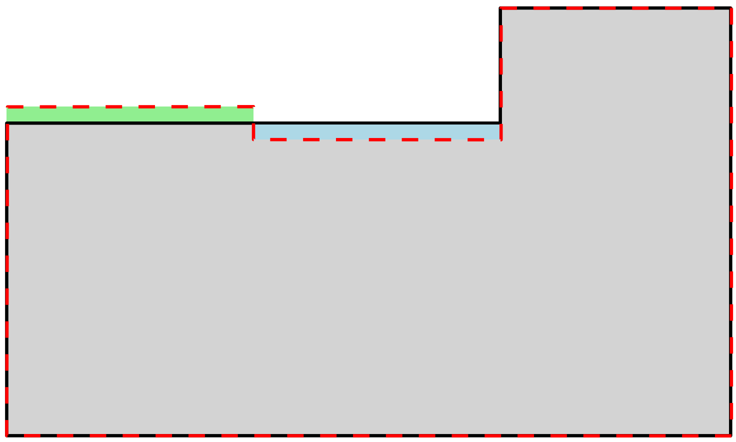


Figure 3.24: Intrusion (green) and extrusion (blue) resulting from generalization, overlaid on the original footprint (red dashed).

3.6.4 Area difference

The area difference is a metric which shows the overall area change of the footprint when generalized. The difference between intrusion and extrusion is the change in area. For the footprint shown in Figure 3.24 the area difference is the area of the green selection minus the area of the blue selection. The difference can be either a positive or a negative value, since the generalization can also produce footprints larger then the original. The area difference metric is required as a value between zero and one, thus the absolute value of the difference is taken. The area difference is calculated using Equation (3.18).

$$AD = 1 - \frac{|Area(O) - Area(G)|}{Area(O)} \tag{3.18}$$

3.6.5 Hausdorff distance

The Hausdorff distance is the maximum distance between two geometries. When the Hausdorff distance is calculated for the original and generalized footprints this denotes the maximum generalized distance. When the distance is much larger then the set generalization distance ($\epsilon_{distance}$), this indicates the removal of a long but narrow part of the building footprint. The distance calculated for the example shown in Figure 3.25 is the black dashed line. The original footprint is the blue plus the red outline where the red part is the are removed due to generalization. The calculated distance for this example is however not the desired one, the preferred result is the Hausdorff distance to be calculated while staying within the building footprint. An adjusted Hausdorff distance calculation is introduced by Filippovska & Kada [2010] and explained in the following section.

Adjusted Hausdorff distance

The adjusted Hausdorff distance is an altered version of the original Hausdorff distance for the use of building generalization quality measurement. The adjustment resides in the

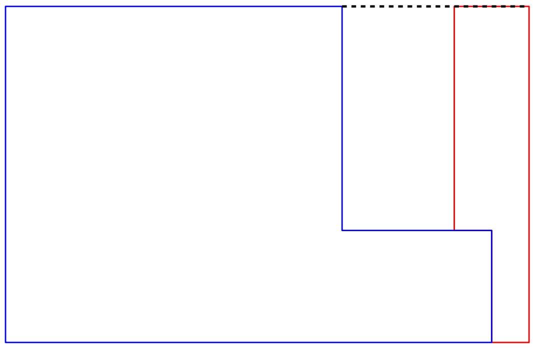


Figure 3.25: The Hausdorff distance (black dashed) between the original (blue with red) and generalized (blue) footprint.

constraint of calculating the maximum distance within the symmetric difference of the building footprint.

The first step is to calculate the intrusions and extrusions of the generalized footprint as explained before, these can be seen in Figure 3.26b. When the intrusions and extrusions are known, the lines corresponding the the original and the generalized need to be found from the contours. The contour of the symmetric difference is intersected with the contour of the original to find the lines corresponding with the original, shown in red in Figure 3.26c. The same is done with the contour of the generalized footprint which results in the blue lines shown in Figure 3.26d. This step is repeated for all intrusion and extrusion parts.

The last step is to calculate the Hausdorff distance between the lines corresponding to the original and the generalized contours. The maximum Hausdorff distance is calculated within all intrusion and extrusion elements of the building footprint. The largest distance from all elements is taken and reported as the adjusted Hausdorff distance as shown with the black dashed line in Figure 3.26e.

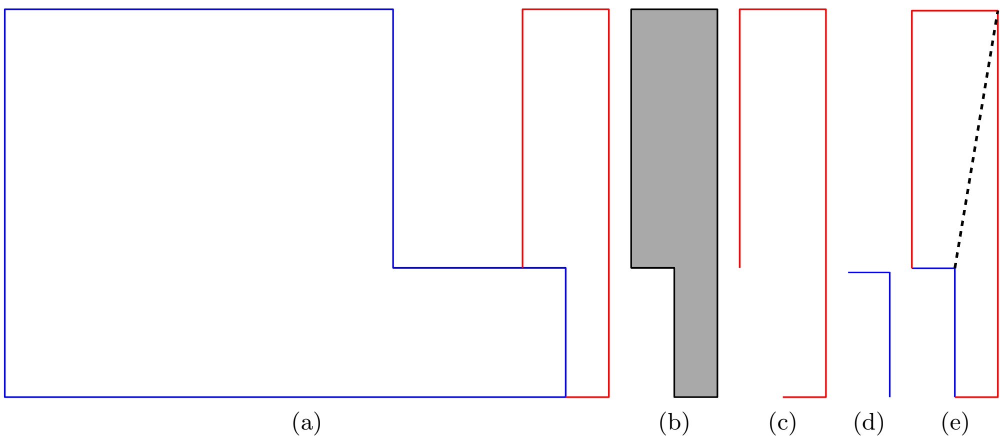


Figure 3.26: Adjusted Hausdorff distance using symmetric difference. (a) Original (red plus blue) and generalized (blue), (b) symmetric difference, (c) contour corresponding to original, (d) contour corresponding to generalized, (e) adjusted Hausdorff distance (black dashed).

Chapter 4

Segmentation of point clouds into planar faces

The segmentation of point clouds into planar faces is described within this chapter. The segmentation algorithm and implementation used is that of Oude Elberink & Vosselman [2009] which is based on the research of Vosselman et al. [2004].

A point cloud consist of points with no knowledge about their relationships. The point cloud only describes the spatial location of measurements, in this case a 3D position. In order to gain insight in the relation between those points they can be grouped in multiple ways. This grouping of points is called segmentation, the creation of segments of points. The segmentation used within this research is the creation of segments of points describing planar surfaces. A planar surface is a surface in 3D space which fits to a flat plane equation.

For this thesis research the existing implementation of the segmentation by Oude Elberink & Vosselman [2009] is used. From their software library the programs for reading, segmentation and writing of points using their own format are linked together to form the process needed for this thesis research. More details on the created program is described in Chapter 6. Even though this existing implementation is used there is still the need for understanding the process. The segmentation depends on multiple parameters needed to be set correctly to obtain results applicable in this research.

The data needs preprocessing before the segmentation of the points and postprocessing of the segments to be usable within the rest of this research. The segmentation workflow is shown in Figure 4.1. The input are the generalized building footprints, as output from Chapter 3, and the point cloud. The output is a list of points containing their segment id and normal vectors of its segment. The process consists of clipping the point cloud to the generalized building footprints which is described in Section 4.1. After clipping the point cloud it is segmented and the normal vectors of the described surfaces are added which is explained in Section 4.2. The last section covers postprocessing of the points list into segments. All covered in Section 4.2 belongs to the existing implementation of Oude Elberink & Vosselman [2009], the rest of the chapter is created for this research.

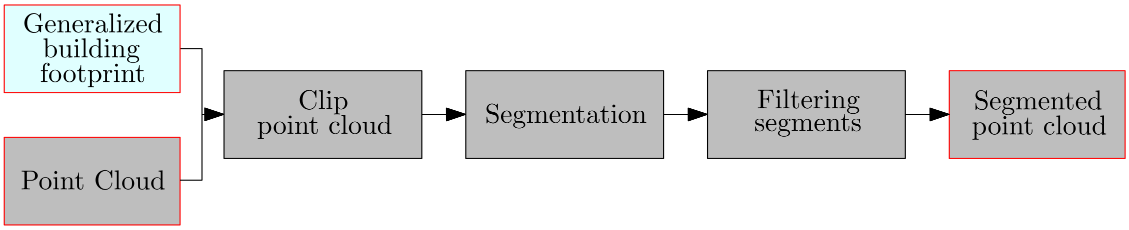


Figure 4.1: Segmentation process.

4.1 Clipping the point cloud

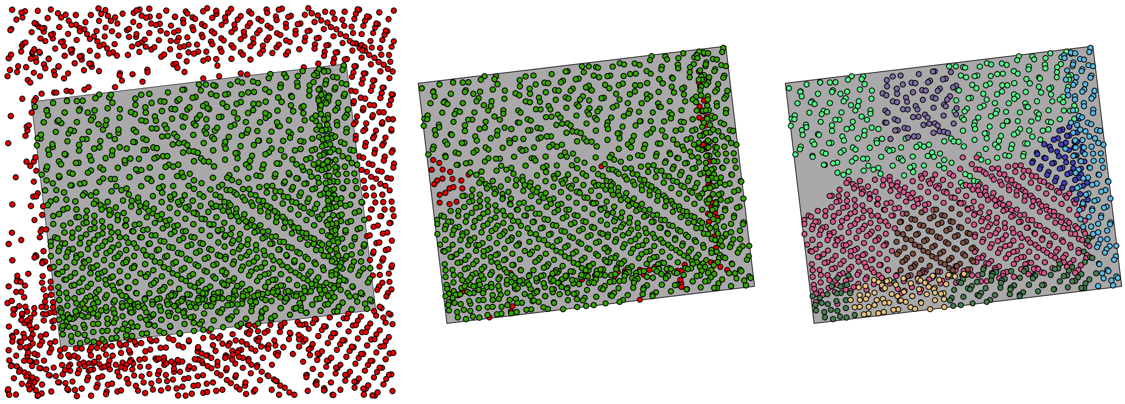
Point clouds often consist of more data then covered by buildings only, points outside the building footprints are not of interest when reconstructing roofs either in LoD1 nor LoD2. Besides the points outside the buildings being unneeded, the points are segmented per building. The reason for segmenting the points per building is that the segments produced are known to belong to that building and do not describe roof planes of other, possibly adjacent, buildings.

The points outside a building footprint are said to be of no use when reconstructing roofs. However roofs can contain overhanging roof parts. These overhanging roof parts are not taken into account within this research. The reason for this choice is that the input data then needs strict requirements. The first is that the building footprint needs to describe the location of the building walls, which is not always the case since building footprints can describe a roof outline. Second the building footprints and the point cloud may not be shifted, as explained in Section 4.3.1, this creates a displacement between the roofs and the building walls. These data set requirements conflict with the thesis requirement that the input data can come from various sources. Besides, generalization of the building footprint would not be possible since this would create displacement of the walls.

The resulting point cloud is one that is clipped to the generalized building footprints and segmented within every footprint separately. An example of a point cloud clipped to a building footprint is shown in Figure 4.2a. After the segmentation additional points are removed due to not belonging to any segment. An example of points that are removed after the segmentation is shown in Figure 4.2b where the resulting segmentation of the clipped points is shown in Figure 4.2c.

When the point cloud is clipped to the generalized building footprint, the points are ready to be segmented. The segmentation process requires the input points to be in ASCII format. The points left after the the clipping are output as ASCII in the following format:

$$x_1, y_1, z_1$$
$$x_2, y_2, z_2$$
$$\dots$$



(a) Point cloud clipped to the generalized building footprint, green points are kept (b) Clipped points (red) overlaid with points left after segmentation (green) (c) Clipped point cloud segmented per building, colored by segment

Figure 4.2: Clipped and segmented point cloud.

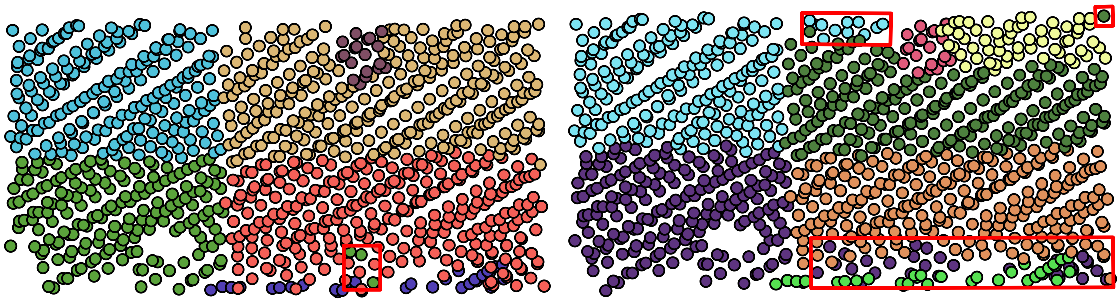
4.2 Segmentation using surface growing

The segmentation used in the existing implementation is a surface growing algorithm with seed detection based on a 3D Hough transform [Vosselman et al., 2004]. The selection on the seed surfaces a 3D Hough transform applied on the k-nearest points of an arbitrarily selected point. The transform indicates if a minimum number of points is within a single plane, whereafter the parameters of the plane are improved by a least squares fit [Vosselman & Dijkman, 2001]. The seed detection accepts a seed point when an acceptable seed surface is found. No search for the optimal seed is performed in order to improve the processing speed of the seed detection.

After the selection of the seed point, the point is used for surface growing. Points are added to the surface when their distance to the estimated plane is smaller then a given threshold. The threshold is used to allow the surface to contain a small curvature and a small amount of noise. The parameters of the estimated plane are recalculated after adding a new point to the surface accept when the distance is very small in which case the parameters are copied for faster processing. If no more points can be added to the surface, the seed point detection is repeated. After processing all possible seed points the segmentation process is completed.

The segmentation algorithm uses k-nearest neighbor (k-NN) to find seed points [Oude Elberink & Vosselman, 2009]. This k-NN can be performed in either 2D or 3D. The difference in segmentation results using 2D or 3D k-NN is compared for usability with decomposition. The one giving the best suitable results is chosen. The results of the segmentation using 2D k-NN are shown in Figure 4.3a which can be compared with the segmentation using 3D k-NN in Figure 4.3b.

The results of the segmentation differs significantly when comparing the segmented points from a top view. Since the segmentation is used in combination with the decomposition, this top view is used for further processing of the segments. In both examples in Figure 4.3 wrongly segmented points are outlined in red, these points are present within the area described by a segments but belong to another segment.



(a) Segmentation using 2D k-nearest neighbor (b) Segmentation using 3D k-nearest neighbor

Figure 4.3: Segmentation of a point cloud within a building footprint, colored per segment.

The actual roof shapes are two gabled roofs with a height jump between the left and right part of the building, where the right gabled roof contains a dormer. A sketch of the actual roof is shown in Figure 4.4. Comparing the 2D and 3D k-NN it shows that the resulting number of segments differs and the 3D k-NN segmentation contains more wrongly segmented points. Due to these points, the segments are not separated by a distinct intersection line but overlap each other. The 2D k-NN example shows a clear separation of the four gabled roof planes, except for the three points of the green segment.

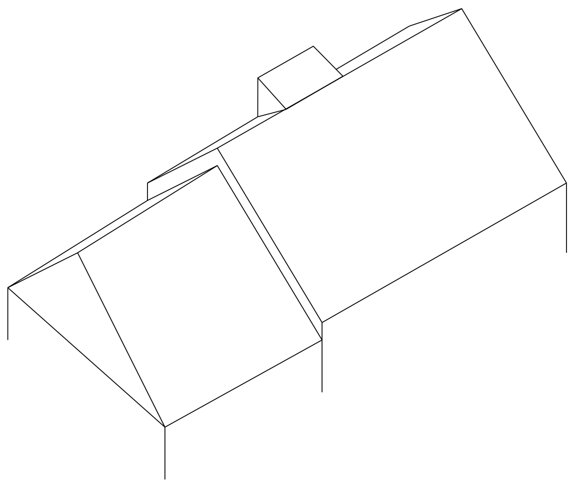


Figure 4.4: Sketch of the roof shape described by Figure 4.3.

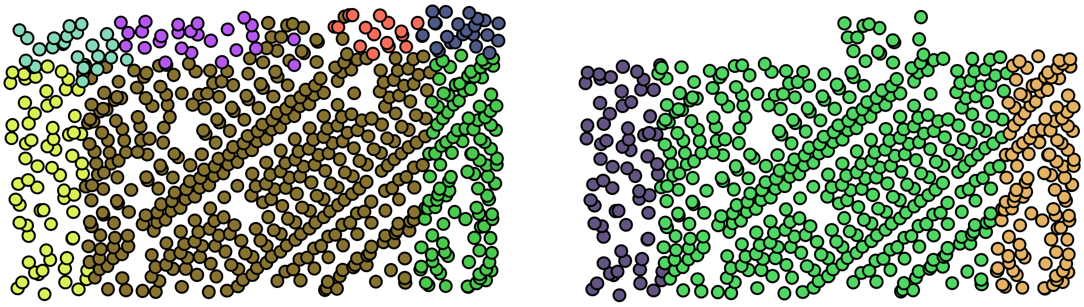
Both examples contain a segment not belonging to the building, being the long small segment on the bottom side of the building. This segment belongs to the building adjacent to it. It falls within this building due to the generalization, georeferencing and other influences described in Section 4.3.1.

The decomposition of the building footprint is done from a 2D perspective in a top view where the point cloud is 3D. The combination of the 2D perspective of the decomposition and the results obtained by the segmentation lead to the conclusion of using the 2D k-NN for the segmentation of the point cloud.

4.2.1 Minimum segment size

The segmentation is used to eliminate small objects which possibly disturb the roof detection. By using a minimum amount of points contained by a segment, small objects can be easily removed. The segmentation writes these points without a segment number by which they can be easily removed.

These segments, describing small planes, do not have additional value to the building reconstruction and are considered noise. When the small segments are removed some information about the roof will be lost but this is such a small amount that the results improve. The selection of the minimum amount of points per segment is depending on the point density of the point cloud and the required amount of detail in the detected roof planes. When using a small number of minimum points, many small roof planes will be detected which can disturb the detection of roof shapes. For a larger number of minimum points only the main planes of a roof are detected, however smaller but important planes will not be found. The segmentation of points using a small value for minimum segment size is shown in Figure 4.5a compared the result of a large value in Figure 4.5b.



(a) Segmentation using a minimum segment size of 10 points

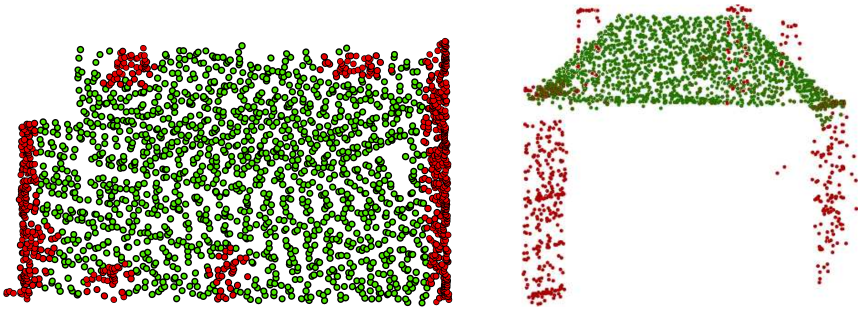
(b) Segmentation using a minimum segment size of 50 points

Figure 4.5: Segmentation of a point cloud using minimum segment size.

4.2.2 Maximum segment slope

Besides filtering on the minimum size of a segment, also the slope can be used for selecting useful segments. Man made structures rarely have roofs with a slope close to vertical, these parts are for example walls or height jumps. Given this knowledge the segments that have a slope close to vertical can be filtered out. The remaining segments are describing roof parts without any vertical planes like walls and height jumps. An example of filtering vertical segments with a slope larger then 70 degrees is shown in Figure 4.6.

Given the knowledge of vertical slopes one is able to select the vertical segments that are close to vertical. These segments could describe walls or height jumps, as noted before. The decomposition could be improved by for example using the vertical segments with a minimum distance from an existing line as an extra decomposition line. However these vertical segments are not taken into account during this research since they interfere with other, non vertical, segments if they describe other shapes like chimneys.

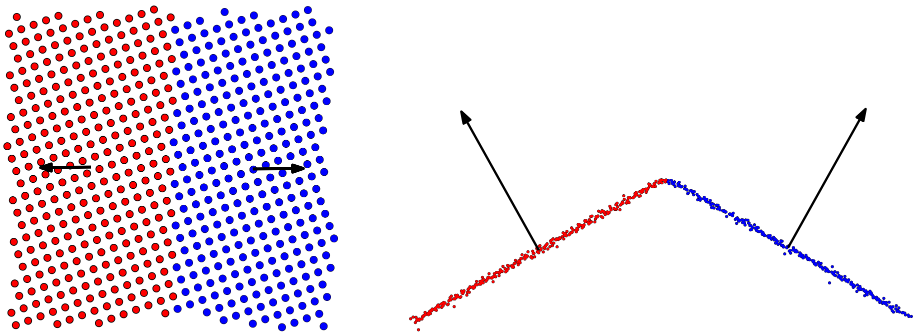


(a) Top view of segments (b) 3D side view of segments

Figure 4.6: Points in segments with a slope larger then 70 degrees detected by the segmentation (red) and other segments (green).

4.2.3 Normal vectors

During the segmentation of the points, a plane is fitted through the points belonging to a single segment and is adjusted using least squares fit. The linear equation of the fitted plane can be used to calculate the normal vector describing the direction of the plane. A normal vector is a direction vector perpendicular to the plane it describes. Using the normal vectors, conclusions can be drawn about the roof shape of a building. Since the normal vectors of the planes are already calculated during plane fitting of the segmentation, these just have to be extracted and added to the segmented points. Figure 4.7 shows an example of a gabled roof with its calculated normal vectors plotted on top of the segmented points.



(a) Top view of segments (b) Side view of segments

Figure 4.7: Normal vectors from segments of points, colored by segment.

4.3 Postprocessing the segmentation

When the segmentation of the points is completed using the existing implementation, the data needs to be post-processed in order to be usable within this research. The output of the segmentation is in ASCII format, same as the input. The output is a list of the points together with their segment number and the normal vector of the segment. The ASCII output is read and the points are grouped using their segment number. The normal vector is stored with the group of points for later use. The format is as following:

$$x_1, y_1, z_1, segment\_number, normal\_x, normal\_y, normal\_z$$
$$x_2, y_2, z_2, segment\_number, normal\_x, normal\_y, normal\_z$$
$$\ldots$$

During the postprocessing the minimum segment size and maximum segment slope, as described in the previous section, are rechecked since the output sometimes still writes these segments. For the rest of the segments extra filtering steps are used to improve the usability of the segmentation. After the filtering of the segmentation is completed, convex hulls are created.

4.3.1 Minimum segment height

Point clouds and building footprints are often created from different sources. These different sources implies various extraction methods which lead to differences in geometric accuracy and georeferencing. Given these facts the point cloud and building footprints can differ in location and shape by using data sets from different sources. The shift of the footprints are for example due to manual processing, using relative or absolute positioning between buildings and the accuracy of the data used to extract the building footprints. An example of different footprint sources overlaid with the points describing the buildings roof is shown in Figure 4.8. The examples also show that the neither footprints cover the correctly segmented roof points.

Since the point clouds can be shifted from the building footprint, segments describing the ground surface can be present in the segmented point cloud. These segments interfere with the building reconstruction thus have to be removed. The segmentation of Oude Elberink & Vosselman [2009] does not handle any kind of minimum segment height.

After the segmentation the average height of the segments is calculated and any segment which height is below a given threshold is removed. An example of segments that are selected based on a 1 meter threshold are shown in Figure 4.9. By doing this the segments that contain only ground points, or points low above the ground in this research fixed at zero, are filtered. The resulting segments contain no more ground points resulting from the shift between the point cloud and building footprint.

4.3.2 Convex hulls

Calculating overlap between a segment of points and a decomposition cell is not useful since only the number of overlapping points can be calculated. Since the area of overlap is wanted, the points need to be converted into an area. Transformation from points to a polygon can be conducted in various ways. The calculation of the convex hull is the most robust point to polygon algorithm available, it describes a set of points using the smallest convex polygon possible [Efron, 1965]. Convex means that from every point in the polygon, any other point in the polygon can be traveled to in a straight line without intersecting its boundary.

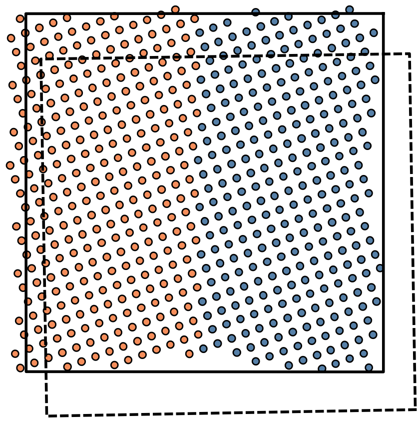


Figure 4.8: Shifted building footprints from different sources.

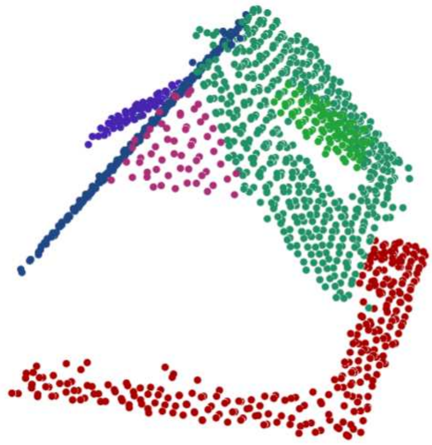
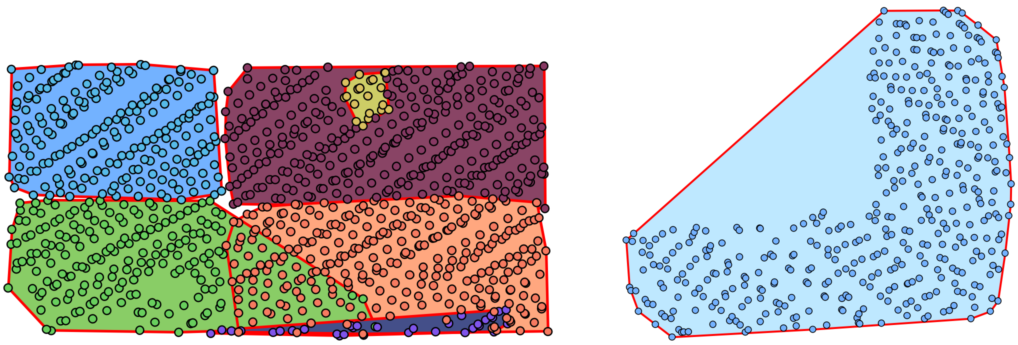


Figure 4.9: Segments with a height lower then 1 meter (red).

Using convex hulls can however introduce problems. When a segment of point contains outliers, like the wrongly segmented points shown in Figure 4.3, the convex hull describes a different area then one would want. Besides outliers, some areas described by a segment, like an L-shape, will not result in a convex hull describing the segment in an optimal way. The convex hull created from a segment with outliers can be seen in Figure 4.10a where in Figure 4.10b the convex hull of an L-shaped segment is shown.



(a) Convex hull of segments containing outliers

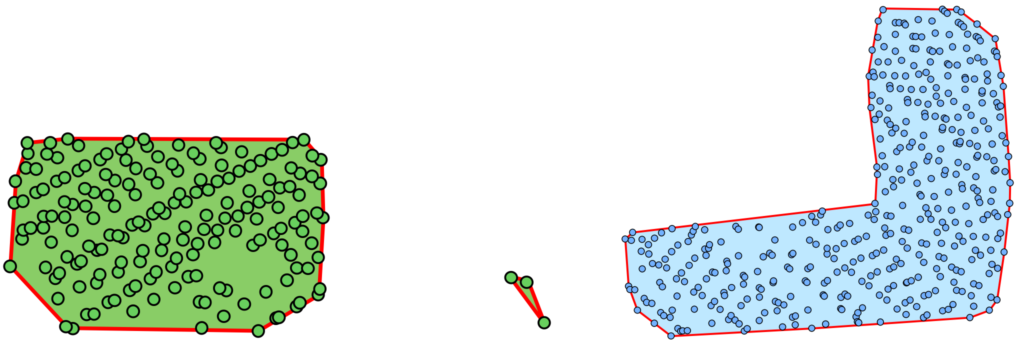
(b) Convex hull of a segment describing an L-shape

Figure 4.10: Convex hulls created from segments of points.

Even though the convex hull in some cases creates unrealistic descriptions of the segments, they are still usable. The results from the given exceptions have do not introduce large problems on the later processing with the decomposition cells as Section 5.2 shows.

4.3.3 Alpha shapes

Some problems introduced by the convex hull could be solved using alpha shapes [Edelsbrunner et al., 1983]. An example of the L-shaped segment represented by an alpha shape is shown in Figure 4.11b. The problem with the alpha shapes is the requirement of a parameter which would be depending on the segments. This parameter might be related to a parameter of the segmentation process, this is however not investigated. The resulting output of the alpha shapes are not predictable given the use of this parameter compared to that of the convex hull. Besides this, the problem given by the outliers is not solved but creates a disjoined polygon describing the segment and the outliers separately. From the lower left segment from 4.10a the alpha shape is created which is shown in Figure 4.11a.



(a) Alpha shape of segments containing outliers

(b) Alpha shape of a segment describing an L-shape

Figure 4.11: Alpha shapes created from segments of points.

Chapter 5

Producing valid 3D geometries

The production of 3D geometries and its validation is described within this chapter. First a description is given on what is considered a valid 3D geometry in Section 5.1, followed by the merging of the decomposition cells of the footprints based on the segmentation of the point cloud in Section 5.2. Thereafter the reconstruction of the geometries in both LoD1 (Section 5.3) and LoD2 (Section 5.4) is explained. The buildings are adjusted based on their adjacency, roof type and height which is also covered within Section 5.6. The final Section 5.7 explains the geometric validation using Oracle Spatial.

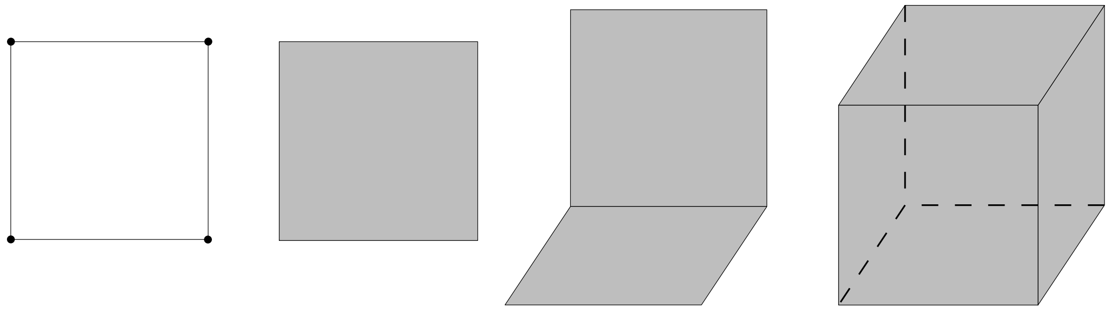
All example figures containing segments given in this chapter are visualized using their points, where in the calculations their convex hulls are used. This is done since the convex hull overlaid on top of the decomposed building footprint is a weaker visualization.

5.1 What is considered a valid 3D geometry

Before describing the creation of 3D geometry it has to be known what is considered to be a valid 3D geometry. A 3D building geometry is modeled as a polyhedron, otherwise referred to as a solid. The International Organization for Standardization (ISO) developed the definition of a solid. A solid consists of multiple boundaries describing its extend, these boundaries are surfaces comparable to its 2D counterpart, a polygon [ISO, 2003].

Let us start from the beginning, one has a collection of points describing a ring where the first and the last point are identical to describe a closed ring. An outer ring combined with zero or more inner rings form a polygon, where is a single polygon is a simple surface. From the surfaces a composite surface is created which together describe a solid. All these steps have to output valid results in order to end up with a valid solid [Kazar et al., 2008]. In Figure 5.1 the steps are visualized using the example of a cube.

As stated before all parts of the solid must be valid. Many validation rules have been described in various publications [Kazar et al., 2008; Ledoux et al., 2009; ISO, 2003]. From the publications a selection is chosen to show the concept of validity. The next sections describe validation rules of all steps used in the creation of a solid.



(a) Ring (b) Surface (c) Composite surface (d) Solid

Figure 5.1: From points towards a solid.

5.1.1 Validation rules for a ring

A ring is described by a set of points from which every set of two points describe a line segment. From the set of points the first and the last point are identical to specify the lines form a closed ring.

Closedness: The first and the last vertices in the ring must be identical.

Planarity: All vertices in the ring must lie in the same plane.

5.1.2 Validation rules for a surface

A surface is described by a single polygon formed by a set of rings. A polygon consists of one outer ring describing the boundaries of the polygon and zero or more inner rings describing holes.

Co-planarity of rings: All rings must be in the same plane.

Orientation: Inner ring must have the opposite orientation of the outer ring.

Contiguous area: The inner rings cannot partition the surface into disjoint areas.

5.1.3 Validation rules for a composite surface

Multiple polygons together form a composite surface. The polygons describing the composite surface do not have to be in the same plane.

Non-overlapping: Any combination of two polygons must not overlap but may only share a (part of a) edge.

Contiguous area: Every surface in the composite should be reachable from any other surface.

5.1.4 Validation rules for a solid

A solid as defined by ISO consists of one composite surface describing the exterior boundaries and zero or more composite surfaces describing the interior.

Single volume: The volume has to be contiguous, the boundary has to be closed and connected.

Orientation: The surfaces are oriented so that the normal vector is pointed outwards, which is defined by the right hand thumb rule.

No inner rings in surfaces: No inner rings are allowed in the composite surfaces of a solid (As implemented by Oracle, strictly speaking it is valid if there are connecting inner boundaries).

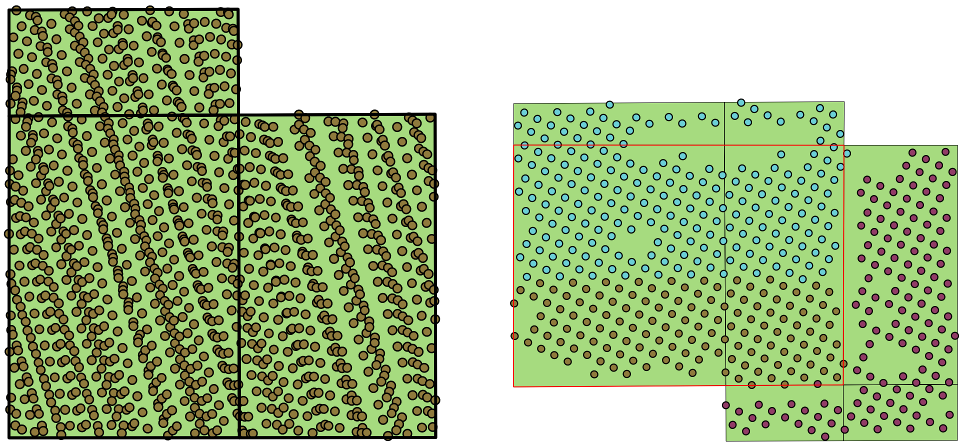
5.2 Merging decomposition cells based on segmentation

The decomposition of building footprints splits the building by extending lines from the footprint outline. Symmetrical features will result from this split. However, a building footprint can be fragmented due to too many decomposition lines. The result of that can be that roof shapes are described by multiple decomposition cells. Since the goal of the research is to create a decomposition which separates the multiple roof shapes and heights, merging these cells is needed.

When a point cloud is used without segmentation, there is no knowledge about the actual roof shapes. Therefore the segmentation results are used to merge the decomposition cells based on their relationship with the segments. A simple example is a segment describing a flat roof which completely covers two decomposition cells, see Figure 5.2a. Of course decomposition cells can only be merged when they are adjacent, which means that they are connected by an edge. Merging disjoint cells into a cell describing a single area is not possible.

Identification of mergeable decomposition cells can be done from two perspectives, one can start from the decomposition cells or from the segments. When looking at the cells, one can identify the segments overlapping a particular cell and check if an adjacent cell has the identical segments overlapping. If these are identical, the cells can be merged since there is no difference in the roof shapes their segments describe, see Figure 5.2b.

Besides starting from the cells, starting from the segments is another possibility. The segments describing a single roof shape can be grouped and the cells that are completely covered by these groups can be merged. For example a gabled roof consists of two non-flat segments that together describe a single roof. The two non-flat segments can be grouped and the cells they cover identified. The cells that are identified can be merged into a cell describing this gabled roof. The problem about this is that knowledge about the adjacency of segments is required. Besides the adjacency, roof shapes have to be described *a priori* and then detected from a set of segments. The adjacency of segments is not taken into account within this research.



(a) Flat segment covering multiple cells

(b) Identical segments (cyan, brown) covering two cells (red outline)

Figure 5.2: Detection of a mergeable cells from segments.

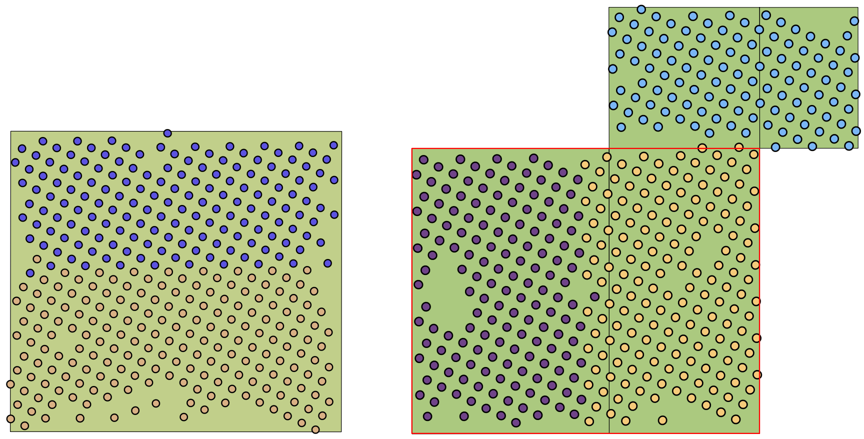
5.2.1 Detection of gabled roof cells

Despite not knowing the adjacency of segments, rules are set up for the detection of a gabled roof. These rules are weaker compared to using segment adjacency but have proven to detect gabled roofs. The gabled roof shape is chosen since this is the simplest roof shape to detect consisting of more then one plane (segment). The gabled roof is described by two planes with normal vectors opposite in x- and y-direction and identical in z-direction, both within a threshold. Two segments that agree these directional terms that are within a building do not necessarily describe a gabled roof but can also describe other shapes like a hipped roof. However when a cell is covered by exactly two segments, it is known that these segments are adjacent. If these segments also agree the directional terms and do not cover any other cells, the cell is said to be gabled. An example is shown in Figure 5.3a. Three rules have to be satisfied for a single cell to be gabled:

- Cell is covered by exactly two segments.
- Segments have normal vectors that are opposite in x- and y-direction and identical in z-direction.
- Segments do not overlap any other cell.

Extending this description, a gabled roof covering two cells can be described. Two cells together describe a gabled roof if one of the two segments is also covering an adjacent cell which is not overlapped by another segment. An example is given in Figure 5.3b. For two cells to be merged because being gabled, the following four rules should be satisfied:

- Cell (1) is covered by exactly two segments (A and B).
- Segments have normal vectors that are opposite in x- and y-direction and identical in z-direction.
- One segment (B) overlaps an adjacent cell (2).
- The adjacent cell (2) is only overlapped by this segment (B).



(a) Gabled roof segments covering a single cell

(b) Gabled roof segments (purple, orange) covering two cells (red outline)

Figure 5.3: Detection of a gabled roof from two segments.

5.2.2 Ordering of merging steps

The detection of mergeable cells is described before, where three different types are described; single segment, identical combination of segments and grouped segments describing roof shapes. The order in which these are applied is important for the results. The ordering of the merging steps is as follows:

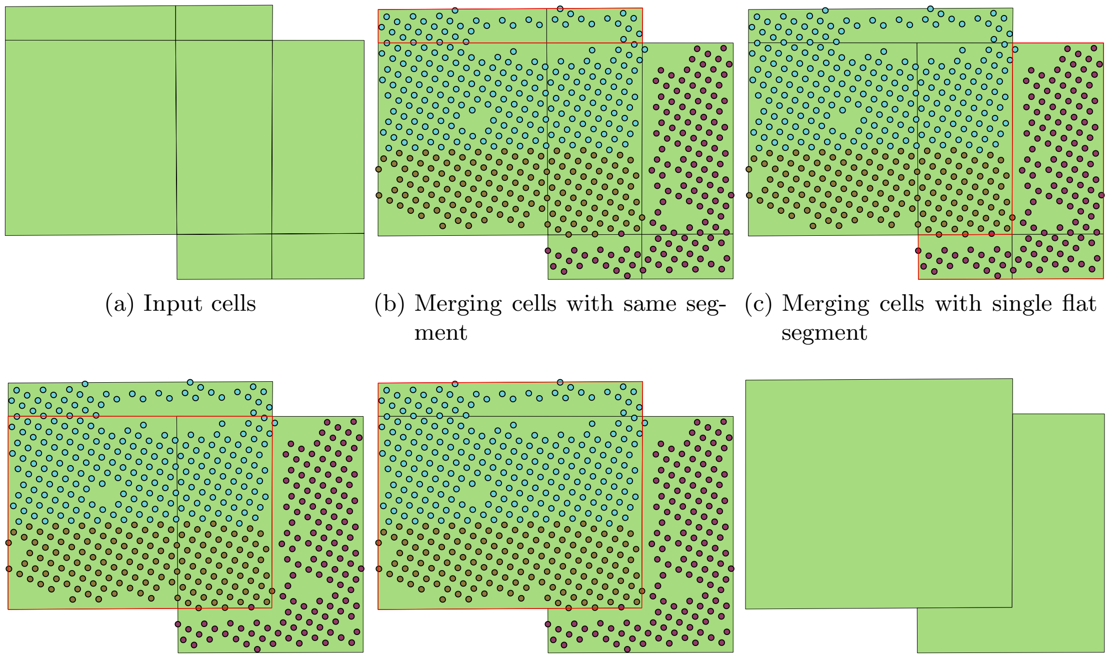
1. Single segment covering multiple cells.
2. Identical combination of segments covering multiple cells.
3. Grouped segments describing roof shapes covering multiple cells.

In a perfect working algorithm where all possible roof shapes are described the first two merging steps would not be necessary. The grouped segments would together describe all roof shapes of the building and thus would result in the merging of all cells in a correct way, assuming the roof shapes are separated by the decomposition which is not always the case. However, since not all possible roof shape are and maybe even cannot be described, the first two merging steps are needed.

When using the steps in the given order, the decomposition is merged into a final decomposition. In Figure 5.4 an example using the ordered merging is shown. The input is the decomposed building footprint, the decomposition is merged based on the segments and the output is the final decomposition.

5.3 LoD1 using extrusion

The creation of a geometric model in LoD1 is performed using extrusion [Ledoux & Meijers, 2011]. Simply said extrusion is raising a polygon to a given height and creating a solid. The solid resulting from the extrusion consists of a bottom, top and multiple sides.



(a) Input cells

(b) Merging cells with same segment

(c) Merging cells with single flat segment

(d) Merging cells with identical segments

(e) Merging two cells with single gabled roof

(f) Output cells

Figure 5.4: Merging decomposition cells based on segmentation.

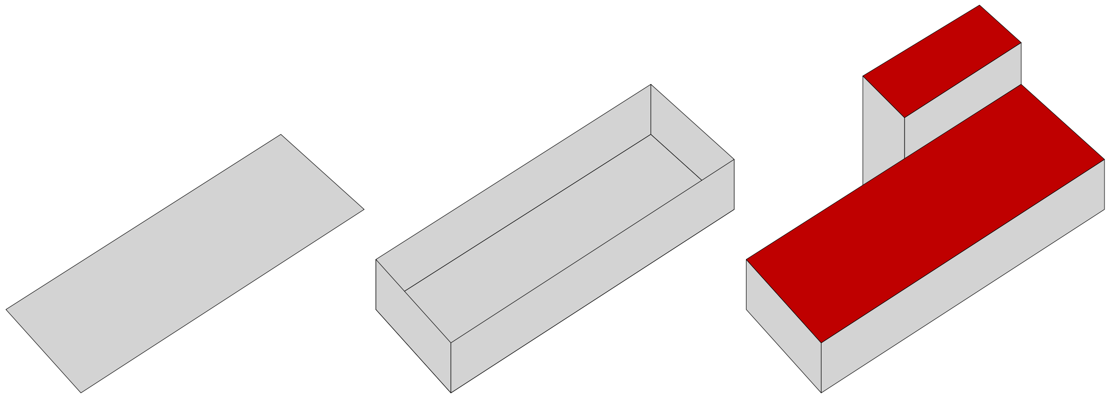
Translating this into terms of a building would mean a floor, roof and multiple walls. The amount of walls depends on the shape of the roof where for a rectangle the amount would be four.

Extrusion of a decomposed building footprint is done by starting with the decomposition cell as the floor. Using the boundary of the floor, walls are created using the height of the floor and the roof and finally a roof is created by adding a height to a copy of the cell. This process is repeated for every cell in the decomposed footprint and result in an extruded building in LoD1 as shown in Figure 5.5. Every decomposition cell creates its own walls, therefore walls of adjacent cells are overlapping.

Before the actual geometric reconstruction is done, the parameters for the reconstruction have to be calculated. In the case of LoD1 these are the heights of the cells which are extruded. The heights are calculated per cell and thereafter averaged if they fall within a given ϵ_{height} threshold. Both the height calculation and the averaging are covered in the following sections.

5.3.1 Acquiring extrusion height

The height a building is extruded to can be calculated in various ways. The height calculation is depending on the use of the model. Mainly there are two options; the average or a fixed proportion of the surface height. There is one exception, flat surfaces, when using either way of height calculation the resulting height of a flat surface should always be the same. Therefore the height calculation for flat surfaces is not covered. To



(a) Floor (b) Floor with walls (c) Extruded building from multiple cells

Figure 5.5: Extrusion of a building footprint.

clarify the height as described here, this is the height of the roof from the lowest to the highest point of the roof without taking into account the height of the walls. The average and fixed proportion calculations are explained in the following.

Average roof height

Before calculating the average height of a decomposition cell, the average height of all segments is calculated and stored. This is done since a segment can overlap multiple cells which would require the same calculation multiple times. The average segment height (ASH) is acquired by using Equation (5.1) where every z is the height of a point within the segment.

$$ASH = \frac{1}{n}\sum_{i=1}^{n} z_i \tag{5.1}$$

Now the average heights of the segments are known these can be used to determine the height of the decomposition cell. Since the segments have different sizes, a weighted average is used. The weight of the segments is the area the segments overlaps with the cell. The 2D convex hull of the segment is used to find this overlap. The average height for the decomposition cell (CH) is calculated using Equation (5.2) where S are the overlapping segments and C is the cell.

$$CH = \frac{\sum_{i=1}^{n} Area(S_i \cap C) \times ASH(S_i)}{\sum_{i=1}^{n} Area(S_i \cap C)} \tag{5.2}$$

Fixed proportion of roof height

The fixed proportion of a cells height means that the height is taken at for example two thirds of the total roof height. This fixed proportion is based on the lowest roof height, the eaves, and the highest, the ridge as shown in Figure 5.6. Calculation of these two heights is difficult since these heights are not exact boundaries in the point cloud. Therefore the heights will have to be approximated.

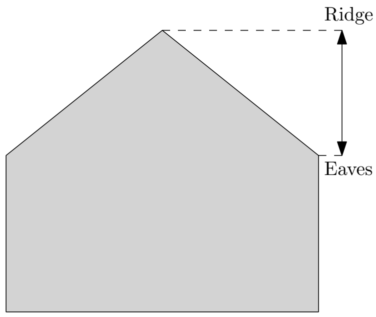


Figure 5.6: Location of eaves and ridge heights.

There are multiple ways of calculating the eaves and ridge heights. One could take the lowest point of all covering segments for the eaves and the highest point of all segments as the ridge height. However when a segment contains outliers, the lowest point is not always representable for the eaves. The same condition applies to the highest point for the ridge height. The calculation time of this is low.

The heights can also be calculated by intersecting 3D planes describing the segments. The intersection between the plane describing a segment and the vertical wall plane created from the cells outline represents the eaves height where the highest intersection between planes describing segments represents the ridge height. In order to calculate the heights this way, the 3D plane equations of the planes fitted to the segments are needed. This approach would probably give the best results.

Another possibility is taking an average of the lowest and highest 5% of the points in a segment. The eaves height will be a bit higher than the lowest points where the ridge will be a bit lower but possible outliers are filtered. The distribution of the points within the segments is unknown which can reflect in the 5% selected points being distributed far from the actual eaves or ridge which disturbs the calculated values. The calculation time is average due to the multiple points and their selection.

After determining the eaves and ridge heights the cell height (CH) calculation is as simple as deducting the eaves height from the ridge height and multiplying it by the given fixed proportion (P) as shown in Equation (5.3).

$$CH = (Ridge\ height - Eaves\ height) \times P \tag{5.3}$$

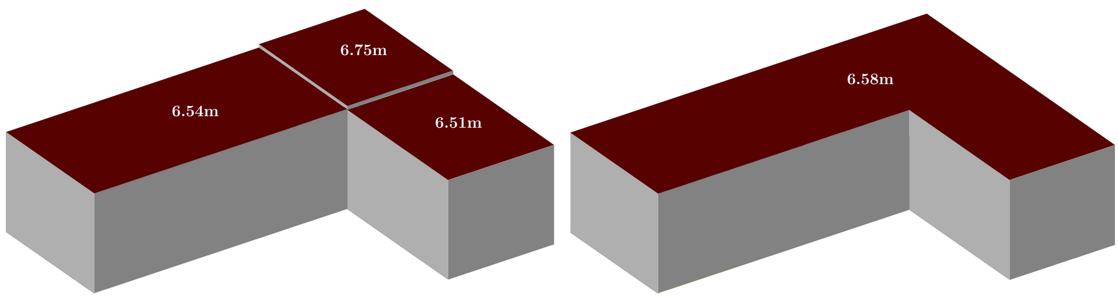
Since the cell heights are adjusted based on their neighboring cells, as described the following section, the minimum and maximum points are used for the eaves respectively the ridge height.

5.3.2 Adjusting cell heights

The heights of the cells are calculated and could be used for extrusion directly. Since the height of the cells is not the real-life height but an approximation, small differences between neighbor cell heights can occur. When these small differences are not averaged, the output model can become less smooth from a visual point of view and add unneeded complexity to calculations made using this model. These small differences can also be a result of the height calculations itself where for example roof points are missing due to incomplete data.

Given the previous reasons the cell heights are averaged if they are within a ϵ_{height} threshold. The new cell height (CH) is calculated using a weighted average where the used weight is the area of the cells within the threshold (C) shown in Equation (5.4). The new heights are assigned to the corresponding cells and the cells that are adjacent and have identical height are then merged into a single cell. A building before the adjustment and the improvements after the adjustment are shown in Figure 5.7b.

$$CH = \frac{\sum_{i=1}^{n} Area(C_i) \times CH(C_i)}{\sum_{i=1}^{n} Area(C_i)} \tag{5.4}$$



(a) Before adjusting cell heights

(b) After adjusting cell heights and merging cells with identical height

Figure 5.7: Improvements using adjustment of cell heights.

5.4 LoD2 using parametric shape fitting

The focus of this research is on the creation of a LoD1 model with different extrusion heights for roof shapes. As described before with the example of the gabled roof in Section 5.2.1, there is a need for detecting the roof shapes in order to correctly merge the decomposition cells. The step from the detection of the roof shape and the actual geometric reconstruction for a LoD2 model is not that large since the parameters are already calculated. After merging the cells describing a single roof shape, the newly created cell can be marked with the correct roof shape since that is already known. The result is that after the merging step the cells are already marked with its detected roof shape.

5.4.1 Detection of roof shapes

The detection of roof shapes is depending on the *a priori* described roof shapes as in parametric shape fitting. To show the usability of this research for the creation of a LoD2 model some roof types are modeled and reconstructed. Within this research a selection of four roof types is used; Flat, Shed, Gabled and Unknown.

The roof types are only applied when it is certain a roof is of that type. If the detection is not certain, the roof will be marked as type Unknown. For example if a gabled roof has a dormer or overlaps part of another cell, the roof is marked as type Unknown. Another example is a cell containing multiple flat segments which are not split during the decomposition are also marked type Unknown.

The described roof types are detected based on the normal vectors of the segments. The types are detected per decomposition cell by analyzing the overlapping segments. The roof types are described as follows:

Flat The cell has exactly one segment which not overlaps any other cells, the normal vector of the segment has a maximum angle of ϵ_{normal} with the vertical axis.

Shed The cell has exactly one segment which not overlaps any other cells, the normal vector of the segment has a minimum angle of ϵ_{normal} with the vertical axis.

Gabled The cell has exactly two segments which not overlap any other cells, the normal vectors of the segments have a minimum angle of ϵ_{normal} with the vertical axis, have an angle of 180° in their x- and y-direction with a variation of ϵ_{normal} and differ maximum ϵ_{normal} in their z-direction.

Unknown Any other possible combination of segments.

NoSegments The cell has no overlapping segments.

5.4.2 Reconstruction of flat roofs

If the detected roof type of a cell is flat then the reconstruction is done the same way as described in the extrusion for LoD1. The average height of the segment is calculated which is used for the height of the roof polygon. The polygon describing the floor is a copy of the roof polygon with a reversed orientation so the normal is pointing downwards and its height set to zero. The walls are calculated iteratively by walking through the ring of the roof polygon. A wall polygon consists of four points, where the upper two are identical to the bottom two except for the z-coordinates. Two points from the ring of the roof polygon are used whereafter these are copied and their z-coordinate set to zero for the third and fourth point of the wall polygon. The example building is shown in Figure 5.8 where the walls are reconstructed with the order shown in Table 5.1. The floor is set to zero since the ground height is unknown. A way to determine this is to use a DTM and find the ground height within the building footprint.

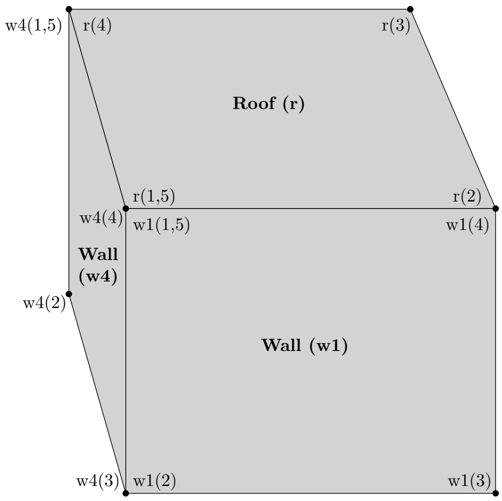


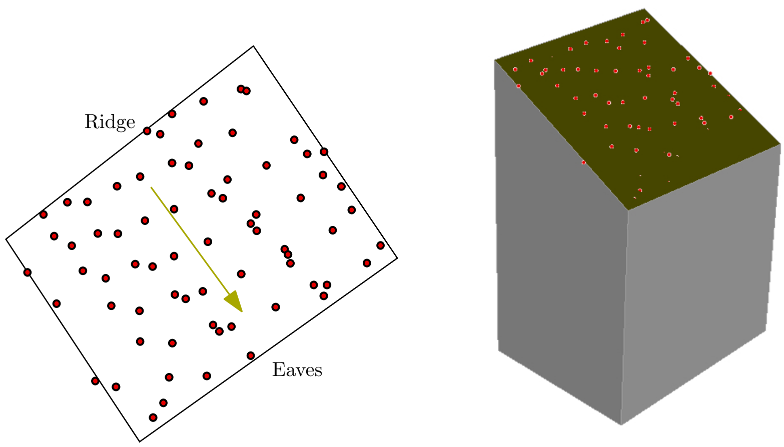
Figure 5.8: Reconstruction of walls.

Wall 1	
w1(1) =	r(1)
w1(2) =	r(1) with z-coordinate at 0
w1(3) =	r(2) with z-coordinate at 0
w1(4) =	r(2)
w1(5) =	w1(1)
Wall 4	
w4(1) =	r(4)
w4(2) =	r(4) with z-coordinate at 0
w4(3) =	r(5) with z-coordinate at 0
w4(4) =	r(5)
w4(5) =	w4(1)

Table 5.1: Wall reconstruction, ordering of points

5.4.3 Reconstruction of shed roofs

The detected shed roofs consist of a single segment, where for the shed roofs these are not horizontal. The shed roof has two different heights, the eaves and ridge. The heights are determined using the previously described minimum and maximum point calculations in Section 5.3.1. When the two heights are known, the roof polygon has to be raised using these. The heights may be known but their location, which side of the polygon is which height, is unknown. To identify the correct eaves side, the direction of the segments normal vector is used. The direction the normal vector is pointing in, in x- and y-direction, is the eaves side where the opposite direction is the ridge side, see Figure 5.9a.



(a) Location detection of eaves and ridge using segments normal vector

(b) Reconstructed shed roof overlaid with points

Figure 5.9: Reconstruction of a shed roof.

The reconstruction of the walls and floor is identical to that of the flat roofs described in the previous section. The walls will automatically use the correct heights since these are copied from the roof polygon. A completely reconstructed building with a shed roof overlaid with its points is shown in Figure 5.9b.

5.4.4 Reconstruction of gabled roofs

The gabled roof is the most complex shape reconstructed within this research. The roof consists of two roof planes with opposite direction and identical angles. The first step in the reconstruction is the detection of the ridge line. The ridge is forced to be in the center of the roof, thus no a-symmetric gabled roofs are reconstructed. Besides the symmetry, non-rectangular cells are not reconstructed at this stage but are detected. The ridge line detection is, identical to the shed roofs, done using the normal vector of the segments. The sides of the polygon where the normal vectors point to are the eaves. The other two sides are split in their center and the two roof polygons are created as in Figure 5.10.

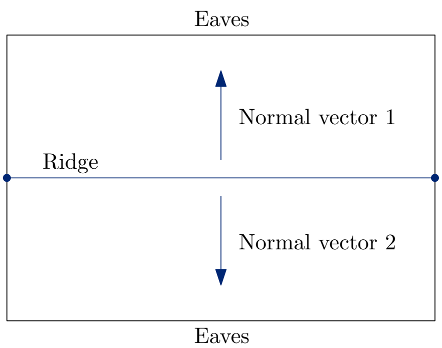


Figure 5.10: Roof polygon split using the calculated ridge line.

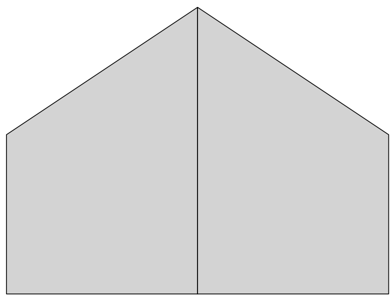


Figure 5.11: Reconstructed ridge side wall using the same approach as for flat roofs.

Since the both the eaves and ridge are known the sides of the two polygons are raised to the respective height. The points that are identical in the two are the ridge. The calculation of the wall polygons is different. The validation of a solid disallows connected polygons in the same plane and requires these to be a single surface. Therefor the wall reconstruction as used for the flat and shed buildings is not usable since this reconstructs the walls at the ridge side in two parts, Figure 5.11. The walls on the ridge side of the building are reconstructed in an alternative manner, the point that describes the ridge is combined with the two points next to it when reconstructing the wall. The example building in Figure 5.12 shows this reconstruction together with the point order described in Table 5.2.

When the roof, all the walls and the floor are reconstructed the reconstruction of a building with a gabled roof is completed. Due to the forcing of the symmetry of the ridge line of the gabled roof, the actual points are differing from the reconstruction as shown in Figure 5.13a. Both there can be a shift from the points towards the reconstructed roof and even the ridge can be a bit shifted. As explained before, the point cloud and the building footprints can be shifted, therefor the choice is made to fix a ridge in the center of a roof. The result for the reconstruction of a gabled roof overlaid with the original points is shown in Figure 5.13b.

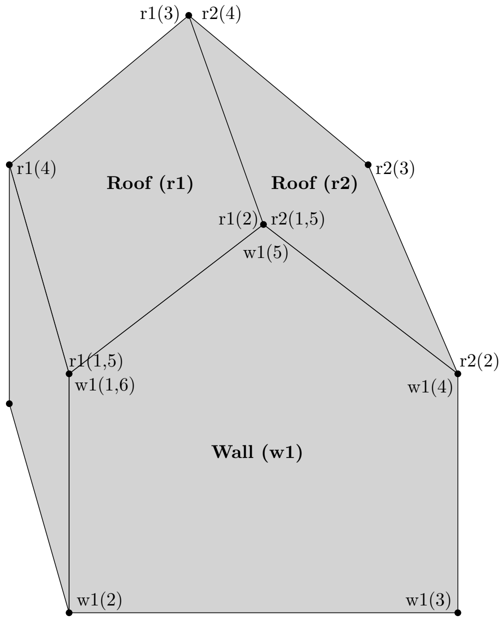


Figure 5.12: Reconstruction of ridge side walls.

Wall 1	
w1(1) =	r1(1)
w1(2) =	r1(1) with z-coordinate at 0
w1(3) =	r2(2) with z-coordinate at 0
w1(4) =	r2(2)
w1(5) =	r2(1)/r1(2)
w1(6) =	w1(1)

Table 5.2: Ridge side wall reconstruction, ordering of points

5.4.5 Reconstruction of unknown roofs

The building cells which are left after the detection of the aforementioned types are reconstructed and labeled as type Unknown. The geometric reconstruction is similar to that of the flat one besides the height which is a weighted average of the segments as described in the LoD1 reconstruction.

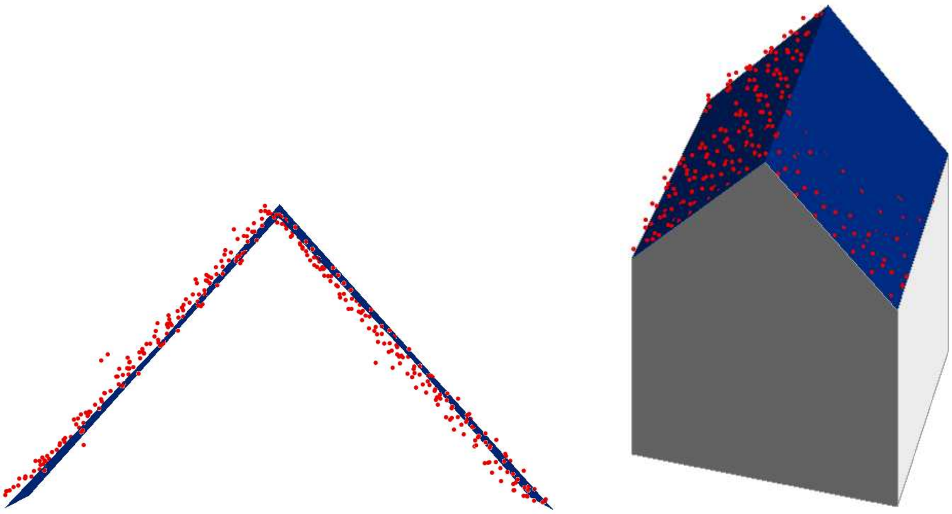
Besides the roofs of type Unknown there is the possibility of a cell which does not have any overlapping segments. When this occurs the cell polygon is kept as is, no walls and roof are reconstructed, and is marked as type NoSegments.

The amount of cells of type Unknown can be reduced in several ways; implementing more roof types, applying the best fitting model and including segment adjacency knowledge, for instance a RAG, to improve the roof type detection. The application of the best fitting model could solve the problems occurring with detection in the presence of dormers.

5.5 Building roof type parameters

During the roof type detection several parameters about the roof are calculated which could be interesting in the use of the model. These parameters are stored as attributes with the geometry of the model in both LoD1 and LoD2. Not every roof type has the same attributes where most here described are used for shed and gabled roofs. During the research the choice made together with iDelft is to store all calculated parameters of the roof shapes. These values are of use when the final model is used for calculations. The attributes stored are the following:

Roof type The type of the roof; Flat, Shed, Gabled, Unknown, NoSegments.



(a) Side view (b) Birdseye view

Figure 5.13: Reconstruction of a gabled roof overlaid with points.

Roof height The calculated height for the extruded building cells, for Flat and Unknown roof types.

Eaves height The calculated eaves height, for Shed and Gabled roof types.

Ridge height The calculated ridge height, for Shed and Gabled roof types.

Roof angle The angle in which the roof is tilted, calculated from the difference between the normal vector of the segment and the normal vector of the horizontal plane, for Shed and Gabled roof types.

X-direction The x-direction the roof planes normal vector, for Shed and Gabled roof types. Where for Gabled roof types the other plane is pointing 180° opposite.

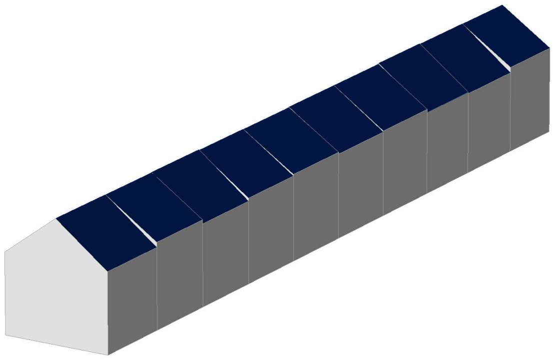
Y-direction The y-direction the roof planes normal vector, for Shed and Gabled roof types. Where for Gabled roof types the other plane is pointing 180° opposite.

5.6 Adjusting building blocks

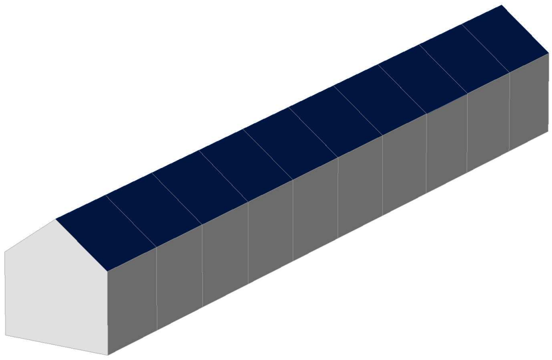
The buildings are reconstructed as single objects as described in the previous sections. However since the possibility of missing data in the point cloud and missing points after segmentation, the heights that are determined are not always the actual heights. The heights between adjacent buildings can differ by a small amount, leaving small differences between the reconstructed roofs. As explained before, the cells within a building are averaged if they fall within a ϵ_{height} threshold. The same approach is used for the adjustment of adjacent buildings with identical roof types. An example of a building block with gabled roofs before the adjustment is shown in Figure 5.14a.

Before the actual geometric reconstruction, the values for all buildings in the block are calculated like roof type and height. These values are adjusted for the building block by averaging adjacent buildings. Adjacent buildings are adjusted if their roof type is similar and the height difference is within the ϵ_{height} threshold. For flat roofs the adjustment is identical to that of the cells within a building. For the shed and gabled roofs the orientation is checked and the eaves and ridge heights are accordingly.

Within this research only roofs with a single roof type are adjusted according to the adjacent buildings. If a building consists of multiple cells describing multiple roof shapes the roofs are not averaged however this is possible. The results of the adjustment of a building block with gabled roofs is shown in Figure 5.14b.



(a) Building block reconstructed as single object



(b) Building block adjusted based on adjacency and roof type

Figure 5.14: Building block with gabled roofs.

5.7 Validation using Oracle Spatial

For the validation of the geometric reconstruction Oracle Spatial 11g is used. The validation rules as described in Section 5.1 and many more are implemented within the Oracle Spatial validator. Before being able to validate the geometry it should be imported into the database which requires the use of the Oracle Spatial geometry format.

The format used within the database is called SDO_GEOMETRY which is enhanced to store 3D data [Kothuri et al., 2008]. It consists of multiple objects such as SDO_GTYPE, SDO_ELEM_INFO and SDO_ORDINATES. The type of geometry is described by the SDO_GTYPE, the multiple elements and the way they are connected are described by the SDO_ELEM_INFO and the actual coordinates are stored in SDO_ORDINATES. The geometry is imported into the database using Structured Query Language (SQL). The following example SQL shows the constructor for a 3D solid.

```
sdo_geometry(
  3008,            -- SDO_GTYPE, 3D solid
  NULL, NULL,      -- SDR_SRID
  sdo_elem_info_array(
    1, 1007, 1,    -- Starting element offset, a solid,
                   -- number of surfaces describing the solid,
```

```
    1, 1006, 7,   -- Starting element offset, a surface,
                  -- number of polygons describing the surface
    1, 1003, 1,   -- Starting element offset, a simple polygon,
                  -- number of rings making up the simple polygon
    16, 1003, 1,
    31, 1003, 1,
    49, 1003, 1,
    64, 1003, 1,
    82, 1003, 1,
    97, 1003, 1
  ),
  sdo_ordinate_array(
    ...           -- The actual ordinates are stored here
  )
)
```

After all the buildings are imported as solids, a metadata table is created describing geometries and their tolerances. The metadata coordinate tolerance value is used as tolerance in the validation process. The following SQL example shows the creation of the metadata table.

```
insert into user_sdo_geom_metadata values (
  'SOLID3D',  -- Name of the table containing the geometries
  'GEOM',     -- Name of the row containing the geometries
  sdo_dim_array(
    sdo_dim_element(
      'X',    -- Name of the coordinate
      308179, -- Minimum of this coordinate direction
      611064, -- Maximum of this coordinate direction
      0.01    -- Tolerance value for coordinate direction
    ),
    sdo_dim_element('Y', 12628, 283595, 0.01),
    sdo_dim_element('Z', 0, 500, 0.01)
  ),
  NULL        -- SDO_SRID
)
```

The actual validation is done using the created metadata table. In the validation statement of the geometries a filter is installed to output only the invalid geometries. In the process marks a geometry as invalid the output is the reason of not passing the validation. Below an SQL statement is shown where the validation of the geometries is executed.

```
select id, status from (
  select id, sdo_geom.validate_geometry_with_context(geom,
    (select diminfo from user_sdo_geom_metadata where table_name='SOLID3D')
  ) status from SOLID3D
) where status!='TRUE'
```

Chapter 6

Implementation, experiments and discussion

This chapter is about the implementation of the algorithm into the prototype and explains engineering decisions. Experiments that are executed for testing the algorithm and finding the most optimal order of generalization steps as described in Chapter 3 are also covered. The most optimal order of generalization steps is found through an iterative process by using the quality parameters.

The data sets used within this thesis research are described in Section 6.1 together with examples of the resulting 3D reconstruction of these areas. The created prototype together with the experiments are explained in Section 6.2 and it also covers the results of the quality parameters of the generalization process. Throughout this same section the model created by the prototype is compared with a 3D model from an existing method based on generalization and decomposition with parametric shape fitting.

6.1 Used data sets

The quality and scale of the used building footprint data sets are of high importance in the results of the algorithm. If the input footprint data set is acquired from data of smaller scale, the results are less good compared to a data set derived from larger scale data, see Figure 6.1. Another difference is if data sets are describing building blocks as single objects or every building as separate objects. Besides the building footprints also the point clouds can be different, for example in the density of points or if they are interpolated into a grid. Within this research multiple data sets are used with varying scale and detail for both the building footprints as the point clouds.

The areas covered by and an explanation of the data sets is given in the following sections. A summary is shown in Table 6.1 outlining some specifications of the used data. The summary shows the complexity and size of the data sets in both the described area and vertices of the building footprints.

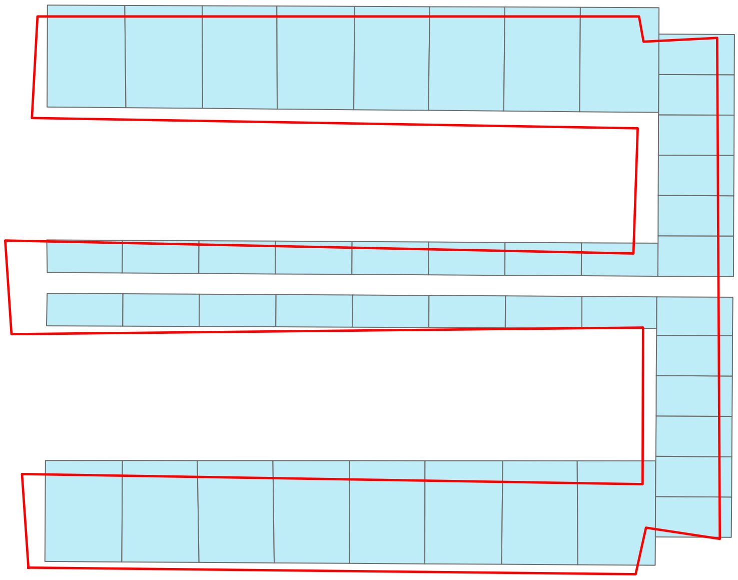


Figure 6.1: Two data sources overlaid, footprints acquired from higher scale data (blue) with footprints from smaller scale data (red outline).

Location	**Area**	**Buildings**	**Maximum vertices**	**Mean vertices**	**Total vertices**
Breda-Noord (BAG)	$4.4km^2$	1564	1819	17.3	27023
Breda-Noord (iDelft)	$3.2km^2$	534	61	9.2	4926
Scheveningen	$0.06km^2$	160	135	12.9	2073
Den Haag	$0.06km^2$	102	75	14.9	1517
Delft	$2.1km^2$	374	64	9.8	3668

Table 6.1: Specifications of data sets used within this thesis research.

The 3D building models in the following sections are reconstructed using the described LoD2 reconstruction method. The roof shapes that are reconstructed are colored based on their type. The colors and corresponding types are shown in Figure 6.2.

Figure 6.2: Roof types and corresponding colors which are used in the 3D visualization.

6.1.1 Breda-Noord

The data sets used for the area covering the north of the city Breda is shown in Figure 6.3. This area is covering an outer city region with terraced houses and an industrial terrain. This area is used especially for the implementation of and testing on the adjusting of the building blocks using the terraced houses.

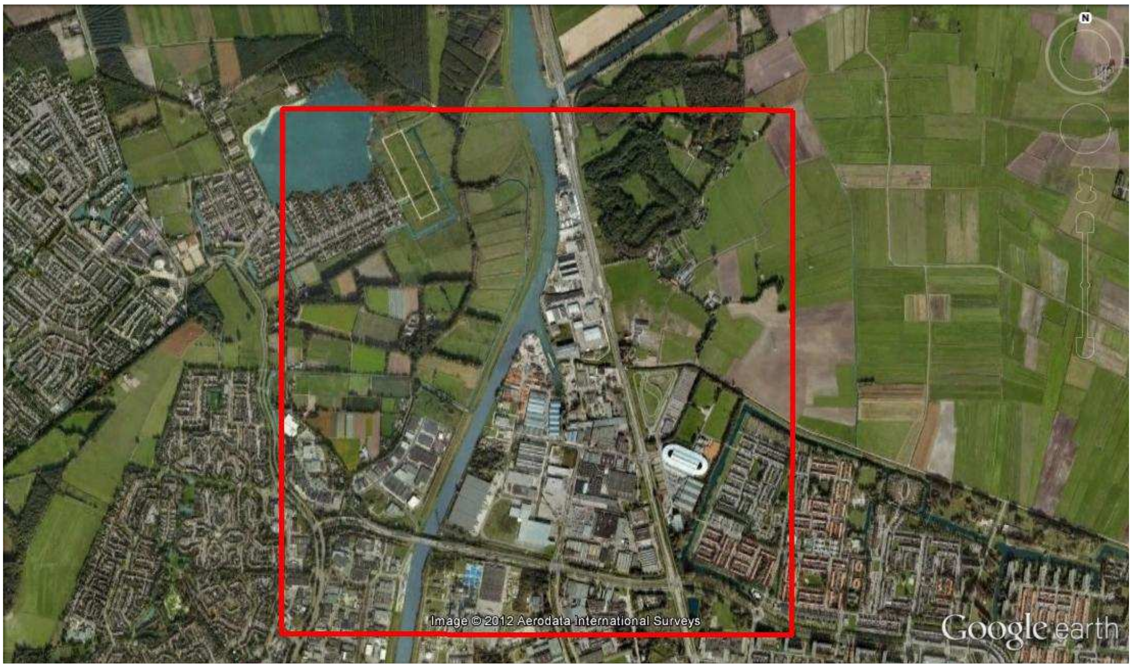


Figure 6.3: Area covering Breda Noord. The imagery is copyrighted by Aerodata International Surveys and Google (2012).

There are two building footprint sets and a point cloud covering this area used in this research. The building footprints are originating from iDelft/Eurosense and from BAG[1] which is a set of building registrations covering the whole of the Netherlands. The set from iDelft is obtained from aerial photography and covers blocks of buildings as single objects digitized by an operator. The BAG data models every building with a unique address as an object thus is of higher detail.

The point cloud available here is a part of AHN2[2] which is obtained with LiDAR and covers the whole of the Netherlands with a high density of points. From this AHN2, the part covering this area in Breda is obtained and the 50cm grid is used.

The LoD2 reconstruction of the complete data set using the BAG data can be seen in Figure 6.4. Since this area consists of quite some buildings a particular part is highlighted. This area benefits most of roof shape fitting and adjustments using adjacent buildings, and is shown in Figure 6.5.

[1] Information on BAG can be found at http://bag.vrom.nl
[2] Information on AHN2 can be found at http://www.ahn.nl

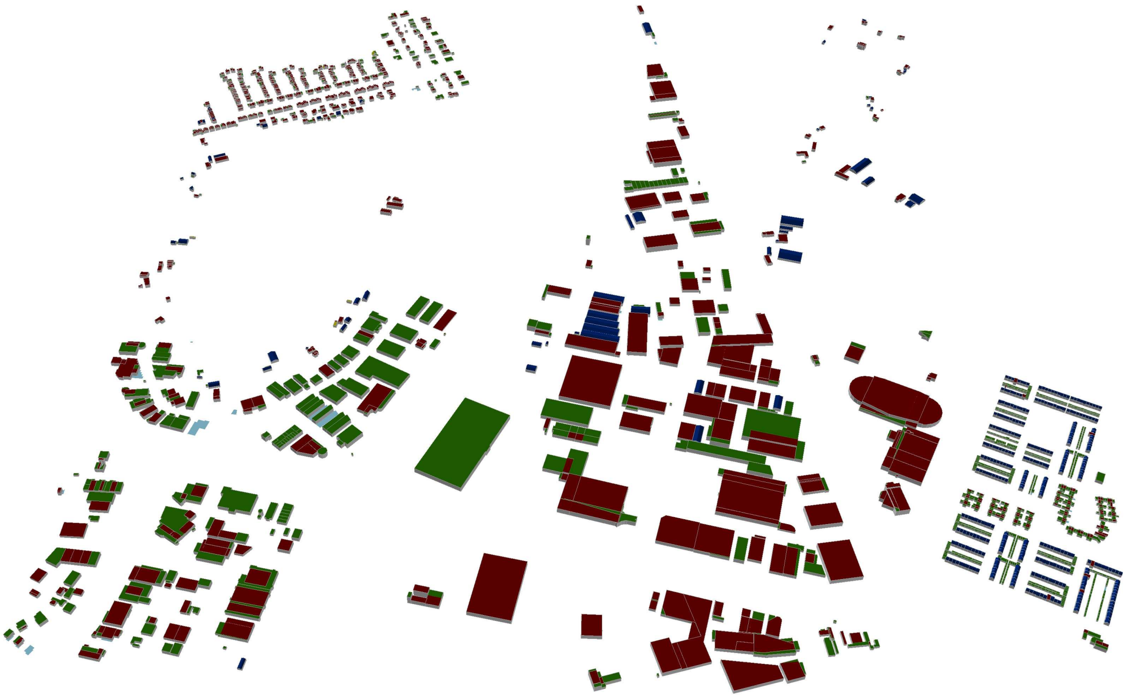


Figure 6.4: North part of Breda reconstructed in 3D.

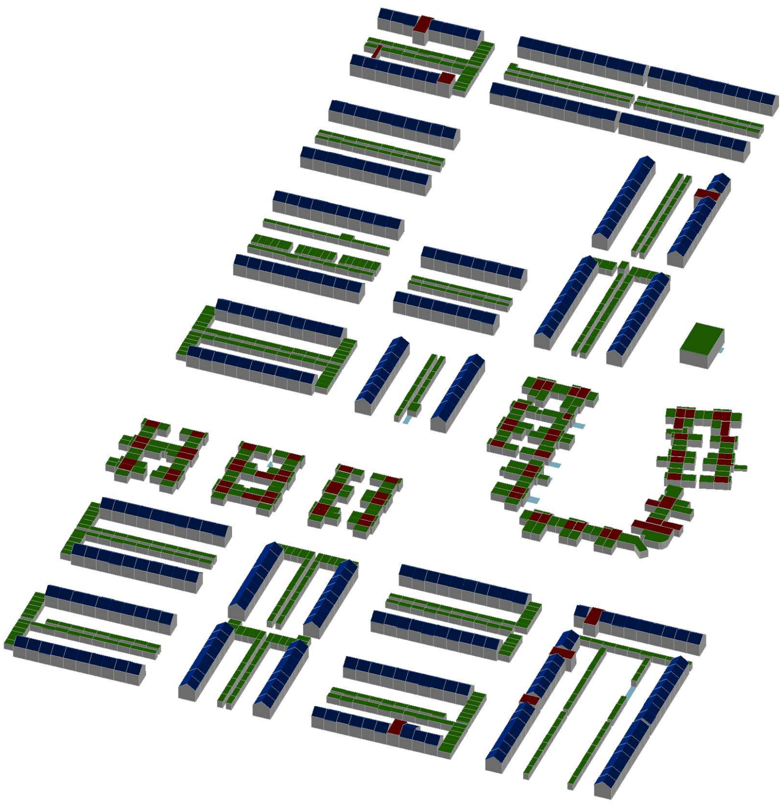


Figure 6.5: Terraced houses Breda reconstructed in 3D.

6.1.2 Scheveningen and Den Haag

The areas of Scheveningen and Den Haag are more complex than the one described before. These consist of inner city areas which have more complex shapes and roofs due to the limited space in the area. The selections as shown in Figure 6.6 and 6.7 are the areas available.

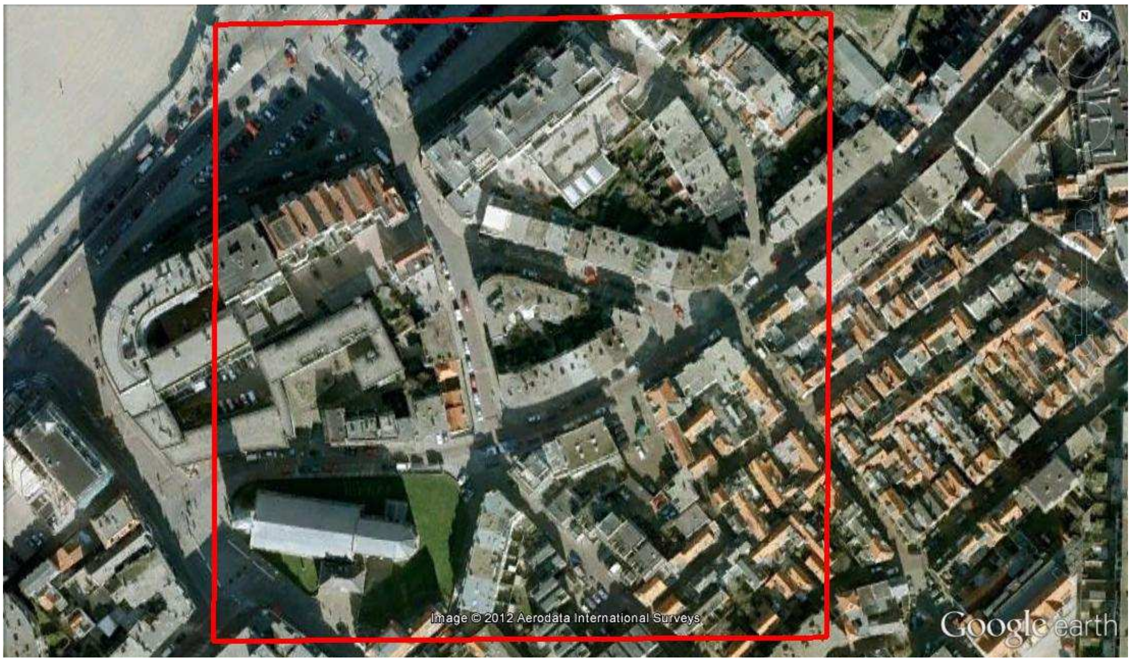


Figure 6.6: Area covering Scheveningen. The imagery is copyrighted by Aerodata International Surveys and Google (2012).

Figure 6.7: Area covering the center of Den Haag. The imagery is copyrighted by Aerodata International Surveys and Google (2012).

The data sets are courtesy of the municipality of Den Haag for the use within this thesis research. The building footprints are obtained and maintained by the municipality of Den Haag. The point cloud used is a part of the AHN2 filtered data set (*AHN2 uitgefilterd*). The AHN2 filtered data contains all raw points that do not belong to the ground surface, leaving points belonging to buildings, cars, trees etcetera.

The results from the reconstruction of the LoD2 3D building models is shown in Figure 6.8 for Scheveningen and Figure 6.9 for the area of Den Haag.

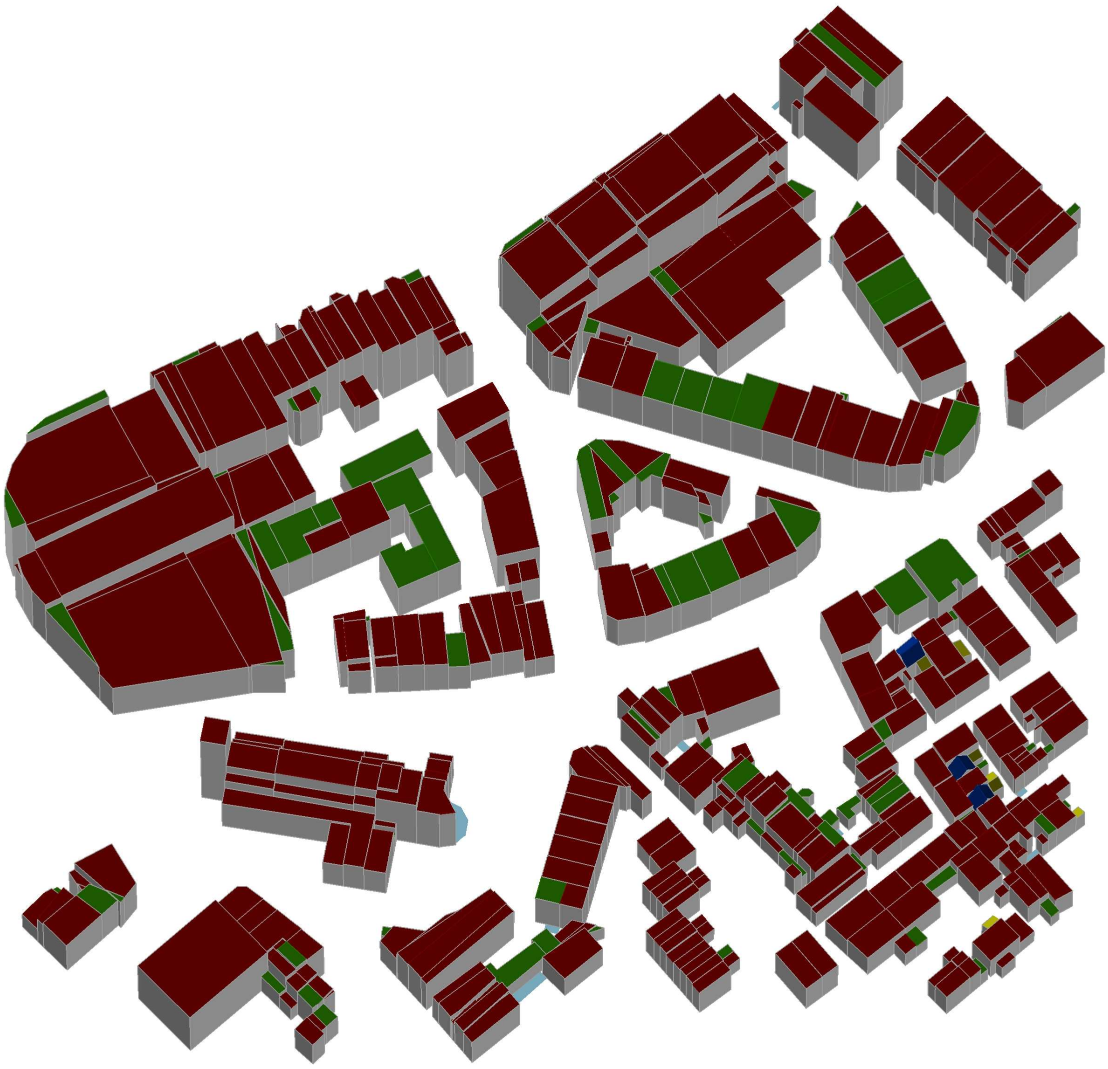


Figure 6.8: Part of Scheveningen reconstructed in 3D.

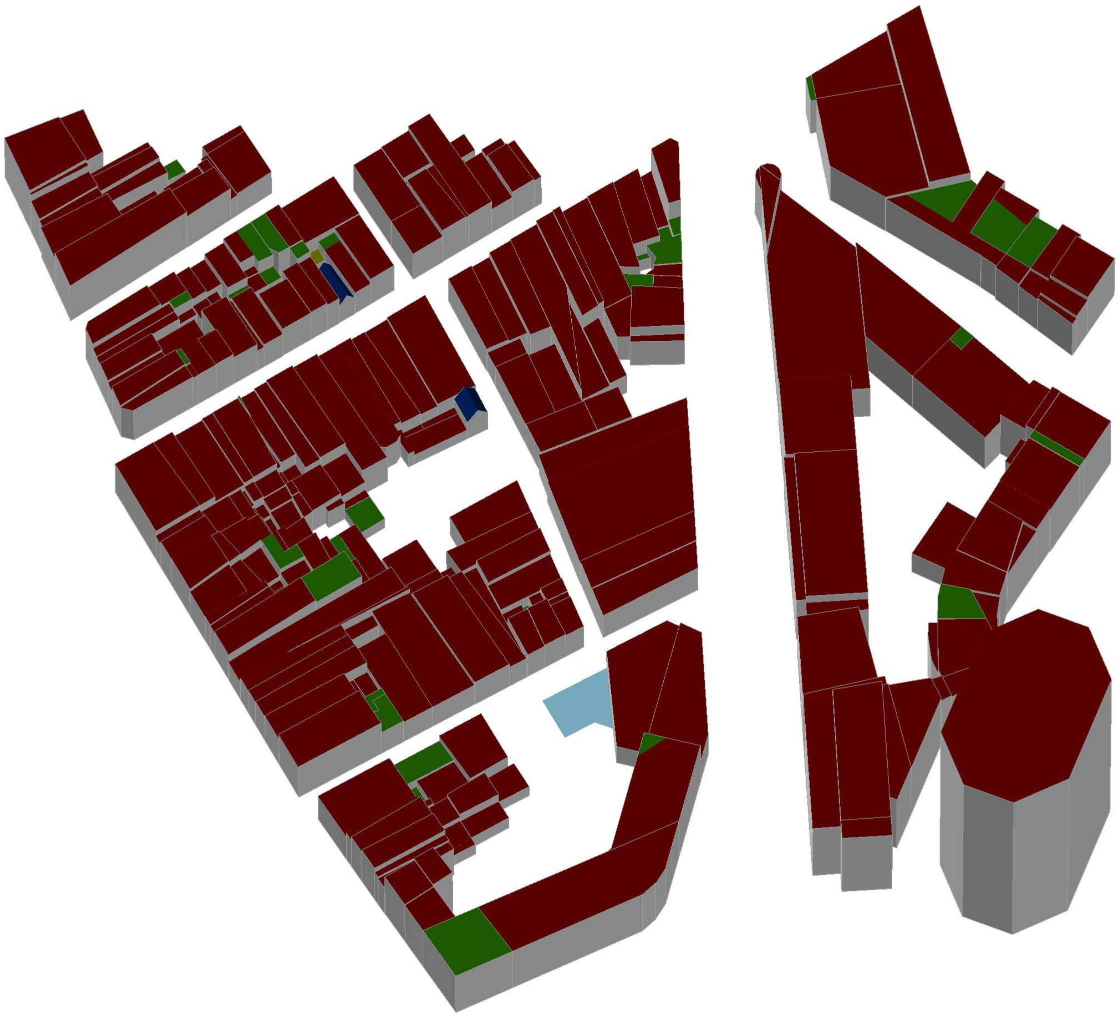


Figure 6.9: Part of the center of Den Haag reconstructed in 3D.

6.1.3 Delft

Since this thesis originates from the Delft University of Technology, the area covering the campus cannot be left untouched. The area covered is shown in Figure 6.10 from which only buildings belonging to the University are selected.

The building footprints are obtained from the TOP10NL[3] data set from which only buildings are used. TOP10NL is a topographic map of the Netherlands consisting of multiple layers describing various objects like roads, water and buildings. The point cloud available for this area is the AHN2 filtered data set (*AHN2 uitgefilterd*).

The reconstruction of the campus of the Delft University of Technology results in the LoD2 model as shown in Figure 6.11. As can be seen a lot of buildings have roofs of type Unknown even though they are actually flat. The roofs contain a lot of small point segments due to air conditioning units and such which disturb the detection of the roof type.

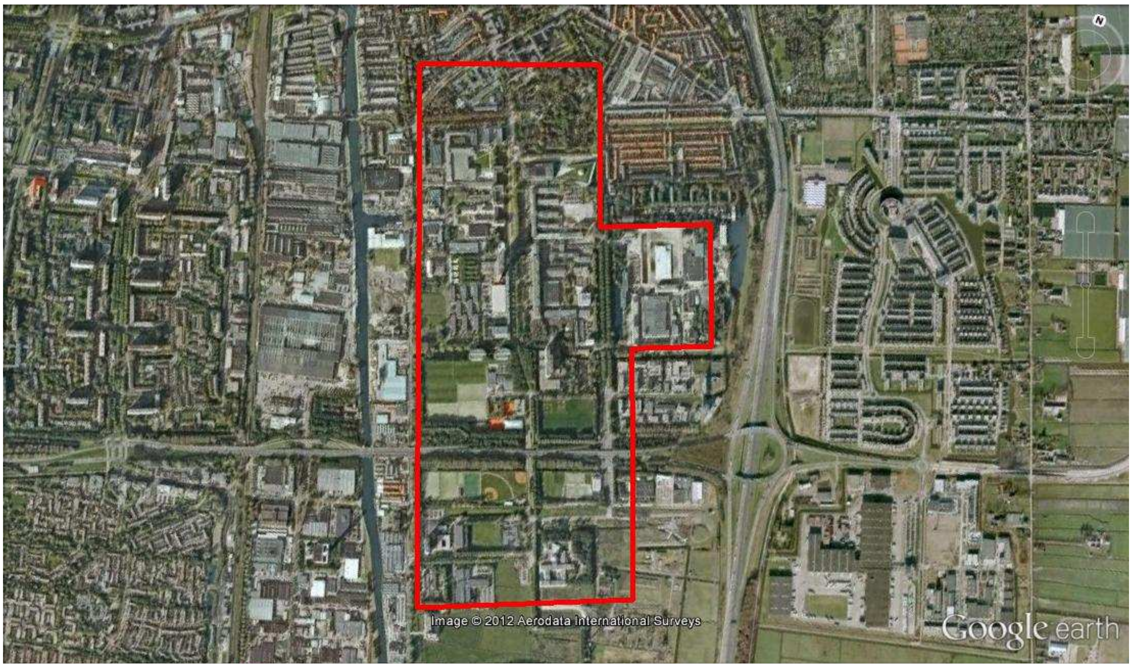


Figure 6.10: Area covering Delft University of Technology. The imagery is copyrighted by Aerodata International Surveys and Google (2012).

[3]Information on TOP10NL can be found at www.kadaster.nl/top10nl

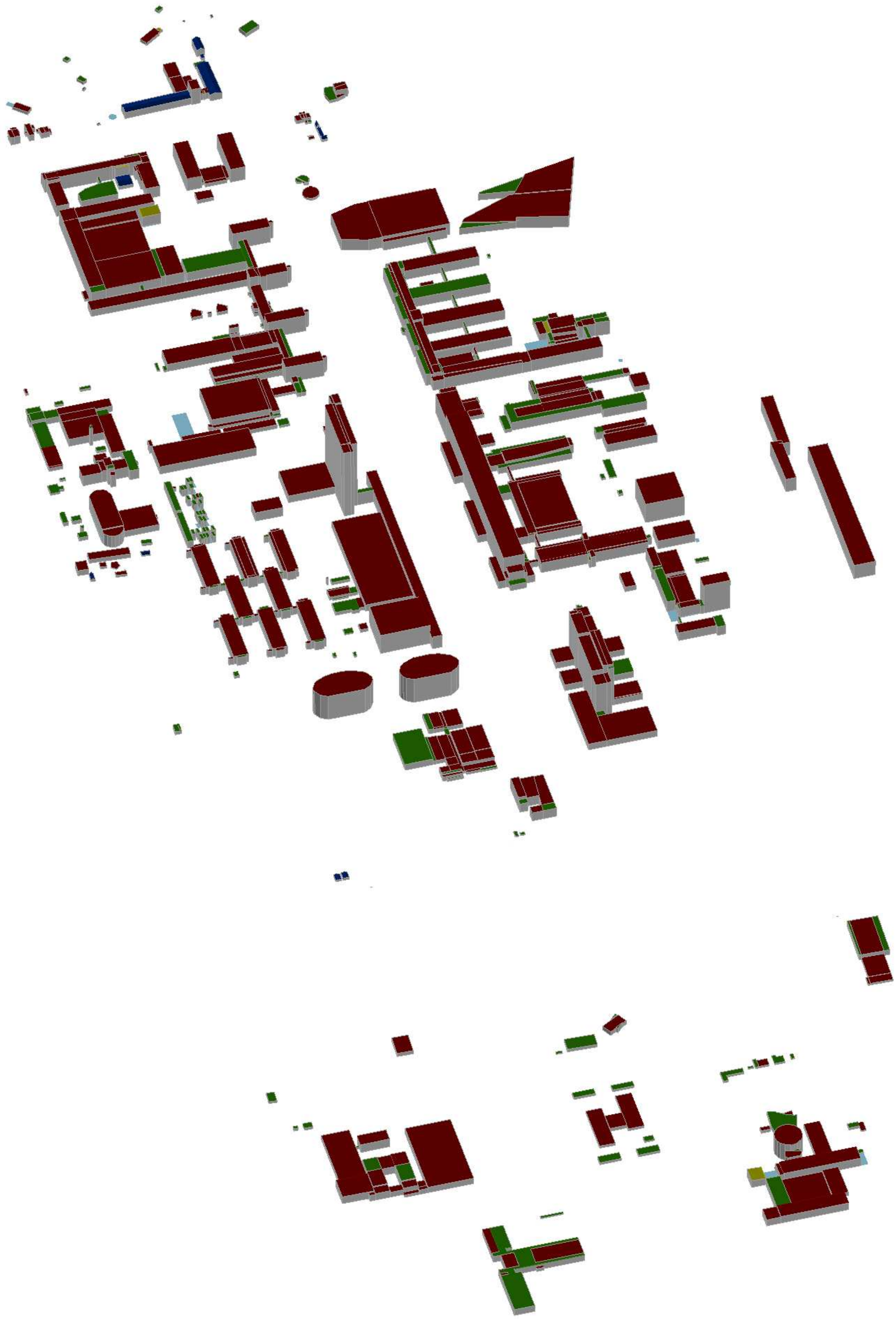


Figure 6.11: The campus buildings of Delft University of Technology reconstructed in 3D.

6.2 Prototype

In order to test and improve the algorithms designed within this thesis a prototype was implemented. The prototype is also a deliverable to iDelft since the request is to design and implement an automatic 3D city model reconstruction algorithm. The implementation of the prototype is divided into three parts which are separately described in the previous chapters. In Figure 6.12 the complete workflow of the prototype is visualized where the three parts are each a vertical group. The segmentation block in the middle group contains the segmentation software of Oude Elberink & Vosselman [2009].

For the prototype a fast implementation is written using the C++ programming language. The choice for C++ is based on previous experience, the creation of program executables and the availability of libraries in the same language. The OGR Simple Feature library is used for the reading and writing part of the data where the GEOS library, which supports a lot of spatial operators as well as topology functions, is used for spatial operations.

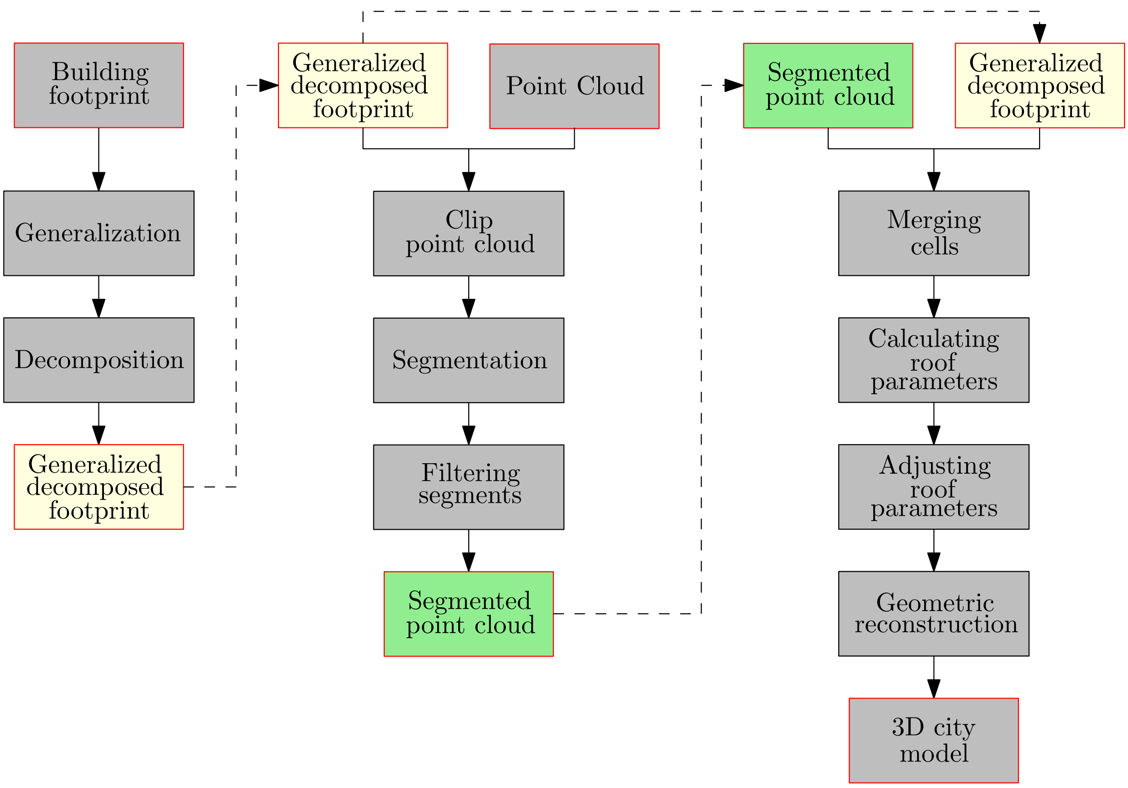


Figure 6.12: Complete workflow in three vertical parts describing the separation within the implemented prototype.

For comparing the results from the prototype with the original method described by Kada & McKinley [2009], a 3D model is obtained from the municipality of Den Haag which is created by software based on the original method. This 3D city model describes the parts of Den Haag and Scheveningen as shown in the previous section. The data described there is the input data used for the creation of this model. In the following sections the results obtained with the prototype are compared to this official 3D model from Den Haag.

6.2.1 Generalization and Decomposition

During the process of generalization of the building footprints multiple steps are taken for the detection of mergeable lines. These steps are all explained in Chapter 3 except for the implemented order of these steps which are shown in pseudocode below.

Algorithm 6.1 Pseudocode of the line generalization process.

Input: List of $Buffers$

Output: List of generalized $Buffers$

```
index ← 0
while index < count(Buffers) do
    for i = index + 1 → count(Buffers) do
        if Buffers(index) and Buffers(i) have same direction & are joinable then
            Join Buffers(index) and Buffers(i)
        end if
    end for
    for i = index + 1 → length(Buffers) do
        if Buffers(i) is includable with Buffers(index) then
            Include Buffers(i) in Buffers(index)
        end if
    end for
    index ← index + 1
end while
index ← 0
while index < count(Buffers) do
    for i = index + 1 → count(Buffers) do
        if Buffers(index) and Buffers(i) are joinable then
            Join Buffers(index) and Buffers(i)
        end if
    end for
    for i = 0 → length(adjacentBuffers) do
        if Buffers(index) and adjacentBuffers(i) are joinable then
            Join Buffers(index) and adjacentBuffers(i)
        end if
    end for
    for i = index + 1 → length(Buffers) do
        if Buffers(i) is includable with Buffers(index) then
            Include Buffers(i) in Buffers(index)
        end if
    end for
    for i = 0 → length(adjacentBuffers) do
        if adjacentBuffers(i) is includable with Buffers(index) then
            Include adjacentBuffers(i) in Buffers(index)
        end if
    end for
    index ← index + 1
end while
```

At the start of these steps the buffer list ($Buffers$) which contains all line buffers is sorted using the total length of the lines within a buffer. At line 1-14 the building lines are combined into one buffer only if they have the same direction, where at lines 15-38 the lines are combined also if they have opposite directions. The terminology joinable and includable, as used at lines 4 and 9 respectively, is identical to that used in Chapter 3. The inclusion of the adjacent buildings' line buffers is done at lines 22-26 and 32-36.

Right angles and parallel lines

The algorithm described in the research of Kada & McKinley [2009] forces all generalized lines in a building footprint to be pairwise parallel and have 90° angles. Since they have the general assumption that ridge and eaves lines should be strictly horizontal, not all roof faces will be planar if the ground shape is not rectangular. Forcing of the rectangular shapes improves the decomposition of the building footprint for the detection of roof shapes.

Within this thesis research tests have been implemented for forcing parallelism and right angles which showed that the generalization created worse results. When forcing these, the topology of the adjacent building footprints is often lost since it creates gaps and overlap between them as shown in Figure 6.13. This is due to generalizing buildings as single objects however it neither improves when including the adjacent buildings since this does not provide the needed information. There are no improvements since the lines are created pairwise parallel and the buildings do not know anything about the generalized lines of their adjacent buildings. Looking at the example in Figure 6.13, the right building forces the middle line to be pairwise parallel with the outer right line where the left building does the same for the outer left line. Since both buildings have no knowledge about the other outer lines, the resulting outlines are overlapping and create a gap.

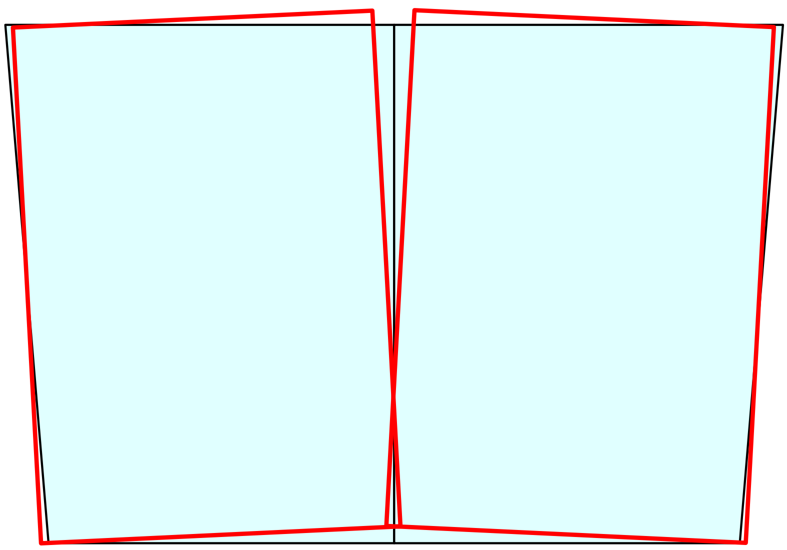


Figure 6.13: Two building footprints before (cyan) and after (red outline) forcing right angles and parallel lines.

Generalization quality

Some experiments used for the improvement of the generalization and decomposition are the quality parameters. A visualization of these parameters is used to identify poorly generalized building footprints. As described before in Section 3.6, the Euclidean distance between the quality parameters is used for the calculation of the total quality parameter (TQ) of the generalization. The generalized building footprints of the north of Breda are shown in Figure 6.15 where the color describes the quality percentage, 100% being identical to the original.

From the statistics of this quality percentage conclusions are drawn on the performance of the generalization algorithm. These statistics are plotted in a histogram showing the amount of buildings in contrast to the percentage in Figure 6.14. From this histogram one can identify that about 14% of the generalized footprints is below 90% quality and even 6% is below 80% quality. Using these statistics the generalization process is and can be further improved to obtain better results.

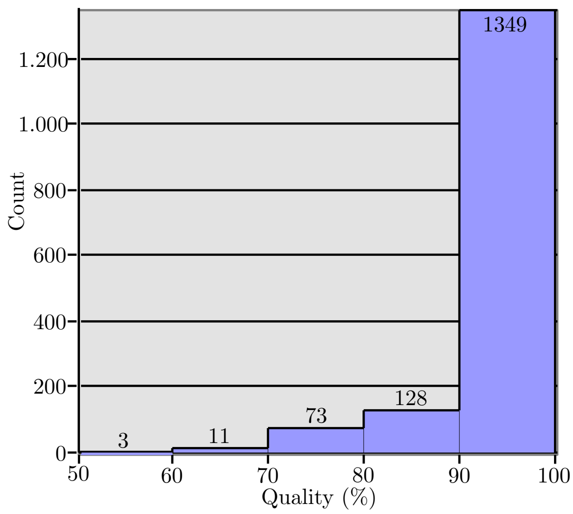


Figure 6.14: Histogram of the generalization quality in percentage for the north of Breda.

During this statistical analysis of the quality it is noticed that even small changes in direction and location of the outlines influence the contour trueness. This is due to the use of a buffer when calculating this parameter. Within this experiments a contour trueness buffer of 1cm is used which could be to small to identify the changes. This buffer is not the generalization distance but the distance in which a contour line is marked identical to the original as described in Section 3.6.2.

Generalization comparison

The generalization method developed and implemented within this thesis research does not always create topologically correct results. The buildings which have adjacent buildings on multiple sides are proven to be problematic. The generalization of these building footprints create gaps between the adjacent buildings. The generalization is compared to that of the official 3D model of Den Haag which does not show these problems at the

Figure 6.15: Total quality parameter of the generalization for the north of Breda.

example location shown in Figure 6.16. The origin for these generalization faults are not identified during this research.

Another interesting observation in the comparison of the produced result and the Den Haag model is the handling of holes within the footprints. The lower right building in Figure 6.16 contains a hole, the prototype produced the hole in the generalization result in contrast to the 3D model of Den Haag where the hole is removed.

From the same Figure 6.16 there is a third observation which lead to somewhat strange conclusions. The buildings on the lower left and the lower right show generalized parts in the Den Haag model. When identifying the size and locations of the generalizations, it looks like the surrounding buildings have other generalization parameters since parts with the same characteristics are not removed. Therefor it seams like the generalization used here is not consistent. Due to not knowing the exact implementation and generalization rules for this model, no real comparison can be made at this point.

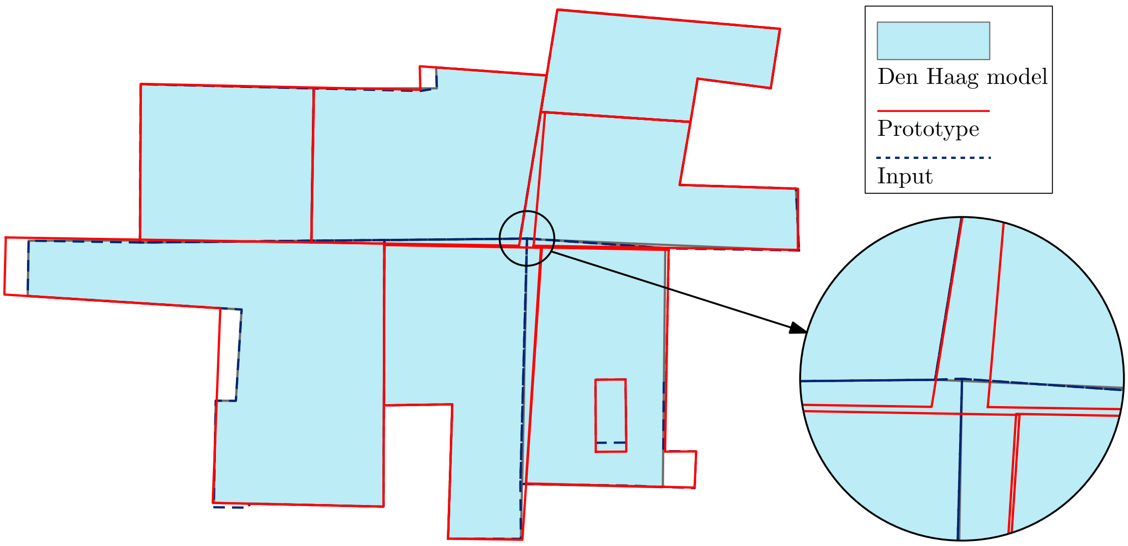


Figure 6.16: Faults due to generalization compared to the official 3D model of Den Haag. Data is copyrighted and courtesy of the municipality of Den Haag.

The results of the prototype are not always worse compared to the Den Haag model. Other parts of the generalization than showed before produce better results. The generalization used in the model of Den Haag also creates gaps between the buildings which is shown in Figure 6.17. In the same example the prototype creates a correct generalization of these building footprints.

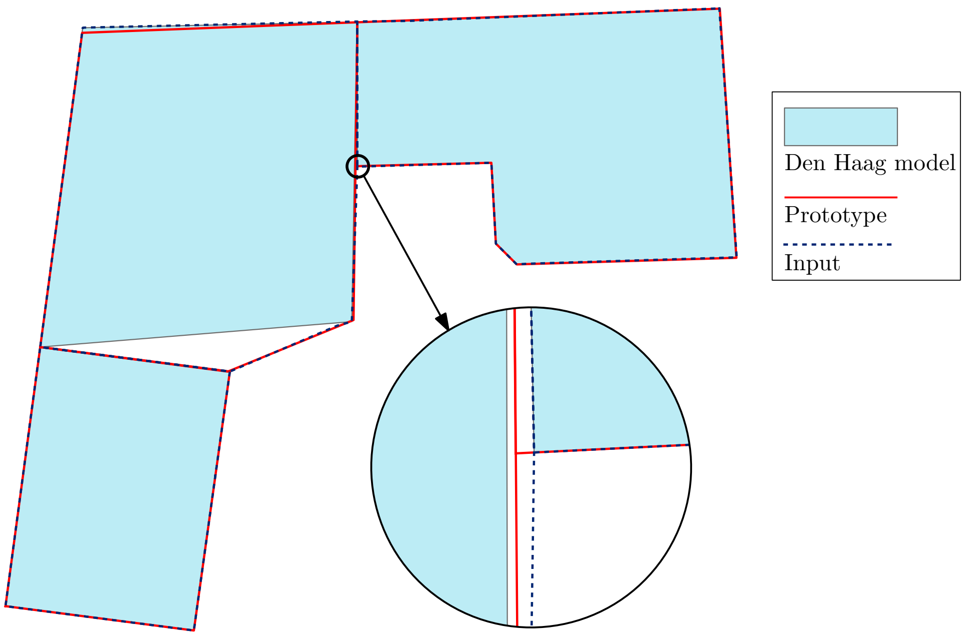


Figure 6.17: Improvements of generalization compared to the official 3D model of Den Haag. Data is copyrighted and courtesy of the municipality of Den Haag.

Besides the better results of the prototype in the previous example Figure 6.17, the model of Den Haag has another generalization artifact. The upper left building misses a large part of the footprint which created a gap between the adjacent building. The reason for this generalization remains unidentified. The observation based on this artifact is the way the lines are generalized, the two outer points of the original lines are kept to create the generalized line. This generalization looks like a Douglas-Peuker algorithm for line simplification but creates topology errors between buildings.

6.2.2 Segmentation

During the implementation of the segmentation the points are clipped to the generalized decomposed footprint. Since the point clouds can consist of millions of points a spatial indexing scheme is used to speed up the processing. If no spatial indexing is used, all points in the point cloud need to be traversed and checked if they are within the footprint. The spatial index used is a quadtree for dividing the point cloud into multiple regions which is available from OGR. Using the spatial indexing only the points within the quadtree cells that overlap the footprint are traversed.

At the start of the process the spatial index is created for the complete point cloud. When the clipping the points to the generalized outline the spatial indexing improves the processing time significantly. During the implementation an improvement of roughly 10x less processing time per building is noticed. This heavily depends on the size of the used data set. Small areas have a limited improvement where larger sets have a bigger gain. Since the creation of the spatial index takes time that is not the total improvement, however in total this is a large improvement.

Combining segmentation software

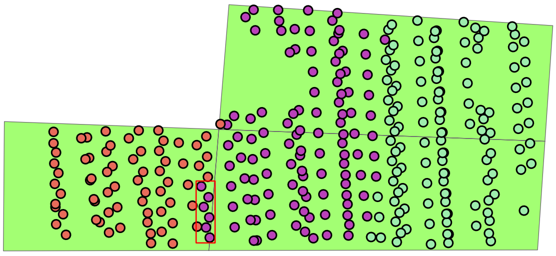
For the segmentation a program is compiled using the available software library of Oude Elberink & Vosselman [2009] which is used within the prototype. As described before in Chapter 4, the segmentation is an existing implementation. That implementation uses its own binary *laser* format for the data which is unusable within this research. In the library however there is a conversion program from ASCII to *laser* and the other way around. The adding of the normal vectors is done after the segmentation in another program. This results in the following steps, of which all are different programs:

- Read ASCII, write *laser*
- Read *laser*, segment points, write *laser*
- Read *laser*, add normal vectors, write *laser*
- Read *laser*, write ASCII

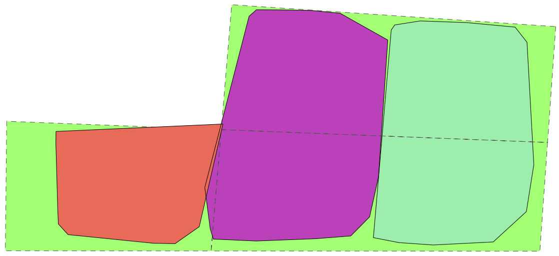
Using these steps, errors can occur more easily compared to having one single program. Besides that the processing time makes a huge difference since a single program does not have to read and write between every consecutive step. Therefor the library is taken and the source code of the respective programs are used to create a single one processing the complete workflow. This program now requires the point cloud input as ASCII and writes the segmented points with their segments normal vector in ASCII format.

6.2.3 3D geometries

For the implementation of the 3D geometry reconstruction the overlap test for the segments with a decomposition cell is executed. During the implementation this is tested using two possibilities, namely the points and the convex hull describing the segment both shown in Figure 6.18. The intersection between the segment and the decomposition cell needs to be identified. The main difference between the two is that when using the points, the intersection of every point with the cell needs to be identified. This is in contrast to the convex hull where only one calculation has to be made. This makes the convex hull better suited and faster for the overlap test, especially when having segments with a large point count.



(a) Visualized using points



(b) Visualized using convex hulls

Figure 6.18: Segmentation visualized using points and convex hull.

The overlap test for the segments uses an error margin for the overlap with the decomposition cells. A segment can contain a couple of points that fall within another cell but does not contribute information to that cell, see the red selection in Figure 6.18a. This is where the error margin comes in, it tests for the percentage of overlap with the cells, if this is too small it is not considered to overlap that cell.

When using the convex hull, its intersection with the decomposition cell can be used, the percentage of this area of the convex hull is then used as the overlap percentage. A side effect is that the convex hull can describe a larger area then the actual point cloud, as explained in Section 4.3.2, which can result in false positives.

If the overlap test is done by means of the points one can identify the amount of points within the decomposition cell and calculate its percentage of the segment point size. As described before this calculation is more time consuming than the convex hull but overcomes the problem of the area described by the convex hull. The problem when using this approach is however the variable density of points. If the density is significantly higher at the edge of the cell it could result in a false positive while the convex hull does not.

Let us say the intersection percentage is more than 5% for a segment to overlap a decomposition cell using the examples in Figure 6.18. Taking the middle purple segment and calculating if it is overlapping the lower left segment. By looking at the distribution of the segments it can be concluded that the segment needs to be marked *not* overlapping the decomposition cell. This middle segment consists of 116 points and intersecting this with the lower left cell results in 6 points overlapping, outlined in red. The resulting percentage of overlap is then 5.2%. Doing the same with the convex hull, the segments area is $11.75m^2$ and the area of the intersection with the lower left cell is $0.276m^2$ resulting in 2.3% overlap. The conclusion here is that using the points, the overlap detection returns a false positive, where using the convex hull it does not. The inverse of this example can also occur but the processing and use of the convex hull is more simple and faster compared to the points, therefore these are used in the prototype.

3D geometry comparison

When looking at the 3D model obtained from the municipality of Den Haag, one thing is directly noticed; the adjacent gabled roofs are not adjusted. As explained in Section 5.6, the adjacent buildings containing gabled roofs which are within a given threshold are averaged in the prototype. This is done since the calculation of the eaves and ridge heights can differ for the adjacent buildings because these can not be determined correctly. The model of Den Haag does not average these adjacent buildings as shown in Figure 6.19.

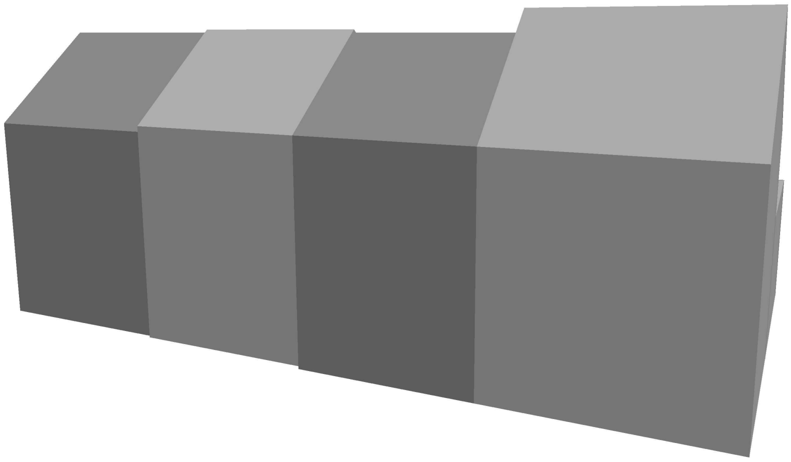
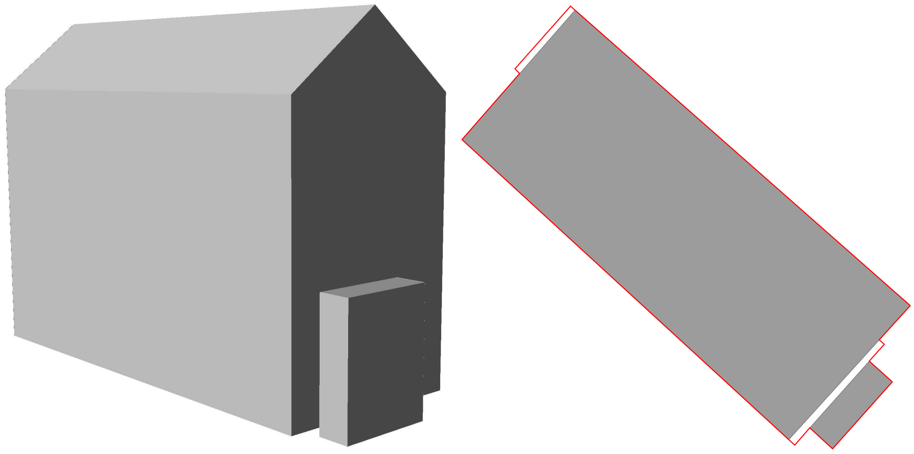


Figure 6.19: Adjacent gabled roofs are not adjusted in the official 3D model of Den Haag. Data is copyrighted and courtesy of the municipality of Den Haag.

An important part of this thesis research is the creation of valid 3D geometries. The geometries created by the prototype are shown to be solid in Section 5.7. During the comparison with the model of Den Haag it is identified that the buildings that are reconstructed are invalid in that model. There are building footprints which are split whereafter the resulting building parts are not attached. This means that even within a single building footprint there are gaps as shown in Figure 6.20. From the top view the detachment of the building parts is directly noticeable. The result of this is that the building as a whole is not a valid 3D geometry.

When the official model is put besides the result of the prototype it shows that the buildings reconstructed by the prototype are starting lower. The reason for this is that the floor of the building is fixed at zero height. Actually the floor of a building should be at ground height. This is not taken into account within this research resulting in the difference in the model. The reason this is not taken into account is that the floor height for a building needs to be calculated from a Digital Terrain Model (DTM). The point clouds that are used for the roof determination do not always contain the terrain points



(a) 3D model of the invalid building

(b) Top view of the 3D model (grey) overlaid with the original footprint (red outline)

Figure 6.20: Invalid building in official 3D model of Den Haag, front part is not attached to the rest of the building. Data is copyrighted and courtesy of the municipality of Den Haag.

of which these heights can be determined. A possible way to solve this is to add the building floor height to the input building footprints as an attribute and handle these in the prototype.

Validation errors

The final part of the 3D reconstruction requires the validation of the created building models, during the prototyping problems are identified in the construction of the geometry which are solved and some of them are outlined here.

The validation requires all surfaces describing the solids to have the correct orientation. The surfaces must have their normal direction pointing to the outside of the solid. During the first validation results it showed that the reconstruction of the walls was ordered incorrectly which is corrected so the normal vectors are pointing outside. This showed to be a bug in the prototype which is solved. The error code of Oracle for this is:

ORA-54502: solid not closed
Cause: The solid geometry was not closed i.e., faces of solid are not 2-manifold due to incorrectly defined, oriented, or traversed line segment because each edge of a solid must be traversed exactly twice, once in one direction and once in the reverse direction.
Action: Correct the orientation of the edges of the neighboring polygons.

Besides the orientation another problem showed, the points of the roof surfaces are not in the same plane. This error is due to the symmetry forced on the reconstructed shed and gabled roof shapes, see Figure 6.21. The eaves and ridge heights are assumed to be horizontal resulting in this error. As described before there is a threshold for the validation in Oracle. This threshold is set to $1cm$ after which there where no more errors.

A solution to the points not being in the same plane is to recalculate the height of the incorrect corner points given the location of the points and the eaves height. After discussing this problem with iDelft this is not desired to change, since the errors are within 1cm this is not considered a problem. Also when looking at the official 3D model obtained from the municipality of Den Haag for comparison, the same problems exist as shown in Figure 6.22. The code associated with this error is:

ORA-54505: ring does not lie on a plane
Cause: The ring was not flat.
Action: Make sure all of the vertices of the ring are on the same plane.

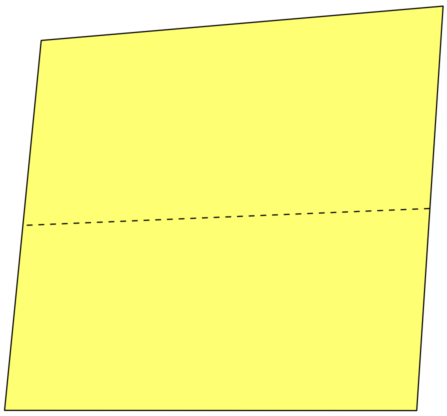


Figure 6.21: Gabled roof split on an asymmetric cell resulting in non-planar gabled roof planes.

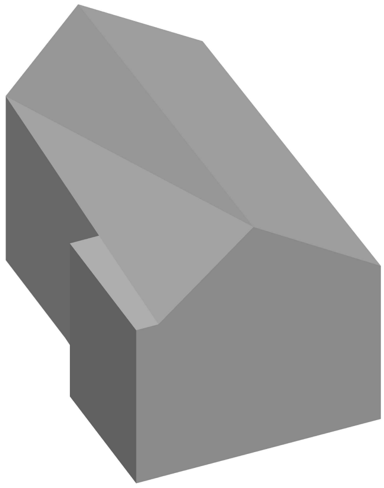


Figure 6.22: Non-planar gabled roof planes from the official 3D model of Den Haag.

As explained before in Section 5.4.4 the first reconstruction attempt of the gabled walls resulted in a validation error. The walls of the ridge side are not allowed to consist of two surfaces which lie in the same plane. A solution is to create a composite surface of the two polygons which would pass the validation. However a more elegant solution is implemented, the walls are reconstructed as a single surface which solves the error. The code belonging to this error is the following:

ORA-54516: adjacent outer rings of composite surface cannot be on same plane
Cause: The conditional flag was set, and a composite surface had at least two outer rings sharing a common edge on the same plane.
Action: Change those outer rings into one larger outer ring.

Chapter 7

Conclusions, recommendations and future work

Conclusions can be drawn from the research throughout this thesis. This final chapter covers these conclusions in Section 7.1. Recommendations based on findings during this research and related to the prototype are presented in Section 7.2. Finally Section 7.3 deals with proposed future work.

7.1 Conclusions

The main research question of this thesis is:
Does the combination of decomposition and point cloud segmentation improve building footprints for the creation of valid 3D geometries?

The answer to this question is yes, combining the two methods improves building footprints for the creation of 3D models. Results obtained from the prototype support this. Reconstructed models describe height differences and roof shapes apparent from the building footprints. Features in building roofs that are not present in the shape of building footprints cannot be identified using this method. Therefore not all height jumps can be detected, nevertheless most of the roof features are detected.

The main research question is answered by completing the main objective: *design and implement a building footprint generalization and decomposition algorithm in 2D combined with point cloud segmentation for the creation of valid 3D geometries*. This main objective is split into sub-research objectives which together result in the main objective. The rest of the conclusions are ordered based on their corresponding sub-research objectives.

Design and implement a method for footprint decomposition and generalization combined with point cloud segmentation.
Improvements on the generalization approach of Kada & McKinley [2009] are made. These are achieved by including the direction of lines and integrating the best fitting line

approach comparable to that of Peter et al. [2008], all covered in Section 3.1. Topology errors in the original generalization method are reduced by including adjacent buildings. By generalizing building footprints with knowledge of connected buildings, a lot of topology errors like overlap and creation of gaps are solved. The generalization is however not completely topologically correct, if a building has multiple neighboring buildings at one side, the generalization does create topology errors. These problems are described in Section 3.1.4. The created generalization gives correct results for decomposition, it removes small details and cleans footprint outlines for the application of decomposition.

Included in both the generalization and decomposition is the preservation of holes. Buildings can contain inner yards which are modeled as holes in the building footprints. By integrating them in the process they are preserved, generalized and even contributing to the decomposition as discussed in Section 3.4. The research of Kada & McKinley [2009] does not present this feature.

Presented in this research is the handling of slant lines. These are defined as lines that are not in the main building direction. They create undesired decomposition lines in footprints. Its problems and solutions are given in Section 3.5. Problems are solved by marking these lines, processing them last and only cut the cells they intersect.

During the research it is noticed that the shift between the used building footprints and point cloud can introduce problems. The point segments and the decomposition cells are not aligned due to the shift between the input data. The accuracy of the building footprints also has a large contribution to this problem. The merging of the decomposition suffers from false positives in the overlap detection of the segments. These false positives result in decomposition cells not satisfying the merging conditions where they should. A solution is created where the algorithm uses an overlap percentage for the segments with the decomposition cells to reduce the return of false positives. This solution is presented in Section 6.2.3.

Design and implement a method for creating valid 3D geometry and validate using Oracle spatial.
Reconstructed buildings that are adjacent can describe an identical roof. By using knowledge of adjacent buildings, small height variations in the roofs -due to averaging of the point heights- can be adjusted to improve the resulting model. The adjustment of roofs in complete building blocks is dealt with in Section 5.6. Results show that with the adjustments, created roofs are more equal and look better.

Validation of the created 3D geometries is executed using Oracle Spatial 11g. A conversion and export option to the Oracle Spatial geometry format is implemented in the prototype. At first the geometry validation returned errors due to bugs in the prototype but after solving these all geometries returned valid. Section 6.2.3 discusses the encountered problems and the implemented solutions.

Test the designed method for the usability with parametric shape fitting.
Experiments in this thesis show the application of parametric shape fitting for the creation of LoD2 models using the improved building footprints, as described in Section 5.4. The segmented point cloud is used for detection of roof shapes for every decomposition cell. Its implementation in the prototype only consists of a limited set of shapes to prove the possibilities. Roof shapes are identified based on the amount and normal

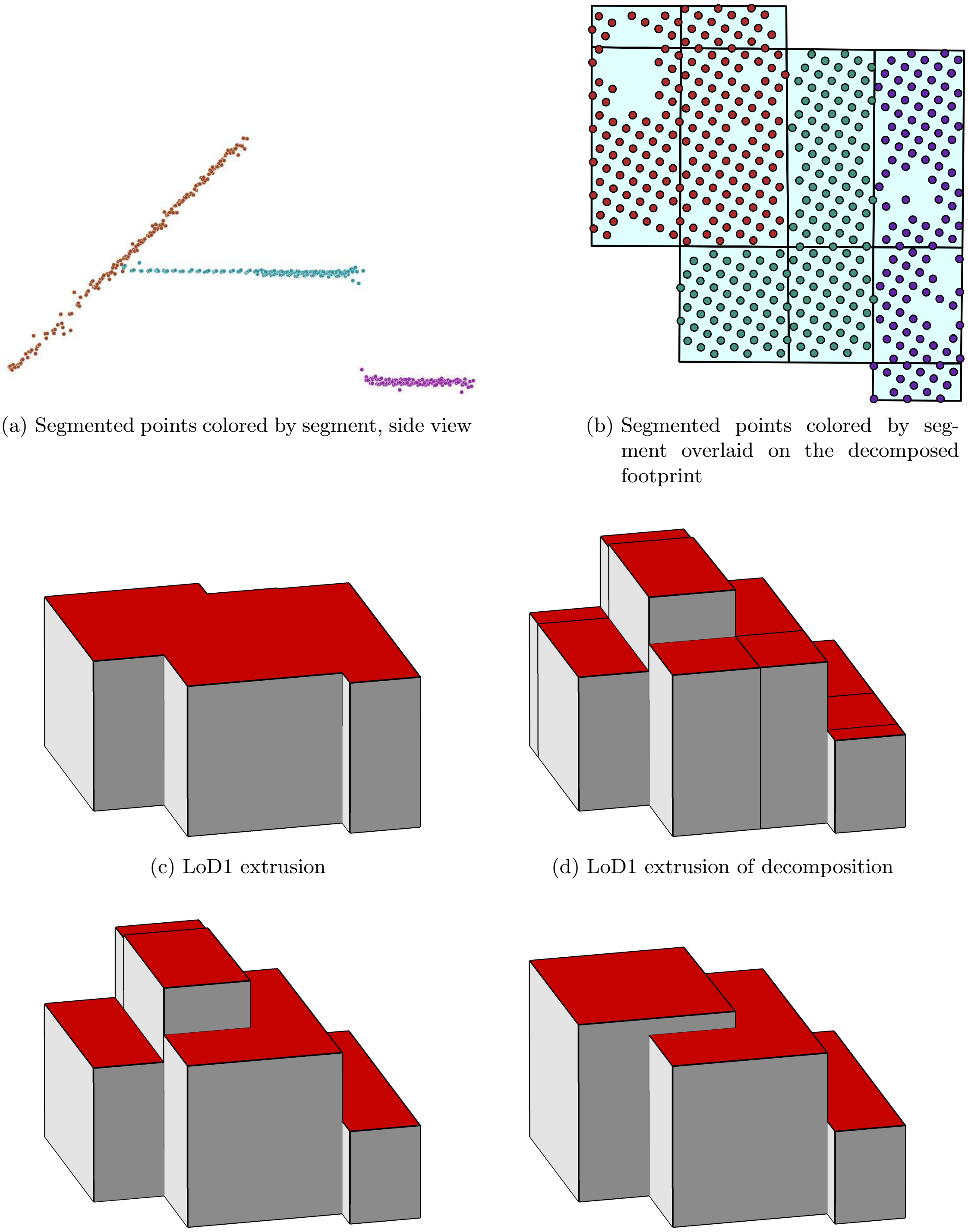
vectors of segments overlapping the cell. Only shapes that completely fit the segments are reconstructed by the prototype.

Perform quality assessment of the created model.
Since this research is aimed at fully-automated processing, quality parameters are included during the generalization. The quality parameters are used to identify buildings which are below a given quality threshold after generalization. Buildings below the given threshold are automatically flagged during the process. If a large amount of buildings is flagged, this implies incorrect generalization parameters as explained in Section 3.6. During this research the quality parameters are also used to inspect the generalization process. Adjusting and improving the process is eased by the use of the quality parameters since the impact of the adjustments are directly comparable.

Compare the prototype results with existing model from another method.
The municipality of Den Haag made their official 3D model, based on the method of Kada & McKinley [2009], available for comparison in this research. The Den Haag model shows topology errors due to the generalization, removal of holes and invalid 3D geometries. Most of the problems, even the generalization problems, are solved within this thesis research. One striking error in the Den Haag 3D model is gaps within a single building footprint, separating the front and the back part of the building, this results in an invalid 3D geometry. With the method described here it is impossible to create gaps within a decomposed building footprint.

Analyze results from the chosen methods and identify their problem cases. The method of Kada & McKinley [2009] does not merge decomposition cells and the method of Oude Elberink & Vosselman [2009] has problems with multiple height jumps between flat surfaces. The decomposition combined with the segmentation of point clouds for the creation of LoD1 models is improved step by step during this research. An example, containing the problems of both methods, is given in Figure 7.1. This example, together with the description below, shows the improvements made with this research.

The building consists of two flat roofs with height jumps and a shed roof next to it, as shown by the segmented points in Figure 7.1a. The segmented points are overlaid on the decomposed footprint which shows a large amount of decomposition cells compared to the segments, Figure 7.1b. The extrusion of the complete footprint to a single height is shown in Figure 7.1c followed by the decomposition extruded per cell in Figure 7.1d. Roughly the height difference are shown but the model consists of many parts. These parts can be joined by averaging their heights if they are within a distance threshold but this does not create a subdivision describing the different roof shapes, Figure 7.1e. By merging the cells based on the segmented point cloud, it does result in a correctly split building footprint and its corresponding extrusion, Figure 7.1f.



(a) Segmented points colored by segment, side view

(b) Segmented points colored by segment overlaid on the decomposed footprint

(c) LoD1 extrusion

(d) LoD1 extrusion of decomposition

(e) LoD1 extrusion of decomposition and averaging cell heights

(f) LoD1 extrusion of decomposition and merging cells based on segments

Figure 7.1: Improvements made on the LoD1 reconstruction step by step.

7.2 Recommendations

Generalization by building block
The best improvement for the prototype is removal of redundancy in the generalization process. As described in Section 3.1.4, line equations are calculated from the building footprints and adjacent building lines are included. Within the prototype, detection of adjacent buildings is repeated for every building together with the transformation from building outlines into line equations. This is redundant which can be resolved by applying the same strategy as used in the 3D geometry reconstruction. The generalization is processed by building block in stead of single objects. The process would then be: detect a building block, transform all lines to line equations and process per building.

Slant line cuts
Slant lines in building footprints are troublesome for the decomposition. Slant lines decompose the footprint where it is most often is unwanted. Improvements made by only splitting the decomposition cells which the slant lines intersect solve for most cases but not all. The merging of decomposition cells which are split by a slant lines could be researched. For example if two cells with slant lines together form a rectangle, they can be merged.

Fixed proportion roof heights
The prototype does not contain an operational fixed proportion operator as explained in Section 5.3. The extrusion height can be obtained from a fixed proportion of the difference between eaves and ridge height. Since the implementation of the LoD1 reconstruction in the prototype contains the knowledge about these two heights, the fixed proportion can be implemented.

7.3 Future work

Shifted footprints and point cloud
As explained in the conclusions, shifts between building footprints and the point cloud introduce problems. Detection and correction of these shifts is not within the scope of this research. The literature found on this topic is quite scarce, revealing the need for further research. The method described here would certainly improve from the automatic adjustment of the shift between the point cloud and building footprints.

Ground height computation
The computation of ground height is not covered nor implemented in the created prototype. The ground height is the height of the earth surface at the location of the building footprint. This height is needed for the floor height of the buildings. It can be calculated using a DTM and intersecting this with the building footprint. Another solution could be using a buffer around the building footprint, selecting the points within the buffer and compute the height from the returned points.

Parallelism and right angles
Problems which are introduced when forcing parallel lines and right angles might be solved by processing buildings per block in the generalization and decomposition process. When forcing parallelism and right angles, the adjacent buildings are overlapping or gaps are

created between them. When processing buildings as a block, there is more knowledge about shared building lines by which it could be possible to overcome these problems.

Topologically correct generalization
The generalization method proposed here is not topologically correct. Adding topology relations in the process will help preserve topology. Lines that contribute to two or more buildings can be flagged and generalized accordingly. This should solve the introduction of gaps and overlap, even in the case of multiple adjacent buildings.

Circular elements and curved surfaces
Presence of circular elements results in problems when decomposing a building footprint. Circular elements should be detected, generalized accordingly and placed back into the decomposed footprint. Detection of roof shapes on circular elements can also benefit from this separate processing. Circular roof shapes like a cone or a sphere are curved surfaces and therefore difficult to identify from planar segments.

Segment adjacency knowledge
Since the decomposition of the building footprints is driven by the shape of the footprint, roof shapes are not always separated by the decomposition process. Height jumps and multiple roof shapes in a line are not identified if not apparent in the shape. A way to optimize the decomposition is to include intersection lines between segments, but only if they describe different roof shapes. For this to work segment adjacency knowledge has to be integrated in the process, for example a RAG.

Roof shape detection
The algorithm and its implementation for LoD2 with parametric shape fitting now only apply the roof shapes if a roof exactly matches the described shape. If a gabled roof contains for example a dormer, the roof is not marked as a gabled roof. Detection of roof shapes could be improved by choosing the best fitting model or loosening the parametric shape description. This can however also result in more false positives in the roof shape detection.

References

Berg, M. de, Kreveld, M. van, & Schirra, S. (1998). Topologically correct subdivision simplification using the bandwidth criterion. *Cartography and Geographic Information Science*, *25*(4), 243–257.

Billen, R., & Zlatanova, S. (2003). 3D spatial relationships model: a useful concept for 3D cadastre? *Computers, Environment and Urban Systems*, *27*(4), 411 - 425.

Bose, P., Cabello, S., Cheong, O., Gudmundsson, J., Van Kreveld, M., & Speckmann, B. (2006). Area-preserving approximations of polygonal paths. *Journal of discrete algorithms*, *4*(4), 554–566.

Buchin, K., Meulemans, W., & Speckmann, B. (2011). A new method for subdivision simplification with applications to urban-area generalization. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (pp. 261–270). New York, NY, USA: ACM.

Chen, D., Daescu, O., Hershberger, J., Kogge, P., Mi, N., & Snoeyink, J. (2005). Polygonal path simplification with angle constraints. *Computational Geometry*, *32*(3), 173–187.

Dorninger, P., & Pfeifer, N. (2008). A comprehensive automated 3D approach for building extraction, reconstruction, and regularization from airborne laser scanning point clouds. *Sensors*, *8*(11), 7323–7343.

Douglas, D., & Peucker, T. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, *10*(2), 112–122.

Durupt, M., & Taillandier, F. (2006). Automatic building reconstruction from a digital elevation model and cadastral data: an operational approach. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *35*, 142–147.

Edelsbrunner, H., Kirkpatrick, D., & Seidel, R. (1983). On the shape of a set of points in the plane. *Information Theory, IEEE Transactions on*, *29*(4), 551–559.

Efron, B. (1965). The convex hull of a random set of points. *Biometrika*, *52*(3-4), 331.

Filippovska, Y., & Kada, M. (2010). Shape assessment of generalized building footprints. In *Proceedings of ASPRS, CaGIS, ISPRS, and AutoCarto Fall Conference, Orlando, Florida, USA.*

Filippovska, Y., Kada, M., & Fritsch, D. (2009, June). Quality evaluation of ground plan generalization. In *Proceedings of 12th AGILE Conference on GIScience* (pp. 1–8).

Filippovska, Y., Walter, V., & Fritsch, D. (2008). Quality evaluation of generalization algorithms. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *37*, 799.

Gröger, G., Kolbe, T. H., Czerwinski, A., & Nagel, C. (2008). *OpenGIS City Geography Markup Language (CityGML) Encoding Standard (OGC 08-007r1)* (Tech. Rep.). Open Geospatial Consortium inc. OpenGIS Encoding Standard.

Haala, N., & Kada, M. (2010). An update on automatic 3D building reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, *65*, 570-580.

Häfele, K.-H. (2010, 12). *CityGML Model of the FZK-Haus.*

Haunert, J.-H., & Wolff, A. (2010). Optimal and topologically safe simplification of building footprints. In *Proceedings of the 18th sigspatial international conference on advances in geographic information systems* (pp. 192–201). New York, NY, USA: ACM.

Hirschmüller, H., & Bucher, T. (2010). Evaluation of Digital Surface Models by Semi-Global Matching. *Publikationen der Deutschen Gesellschaft fr Photogrammetrie, Fernerkundung und Geoinformation*, *19*, 571–580.

ISO. (2003). *ISO 19107:2003 Geographic information – Spatial schema* (Tech. Rep.). International Organization for Standardization (TC 211).

Kada, M. (2010). Generalization of 3D Building Models for Map-Like Presentations. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *37*, 399.

Kada, M., & Luo, F. (2006). Generalisation of building ground plans using half-spaces. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *36*, on CD.

Kada, M., & McKinley, L. (2009). 3D building reconstruction from lidar based on a cell decomposition approach. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *38*, 47–52.

Kada, M., Peter, M., & Filippovska, Y. (2009). 3D building generalization under ground plan preserving quality constraints. *Spatial Data Quality - From Process to Decisions*, 73–85.

Kazar, B., Kothuri, R., Oosterom, P. van, & Ravada, S. (2008). On valid and invalid three-dimensional geometries. *Advances in 3D geoinformation systems*, 19–46.

Köninger, A., & Bartel, S. (1998). 3D-GIS for urban purposes. *Geoinformatica*, *2*(1), 79–103.

Kothuri, R., Godfrind, A., & Beinat, E. (2008). *Pro Oracle Spatial for Oracle Database 11g.* A1bazaar.

Ledoux, H., & Meijers, M. (2011). Topologically consistent 3D city models obtained by extrusion. *International Journal of Geographical Information Science*, *25*(4), 557–574.

Ledoux, H., Verbree, E., & Si, H. (2009). Geometric Validation of GML Solids with the Constrained Delaunay Tetrahedralization. In *Proceedings of the 4th International Workshop on 3D Geo-Information* (pp. 143–148).

Lee, J., & Zlatanova, S. (2008). A 3D data model and topological analyses for emergency response in urban areas. *Geospatial Information Technology for Emergency Response*, 143–168.

Oude Elberink, S. (2008). Problems in automated building reconstruction based on dense airborne laser scanning data. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *37*, 93–98.

Oude Elberink, S. (2009). Target graph matching for building reconstruction. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *38*, 49–54.

Oude Elberink, S. (2010). *Acquisition of 3d topgraphy : automated 3D road and building reconstruction using airborne laser scanner data and topographic maps.* NCG, Nederlandse Commissie voor Geodesie.

Oude Elberink, S., & Vosselman, G. (2009). Building reconstruction by target based graph matching on incomplete laser data: Analysis and limitations. *Sensors*, *9*(8), 6101–6118.

Peter, M., Haala, N., & Fritsch, D. (2008). Preserving ground plan and facade lines for 3D building generalization. In *The xxi congress of the international society for photogrammetry and remote sensing (isprs 2008), beijing, china* (pp. 481–486).

Podolskaya, E., Anders, K.-H., Haunert, J.-H., & Sester, M. (2008). Quality assessment for polygon generalization. *Quality Aspects in Spatial Data Mining*, 211–220.

Schulte, C., & Coors, V. (2008). Development of a CityGML ADE for dynamic 3D flood information. In *Joint ISCRAM-CHINA and GI4DM Conference on Information Systems for Crisis Management.*

Taillandier, F. (2005). Automatic building reconstruction from cadastral maps and aerial images. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *36*, 105–110.

Vallet, B., Pierrot-Deseilligny, M., Boldo, D., & Brédif, M. (2011). Building footprint database improvement for 3D reconstruction: A split and merge approach and its evaluation. *ISPRS Journal of Photogrammetry and Remote Sensing*, *66*, 732–742.

Vosselman, G. (1999). Building reconstruction using planar faces in very high density height data. *International Archives of Photogrammetry and Remote Sensing*, *32*(3; SECT 2W5), 87–94.

Vosselman, G., & Dijkman, S. (2001). 3D building model reconstruction from point clouds and ground plans. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *34*(3/W4), 37–44.

Vosselman, G., Gorte, B., Sithole, G., & Rabbani, T. (2004). Recognising structure in laser scanner point clouds. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *46*(8), 33–38.

Vosselman, G., & Suveg, I. (2001). Map based building reconstruction from laser data and images. *Automatic Extraction of Man-Made Objects from Aerial and Space Images (III)*, 231–239.

Wahl, R., Schnabel, R., & Klein, R. (2008). From detailed digital surface models to city models using constrained simplification. *Photogrammetrie, Fernerkundung, Geoinformation*, *3*, 207–215.